Fault Modeling and Testing of Flash Memories

Olivier Ginez

To cite this version:

Olivier Ginez. Fault Modeling and Testing of Flash Memories. Micro and nanotechnologies/Microelectronics. Université Montpellier II - Sciences et Techniques du Languedoc, 2007. English. NNT : . tel-00194584

UNIVERSITE MONTPELLIER II
SCIENCES ET TECHNIQUES DU LANGUEDOC

T H E S E

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE MONTPELLIER II

Spécialité : Microélectronique.

Formation Doctorale : Systèmes Automatiques et Microélectroniques.

Ecole Doctorale : Information, Structures et Systèmes.

présentée et soutenue publiquement

par

Olivier GINEZ

le 29 Novembre 2007

Modélisation de Fautes et Test des Mémoires Flash

(Fault Modeling and Testing of Flash Memories)

JURY

M. Dominique Dallet, Professeur, IMS, ENSEIRB, Bordeaux	Rapporteur
M. Jean-Michel Portal, Maître de Conférences, L2MP, Polytech' Marseille	Rapporteur
M. Jean-Michel Daga, Sté ATMEL, Rousset	Examinateur
M. Arnaud Virazel, Maître de Conférences, Université Montpellier II – LIRMM	Examinateur
M. Joan Figueras, Professeur, Politecnica de Catalunya, Barcelone	Membre Invité
M. Patrick Girard, Directeur de Recherche CNRS, LIRMM	Directeur de Thèse
M. Serge Pravossoudovitch, Professeur, Université Montpellier II – LIRMM	Co-Directeur de Thèse

A ma fille et à ma femme dont le soutien et la présence m'ont permis de mener cette thèse à bien.

A mon grand-père qui je l'espère est fier de moi même si parfois le chemin que je trace n'est pas toujours droit.

A mes parents qui ont su se montrer patients avec moi.

Remerciements

Cette thèse a été effectuée au Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM), dirigé par Monsieur Michel Robert, Professeur à l'Université de Montpellier II, dans le département Microélectronique dont le responsable est Monsieur Lionel Torres, Professeur à l'Université de Montpellier II. Je les remercie de m'avoir accueilli.

Toute ma gratitude va à mon directeur de thèse, Patrick Girard, Directeur de Recherche au CNRS, dont l'encadrement très professionnel, la présence et les encouragements constants m'ont permis d'évoluer en thèse en toute sérénité. Je voudrais qu'il trouve ici toute l'expression de ma reconnaissance.

Je tiens à remercier mon co-directeur de thèse Serge Pravossoudovitch, Professeur à l'Université de Montpellier II, dont la pédagogie a suscité en moi l'envie d'enseigner mais aussi d'avoir bien voulu présider le jury chargé de juger ma thèse.

Je remercie aussi Christian Landrault ainsi qu'Arnaud Virazel pour m'avoir encadrer avec beaucoup d'enthousiasme durant ces trois années mais aussi pour m'avoir tant apporté sur le plan scientifique et technique. La thèse nous prouve parfois qu'il ne s'agit pas forcément que d'une histoire de scientifiques mais aussi d'hommes !

Je voudrais également remercier Jean-Michel Daga, Manager de l'équipe conception de mémoires non-volatiles dans la société ATMEL, qui m'a beaucoup aidé tout au long de cette thèse tant sur l'aspect technique qu'industriel de mes travaux. Sa vision pragmatique et son esprit de synthèse ont été très précieux pour mener à bien cette thèse. Je lui adresse ici mes remerciements les plus sincères.

Je remercie également Messieurs Dominique Dallet, Professeur à l'ENSEIRB de Bordeaux, Joans Figueras, Professeur à l'Université de Catalogne, et Jean-Michel Portal, Maître de Conférences à Polytech' Marseille, de s'être intéressés à ce travail de thèse et d'avoir accepté d'en être les rapporteurs. Qu'ils trouvent ici toute l'expression de ma reconnaissance.

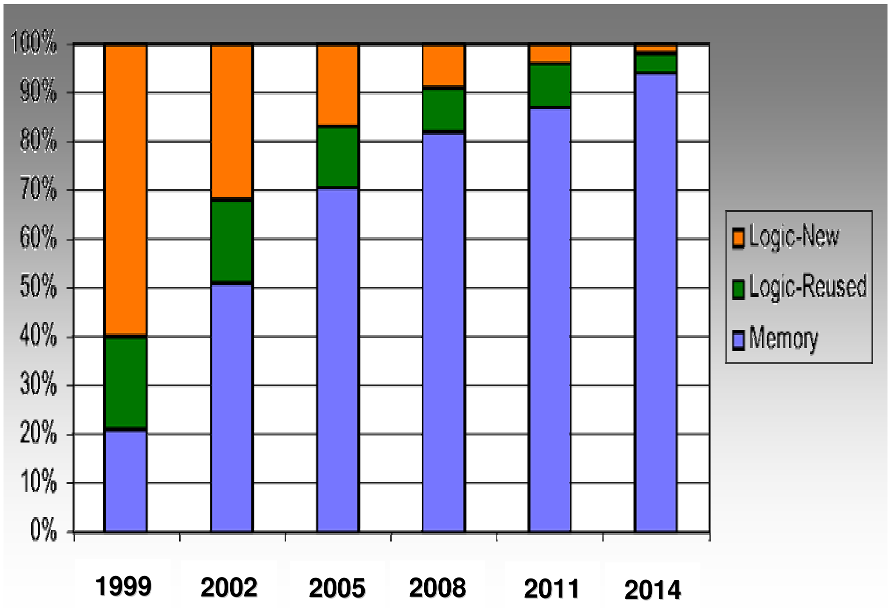
Enfin, je terminerais en remerciant tous mes collègues du laboratoire pour les bons moments que nous avons passés ensemble, Grand Nico, Petit Nico, Marie, Lio, Alex N, Alex R, Fab, Alin, Seb, Robin, Olivier B, Nico H, JE, Julien V, Julien D, Olivier L, Marion, Mehdi, Olivier H, Luigi, Norbert, Réou, JB, JD, Diego.

A tous Merci…

Index

General Introduction

Since the transistor invention, a lot of microelectronics components have been developed in academic laboratories or in semi-conductor companies. One of the most used microelectronic components is the memory circuit. The memory business has started to increase at the beginning of the 70's with the arrival of the first microprocessor (4004) developed in 1968 by Intel. Nowadays, semi-conductor memories are not only used in microprocessors but also in each electronic device developed for information or communication application fields. The memory market has continued to increase drastically in the last years and now achieves 25% of the total microelectronics market. This growth is due to the high demand of customers for personal computers, mobile phones and other high technology devices available on the market since 2000. However, in memory chips many factors have to be considered to analyze their growth and their presence in electronic devices. We know that DRAM memories are present in stand alone packages in a personal computer for instance. Similarly, high density Flash memories are used in stand alone for data storage in camera or mobile phones whereas low density embedded Flash memories are present in microcontroller to store the embedded operating systems or application codes. Therefore, the memory market must be divided according to two main families, namely the stand alone and the embedded memory families. To illustrate the memory market growth, the stand alone Flash market represented close to 20 billions dollars in 2005. This market has grown faster than any other market in all the semi-conductor history due to the high demand of storage capability in mobile phones, MP3 players and cameras. Concerning the embedded memories market, the volumes are less important compared to stand alone memories but remains very important for semi-conductor actors who propose special embedded systems solutions to their customers. The interest for embedded memories is confirmed by the SIA Roadmap which forecasts a memory content approaching 94% of System on Chip (SoC) silicon area in the next years [SIA03].



Repartition of silicon area in a SoC – SIA forecasts

In addition, semi-conductor memories are considered as a vehicle for CMOS process technology development. Thus, advances in their fabrication, through the scaling for higher densities and faster speeds, is helpful for the performance establishment of other digital circuits. For all the above reasons, there is a need to develop accurate methods and solutions to design, manufacture, and test semiconductor memories. Design and manufacturing of semiconductor memories represent a consequent part of microelectronics research contributions whereas an effort must still be done in the field of memory testing [SIA06].

Considering the above context, the main focus of this thesis is on memory testing. A particular type of semiconductor memory called Flash has been investigated, and particularly Flash memories embedded in complex System-on-Chip (SoC) like a MCU (Microcontroller Unit) or an ASIC (Application Specific Integrated Circuit). This type of embedded memories has the particularity to be non volatile and thus to keep its memory content without any supply source.

Nowadays, a large portion of the SoC area is dedicated to memory circuits. Embedded Flash (eFlash) memories are very used in SoC due to their flexibility that allows to program and erase memory content electronically. Consequently, the test of embedded Flash memories is becoming a very important step in SoC testing. The high integration density of eFlash memories and their particular manufacturing process steps based on the floating gate principle make them more and more prone to inter or intra core cell defects. In addition to its high integration density, the non-volatile nature of eFlash memory requires a high electric field during the programming mode. According to its sensitivity, such high electric field may disturb the content of an eFlash core cell and in some cases may lead a logic flip.

Tests currently used for eFlash validation are very time consuming and the fault tested are not always realistic. Moreover, the most known solutions currently used to test RAM memories, which focus on the detection of static faults, like Stuck-At Faults, Transition Faults or Coupling Faults, are not applicable to test eFlash memories due to the slow programming time of such memories.

Therefore, the objective of this thesis has been to develop new and efficient solutions for testing most of actual defects occurring in eFlash memories.

This objective has required the initial study of actual defects and related failure mechanisms occurring in eFlash memories and particularly in eFlash memories built with FloTOx (Floating gate Tunnel Oxide) core cells.

This thesis is organized as follows:

Chapter 1 is dedicated to the presentation of the various types of memories existing today and gives details about technologies and design of these components. Volatile memories corresponding to the RAM family are simply described whereas Flash and EEPROM memories are more detailed. Some comparisons between non-volatile memories with respect to criteria like size, speed or reliability, are carried out. Finally, an overview of our work environment based on embedded Flash memories built with FloTOx core cells is shown.

Chapter 2 provides a general background of memory fault modeling and testing. Conventional RAM and eFlash test approaches are compared and the eFlash testing specificities are highlighted. With the help of a home made fault simulation tool, fault coverage measures obtained with eFlash test solutions, currently used in industry, are provided. These measures reveal the need to develop new and efficient test solutions for current eFlash memories.

In Chapter 3, we analyze actual defects and related failure mechanisms occurring in eFlash memories and especially in the core cell array. Then, we explain the interest of developing a SPICE-like model for faulty behavior prediction and how to develop such model. Next, with the help of our FloTOx SPICE-like model and by performing electrical simulations, a classification and a modeling of the faulty behaviors resulting from actual defects in eFlash are provided. We then discuss the test solutions that can be used for these defects and conclude on the need to provide dedicated test solutions for oxide thickness variations and address decoder faults.

In Chapter 4, the case study of a hard to detect actual defect and the related test strategy are proposed. This defect relates to a variation of the oxide thickness in the floating gate transistor performing the memory action. Such a defective mechanism has been modeled as a State Coupling Fault. The solution to detect such defect uses the coupling phenomenon existing between two adjacent bit lines and is based the application of two patterns (CKB and CKBI) to sensitize and observe the fault.

In Chapter 5, a particular fault model called Address decoder Fault (AF) is studied. The AFs are the most constraining faults in an eFLash test flow because their detection is very time consuming. Once again, a test solution based on the eFlash specificities is proposed to reduce the number of programming operations and to considerably reduce the total testing time. Moreover, we explain the advantages of our proposed solution to detect others fault models that are currently tested with others functional test flow. A comparison with a typical eFlash test flow is provided and shows significant improvement in test time and efficiency.

This work has been carried out in collaboration with ATMEL (Rousset), a company specialized in ASIC, MCU and embedded Flash memory designs, under the framework of a CIFRE contract. This partnership has allowed an industrial validation of the results obtained in this thesis. This work has also been validated by several publications in international conferences specialized in the test domain.

CHAPTER 1: IN THE FIELD OF NON-VOLATILE MEMORIES

This first chapter is dedicated to the volatile and non-volatile memory descriptions. First, the available and most often used semiconductors memories are described. Volatile memories (SRAM, DRAM…) are quickly described whereas non-volatile memories (FLASH, EEPROM…) are treated in more details. According to functionality criteria, we compare the different types of non-volatile memories. This is a hot topic of interest due to the high efforts carried out by industry and academia to develop a low cost (small size and reduced process steps) and reliable (high endurance and infinite retention time) non-volatile memory technology. On another hand, this non-volatile memory technology must be adapted for fast applications (low programming time) and be CMOS process compliant. Nowadays, almost all memories are able to fit with few of all previous criteria but some efforts remain. Then, the chapter is concluded by a top down overview of our work environment based on embedded Flash memories built with FloTOx (Floating gate Tunnel Oxide) core cells [YAR82].

1.1 SEMICONDUCTOR MEMORIES: STATE OF THE ART

Since the arrival of the first integrated circuits in 1959, a lot of memories have been developed. This wide panel of available memories is due to the high demand of storage in a lot of applications and systems. In addition, according to the system or to the application, several specificities to store data are needed. Sometimes, the data retention and the current consumption must be considered but sometimes the target is the memory core cells size or the access time. There are a lot of existing different memories proposed by researcher. Note that the solution of a given type of memory for a given application is performed with respect to the system design specifications.

Semiconductors memories can be divided in two families, the volatile memories RAM (Random Access Memories) and the non-volatile memories based on the ROM concept (Read Only Memories).

1.1.1 Volatile Memories

Volatile memories (RAM) have the particularity to lose their memory content (logic data) when they are disconnected from the supply voltage. In the RAM family, there are two kinds of memory, namely the Static RAM (SRAM) and the Dynamic RAM (DRAM).

1.1.1.1 Sram

Compared to other semiconductor memories, the SRAM is the fastest one but it remains the most expensive to produce. The SRAM is designed with the well known six transistors structure (inverters loop with select transistors) and keeps its memory cell content as long as a supply voltage is provided to these six transistors. Nevertheless, the logic data stored in this memory does not require refresh mode compared to a DRAM for example. In almost all cases, the SRAM is used as a shadow or a cash memory, where the requirements are to have a fast but small size memory.

1.1.1.2 Dram

The DRAM is built with one MOS transistor and one capacitance (Figure1.1). To program the DRAM memory cell with a logic ‘1’ or a logic ‘0’, the desired logic value must be applied on the bit-line and an activation voltage has to be applied on the word-line. Thanks to the word-line activation, the MOS transistor is ‘on’ and the capacitance is loaded with the desired logic data, either ‘1’ or ‘0’. To read the memory cell content, the MOS transistor must be activated and the capacitance voltage must be directly applied to the bit-line node. With a specific peripheral circuitry called *Sense Amplifier*, the capacitance potential is interpreted as a logic ‘1’ or a logic ‘0’.

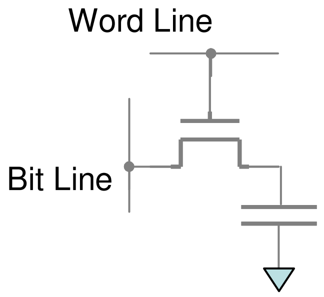


Figure 1.1: 1T-DRAM.

This DRAM structure is a low cost solution due to its small size allowing a high integration density. However the DRAM memory cell features two problems. The first one is due to the read principle during which the capacitance is discharged, and thus leading to the memory content loss. To avoid this problem, designers use particular circuitry to rewrite the logic content in the DRAM cell after the read operations. The second problem is related to the capacitance leakage involving once again the loss of the memory cell content. Thanks to a periodic data refresh system, this problem is avoided.

1.1.2 Non-volatile Memories

Unlike to volatile memories, the main advantage of non-volatile memories is the long storage duration (up to 10 years) of logical data even without supply energy. In most cases, non-volatile memories are evaluated according to their technology limitations, to the customer demands and economical constraints. In this section, the major non-volatile memories, that are the most often used in recent systems and applications, are described. There are four main kinds of non-volatile memories, namely the Read Only Memories (ROM), the Programmable ROM (PROM), the Erasable PROM (EPROM) and the Electrical EPROM (EEPROM). Note that Flash memories belong to the EEPROM memories family and their difference is based on their data access granularity.

1.1.2.1 Rom

The ROM memory is accessed only for read operations. There are two kinds of ROM technologies, the pre-diffused ROM and the programmable ROM. The ROM is programmed either during the production process or by the user thanks to fuse-based structures. This memory is often used to store microinstruction codes data in microprocessors or in microcontrollers but also to store user application codes in OTP (One Time Programming) microcontroller.

1.1.2.2 EPROM

The EPROM is electrically written and is erased by UV beam thanks to a special window present on its packaging. The duration of the write operation takes 100µs to 1ms whereas the erase phase duration takes more than 20 minutes under UV beam [BRO98]. To write an EPROM core cell we use a particular programming mechanism called Hot Electron Injection (HEI) [SHA97] through the oxide between the floating gate and the channel of the EPROM core cell. Note that the EPROM core cell is only built with a double gate transistor represented in Figure 1.2. Thanks to this property, an EPROM has the same integration capability as a DRAM [MAE89].

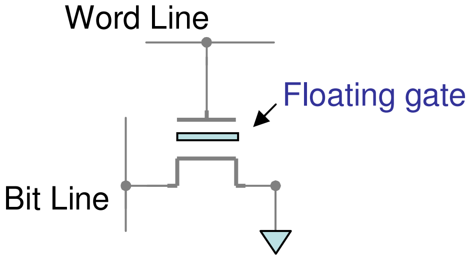


Figure 1.2: Scheme of EPROM core cell

1.1.2.3 EEPROM

The main drawback of the EPROM is the erase duration that takes close to 20 minutes under UV beam. To avoid this problem there is a particular memory core cell called EEPROM which is based on the EPROM principle but that can be written and erased electrically.

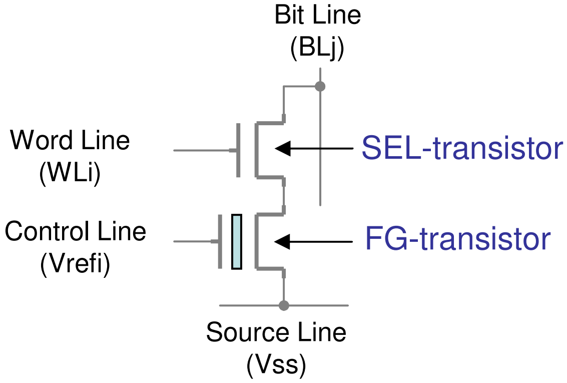


Figure 1.3: EEPROM core cell.

Figure 1.3 presents an EEPROM core cell composed of two serial transistors. One of the two transistors is the select transistor (SEL-transistor) allowing to select the core cell from the bit-line. Note that this transistor is built with a standard NMOS transistor. The second one is the floating gate transistor (FG-transistor) that performs the memorization of the logic data [YAR82].

Unlike to the EPROM which is built with a single transistor, the EEPROM needs more area due to the presence of two transistors. Consequently, this memory is never chosen for applications that require high integration density.

1.1.2.4 FLASH

To deal with Flash memories is not an easy talk because describing this kind of memory requires to consider several parameters. Flash memories are not linked to a technology but to a structure and more precisely to a granularity. In fact, the name Flash comes from the particularity of such memory to erase a block or a page in one time. However, in almost all cases the Flash memory is built with a particular core cell technology based on a floating gate transistor. This core cell technology offers the advantage to a reduced area and can be compared to the EPROM scalability. To write a logic content in this floating gate transistor, the HEI phenomenon is used whereas to erase this core cell, the tunneling effect called Fowler-Nordheim [SHA97] is preferred. This typical Flash core cell is presented in Figure 1.4 during the write and erase operations.

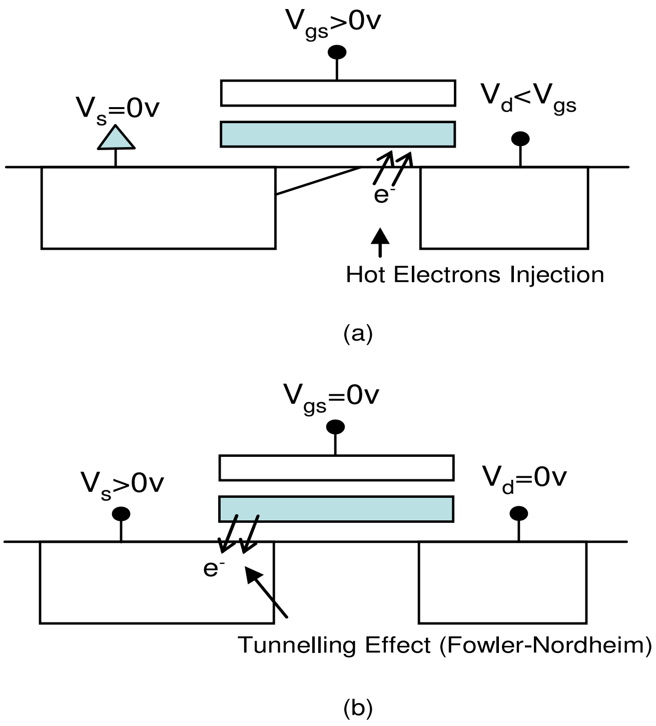


Figure 1.4: Flash core cell written by HEI (a) and erased by Fowler-Nordheim tunneling effect (b).

Note that the Flash memory can also be built with the same core cell as the EEPROM (2 transistors) and this particular case is used for instance to design Flash memories for embedded systems.

1.1.3 Future Non-volatile memories

Researchers try to develop a new non-volatile memory technology taking into account all advantages from the previous described non-volatile memories (data retention, endurance …) but also from the volatile ones (high integration density, fast write speed …). Two types of memory technology seem have many of these advantages, namely the FeRAM for Ferro-electric RAM and the Phase Change RAM also called Ovonic Unified Memory (OUM).

1.1.3.1 Phase Change Memories

The phase change memories (PCRAM) or OUM are one of the most interesting future non-volatile memories. The PCRAM uses the particular behavior of chalcogenide glass that can switch between two states (crystalline or amorphous) thanks to a heat application. The chalcogenide [YAM91] belongs to the same material family than that used to build re-writable optical media as CD-RW or DVD-RW. In optical media disc, a low power laser beam focuses the media to locally heat its surface and switch the material from a crystalline to an amorphous state. This phase change from a crystalline to an amorphous state corresponds to a binary data. The difference between these two states is sensed in an optical way based on the reflectivity behavior of the memory point which is either written or erased. In the PCRAM the principle of erasing and writing the core cell is different because the logic data corresponding to the material state is obtained electrically. In fact, the PCRAM undergoes a phase change (crystalline to amorphous) due to a current passing through the material and involving a joule effect. This kind of core cell is represented by a cross-section scheme in Figure 1.5. Before the joule effect, the material has a low resistance whereas after the joule effect, the material has a high resistance value due to its amorphous state. To switch the material from one state to the other, it takes less than 30ns and some prototypes built by Samsung Company achieve a phase change speed close to 5ns.

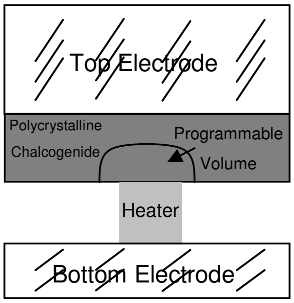


Figure 1.5: Cross section of a PCRAM core cell.

To forecast the endurance of a PCRAM core cell a lot of cycling tests have been carried out by researchers and have shown that after 10^{12} cycles the resistance value of the material is almost unchanged. Note that this endurance has never been achieved with other non-volatile memories. Since 2001, research teams develop phase change memories prototypes [LAI01] [GIL02] and they are facing two main problems which are the programming current reduction and the design of high density array. Currently, a few companies begin to design PCRAM array containing more than 512Mbits.

1.1.3.2 FeRAM

The Ferroelectric RAM (FeRAM) cell is designed with the same elements as those widely used DRAM cells, namely the access transistor and the cell capacitor. Note that this cell structure is called 1T-1C. In a DRAM, the cell capacitor is built with a linear dielectric whereas in FeRAM, the cell capacitor is based on a dielectric structure including ferroelectric material, typically *Peroveskite Zirconate Titanate* (PZT). Figure 1.6 describes the FeRAM core cell.

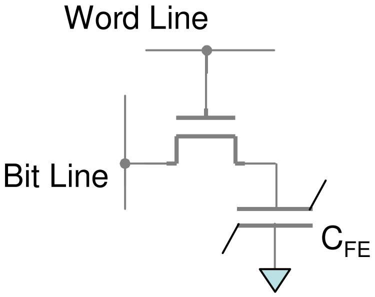


Figure 1.6: 1T-C FeRAM core cell.

Operationally, a FeRAM is similar to a DRAM. To write the FeRAM cell, an electric field is applied across the ferroelectric layer by charging the plates on each side of it. The presence of the electric field forces the atoms inside into a *up* or *down* orientation according to the polarity of the charge applied and thereby a binary information can be stored, a logic ‘1’ or ‘0’. Reading the

FeRAM cell is different than reading a DRAM cell because thanks to the access transistor, the FeRAM cell is forced into a particular state, called '0'. If the FeRAM cell already contains a logic '0', nothing occurs at the output lines. If the cell contains a logic '1', the re-orientation of the atoms in the material causes a brief pulse of current through the output line as they push electrons out of the metal on the down side. The presence of this pulse is sensed and interpreted as a logic '1'. Moreover, the read operation has to be considered as aggressive because after this operation, if the FeRAM cell contains a logic '1', this value will be over-written and replaced by a logic '0'. To repair the read operation the FeRAM cell rewrite to a logic '1' has to be performed. However, note that in a FeRAM memory, write and read operations are quite similar to DRAM. In a near future, due to its high integration density, to its fast write and read speed but also to its high endurance and data retention characteristics, the FeRAM could be very interesting.

1.2 NON-VOLATILE MEMORY TECHNOLOGY

The non-volatile memory principle is to keep data information without any electrical supply voltage. Researchers have found a lot of solutions to develop non-volatile memory core cells that are compliant with CMOS technology. The most known solutions are based on electrical charges storage for modulating the electrical characteristics of a custom transistor such as its threshold voltage value. For instance, if two different threshold voltage values are reached in a transistor two different logic states are obtained, a logic '1' or a logic '0'. Then, thanks to a particular polarization these two states are sensed in an electrical way. Currently, there are two charge storage technologies for non volatile memories [BRO98], the floating gate technology and the charges trap technology. In the near future, other type of storage technology for non-volatile memory, like the ferroelectric technology mentioned in the previous section, will be possibly used in production.

1.2.1 FLOATING GATE TECHNOLOGY

The floating gate principle is to catch electrical charges into an isolated gate of a double gate MOS transistor. Note that this isolated gate, called floating gate, is built with a conductor or semi-conductor. This structure is represented in Figure 1.7. With the stored electrical charges, the threshold voltage value of the MOS transistor is modulated. In almost all cases, charges are injected through a dielectric built with an oxide like SiO_2. This oxide is located between the floating gate and the channel of the double gate transistor. A second gate, called control gate, is placed under the floating gate and corresponds to the basic gate of a MOS transistor. These two gates are separated by a thick oxide layer called ONO (Oxide-Nitride-Oxide) [BRO98]. Programming a memory core cell built with the floating technology can be done by using two types of charges injection. The first one is the Hot Electron Injection (HEI) used by the technology SIMOS (Stacked gate Injection MOS) often used to build EPROM. In this technology SIMOS, the floating gate and the control gate are built in polysilicon [ROS77]. The floating gate of this SIMOS cell is charged by HEI and the charges are removed by performing UV beam insulation. During an HEI operation there is a consequent current consumption due to the conduction state of the SIMOS core cell. Moreover, the efficiency of the programming operation by HEI is related to the doping quantity present in the bulk, to the channel length and to the overlap size between the floating gate and the drain diffusion [BRO98].

The second mechanism allowing to program a floating gate core cell is the Fowler-Nordheim tunneling effect. This is a quantum phenomenon that allows injecting charges in the floating gate through a thin oxide of about 8 to 10nm. This programming mechanism is used for the specific FloTOx core cell (Floating gate Tunneling Oxide) in EEPROM. Note that a mixed solution is used to program ETOX (EPROM Tunnel Oxide) core cells used to build Flash memories. In this case, a HEI is performed to erase the core cells and a Fowler-Nordheim tunneling effect is performed to write the core cells.

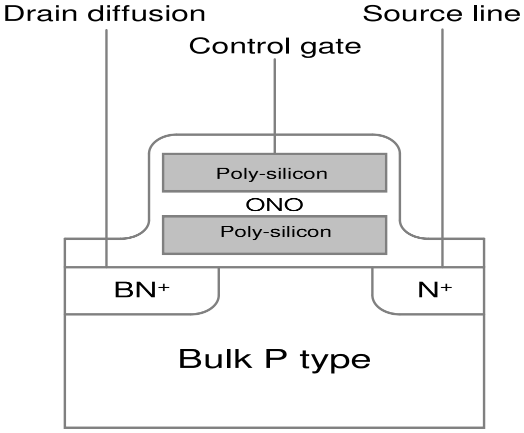


Figure 1.7: Cross section of a basic floating gate structure

Figure 1.8 details a FloTOx structure in which the Fowler-Nordheim principle is used to inject or remove charges from and into the floating gate. This quantum charge injection principle was used initially in a non-volatile RAM core cell [HAR78] and was the first step before the development of the EEPROM at the beginning of 80's. The FloTOx technology can be either double or simple polysilicon but the most often used in EEPROM is the double polysilicon technology.

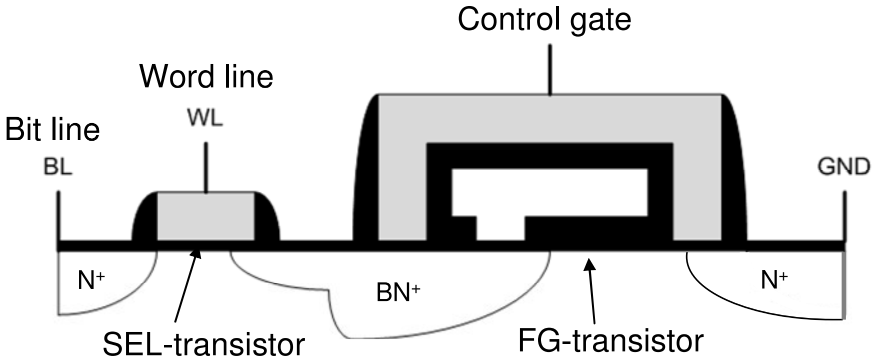


Figure 1.8: FloTOx core cell with double polysilicon technology.

In the FloTOx structure built with double polysilicon technology the second polysilicon gate, called control gate, is above the floating gate and separated from this one by an oxide called ONO (Oxide Nitride Oxide). Whatever the technology used to build FloTOx core cells, there is always a tunnel window in which the oxide is thinner (8 to 10nm) in the oxide between the floating gate and the channel. Electrons go through this thin oxide layer tanks to the tunneling effect when an electric field is applied between the floating gate and the drain diffusion. According to the value of the electric field, electrons are injected or removed into and from the floating gate. Hence, this charge quantity increases or decreases the threshold voltage value of the FloTOx core cell.

1.2.2 CHARGE TRAP TECHNOLOGY

In this section, the non-volatile memory technology based on the charge traps principle in insulator is described. There are two main memories built with this particular technology namely the MNOS and SONOS memories [SHA97].

1.2.2.1 MNOS TECHNOLOGY

Concerning the MNOS (Metal Nitride Oxide Semiconductor), instead of storing electrical charges in a floating gate, charge traps situated in a nitride layer are used. The electrons are injected through a thin oxide layer of about 1.5nm to 3nm thickness thanks to a tunneling effect. Once the electrons are trapped, the threshold voltage value of the MNOS core cell is modified.

Figure 1.9 describes an EAROM core cell (Electrically Alterable semiconductor ROM) in MNOS technology. Like basic transistors, this EAROM core cell has a control gate in metal. The charges are injected through the central oxide layer between the channel and the nitride layer Si_3N_4. However, this kind of non-volatile memory core cell features some drawbacks, such as its low speed, its limited density and the using of several electric potentials to run it.

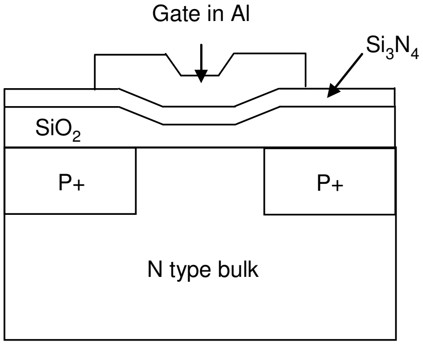


Figure 1.9: EAROM core cell based on MNOS structure.

In the 80's, the MNOS technology has been improved with the SNOS technology (Silicon Nitride Oxide Silicon) based on the LPCVD (Low-Pressure Chemical Vapor Deposition) method that stacks the polysilicon on the nitride layer and the pre-metallization by an hydrogen annealing at a high temperature to enhance the nitride to thin oxide interface [BRO98].

1.2.2.2 SONOS TECHNOLOGY

The SONOS (Silicon Oxide Nitride Oxide Silicon) technology has been developed in order to reduce the charge injection from the gate to the nitride compared to the MNOS technology. An oxide layer of about 2nm to 3nm is introduced between the nitride and the polysilicon gate [CHE77]. This oxide layer is obtained by nitride oxidation, thus reducing its thickness. However, this nitride thickness reduction involves a hole leakage to the gate. In fact, as the trapping length in the nitride is higher for holes (15 to 20nm) than for electrons (5 to 10nm), reducing the nitride thickness (<20 nm) makes that the holes are trapped at the nitride to gate interface whereas the electrons are caught in the nitride. Several holes are lost through the gate even in presence of a thin oxide layer between the nitride and the gate. This involves a decrease of the threshold voltage value.

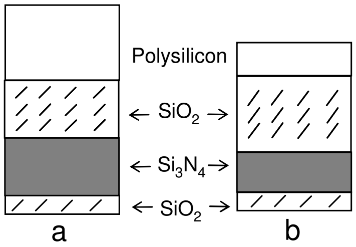


Figure 1.10: SONOS technology comparison between the old (a) and the new SONOS concept (b).

To avoid this trapped holes leakage through the gate, there is a new concept of SONOS structure keeping the same specificities [SUZ83] [CHA87] [DEL87]. This new concept of the SONOS structure layers is presented in the right part of the Figure 1.10. This new structure is thus composed by a thin oxide layer, a thin nitride layer (<10nm) and a thicker oxide layer (>3nm) [SUZ83] [DEL87]. These new characteristics allow to considerably reduce the SONOS structure thickness (Figure 1.10) and thus the necessary programming voltage (<5V). Moreover, the retention and the structure reliability are enhanced.

1.3 EMBEDDED FLASH (EFLASH) MEMORIES BUILT WITH FLOTOX CORE CELLS

1.3.1 EFLASH MEMORIES ARCHITECTURE

The functional scheme of an eFlash memory is presented in Figure 1.11. Like all other memories, the eFlash is composed of a core cell array (CORE), data latches (DLATCH), bit-line and word-line decoders and sense amplifiers. In addition, eFlash memories need two particular building blocks to perform dedicated functions: a Charge Pump device for the High Voltage Generation (HVG) allowing the write and erase operations and a Sense Reference Voltage used during the read operation.

Two types of core cell array can be used to realize an eFlash memory; NOR and NAND-based structures [SDC91]. Here, only the NOR-based structure is presented as this is the most often used structure in high-speed applications. In a NOR-based eFlash, core cells are placed in parallel as shown in Figure 1.12. A word-line (WLi) is shared by all the cells in the row. All the cells of one row are addressed together to form a page.

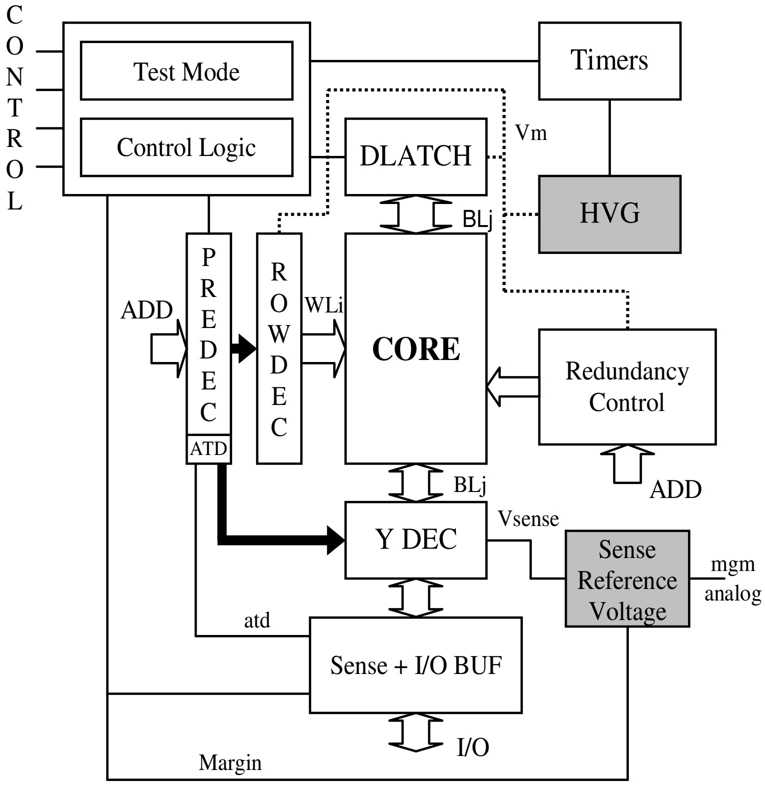


Figure 1.11: Functional scheme of eFlash memories

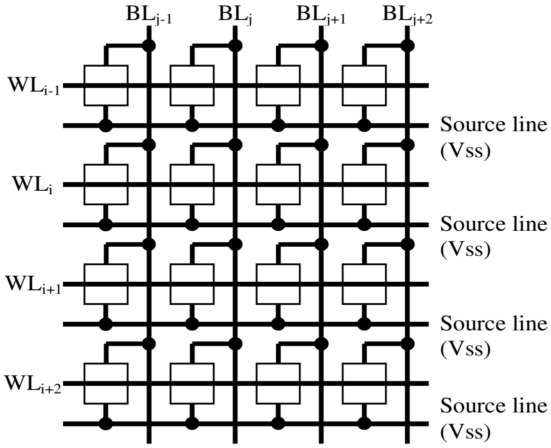


Figure 1.12: NOR eFlash structure

The eFlash core cell is based on the floating gate (FG) concept. As discussed previously, there are two typical mechanisms to transfer electric charges from and into the FG: hot carrier injection (HCI) [SHA97] and the Fowler-Nordheim (FN) tunneling effect [SHA97]. The FN tunneling effect is used for charge injection or removal in FloTOx core cells. In our work, we consider the FloTOx core cell structure presented in Figure 1.13 [SDC91] [SHA97]. The memory cell is composed of a select transistor (SEL-transistor) and a floating gate transistor (FG-transistor). The SEL-transistor having the word line connection (WL) allows the selection of the targeted cell. It also avoids disturbances from the high voltage on the bit-line when the cell is not selected. The FG- transistor contains the floating gate (FG) and the control gate.

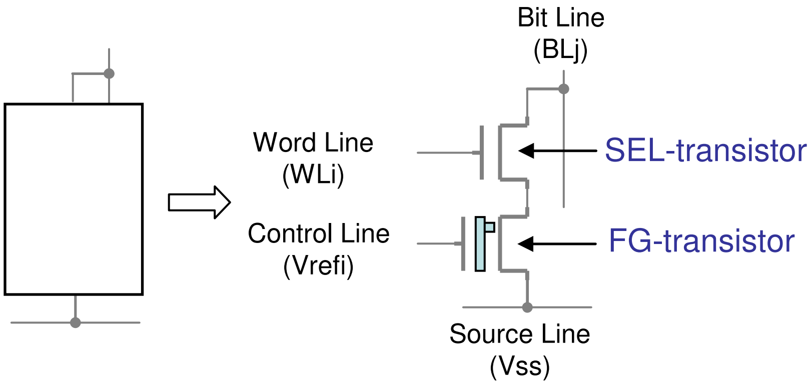


Figure 1.13: 2T-FloTOx core cell

1.3.2 EFLASH MEMORY FUNCTIONING

Three different operations can be performed on an eFlash: Erase, Write and Read. In the following, the cell(i,j) in the NOR-based array in Figure 1.12 is considered as the target cell. The different voltage levels required for the erase, write and read operations of cell(i,j) are reported in Table 1.1. The Erase operation consists in injecting charges in the FG with a specific high voltage combination. To inject charges in the FG, the high voltage must be applied on the Vrefi node of the sense transistor while its drain must be maintained at ground. During the Erase operation, the core cell is 'on' and allows the node BLj to be pulled-down at the Vss potential.

	ERASE	WRITE	READ
BLj	0v	Vpp	1v
BLx	0v	HZ	HZ
WLi	15v	15v	3.3v
WLy	0v	0v	0v
Vss	0v	1.2v	0v
Vrefi	Vpp	0v	0.7v
Vrefy	HZ	HZ	HZ

with x≠j and y≠I Vpp ≈ 12.5v

Table 1.1: Voltage levels for Erase, Write and Read operations

At this point, it is important to notice that the erase operation is performed simultaneously on all the cells of the same page, equivalent to all core cells belonging to a word line, and not cell by cell. At the end of the erase operation, charges in the FG have changed the VT of the sense transistor to a high VT (VTH in Figure 1.14). From a functional point of view VTH corresponds to logic '1'.

The Write operation consists in removing charges from the FG by putting the Vref node at ground while maintaining BLj at a high voltage (see Table 1). With this operation, charges of the FG are removed and so, the sense transistor has a low VT (VTL in Figure 1.14) which corresponds to logic '0'. The difference between VTH and VTL is called VT window.

As for all other kinds of memory, the read operation is performed by a sense amplifier but this time working in a current measurement mode. The Vref node is set around 0.7v during the read operation (see Table 1.1). If the sense transistor has a low VT (VTL), it delivers a current (between 10µA to 30µA) and the sense amplifier provides a logic '0' on its output. On the other hand, with the same Vref value, if the sense transistor has a high VT (VTH) there is no current through the bit-line and hence the sense amplifier gives a logic '1'.

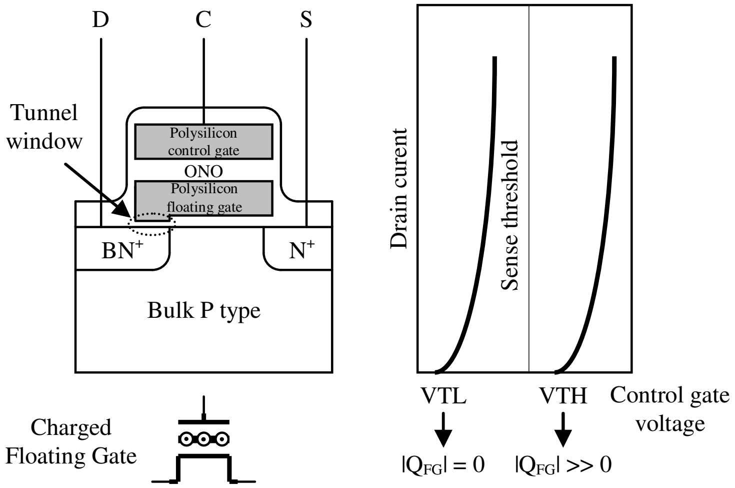


Figure 1.14: VT modulation with charge quantity

1.4 CONCLUSION

In this first chapter an overview of volatile and non-volatile memories has been presented. All these memories have several specificities and characteristics and allow to cover a large panel of electronic systems and applications. To summarize this overview, Table 1.2 presents a summary of embedded RAM and embedded Flash memories performances and, in Table 1.3 there is a characteristics comparison between actual Flash memories and future non volatile memories.

The unique memory able to be used in all of electronics systems does not exist yet and researcher work hardly in this field. Moreover, this chapter has introduced the context of our work which is related to embedded Flash memories (eFlash) built with FloTOx core cells. Due to its low access time, low power consumption and to its high density, this kind of non-volatile embedded memory is popular for portable devices. However, the high integration density of the eFlash memories and their particular process steps make them more and more prone to inter or intra core cell defects that will be analyzed in the next chapter.

	eDRAM	**eSRAM**	**eFLASH**
Cell size	25-30F^2	120-140F^2	15-30F^2
Extra masks	+5	0	+10-12
Read Speed	Up to 200MHz (4Mbits&90nm)	Up to 1GHz (512Kbits&90nm)	Up to 50MHz (4Mbits&90nm)
Comments	Large amounts of cache to justify process cost Large overhead	Leakage issue requiring process options (VT, Tox) for potable applications	Low speed programming
Potential issues	Leakage Stack capacitance value	Leakage SNM Soft errors: SEU	Tunnel oxide L_{eff} scaling with HEI Gate coupling
Solutions to push the limits	High K materials MIM cap ECC	Design techniques ECC materials	3D structures High K materials Design techniques ECC
ITRS prospects (2015)	Cell size: $0.0038\mu^2$ Up to 10Gbits	Cell size: $0.15\mu^2$	Cell size: $0.013\mu^2$

Table 1.2: Summary of embedded RAM and Flash performances

	Flash	**MRAM**	**FeRAM**	**PCRAM**
Current status				
Cell size	2.3-40F^2	20-60F^2	15-200F^2	12-40F^2
Write time	µs	<100ns	<100ns	<100ns
Mask count(*)	+5-10	+2-4	+2-3	+5-6
Endurance	10^5-10^6	>10^{14}	10^{10}	10^{12}
Maturity	130nm Volume Prod	180nm Test chips	Limited prod 0.35um	Test chips
Scalability	FAIR Down to 45nm	??	??	Good
R&D	SONOS Nanocristal Improve scalability and write time	TAS CIMS Improve scalability	1T,1C stacked cell, 2D-3D FeCAP Improve scalability	Decrease reset current Materials

(*) depends on technology and high voltage requirements, does not reflect automatically the process complexity (MTJ stack complexity on MRAM for example)

Table 1.3: Comparison between emergent non volatile memories and Flash

CHAPTER 2: GENERAL ASPECTS ON MEMORY FAULT MODELING AND TESTING

In the previous chapter, we have seen that stand alone memory market increases drastically and that embedded memory resources are continuously increasing and will approach 94% of the System on Chip (SoC) silicon area in the next ten years [SIA03]. These statements highlight the main issue that we are currently facing in the field of memory design and test. In fact, because of their high density, memories are considerably impacting test time and for the embedded ones they are becoming the main contributor of the overall SoC yield loss. Consequently, efficient test solutions and repair schemes for these memories are needed. This chapter proposes a summary of existing state of the art solutions for memory testing. Special interest will be given to eFlash memory testing.

2.1 BACKGROUND OF RAM FAULT MODELING AND TESTING

The high integration density of memories and their particular manufacturing process steps make them more and more prone to defects. Memory device failures can be classified according to three families [SHA97]:

- the infant mortality failures associated to designs errors or to process defects and to bad process steps during the production flow,
- the usage failures caused by a reliability missing, an electro-migration phenomenon or a perturbing event like a SEU (Single Event Upset),
- the wear-out failures due to operational conditions, to the aging effect of electrical contacts for instance or to mechanical stress during the device use.

To detect these failures in integrated circuits or systems, the use of comprehensive fault models is required and adequate stimuli to detect these faults have to be found. Note that fault models can be defined according to different abstraction levels as behavioral, functional, structural or electrical description levels. From a memory modeling point of view, the functional scheme of a RAM (Figure 2.1) can represent almost all semiconductor memories because the operations performed are always the same, i.e. writing a data in a core cell, storing a data or providing a data read-out from a random core cell location.

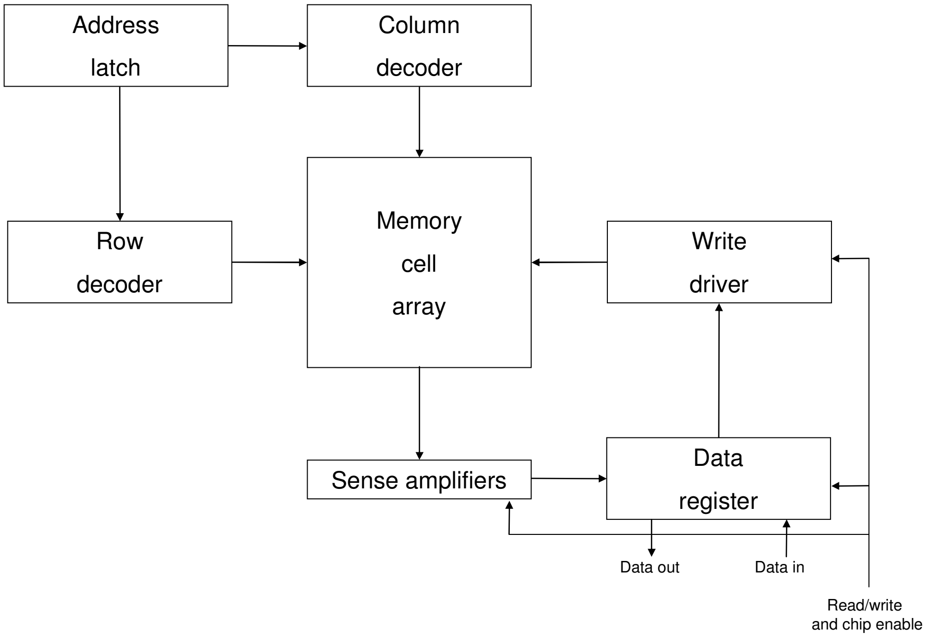


Figure 2.1: Functional scheme of a RAM chip.

2.1.1 RAM FAULT MODELS

In semiconductor memories and more especially for the RAM family, several types of physical faults can appear in a chip. Based on the general memory scheme described in Figure 2.1, we can expect to observe functional faults in different parts of a memory device:

- Cells stuck, a memory cell storing a binary value can be stuck to a logic ‘0’ or to a logic ‘1’.
- Read/write lines stuck, the lines driving the read or write signals can be stuck to a logic value ‘0’ or ‘1’ so that inhibits the read or write operations.
- Chip select line stuck, the signal ‘Chip select’ that switch ‘on’ the memory device can be stuck to a logic value and thus the memory chip can be selected or unselected every time.
- Wrong memory access, this fault induces a decoding problem in the address decoder logic and access problems occur during the write or read operation performed on victim memory cells.
- Etc …

This functional fault description is not enough to test precisely memory device like RAM and this is the reason why a logical fault modeling has to be performed. This modeling allows not only to test memories but also to examine precisely the physical causes of the fault based on a logical comparison with the known good dies. Note that a physical examination of the fault in a memory is not always easy due to the high cost of examination process and equipments.

In the RAM testing literature [VAN00] [ALA01], there are two main families of fault models. The first family relates to a static faulty behavior whereas the second relates to a dynamic faulty behavior.

* ***Static faults*** are sensitized with only one operation (read/write). The most known fault of this family is the simple *Stuck-At Fault* (SAF) model that can alter the content of a core cell. For instance, imagine that a core cell Cx is Stuck-At zero (SAF0). This means that irrespective of the operation acted on cell Cx, its contents will remain at ‘0’. To sensitize this fault, only one operation has to be applied, a write ‘1’ in the faulty core cell. In fact, at the end of the write ‘1’ operation, the core cell has not flipped from a logic ‘0’ to a logic ‘1’, as expected. To observe the result of the write ‘1’ operation, a read ‘1’ of the core cell Cx is needed and due to the SAF0 the expected ‘1’ will never be read.

This family includes some type of other fault with a faulty behavior that can be harder to detect. There are the Address decoder Faults (AFs) present in the address decoder circuitry of a memory chip. In this fault model several combinations of addresses are possible at the same time, involving multiple core cells to be addressed simultaneously. There are also fault models involving a unique core cell at a time, like the Stuck-At Faults (SAFs), the Stuck-Open Faults (SOFs), the Transition Faults (TFs) or the Data Retention Faults (DRFs). Then, there are the fault models describing a coupling phenomenon between two core cells, an aggressor and a victim. This sub-family includes the so called Coupling Faults (CFs).

Note that the static fault models can be more complex if we consider for example that two coupling faults with two different aggressor core cells can be linked and affect the same victim core cell. The faults that occur in this configuration are called *Linked Static Faults*.

* ***Dynamic faults*** are sensitized with at least two operations (read/write). These faults are harder to detect compared to the previous static faults because they appear after a particular sequence of read/write operations performed on the memory cells. These faults are currently among the main problematic in the field of RAM fault testing and a lot of research works have

been done to primarily address the problem of dynamic fault modeling [BOR05]. The well known dynamic fault models are the Address Decoder Open Faults (ADOFs), the dynamic Read Destructive Faults (dRDFs), the dynamic Data Retention Faults (dDRFs), the Un-Restored Write Faults (URWFs) and the Un-Restored Read Faults (URRFs).

To illustrate this particular fault family, the most known dynamic fault, called dynamic Read Destructive Fault (dRDF), is described. First of all let us consider the definition of dRDFs established by [VAN00]:

- *A cell is said to have a* ***dRDF*** *if a write operation immediately followed by a read operation performed on the cell changes the logic state of this cell and returns an incorrect value on the output.*

To well understand this particular fault, the analysis of what are their probable origins is required. In [DIL05] an exhaustive analysis of actual resistive open defect occurring in an SRAM core cell has been carried out. In this work, authors shown that a particular location of a resistive open defect in an SRAM core cell could impact the logical core cell behavior and can be modeled as a dRDF. Moreover, the author showed that according to the resistive open values, the number of read operations (read destructive) required to change the logic state of the cell, is different. This statement confirms that such resistive defect configuration involves a dynamic fault.

As explained previously, dynamic faults such as dRDF are harder to test than static faults due to their nature and particular behavior. They are currently the main problematic in RAM testing.

2.1.2 RAM TEST ALGORITHMS

In the previous sections the various fault models available for memories, especially for RAM. In this section we deal with methods to test these fault models. Remember that a memory is a particular chip having a large quantity of internal states related to its size, i.e. 2^n with n the number of bits in the memory. Because of time constraints, the test of all possible internal states of memory is not possible. Currently, memories achieve more than 1Gbits of storage capacity. For instance, with a $O(n^2)$ test procedure, a 4Mbits SRAM would be tested in 500 hours. Thus, based on their regular structure and on their functional fault models, researchers have developed new test methods and algorithms with a O(n) complexity.

The first test method with a O(n) complexity was just based on a unique pattern consisting in writing in all the memory chip. In this test category, there are the Checkerboard Pattern and its inverse used to test direct shorts between topological adjacent core cells. There are other patterns like Column Bars or Row Bars. Although simple to implement and test time advantageous, these patterns present a low fault coverage and only the SAF detection is guaranteed [VAN98a].

For these reasons researchers have developed new methods, called March tests, achieving a high coverage for SAF, TF, AF or CF. March algorithms have a low complexity (O(n)) and more flexibility thanks to their Degree of Freedom (DOF) [NIG98]. We assume the definition of a March test described by [SUK81]:

> ***A March test consists of a finite sequence of March elements. A March element is a finite sequence of operations applied to every cell in memory before proceeding to the next cell. The latter can be done in either one of two address orders: an increasing (⇑) address order (e.g. from address 0 to address n - 1), or a decreasing (⇓) address order which is the opposite of the ⇑ address order. When the address order is irrelevant the symbol ↕ is used. An operation can consist of: writing a '0' into a cell (w0), writing a '1' into a cell (w1), reading a cell with expected value '0' (r0), and reading a cell with expected value '1' (r1). Note that all operations of a March element are performed at a certain address, before proceeding to the next address.***

To illustrate the definition of March test algorithms, the well known March C- [MAR82] used by several semiconductor companies to test RAM is described. This is a 10N linear test, which is effective to detect Stuck-At Faults, Transition Faults and Coupling Faults. March C- has the structure shown in Figure 2.2.

$$\updownarrow(w0) ; \Uparrow(r0,w1) ; \Uparrow(r1,w0) ; \Downarrow(r0,w1) ; \Downarrow(r1,w0) ; \updownarrow(r0)$$

Figure 2.2: March C- structure

However, to test each kind of fault model described previously, a March test must satisfy certain conditions. For instance, to build a March test allowing to detect a Transition Fault <↑/1>, a particular sequence of predefined elements has to be implemented. Let us consider the state diagram presented in Figure 2.3 and Figure 2.4, the first diagram shows the good behavior of a memory cell whereas the second scheme describe a memory cell presenting a Transition Fault <↑/1>.

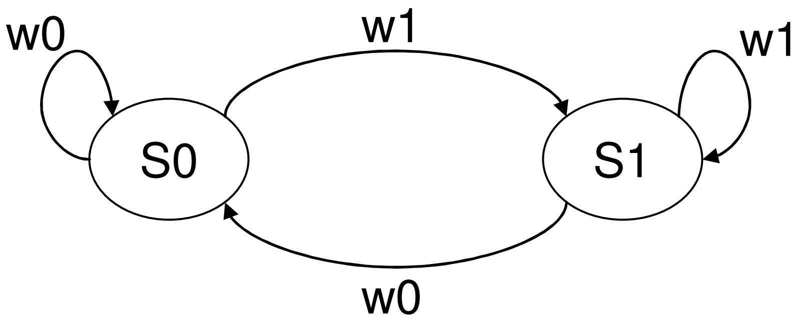


Figure 2.3: State diagram of a fault free SRAM core cell.

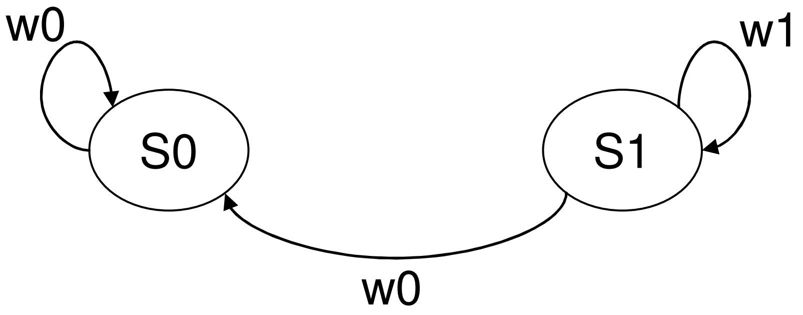


Figure 2.4: State diagram of a faulty SRAM cell presenting a TF.

By looking at these two state diagrams, we can easily develop the particular sequence to implement in the March algorithm in order to detect a TF <↑/1>:

- (…, w1,r1, …)

Indeed, the presence of a TF<↑/1> on a victim memory cell can be detected by performing a write ‘1’ on the victim cell immediately followed by a read ‘1’ operation of the same cell. So that all TF<↑/1> are detected, the element (…, w1,r1, …) must be in the March test algorithm. Note that the dots (…) correspond to any operation before and after the element “w1, r1” detecting the TF<↑/1>. For each fault model like Stuck-At, Coupling Fault, etc…, we have to reproduce the same reasoning as for the TF<↑/1> and find elements to implement March test algorithm.

Table 2.1, presents a list of four March algorithms with their respective fault coverage for SAF, AF, TF and CF. We have also given the test time if we apply each algorithm in a 1Mbits SRAM.

March Algorithms	**Fault Coverage**				**Test Time**	
	SAF	AF	TF	CF	Complexity	1Mbits SRAM
MATS+ ↕(w0) ; ⇑(r0,w1) ; ⇓(r1,w0)	100%	100%			5n	0.52s
MATS++ ↕(w0) ; ⇑(r0,w1) ; ⇓(r1,w0,r0)	100%	100%	100%		6n	0.63s
March C- ↕(w0) ; ⇑(r0,w1) ; ⇑(r1,w0) ; ⇓(r0,w1) ; ⇓(r1,w0) ; ↕(r0)	100%	100%	100%	100%	10n	1s
March LR ↕(w0) ; ⇓(r0,w1) ; ⇑(r1,w0,r0,w1) ; ⇑(r1,w0) ; ⇑(r0,w1,r1,w0) ; ⇑ (r0)	100%	100%	100%	100%	14n	1.5s

Table 2.1: Comparison of four different March algorithms.

2.2 BACKGROUND OF FLASH MEMORY FAULT MODELING AND TESTING

Exploiting their similarities with RAM, some fault models have initially been adapted to Flash memories [SHA97]. As there are structural differences between RAM and Flash memories, all RAM fault models are not realistic transposed in a Flash context. Only four main fault models can be found in the Flash test literature: the Stuck-At Fault model, the Transition Fault model, Address decoder Fault and State Coupling Fault. Note that certain Coupling Fault models like the Idempotent Coupling Fault model are not present due to the specificities of Flash memories allowing to avoid such faults. In this field, there is only few research works dealing with Flash fault modeling as well as Flash testing.

2.2.1 THE 7-PATTERNS TEST SEQUENCE

To test their Flash memories, semiconductor companies exploit several functional patterns which exercised the chip to detect any address sequencing and data pattern sensitivities. In the literature we often find the most known 7-patterns sequence allowing to test Flash memories (or EEPROM) like a Static RAM by performing the following programming (erase and write) and read operations:

- Write all 0s and Read all 0s – This pattern ensures the operability of the EEPROM or Flash memory page mode. Moreover this pattern preconditions the memory for the next pattern, i.e. GalPat0.
- GalPat0 (Galloping Pattern) – All the memory is set to “0” by the previous pattern.

Thanks to the GalPat0 the read disturbances is tested between byte or word by reading "00" alternately in byte or word sharing the same column or row.

- Bit Checkerboard –At first, a succession of '0' and '1' is written to obtain an opposite state between two adjacent core cells in the memory array. Then, there is a second pattern corresponding to the complement of the previous one. The Bit Checkerboard pattern ensures that every cell can be written to a '1' or '0' but also that the adjacent core cells which are written with an opposite state do not affect the state of the cell under test.
- Columns Pattern – This pattern writes alternating columns composed of a byte or a word with "00" and "FF" to precondition the memory for the next pattern.
- DRAD (Diagonal Read of Alternating Data) – During the DRAD, no write operations are performed but only read operations. Indeed, in this pattern only the bytes or word belonging to the memory array diagonal are read. This configuration is extreme because for each read cycle the address decoders change and the data read is the complement of the previous one. Moreover, this pattern is often used for AC parametric test because it corresponds to the worst case of read access time.
- Bit unique test – This pattern verifies that every bit within a byte is unique and ensures that two bits are not shorted
- March element – Based on a byte or word write operation granularity, this pattern writes a March algorithm element in the entire Flash or EEPROM memory chip. Considering the March LR algorithm, this March element pattern could be executed using the element $\Uparrow(r1,w0,r0,w1)$. However, as the Flash or EEPROM granularity is byte even word, the March element is performed with a data background "FF" or "00" instead of '1' and '0' respectively. The March element is used to test address uniqueness and multiple selections.

The functional Flash test based on a 7-patterns sequence is time consuming. To reduce test time, some special functions allowing the mass (parallel) chip erase/write modes have been implemented in Flash or EEPROM. However, these particular programming modes present some limitations in the detection of CF for example due to the limited number of combinations performed in one time. Moreover, due to the particular address decoder configuration and to the electrical features required during these parallel programming modes, some new failures mechanisms can appear. To avoid this type of problematic, accurate electrical characterizations and estimations of these modes have to be performed.

2.2.2 THE 5-STEPS TEST SEQUENCE

From a test point of view, because of the intrinsic very low speed of programming operations, Flash memories are very different from other types of memories. Only a very limited number of patterns can be used to test Flash memories in order to keep the testing cost acceptable. Assuming a Functional Write mode (FW) with 2ms to erase and 2ms to write, it takes 4ms to program a data in a word. Note that almost all Flash memories have also a Page Write mode (PW) allowing to program a page in the same duration than for a FW.

For instance, if we consider a 256 bytes per page architecture, 512 pages are necessary to build a 1Mbits memory. Using a Page Write (PW) mode, close to 2 seconds are mandatory to write one pattern to the 1Mb Flash. Consequently, programming a set of basic patterns such as “00”, “FF”, checkerboard and inverse checkerboard using PW mode will result in a testing time close to 10s per die. Decreasing the testing time per die is technology dependent, as parallel access to full or large portions of the array is possible to speed up the programming of test patterns. Programming a huge number of cells in parallel is only possible if a very low current programming mode such as FN tunneling is used. This is the case in embedded Flash (eFlash) built with FloTOx core cells. By executing dedicated test modes, a one time programming of large sectors to “00” or “FF” is possible in a few milliseconds. A Checkerboard pattern can also be programmed in a few ms using a Concurrent Chip Write Pattern (CCWP) that allows programming in one time the data loaded in the Flash page buffer in several memory blocks. For the Checkerboard pattern, the CCWP selects either odd pages or even pages to program in one time and reduces the global programming time. In any case, even if only one test pattern is programmed using the user mode (FW), testing time of medium to large Flash memories will be in the range of seconds, to compare with milliseconds in ROM testing.

In many companies, a 5-steps test sequence is typically used to test eFlash memories. From an algorithmic point view, the 5-steps testing flow is composed as follows:

- 1st step: W_{CCWP} CKI + Read CKI
- 2nd step: W_{FW} CKB + Read CKB
- 3rd step: W_{FW} Diag0 + Read Diag0

- 4th step: CE + Read all '1'
- 5th step: CW + Read all '0'

Note that CKB (CKI) stands for checkerboard (inverse checkerboard), and Diag0 is a diagonal pattern of '0'. W_{CCWP} is a test mode allowing high speed programming of the CKB (CKI). Chip Erase (CE) and Chip Write (CW) are specific test modes allowing one time programming of the full array to 'FF' and '00' respectively. W_{FW} is the Functional Write mode which corresponds to the user mode.

The fault coverage analysis of this 5-steps sequence is summarized in the last section of this chapter but we can anticipate that this kind of sequence can reach a good coverage to test Stuck-At Faults, Stuck-Open Faults, Addressing decoder Faults and Transition Faults. But due to the particular programming mode used during 5-steps test sequence, the difficulty to detect more complex faults can be easily imagined. Indeed, the writing of a large amount of cells in parallel does not allow testing particular coupling faults. For example, we can imagine that all combinations of State Coupling Faults (SCF) are not detected by the 5-steps test sequence. Now, if we consider the 1Mbits eFlash memory architecture cited above (256 bytes on 512 pages), this 5-steps sequence will take close to 5sec to test the whole memory chip. In comparison, an SRAM memory, with the same storage capacity, is tested by a 14n March LR algorithm in 1,5s.

Such test sequence will have an impact on the test quality and cost of eFlash memories. For instance if the previously detailed complex fault model (SCF) is considered, the 5-steps test sequence presents some limitations to detect this kind of fault. In this 5-steps test sequence, the patterns CKB and CKI have the ability to detect certain SCF but these patterns do not perform all possible SCF combinations in the matrix due to the particular programming modes used. In Figure 2.5, the application of the two first patterns (CKI and CKB) from the 5-steps test is presented in a small array of 3*3 core cells.

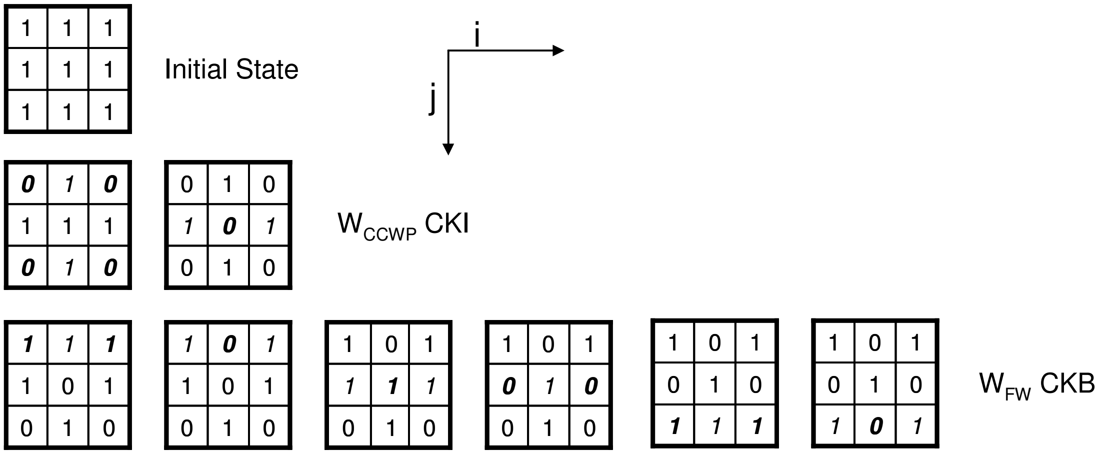


Figure 2.5: CKI and CKB patterns from the 5-steps test sequence.

In Figure 2.5, the pattern written between two steps is represented in italic whereas the bits changing are highlighted in bold. Note that the coordinates i and j give the position of the cell(i,j) in the matrix, the cell (0,0) is the cell at the top left corner. Considering for instance the first page of the memory, the SCF (0,0), in which the cell (0,0) is the aggressor and the cell (2,0) is the victim cell, is never tested. Indeed, along the two patterns execution, these two cells are always in the same logical state.

2.2.3 An eFlash Fault Simulator to Evaluate a Basic Test Flow

Due to the long programming time of an eFlash memory, an optimized solution or algorithm to test this kind of memory has to be chosen. For this purpose, a functional fault simulator has been developed to evaluate the quality degree of different sequences used to test actual fault models occurring in eFlash and the possibility to optimize the global test sequence. The quality degree of a test sequence is essentially given by the coverage of this sequence when a particular list of faults is considered. Due to the huge number of faults to consider, an assumption is made that only one fault at a time can occur at once during the fault simulation process. This simulator has been developed in C language and is composed of:

- A fault dictionary with the actual list of faults in eFlash described previously: SAF, SOF, TF, AF and SCF.
- A pattern dictionary containing the basic patterns present in the 5-steps test sequence detailed previously (CKB, CKI, Diag0, Diag1 …).
- A predetermined eFlash memory topology that can be modified according to the user requirements.

Moreover, the simulator presents the possibility to customize the patterns or the test sequences to apply to the eFlash memory in order to give more flexibility to the user. Now thanks to this simulator we are able to evaluate and analyze more precisely different sequences or algorithms used for eFlash memories testing.

With this fault simulator the 5-steps test sequence, presented in the previous subsection, has been analyzed for a predetermined list of fault models. This is summarized in Table 2.2 below.

Fault Models	Combinations	Coverage rate (%)
SAF	0	100
	1	100
SOF	0	100
	1	100
TF	↑	100
	↓	100
SCF	<0,0>	25
	<0,1>	100
	<1,0>	25
	<1,1>	25
AF	All combinations	100

Table 2.2: Fault coverage rates of the 5-steps test sequence.

In this table, all basic faults like SAFs, SOFs, TFs and AFs are tested by the 5-steps test flow. However, this table shows us that the weakness point of the 5-steps test sequence states in the SCF detection and more precisely for SCF configurations <0,0>, <1,0> and <1,1>. The bad detection of such faults is directly related to the concurrent methods used during pattern programming operations. In the next chapter a proposition to enhance the detection of possible SCF will be done based on eFlash electrical specificities.

2.3 March Algorithms Applied to eFlash Memory

In this section, the difficulties to transpose the SRAM or DRAM test methods in the eFlash context due to their technology differences are presented. First, RAM and Flash memories are compared. Then, a March algorithm applied in the Flash memory context is evaluated.

2.3.1 eFlash Compared to RAM

From a technological point of view, RAM and Flash memories are very different (see Chapter 1). As their core cells (memory points) are different, the failure mechanisms are different and the resulting fault models are quite different. Moreover, the use of a high voltage during the eFlash programming operations involves an increase of device stress, especially for their gate oxide, that does not occur in RAM. Concerning the coupling phenomena involved by the technology over scaling in RAMs, this occurs in Flash but the activation mechanisms are not the same due to the difference of electrical conditions between RAM and Flash. From a functional point of view, all coupling fault models available in RAM are not directly applicable in an eFlash context and some restrictions have to be done.

We also know that writing a Flash core cell takes much more time than writing a RAM core cell (about 2ms for a Flash memory compared to few ns for a RAM). This implies that fault models which are likely to occur have to be targeted first if the global test time of the memory chip has to be optimized. Thus, an exhaustive approach to test Flash memories is not suitable.

From an architectural point of view, we have seen that the functional scheme of a RAM corresponds to the one of an EEPROM or a Flash memory. Thus, the peripheral circuits of a Flash memory can be subject to the same functional faults than those of a RAM. However, an important issue to consider with the Flash architecture is the granularity. In fact, Flash memories are always word-oriented even page-oriented, compared to RAM that are bit-oriented or sometimes word-oriented. This means that the test strategy will consider inter-word or inter-page possible faults but also intra-word or intra-page possible faults.

2.3.2 EVALUATION OF MARCH ALGORITHMS IN EFLASH

Now, let us consider a particular fault model that can occur in a eFlash memory context and which is listed in the previous section. The State Coupling Fault (SCF) model is chosen. In the case of a SCF, one core cell Ci is called coupled (victim) if this core cell is forced to a logic value x when the aggressor cell Cj is in a particular state y [DEK90]. Here, the four possible SCF configurations between two core cells Cj and Ci are

- <1,0> - when the cell Cj contains a logic ‘1’ the cell Ci is Stuck-At a logic ‘0’.
- <1,1> - when the cell Cj contains a logic ‘1’ the cell Ci is Stuck-At a logic ‘1’.
- <0,1> - when the cell Cj contains a logic ‘0’ the cell Ci is Stuck-At a logic ‘1’.
- <0,0> - when the cell Cj contains a logic ‘0’ the cell Ci is Stuck-At a logic ‘0’.

From a test point of view, to detect all possible combinations for this kind of fault, the use of March algorithms [VAN96], that is the most often used in a RAM context, is mandatory. Previously we have seen that eFlash memories are always word-oriented or page-oriented , and in order to detect intra-word Coupling Fault, a ‘Bit Oriented Memory’ (BOM) March test could be converted to a ‘Word Oriented Memory’ (WOM) March test. This could be done by replacing the bit wide operations (`r0', `r1', `w0' and `w1') by operations reading and writing a data background of n bits [VAN98b]. In the case of SCF detection, the number of data background (NB_{DB1}) to apply to the memory has to be defined. In [VAN98b] the formula giving the number of data background to detect all SCF is the following:

$NB_{DB1} = 3 + 3*\log_2(B)$ *where B is the number of bits in a word* (1)

Note that a WOM March is the concatenation of the inter-word March test with the intra-word March test. This means that for each write or read operation in a March element, the values ‘0’ and ‘1’ are replaced by the different data background. Now, let us consider a Flash memory composed of 64K words of 32 bits. The memory has 1024 pages of 64 words composed of 32 bits. To detect all possible coupling faults occurring in a semiconductor memory, a general solution is given in [VAN98b] to modify any BOM March test in WOM March test. Finally the number of data backgrounds (NB_{DB2}) to generate is the following:

$NB_{DB2} = (10 + 6*\log_2(B))$ *where B is the number of bits in a word* (2)

To simplify our eFlash example, a global page can be considered as a word in order to save test time because writing a page takes the same duration as a word. Thus, if B corresponds to 2048 bits, according to the general expression (2) the number of data backgrounds to generate is 76. If we choose a March C-, described previously, to test our eFlash, we know that its length is 10N with N the number of bits in a memory and 10 corresponding to 5 read and 5 write operations. In an eFlash context due to its specificities, the read operation duration is always negligible compared to the write operation. Based on a page programming approach with a duration close to 4ms (erase + write), the test of our 64K words of 32 bits eFlash with the March C- will take: 4ms*1024*76*5=1556sec with 1024 the number of pages in the memory example. From this estimation, it is evident that March algorithms are not a cost effective solution to test Flash memories even if its coverage rate is 100% for all actual faults occurring in eFlash. In [YEH02], the authors have developed a solution called March-FT algorithm based on the March tests but oriented to Flash memories. Their solution, allowing the testing of almost all basic faults in Flash, requires only 2 write operations and 2 mass erasures of the memory chip. Once again because of the Flash granularity this approach is still unsuitable and compared to the March C- solution evaluated below, the testing time is just reduced by a factor 2.5.

2.4 Conclusion

Among the contributors to the eFlash memory development costs, testing cost must be carefully evaluated. This is mainly due to the intrinsic characteristics of the FG device, resulting in a very low speed programming operations. As a result, the eFlash testing problematic is the following: how to guarantee acceptable fault coverage with a limited number of programming operations?

First, the March algorithms have been evaluated in an eFlash context and the conclusion was the unfeasibility to use this kind of algorithms to test high density Flash even with particular minimizations of write operations [YEH02]. Secondly, to find optimal test solutions this requires a very good understanding of eFlash particular defects, resulting in a realistic list of faults, to be considered for the test pattern generation. These aspects will be illustrated in the next chapter by detailed descriptions of failure mechanisms related to an eFlash core cell matrix. The fault coverage based on a cost effective 5-steps test sequence has been provided for particular list of faults known in the literature and its weakness points have been highlighted.

CHAPTER 3: ANALYSIS AND MODELING OF ACTUAL EFLASH DEFECTS AND RELATED FAILURE MECHANISMS

In this chapter we first introduce actual defects and related failure mechanisms occurring in embedded Flash memories (eFlash) built with FloTOx core cells. The objective of such a study is to perform a functional fault modeling based on faulty behaviors involved by defects. During the functional fault modeling process, we need to perform electrical simulations to accurately describe a failure mechanism. So, the development of an electrical model of a FloTOx core cell is proposed and detailed in the second part of this chapter. Then, typical defect injections with electrical simulations are presented and a list of functional fault models is reported.

The last part of this chapter is dedicated to the detailed analysis of a defect that corresponds to the defectiveness of the tunnel window oxide. We show that this defective oxide thickness impacts erase or/and write operations as well as retention and reliability of FloTOx eFlash memories. A fault model and a solution to detect such a defective oxide are proposed. The detection method is based on the coupling phenomenon existing between bit lines.

3.1 QUALITATIVE ANALYSIS OF FAILURE MECHANISMS IN FLOTOX CORE CELLS EFLASH

In this part, actual failure mechanisms that may appear in eFlash memories built with FloTOx core cells are described. We begin with a presentation of the experimental conditions and then, possible defects that may appear in eFlash are reviewed.

3.1.1 EXPERIMENTAL SET-UP AND FAILURE CLASSIFICATION

The defect and failure mechanism analyses have been done on a 150nm FloTOx eFlash technology provided by a Semiconductor Company. Let us have a look to Figure 3.1 describing a 3*3 FloTOx core cells array with a NOR arrangement of core cells.

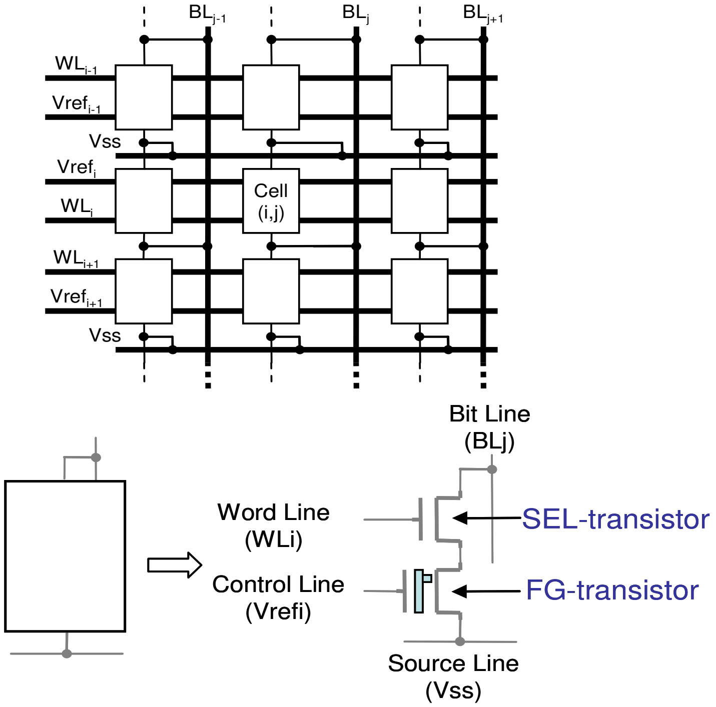


Figure 3.1: Small eFlash array (3*3) built with FloTOx core cells.

In Figure 3.1, the cell(i,j) is the cell corresponding to word line WL$_i$ and bit line BL$_j$. Note that in our experimental conditions, the cell(i,j) will be the only one on which write or read operations will be applied. For our analysis, the specific voltage values given in Table 3.1 for the Erase, Write and Read operations are used. For example, a Write operation on cell(i,j) requires 14v on WL$_i$ and 12v on BL$_j$, 1.2v on Vss and 0v on Vref$_i$. The other bit lines (BLx, x≠j) and Vref lines (Vrefy, y≠i) of the array must be set to high impedance (HZ) and the other word lines (WLy, y≠i) must be set to 0v.

	ERASE	WRITE	READ
BLj	0v	12v	1v
BLx	0v	HZ	HZ
WLi	14v	14v	3.3v
WLy	0v	0v	0v
Vss	0v	1.2v	0v
Vrefi	12v	0v	0.7v
Vrefy	HZ	HZ	HZ

Table 3.1: Experimental voltage conditions

Another important value is the VT window (VTW) defined as the difference between VTH and VTL (Chapter 1). In almost all eFlash memories the VTW value is close to 4v or 5v and the measurement of this value must be carried out to ensure good erasing and writing operations. All these important parameters have been used to analyze the functional impact of physical defects in eFlash. The defects we were looking for represent the most frequently encountered ones. These defects are divided in four classes according to their nature and their origin:

- **Hard defects:** pure open and short defects in the array.
- **Resistive defects:** resistive open and resistive short defects in the array.
- **Floating gate transistor defects:** oxide thickness variation or defectiveness.
- **Disturb phenomenon:** disturbances induced by the over scaling between two bit lines.

In the next subsections, each defect family is detailed and their possible faulty behavior are analyzed.

3.1.2 Hard Defect Related Failures

According to layout dependencies and defect probability related to the manufacturing process, open and short defects have been injected on the 3*3 NOR-based eFlash memory array as shown in Figure 3.2.

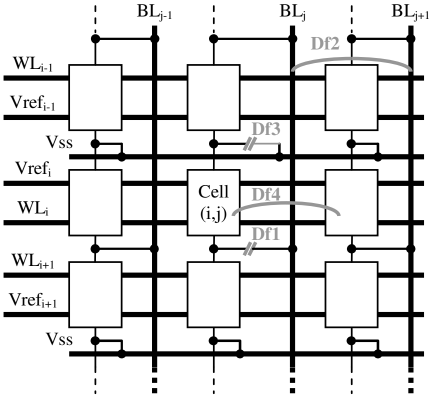


Figure 3.2: Hard defects in a NOR-based eFlash memory.

Df1 - Opened contact on the bit line

In the layout view, two adjacent core cells share the same bit line contact. With this contact opened (R = ∞Ω), the drain junction of two adjacent SEL-transistors are disconnected from the bit line. In presence of such a defect, the erase operation remains possible but the write cannot be performed. Moreover, during a read operation, whatever the cell data content, no current will be delivered and thus a logic ‘1’ will be observed for the two cells: cell(i,j) and cell(i+1,j). From a functional fault point of view this corresponds to simple fault called: Stuck-At ‘1’ Fault (SAF1).

Df2 - Short between 2 bit lines (at the 1st metal layer)

According to operating conditions shown in Table 3.1, this kind of defect is only sensitized during a write operation concerning one of the core cell driven by one of the two shorted bit line. In fact, during the cell(i,j) write operation, the targeted bit line BL_j (see Table 3.1) is fixed to 12v and all the other bit lines to HZ. In presence of Df2, 12v is also applied to the shorted bit line BL_{j+1} and implies a write operation on cell(i,j+1).

In this case the functional fault modeling is harder due to the implication of two adjacent core cells. Thus, Df2 induces a Coupling Fault (CF) due to the presence of an aggressor and a victim core cell. More precisely, this is a State CF (SCF) due to the implication of the aggressor core cell state on the victim core cell state. As the faulty behavior occurs during the write operation, the fault model is a State Coupling Fault (0,0). Remember that an erase operation is performed simultaneously in a whole eFlash page (see Chapter 1) and thus the opposite SCF behavior occurring during the erase operation (SCF(1,1)) is avoided.

Df3- Contact source opened

As for Df1, two core cells, cell(i,j) and cell(i-1,j) share the same Vss contact point. In presence of this open defect, the erase and write operations are still possible but as the source line contacts are disconnected, no current can be measured by the sense amplifier during the read operation. The two cells always present a logic ‘1’ and this can be functionally represented by a SAF1 as for Df1.

Df4- Floating gate short (at the 1st poly-silicon layer)

Due to the aggressive scaling between two floating gate transistors along a word line, two adjacent floating gate (FG) can be shorted. It implies double cells or multiple cells with the same state along the considered word line WL_i. As for Df2, a write operation on cell(i,j) induces a write on cell(i,j+1). The related fault model for this defect is a SCF(0,0) if the same reasoning as for Df2 is performed.

3.1.3 Resistive Defect Related Failures

In this section, the cell(i,j) behavior is analyzed in presence of resistive shorts in the array. The defect injection is performed on the 3*3 NOR-based eFlash memory array described in Figure 3.3. From a layout point of view, such defects may occur on the same layer (poly/poly, *i.e.* resistive defect between WL and Vref); this corresponds to defects Df8 and Df9, or between two different layers (metal/poly, *i.e.* resistive defect between WL and BL); this corresponds to defects

Df5 to Df7. Note that each defect configuration involves a net (bit line or word line) belonging to the target core cell of our study, i.e. cell(i,j).

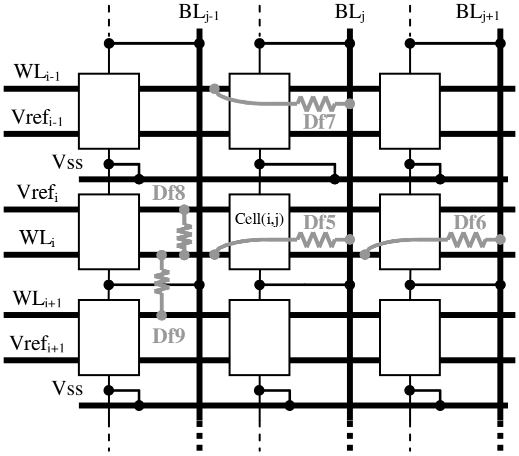


Figure 3.3: Resistive short defects in a NOR-based eFlash memory

In order to analyze the eFlash behavior in presence of a resistive short, five defects (see Figure 3.3) have been injected:

- **Df5:** resistive short between WL_i and BL_j. This defect concerns the cell(i,j).
- **Df6:** resistive short between WL_i and BL_x. This defect concerns the cell(i,x) sharing the same word line than cell(i,j) but not on the same bit line. In our case, the defect between WL_i and BL_{j+1} has been injected.
- **Df7:** resistive short between WL_y and BL_j. This defect concerns a cell(y,i) sharing the same bit line than cell(i,j) but not on the same word line. In our case, the defect between WL_{i-1} and BL_j has been injected.
- **Df8:** resistive short between WL_i and $Vref_i$. This defect concerns all the cells sharing the same word line and the same control line.
- **Df9:** resistive short between WL_i and WL_{i+1}. As for Df8, this defect concerns all the cells sharing the same word line.

However, note that a resistive defect between two Vref lines is not represented in Figure 3.3. This is due to the particularities of eFlash memories in which all Vref lines are driven in a concurrent way and thus no failure mechanism can be related to a resistive short between these two nets. Now, based on this defect injection configuration, a qualitative analysis of the failure mechanisms related to these defects in the 3*3 eFlash array is performed. This analysis was carried out for the three following operations:

- Erasing WL_i (erasing all cells $\in$ WL_i)
- Writing cell(i,j)
- Reading cell(i,j)

Erasing word line WL_i

During the erase operation, 14v is applied on WL_i and 0v on BL_j. Then, in presence of Df5, a high current pulse may appear that involves a drop of the high voltage generation (HVG < 14v). Consequently, the erase operation is not well performed on the cell(i,j) because less charges are injected in the FG. The VTH of cell(i,j) is affected. For example, a defect Df5 of about 500Ω decreases the HVG (14v) of about 1v that impacts the VTH of 860mV. The same behavior occurs in presence of Df6 because this defect also produces a resistive short between the HVG and the ground. On the other hand, Df7 and Df8 have no impact on the entire memory behavior. Note that, the presence of Df8 even enhances the erase operation on the entire word line. Concerning Df9, a drop on the high voltage apply to WL_i during the page selection can occur depending on the defect size. This results in a bad page erasing.

Writing cell(i,j)

During a write operation, the presence of Df5 has no impact on the state of cell(i,j) because the high voltage is applied on both lines WL_i and BL_j. On the other hand, with Df6, the word line voltage of WL_i may be transmitted on BL_{j+1}. Depending on the defect size, the voltage level of BL_{j+1} may be efficient enough to operate a write action on cell(i,j+1) but without impact on the content of cell(i,j). Df7 has the same impact than Df5 in erase mode. In fact, it makes a resistive short between the HVG and ground that may fall-down the HVG and so, impact the VTL. Df8 disturbs the write operation on cell(i,j) because it loads the control line ($Vref_i$) node to a high voltage and thus reduces the electric field between the floating gate and the drain diffusion of the cell(i,j). Finally, Df9 induces (as for the previous operation) a drop of the high voltage generation and thus the write operation on the cell(i,j) is not well acted.

Reading cell(i,j)

In presence of Df5, the node BL_j is charged by the current through the defect. This charge may mask the read only when cell(i,j) contains a logic ‘0’. In that case, the current consumed by cell(i,j) (due to the logic 0) may be masked by the current flowing through Df5. Then, no current may be measured by the sense amplifier and so, a logic ‘1’ may be read instead of ‘0’. The current consumed by the defect is equivalent to the current consumed by the cell when Df5 is about 100kΩ on the ATMEL technology. Df6 involves an increase of the WL_i load (capacitance) as it makes a resistive short between WL_i and BL_{j+1}. Then, the time required to set WL_i at 3.3v will be higher in presence of this defect. The read operation is thus delayed. Such a defect induces a dynamic fault. On the other hand, Df7 may lead to a faulty behavior but only in the case where cell(i,j) contains a logic ‘1’. In that case, no current has to be measured by the sense amplifier. But, as the defect makes a resistive short between BL_j and the ground, the resulting current is measured by the sense amplifier. Then, a logic ‘0’ may be observed instead of ‘1’. Finally, Df8 and Df9 have no impact on the entire memory behavior.

3.1.4 FLOATING GATE TRANSISTOR RELATED FAILURES

This kind of failures should not have immediate impact on the eFlash behavior. In fact, these defects may change some parametric characteristics of the eFlash memory and thus impact their reliability. These characteristics are the data retention, the endurance and the stability of the VTW (VT window). With the help of the cross section of a FloTOx core cell presented in Figure 3.4, an analysis of related defects has been performed. Two possible process variations are analyzed: the tunnel window thickness and the oxide nitride oxide (ONO) short. As for previous defects, these possible process variations may occur on actual eFlash due to their high integration density but also to the high accuracy required to perform the particular process steps used to build a FloTOx core cell.

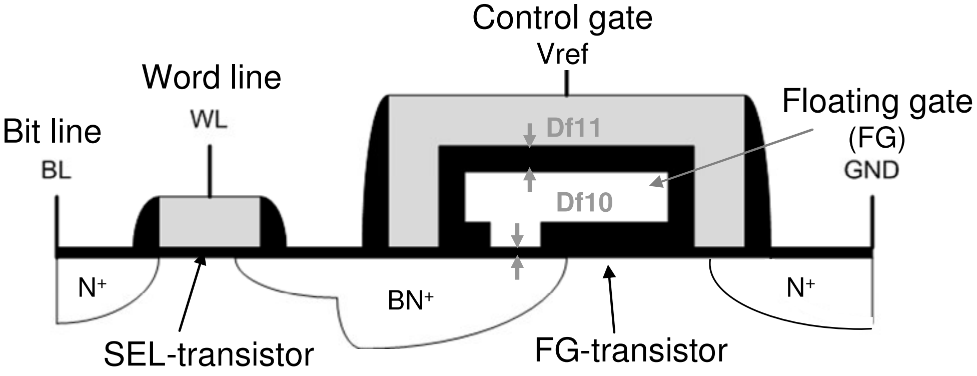


Figure 3.4: Cross section of a FloTOx core cell

Df10- Tunnel Window

In the tunnel window, the tunneling effect occurs under a high electric field. When this tunnel window is not well defined due to a non-nominal process step, the write or erase operation cannot be efficient [PAR98]. If there is no tunnel window designed between the floating gate (FG) and the channel, the single bit fails due to the inhibition of the write and the erase operations. The cell remains close to its virgin state with a standard threshold voltage. On the other hand, with a short between the FG and the channel, there is a very good bit programming but there is no data retention because the floating gate is not isolated.

The two previous cases are extreme ones because in almost all practical cases there is only a little variation +ΔTox or -ΔTox of the tunnel oxide thickness. In the case of -ΔTox (small reduction of the tunnel window), a bad retention of charges by the FG is achieved but a good VTW. It means that the duration of the stored information is not maximal. Moreover, a little negative ΔTox variation can increase the stress of the oxide and so, affect the reliability of the core cell. When the tunnel window thickness variation is positive (+ΔTox), the electric field is smaller and the charges are less injected or removed in the FG. So, the read margin defined by the VTW parameter (VT window) is affected.

Df11- ONO Short (inter poly-silicon oxide)

When a ONO (Oxide Nitride Oxide) short occurs, the two poly-silicon layers, floating gate and control gate, are shorted. The result of this defect is a single bit failed because the poly-silicon does not keep the trapped charges and a gate leakage current occurs. The threshold voltage of the core cell remains virgin due to the leakage of the FG charges.

3.1.5 DISTURB PHENOMENON

Until now, all analyses of disturb mechanisms have been done on Flash memory architectures built with ETOX core cells [MOH01] [MOH03a] [MOH03b]. In this section, we show that a disturb phenomenon may also appear in NOR-based eFlash architectures with FloTOx core cells.

A disturb phenomenon appears in a memory array when a Write, Erase or Read operation on a targeted cell affects the state of its neighborhood. Most of the time, the disturbances are due to the presence of a high voltage on the core cell nodes. Thanks to its structural specificities, the eFlash memories built with FloTOx core cells may be affected by only one disturb mechanism.

This disturb is due to the bit line coupling between a targeted core cell and a victim core cell sharing the same word line (Figure 3.5). The aggressive over-scaling of eFlash technology enables two adjacent bit lines (at a layout point of view) to create a non negligible coupling capacitance (C1). This coupling phenomenon creates a capacitive divider bridge with the equivalent bit line capacitance (C2) and thus induces a disturb voltage on the unselected bit line node. Note that this disturb occurs only during a write operation (logic '0') on the targeted core cell whereas the victim core cell is at a logic '1' (VTH).

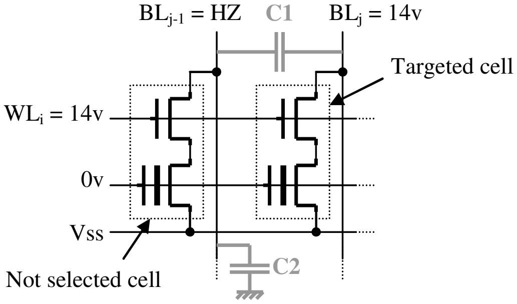


Figure 3.5: Bit line coupling phenomenon between FloTOx core cells.

In our experiment performed with a 150nm eFlash memory technology, we have measured C1 = 0.11 fF and C2 = 0.2 fF. Based on these two values, the contribution of the coupling effect on the drain voltage can be easily calculated. For example, when the bit line BL_j is set to 14v thus inducing a non negligible voltage on the bit line BL_{j-1}, there is:

$$BL_{j-1} = [(0.11) / (0.11 + 0.2)] * BL_j = 4.97v$$

Moreover, as the select gate of the victim cell is 'on', the disturb voltage due to the capacitive divider bridge on the bit line is directly applied on the drain of the FG-transistor. This voltage creates a variation of the electric field between the floating gate and the drain diffusion (BN^+) (see Figure 3.4). To have a well understanding of this phenomenon, the Fowler-Nordheim current equation (1) which realizes the programming operation in a FloTOx core cell must be taken into consideration:

$$I_{FN} = A * \alpha * E^2_{ox} * \exp(-\beta/E_{ox}) \qquad (1)$$

with:

A = Tunnel window area

α = Fowler Nordheim constant

β = Fowler Nordheim constant

E_{ox} = Oxide electric field

The threshold voltage (VT_{cell}) of the core cell depends on the charge quantity stored in the floating gate. This quantity is given by the integration of the Fowler-Nordheim tunneling equation (1) during the write time Tp:

$$VT_{cell} = VT_0 - (Q_{FG} / Cc) \qquad (2)$$

with:

$Q_{FG} = Q_{FG0} + \int^{Tp} I_{FN}*dt$

VT_0= Initial threshold voltage of the FloTOx core cell

C_c = ONO (Oxide Nitride Oxide) capacitance

The interest of these two previous equations is to create a direct relation between the VT variation of a victim erased (ΔVTH) core cell with the disturb voltage and the exposition duration. Based on this relation, we want to know the VTH variation whatever the disturb voltage value and its duration. However, note that equations (1) and (2) describe a continuous phenomenon and to develop a first order numerical model of the core cell behavior, these two expressions have to be digitized. After the expression digitization, the new theoretical and analytical model has been calculated with the help of constant extracted from 150nm FloTOx silicon data. This first order model (Table 3.2) has been compared to measurements done on a 150nm eFlash technology (Table 3.3) and for different experimental set-up. In Tables 3.2 and 3.3, the first column gives the disturb duration and the next ones give the VTH values for different disturb voltages. The original state of the victim core cell is erased (logic ‘1’) with a VTH fixed at 2.5v.

Time **(sec)**	**VTH** **BL_{j-1}=3.5V**	**VTH** **BL_{j-1}=5V**	**VTH** **BL_{j-1}=6V**	**VTH** **BL_{j-1}=7V**	**VTH** **BL_{j-1}=8V**
0.001	2.5	2.49	2.49	2.44	1.95
0.01	2.5	2.49	2.48	2.15	1.27
0.1	2.49	2.49	2.38	1.61	0.62
1	2.49	2.48	2.03	1.06	0.06
10	2.49	2.40	1.58	0.58	-0.41

Table 3.2: Theoretical estimation of VT disturb due to bit line coupling

Time (sec)	VTH BL_{j-1}=3.5V	VTH BL_{j-1}=5V	VTH BL_{j-1}=6V	VTH BL_{j-1}=7V	VTH BL_{j-1}=8V
0.001	2.49	2.49	2.49	2.46	2.02
0.01	2.49	2.49	2.48	2.24	1.53
0.1	2.48	2.49	2.39	1.81	1.03
1	2.48	2.47	2.11	1.38	0.58
10	2.48	2.37	1.75	0.98	0.17

Table 3.3: Measurement of VT disturb due to bit line coupling

From these values we can validate our model to estimate the VT variation due to a disturb voltage because the error between experimental and theoretical data is very low. Moreover, these results show that the VT value is more sensible to the variation of the disturb voltage than to the exposition duration. In fact, with a disturb voltage of about 3.5v or 5v, the VT remains close to its initial value whatever the exposition duration. Whereas with a high disturb voltage (*i.e.* high bit line coupling value, C1 in Figure 3.5) of about 7v or 8v, the core cell state is changed.

3.2 A Spice-like FloTOx Electrical Model for Faulty Behavior Prediction

In the previous section, a qualitative analysis has been proposed on actual defects that may occur in an eFlash memory array built with FloTOx core cells. The faulty behaviors associated to the hard defects can be easily modeled without the use of electrical simulations.

Conversely, electrical simulations have to be performed in order to validate the possible faulty behaviors of the eFlash in presence of defects such as resistive shorts. Electrical simulation models are needed to perform such evaluations during the read operation but also during the write and erase operations. The read operation is easy to simulate as it consists in a current measurement through the core cell. This operation can be simulated with a SPICE-like description including different VT values (VTL and VTH) of the floating gate transistor and appropriate voltage levels on core cell nodes. On the other hand, erase and write operations have to be performed in presence of defects. In this case, the electrical simulation model has to take into account the Fowler-Nordheim tunneling effect which is more difficult to represent.

In [MOH01] [MOH03a], a SPICE electrical model based on 1T Flash bit-cell is presented to evaluate the *Program Disturb Fault* in Flash memories. This model implements the two possible mechanisms (Channel Hot Electron Injection and Fowler-Nordheim tunneling effect) to write and erase a FG-transistor. This model is limited to a static description of write and erase operations represented by a fixed current and voltage sources. Therefore, with such a model, it is impossible to simulate successive operations (write and/or erase) performed on a core cell. Others academies proposed a dynamic solution to simulate a FloTOx core cell behavior based on a MOS Model 9 [POR02]. This model achieves good performances but requires complex set-up parameters. In this section, the development of a SPICE-like FloTOx model writable and erasable by Fowler-Nordheim tunneling effect is proposed. Our proposed simulation model will have a dynamic and autonomous electrical behavior depending on the voltages applied on its nodes. In the following, the proposed electrical model is described and then, comparisons with silicon data to validate this model are provided.

3.2.1 Description of the Simulation Model

To develop such a model, the eFlash FloTOx core cell presented in Figure 3.6 is considered.

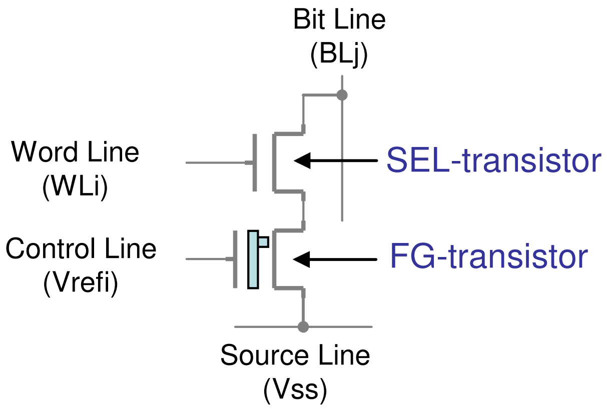


Figure 3.6: eFlash FloTOx core cell.

From this scheme, the development of the model consists in representing the SEL-transistor and the FG-transistor.

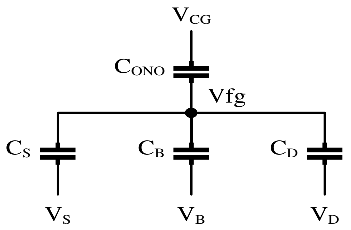
The SEL-transistor is modeled by a particular device with high voltage properties and available in almost all eFlash memories in a 150nm technology. This particular transistor will be called NMOSHV for NMOS *High Voltage*.

Concerning the FG-transistor, the description is more complex due to particular coupling effects and to the Fowler-Nordheim tunneling phenomenon.

To model the coupling effects, the double-gate device presented in the cross-section of Figure 3.4 is considered. One gate is a floating gate (FG) and the other is the control gate (CG) that is connected to the Vref node. Due to the ONO (Oxide Nitride Oxide) dielectric between the two gates, a part of the voltage applied on the Vref node (CG) also occurs on the FG. This phenomenon is called the gate coupling effect. This effect is characterized by the ratio of the

ONO capacitance over the total capacitance of the FG-transistor. This ratio is referred to here as Kg. To have a good understanding of this coupling effect, Figure 3.7 represents the different capacitances of the FG-transistor. The different nodes of the FG-transistor are also represented, i.e., the control gate, the floating gate, the drain, the bulk and the source nodes.

Due to the gate coupling effect, the high voltage applied to the control gate during the erase operation is important enough to create a high electric-field between the floating gate and the drain node. In presence of this electric-field, the Fowler-Nordheim tunneling effect can occur. The other capacitances (C_S and C_B) do not contribute significantly to the floating gate voltage due to their low values. The drain coupling effect (C_D) is due to the oxide tunnel window drawn above the drain diffusion where the oxide thickness is very small (see Figure 3.4). It is referred to here as Kd.



$$C_{TOT} = C_{ONO} + C_S + C_B + C_D$$

$$Kg = \frac{C_{ONO}}{C_{TOT}}$$

$$Kd = \frac{C_D}{C_{TOT}}$$

Figure 3.7: Coupling effects in the sense transistor

We have seen the importance of capacitive effects in order to model the FG-transistor. Now, the charge injection or removing mechanisms in the floating gate is described. Remember that the erase or write operations of a FloTOx core cell use the Fowler-Nordheim tunneling effect to modulate the charge quantity in the floating gate and thus the threshold voltage value of the FG-transistor. The FG can be easily represented by a capacitive charge quantity varying with the Fowler-Nordheim tunneling effect. Following the law Q = C * U, the voltage of this capacitance is proportional to the charges injected or removed. From a physical point of view, the equivalent capacitance value (C_{TOT}) represents the total capacitance of the FG-transistor.

We have seen in the previous section that the charge quantity stored in the FG impacts the VT value of the FG-transistor. When the core cell is erased, a high VT (VTH) is achieved and when the core cell is written, a low VT (VTL) is obtained. Thus, to create the model, a transistor whose VT value changes according to the quantity of charges stored in C_{TOT} is required. This principle can be easily implemented in an electrical SPICE-like model.

Now, the physical phenomenon of Fowler-Nordheim charge transfer in the floating gate is described. The equation (1), explained previously in this chapter, describes the Fowler-Nordheim tunneling effect. This tunneling effect is equivalent to a current source controlled by an electric-field (Eox). This electric-field is due to the voltage between the drain diffusion and the floating gate at the tunnel window interface. In a SPICE-like simulator, such current source can be easily described if all parameters of expression (1) are known. From silicon measurements on the 150nm eFlash technology, all these parameters have been extracted; A, α, β and the oxide thickness of the tunnel window used to calculate the electric field Eox.

With all these elements a first order SPICE-like has been implemented of which the block scheme is presented in Figure 3.8.

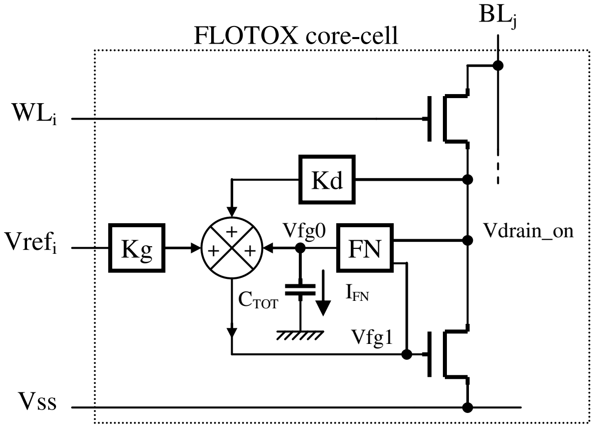


Figure 3.8: FloTOx electrical simulation model

For the SEL-transistor, as mentioned before a NMOSHV transistor is used. The blocks Kg and Kd represent the coupling factors due to the different capacitances. The capacitance C_{TOT} is used to store the charges provided by the block FN representing the Fowler-Nordheim tunneling effect. We sum these three effects (Sum block) to control the gate of a high voltage transistor NMOSHV representing the FG-transistor. The voltage Vfg1 represents the equivalent floating gate voltage of a FloTOx core cell under Fowler-Nordheim and coupling effects. As seen previously, the VT value of an erased or written FloTOx core cell depends on the charge quantity stored in its floating gate. This charge quantity is represented by the Vfg1 potential in the proposed model. During an erase operation, as charges are injected to the floating gate, Vfg1 becomes negative. Conversely, the Vfg1 potential becomes positive during a write operation as charges are removed from the floating gate. From a read operation point of view, when Vfg1 is negative, a high voltage on Vref_i is needed to create a population inversion in the NMOSHV device and hence appearance of a current through the bit line. The VT value is high and the cell is erased (VTH). When Vfg1 is positive, even a small voltage on $Vref_i$ node allows the NMOSHV device conduction, thus corresponding to a low VT value (VTL). Note that the model is represented as a linear feedback system because the electric field depends not only on the drain voltage but also on the floating gate voltage which varies with the C_{TOT} charge quantity and thus with floating gate potential.

3.2.2 Experimental Validation of the Model

To validate our proposed FloTOx core cell model, some simulations have been performed for erase, write and read operations. These results are presented in Figure 3.9. In this figure, the five waveforms represent WL_i, $Vref_i$, BL_j, Vfg1 and the current through the bit line (I_{BLj}) respectively.

The first 2ms of the simulation corresponds to an erase operation. This operation consists in applying a high voltage to WL_i (≈14v) and $Vref_i$ (≈12.5v) nodes, and in assigning Vss and BL_j nodes at ground. At the beginning of the erase operation, due to the coupling effects (Kg and Kd), the floating gate voltage (Vfg1) follows the $Vref_i$ node ($Vfg1 \approx Kg * Vref_i$). Afterwards, the Fowler-Nordheim tunneling effect begins and induces a charge injection in the floating gate. In the model, charges are stored in C_{TOT} that leads to a negative voltage level at Vfg0 node. Thus, the resulting level of Vfg1 decreases until the end of the erase operation. The erase operation is followed by a standby phase of 1ms duration. During this phase, the Vfg1 level is measured about -1.32v, which corresponds to the VT value (VTH) of an erased FloTOx core cell.

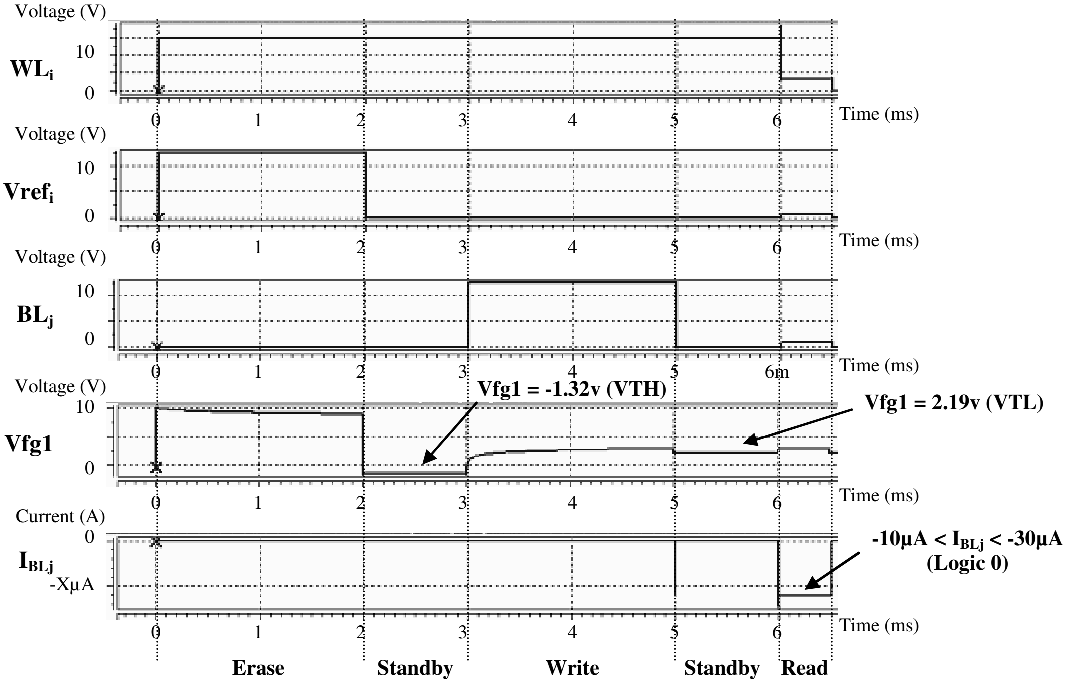


Figure 3.9: Electrical simulations of Erase, Write and Read operations

In the next phase (from 3ms to 5ms) which is a write operation, nodes WL_i and BL_j are set to a high voltage (≈14v and ≈12.5v respectively) while nodes $Vref_i$ is set to ground. At the beginning of this write operation, the coupling effects (Kg and Kd) and the voltage levels on the core cell nodes make the floating gate level close to its previous level. Then, the Fowler-Nordheim tunneling effect starts to remove charges stored in the floating gate. According to the model, charges are removed from C_{TOT}. The Vfg1 level increases as a capacitance loading. The following phase, as previously, is a standby phase. The Vfg1 level measured in this phase is about 2.19v which corresponds to the VTL of a written FloTOx core cell.

Finally, the last operation (from 6ms to 6.5ms) corresponds to a read operation. The different nodes of the core cell are set as follows: ≈3.3v for WL_i, ≈1v for BL_j, ≈0.7v for $Vref_i$ and ground for Vss. As the core cell has been written (VTL), it contains a logic '0'. In this case, during the read operation the sense amplifier has to measure a current through the bit line. The sense amplifier measures the current through the bit line, with an error below 10%. An erase operation followed by a read has been also simulated (not represented in Figure 3.9). In this case, the core cell is erased (VTH which represents a logic '1'). The simulation confirms that no current passes through the bit line.

After these basic simulations of the FloTOx core cell model, we now present some comparisons with silicon data obtained on the 150nm eFlash technology. The VTH and VTL values are analyzed after the erase and write operations for different programming times and voltages. Table 3.3 gives a representative set of these comparisons between Silicon data and the proposed model. In this table, the two first columns give the programming time (Tp) and voltage conditions (Vpp). The following columns present the VT window (VTW = VTH – VTL) obtained after an erase and a write operation for both our model (SPICE-like Model) and measurements (Silicon Data). The last column of the table gives the difference between the model and silicon data. VTH and VTL values obtained with the model and measured on an eFlash memory are not reported on this table for confidentiality reasons.

These comparisons show that for a Vpp value close to the nominal condition (12.5v), the SPICE-like model and silicon data are always quite similar (the error is less than 2%) irrespective of the considered programming time (from 1ms to 5ms). To comment these results, it is important to notice that the establishment of the model has required the extraction of the Fowler-Nordheim constants from silicon data measurements. These constants have been calculated from erase and write operations in nominal conditions; a programming time of 2ms and a high voltage Vpp of 12.5v. So, the matching between the model and silicon data was quite predictable for a nominal value of Vpp. Even if this proposed model is a first order model, the resulting simulation represent, with good accuracy, the electrical behavior of FloTOx core cells.

Timing and voltage conditions		Threshold voltage VT (v)		Error (%)
		SPICE-like	Silicon Data	(VTWm – VTWd)/VTWd*100
Tp	Vpp	VTWm	VTWd	
1ms	12.5v	3.67	3.64	0.82
1ms	12v	2.69	2.77	-2.89
1ms	11.5v	1.84	1.90	-3.16
1ms	11v	1.15	1.04	10.58
2ms	12.5v	4.28	4.19	2.15
2ms	12v	3.26	3.32	-1.81
2ms	11.5v	2.32	2.45	-5.31
2ms	11v	1.52	1.58	-3.8
5ms	12.5v	4.98	4.88	2.05
5ms	12v	3.93	4.00	-1.75
5ms	11.5v	2.93	3.13	-6.39
5ms	11v	2.04	2.26	-9.73

Table 3.3: Characterization of the FloTOx model.

3.2.3 Further Improvements of the Model

The objective of our model was reached for nominal conditions as there is a good matching with silicon data measurements. However, our characterization study of the proposed model can be extended to other conditions. These conditions could be those of new Flash memory utilization or those induced by the presence of a particular defect. Thus, some simulations with a Vpp not in nominal conditions (see Table 3.3) have been performed. For these extremes conditions, the decrease of the Vpp voltage involves a bad matching of the proposed model with silicon measurements. This is true especially when we measure VTH and VTL obtained with the model and silicon data. As for the nominal conditions characterization, these results were predictable. In fact, the Fowler-Nordheim constants calculation has been performed in nominal conditions and silicon measurements show that their values can depend on the electrical conditions applied on the FloTOx core cell. For example, the β parameter of (1), which depends on the SiO2 energy barrier at the tunnel window interface, varies with the initial electric-field E_{OX} intensity and so, with the Vpp value. These discrepancies of the model do not appear when the VT window (VTWm and VTWd) is only considered as presented in the last column of Table 3.3 where a maximum error of 10.6% is obtained. This phenomenon can be explained by an offset problem for which the value is correlated to the extremes Vpp and Tp conditions.

A possible improvement of this work would be to focus on the setting up of a new model matching more precisely the silicon measurements irrespective of the experimental conditions. Due to the high voltage, the FloTOx environment is aggressive and thus requires the use of particular devices (NMOSHV) to realize the model. A model could be proposed where these NMOSHV devices could be characterized for a larger range of operations. Moreover, our proposed model takes place in the 150nm eFlash technology model card; this induces a low speed simulation that can be improved using a compact model [POR02].

Nevertheless, it is important to emphasize that the model proposed in this paper is pertinent enough to describe the electrical behavior of a FloTOx core cell under defect injection or for failure analysis.

3.3 Resistive Defect Injection in the FloTOx Core cell Array

In Section 3.1, the qualitative analysis of possible defects occurring in the FloTOx core cell and in the eFlash array has been performed. All these resistive defects have been reported from a

150nm eFlash technology.

From this first analysis, we have seen that resistive defects in the core cell array involve complex behaviors during erase, write and read operations. Now, with the help of the first order electrical model of a FloTOx core cell, resistive defects have been injected and their corresponding behaviors have been analyzed. The objective in this section is to demonstrate the interest of our electrical model for defect analysis and faulty behavior prediction useful to give comprehensive fault models. In fact, the proposed model is used to simulate the erase, write and read operation in presence of defects. These simulations allow the evaluation of electrical levels on important nodes of the core cell, i.e. Vfg1 and I_{BLj}. These levels allow predicting the resulting faulty behavior of the core cell and the range of defects that involves this faulty behavior.

In the following, we first present the resistive defect injection and the simulation environment used to simulate the considered defects with our model. Then, each defect with electrical data has been analyzed in detail and a functional fault model is proposed for each faulty behavior.

3.3.1 DEFECT INJECTION SET-UP

First, a simulation environment is configured close to the silicon reality. A 3*3 eFlash core cell array based on our FloTOx electrical model is described. Moreover, in this description two blocks are added; a logic decoder and a high voltage generator. The logic decoder drives the high voltage on the WL_i, $Vref_i$ and BL_j nodes according to the operation acted on the selected core cell. Details on this circuit are not given here as it is composed of standard decoder circuits using pass-gates with high voltage properties.

In eFlash memories, a charge pump based on the Dickson principle [DIC76] is embedded for high voltage generation. In order to simplify the simulation environment, the equivalent scheme of the charge pump (Figure 3.10) is used.

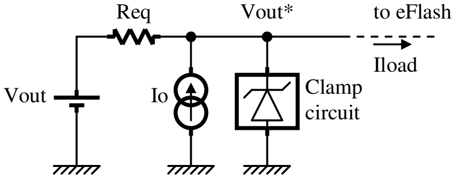


Figure 3.10: Charge pump first order equivalent circuit

A charge pump allows providing a high voltage level from a standard low voltage supply by a capacitive energy transfer [DIC76]. There is a linear relation between the size of the charge pump (N), the low voltage supply (Vdd) and its output voltage (Vout), Vout = (N + 1) * Vdd. In this relation we can add a statement introducing that the charge pump device has a drop voltage when its load becomes too important:

- Vout = (N + 1) * Vdd – (Iload * Req).

Note that the charge pump equivalent resistance (Req) depends on the size of its building elements and on its working frequency [DIC76]. In our simulation environment, an equivalent charge pump circuit embedded in a 2Mbits eFlash memory is used. The equivalent resistance of this charge pump is 42kΩ and the output clamped voltage is about 15V. For the defect injection simulation, especially for resistive shorts, the previous statements on the electrical charge pump characteristics must be considered.

With the complete simulation environment, including the core cell array, the logic decoder and the charge pump, the considered defects can be injected and thus the prediction of their impact on different node voltages is possible. This was done for different defect values but also for the following sequence of operations:

Op.1: Erasing WL_i and Writing cell(i,j)

Op.2: Reading cell(i,j), cell(i,j+1) and cell(i-1,j)

Op.3: Erasing WL_i

Op.4: Reading cell(i,j), cell(i,j+1) and cell(i-1,j)

Note that only cell(i,j) is written in Op.1 and the other cells of the array are considered as erased (VTH → the cells contain a logic ‘1’). Moreover, in Op.1 an erase of the entire word line WL_i is performed before a write on cell(i,j). In the functional work of a FloTOx core cell, a write operation is always preceded by an erase operation. Only the first erase operation is performed from a pseudo-virgin state where the core cell has its threshold voltage between VTH and VTL. Concerning the read operation, the neighborhood cells of cell(i,j) sharing the same word line are also read to analyze the possible coupling effect induced by each resistive defect. Moreover, simulations have been performed with the following parameters:

- Process: fast
- Supply voltage: 1.8V for logic parts
- Temperature: 25°C
- Programming time (erase and write operations): 2ms

In the following, the impact of each resistive defect is detailed during the four operations (Op.1 to Op.4). Simulations have been performed on a 3*3 eFlash array built with our proposed FloTOx model for which the following electrical values have been extracted:

- Vm corresponds to the charge pump output voltage
- WL_i, $Vref_i$, and BL_j correspond to the inputs of cell(i,j)
- VT value (VTH or VTL) calculated from the floating gate voltage Vfg
- I_{BL} value corresponding to the read current of a cell viewed by the sense amplifier

3.3.2 Resistive Short between Metal and Poly-Silicon Layers

In this part, the resistive defects corresponding to the configuration Df5, Df6 and Df7 described in Section 3.1.3 are analyzed. These three defects are represented again in Figure 3.11.

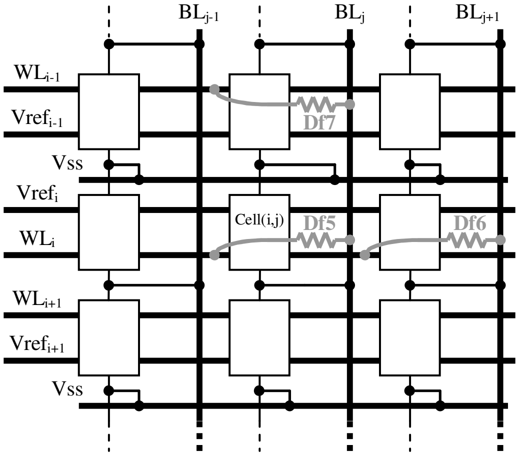


Figure 3.11: Resistive defects configuration between metal and poly-silicon layers

Df5-Aanalysis

The impact of defect Df5 (resistive short between WL_i and BL_j) on the behavior of cell(i,j) and its neighbors cell(i,j+1) and cell(i-1,j) is analyzed. Electrical measurements provided by the model are reported in Table 3.4.

Op.1: Df5 impact during a Write operation

In presence of Df5 during a write operation, the voltage of BL_j node increases. This effect is due to nominal conditions for which the WL_i voltage is always higher than the BL_j voltage. For small defect size, Df5 = 10Ω for example, the BL_j voltage becomes equivalent to the WL_i level. Concerning the VTL value of cell(i,j), we see in Table 3.4 that whatever the size of Df5, it remains stable around -1.1v. Cell(i,j+1), sharing the same word line as cell(i,j), has its VTL value that begins to be impacted for Df5 less than 100kΩ, from 2.66v to 1.47v. The VTL of the cell(i-1,j) remains unchanged whatever the size of Df5.

Op.2: Df5 impact during a Read operation

For Op.1, we have seen that the VT value of the defective written core cell (VTL) keeps its expected value. During the read operation, the current that passes through the core cell must correspond to this VTL value, i.e. the core cell must provide a current between 5µA to 30µA. But the current induced by the defect may mask the current provided by cell(i,j). Thus, a logic '1' may be read instead of a logic '0'. As shown in Table 3.4, such a faulty behavior does not occur when Df5 is higher than 1MΩ. A faulty behavior of the read operation on cell(i,j) begins to occur for a defect size less than 1MΩ. During Op.2, the current read through the cell(i,j+1) corresponds to a logic '1' whatever Df5 size as the defect has no influence on the bit line j+1. On the other hand, during the read operation of cell(i-1,j), the current that passes through the defect makes the sense amplifier providing a logic '0' instead of a logic '1' (for Df5 < 100kΩ).

Op.3: Df5 impact during an Erase operation

During the erase operation, Df5 involves two main impacts on cell(i,j); the first one on the BL_j node and the second one on the high voltage generation Vm. Without defect during the erase operation the two transistors of the core cell are on. As Vss node is set to ground, BL_j is pulled-down to this level. Now, in presence of Df5, and depending on its size, BL_j is pulled-up to the WL_i level. As the two core cell transistors are on (linear mode), a part of the BL_j voltage is present on the drain node of the sense transistor. When the BL_j level becomes equivalent to the WL_i level, the select transistor becomes saturated and starts to deliver a current through the core cell. This current passes through the core cell but also through the defect. The charge pump delivers this defective current. When it reaches a certain value, a drop on the high voltage generation is observed. In Table 3.4, we observe the increase of the BL_j node and the drop on the high voltage generation, i.e. Vm node. For a defect size of about 10kΩ the clamp of the Vm level is observed. With the increase of BL_j level, the drain level of the FG-transistor increases too. This involves a reduction of the electric field E_{ox} used in the current tunnel equation. Consequently, the Fowler-Nordheim tunneling effect is not enough efficient to inject charges in the floating gate. Due to this bad electric field, the VT value of cell(i,j), obtained at the end of the erase operation, remains close to its previous state (VTL). The core cell does not reach the expected VTH (about +2.66V for a fault free core cell). For example, with Df5 = 10Ω, the electrical model gives a VTH of about -1.04V. The VT values of cell(i,j+1) and cell(i-1,j) remain unchanged, i.e. the ones obtained with Op.1.

Node levels during Opi		Df5 value (Ω)							
		10	100	1E+3	1E+4	1E+5	1E+6	1E+7	∞
WRITE Op.1	Vm(V)	15.2	15.2	15.2	15.2	15.2	15.2	15.2	15.2
	WLi(V)	14.4	14.4	14.4	14.5	14.6	14.6	14.6	14.6
	Vrefi(V)	0.351	0.351	0.351	0.351	0.351	0.351	0.351	0.351
	BLj(V)	14.4	14.4	14.4	14.2	12.8	12.5	12.5	12.5
	Vfg1(V)	1.76	1.76	1.76	1.77	1.8	1.86	1.86	1.86
	VTL(V)	-1.04	-1.04	-1.04	-1.05	-1.09	-1.16	-1.16	-1.16
	VTL(V) of cell(i,j+1)	1.47	1.47	1.47	1.47	2.11	2.66	2.66	2.66
	VTL(V) of cell(i-1,j)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
READ Op.2	WLi(V)	2.04	2.07	2.29	2.96	3.26	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	2.03	2.01	1.8	1.23	0.991	0.958	0.955	1
	Iblj(A)	-5.9E-4	-5.75E-4	-4.66E-4	-1.42E-4	5.56E-6	2.6E-5	2.8E-5	2.8E-5
	Ibl of cell(i,j+1)	7.29E-8	7.29E-8	7.29E-8	7.29E-8	0	0	0	0
	Ibl of cell(i-1,j)	2.98E-4	2.9E-4	2.32E-4	7.6E-5	9.97E-6	9.97E-7	0	0
ERASE Op.3	Vm(V)	10.7	10.7	10.7	10.9	14.6	15.2	15.2	15.2
	WLi(V)	9.44	9.44	9.24	9.64	13.5	14.5	14.6	14.7
	Vrefi(V)	8.78	8.79	8.82	8.95	12.2	12.8	12.8	12.8
	BLj(V)	9.44	9.42	9.24	7.31	1.45	0.147	0	0
	Vfg1(V)	1.76	1.76	1.76	1.76	0.033	-1.17	-1.27	-1.28
	VTH(V)	-1.04	-1.04	-1.04	-1.04	1.06	2.53	2.65	2.66
	VTH(V) of cell(i,j+1)	1.47	1.47	1.47	1.47	2.11	2.66	2.66	2.66
	VTH(V) of cell(i-1,j)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
READ Op.4	WLi(V)	2.04	2.07	2.3	2.96	3.26	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	2.03	2.01	1.81	1.23	1.03	1	1	1
	Iblj(A)	-5.89E-4	-5.75E-4	-4.66E-4	-1.42E-4	-2.02E-5	-2.29E-6	-2.3E-7	0
	Ibl of cell(i,j+1)	7.29E-8	7.29E-8	7.29E-8	7.29E-8	0	0	0	0
	Ibl of cell(i-1,j)	2.98E-4	2.9E-4	2.32E-4	7.6E-5	9.97E-6	9.97E-7	0	0

Table 3.4: Electrical simulation results for Df5 injection

Op.4: Df5 impact during a Read operation

According to Df5 values, the expected VTH can never be reached by cell(i,j) and thus keeps its previous state, i.e. a low VT. As in read mode the control gate voltage is 0.7v, a positive current must be observed through the core cell. Due to the presence of Df5 and to the voltage conditions between WL_i and BL_j, we do not observe this current caused by the bad erasing of cell(i,j). But a negative current through Df5 is observed with a value depending on its size. According to Df5 size, this defective current may inhibit the current delivered by the core cell. In Table 3.4, we observe that the current read through the bit line never exceeds the sense amplifier threshold. The sense amplifier always reads a logic ‘1’ and that for any Df5 values. Concerning cell(i,j+1) and cell(i-1,j), the same explanation as for Op.2 can be done.

Conclusion and fault modeling

From a quantitative point of view, the presence of Df5 is always problematic whatever its size. In fact, a faulty behavior occurs for both small and high defect values. Based on the electrical simulation performed with the proposed model, the different faulty behaviors can be associated to functional fault models.

For cell(i,j), as shown before, the write operation is correctly acted while, due to a drop on the high voltage generation, the erase operation is not performed. A read operation after a write will provide a logic ‘1’ due to the current through the defect even if the core cell has a low VT. In the same way, a logic ‘1’ is read after an erase operation even if the core cell has a low VT. Thus, Df5 behaves like a Stuck-At ‘1’ Fault (SAF1).

The presence of Df5 has no impact on cell(i,j+1) behavior while it impacts cell(i-1,j). Cell(i-1,j) is erased (VTH) but any read operation will give a logic ‘0’ as the sense amplifier measures the defective current that passes through the defect. Then, a Stuck-At ‘0’ Fault (SAF0) occurs on cell(i-1,j).

Df6- Analysis

In this section, the impact of defect Df6 (resistive short between WL_i and BL_{j+1}) on the behavior of cell(i,j) and its neighbors cell(i,j+1) and cell(i-1,j) is analyzed. Electrical measurements provided by the model are reported in Table 3.5.

Op.1: Df6 impact during a Write operation

The presence of Df6 in the eFlash array during a write operation does not impact the behavior of cell(i,j). The threshold voltage of cell(i,j) remains almost unchanged with the decrease of Df6 size: VTL varies from –1.16v for Df6 = 10MΩ to -0.98v for Df6 = 10Ω. On the other hand, the VT of cell(i,j+1) is affected during the write of cell(i,j) with a minimum value of -1.43v for Df6 = 100kΩ. Finally, cell(i-1,j) is not impacted by the defect and thus its VT remains unchanged.

Op.2: Df6 impact during a Read operation

We have seen in the previous operation analysis that the VT value (VTL) of cell(i,j) keeps roughly its expected value. During the read operation, the current that passes through the core cell corresponds to this VTL and its value is around 20µA. However, we can observe that the word line potential WL_i goes down with the decrease of Df6 and thus induces a small decrease of the read current provided by cell(i,j). From a logic point view, cell(i,j) after a write operation and in presence of Df6 gives a logic ‘0’ whatever the defect size.

The impact of Df6 on the read operation of cell(i,j+1) is more complex. As seen previously, the write performed on cell(i,j) has impacted the VT of cell(i,j+1). This corresponds to a flip from VTH to VTL. Thus, the read operation of cell(i,j+1) should provide a logic ‘0’. Data reported in Table 3.5 show that the cell provides a current that corresponds to a logic ‘0’ for a defect size of about 100kΩ. For lower defect sizes, the current that passes through the defect masks the current of the cell and then a logic ‘1’ is read. This phenomenon is represented in Figure 3.12 where waveforms of VT and I_{BL} of cell(i,j+1) are drawn. From these waveforms, we show that a logic ‘0’ is read when 100kΩ < Df6 < 400kΩ.

For read operations performed on cell(i-1,j), no current is provided by this cell through the bit line and thus a logic ‘1’ is always read.

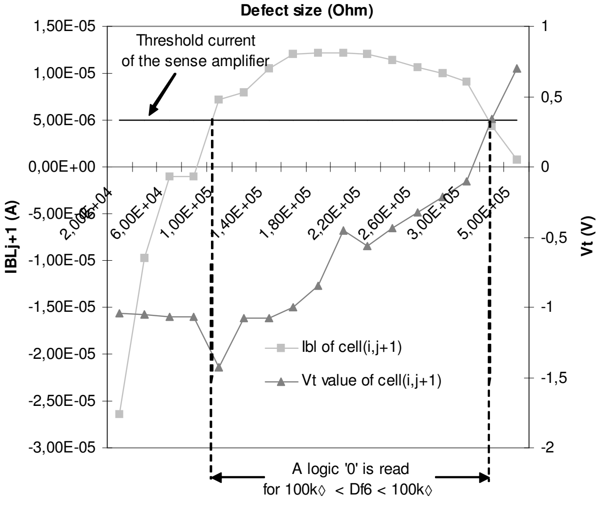


Figure 3.12: VT and I_{BL} of cell(i,j+1) function of Df6 size after a write operation on cell(i,j)

Op.3: Df6 impact during an Erase operation

Like for Df5 injection, the high voltage generation falls down with the decrease of Df6 during the erase operation. In Table 3.5, the voltage drop of the Vm node for a defect less than 10kΩ is observed. This voltage drop on Vm induces a reduction of the electric field Eox used in the current tunnel equation. Consequently, the Fowler-Nordheim tunneling effect is not effective to inject charges in the floating gate. Due to this bad electric field, the VT value of cell(i,j) obtained at the end of the erase operation (VTH) remains close to its previous state. The core cell does not reach the expected VTH (about +2.66V for a fault free core cell).

Node levels during Opi		Df6 value (Ω)							
		10	100	1E+3	1E+4	1E+5	1E+6	1E+7	∞
WRITE Op.1	Vm(V)	15.1	15.1	15.1	15.2	15.2	15.2	15.2	15.2
	WLi(V)	14.4	14.4	14.4	14.5	14.6	14.6	14.6	14.6
	Vrefi(V)	0.351	0.351	0.351	0.351	0.351	0.351	0.351	0.351
	BLj(V)	12.3	12.3	12.3	12.3	12.4	12.4	12.4	12.5
	Vfg1(V)	1.71	1.71	1.72	1.72	1.78	1.86	1.86	1.86
	VTL(V)	-0.98	-0.98	-0.992	-0.992	-1.07	-1.16	-1.16	-1.16
	VTH(V) of cell(i,j+1)	-1.38	-1.38	-1.38	-1.38	-1.43	1.38	1.81	2.66
	VTH(V) of cell(i-1,j)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
READ Op.2	WLi(V)	2.04	2.07	2.29	2.96	3.26	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	0.969	0.969	0.966	0.96	0.957	0.955	0.955	1
	Iblj(A)	1.8E-5	1.9E-5	2.1E-5	2.5E-5	2.67E-5	2.8E-5	2.8E-5	2.8E-5
	Ibl of cell(i,j+1)	-6E-4	-5.8E-4	-4.8E-4	-1.44E-4	7.2E-6	0	0	0
	Ibl of cell(i-1,j)	0	0	0	0	0	0	0	0
ERASE Op.3	Vm(V)	10.7	10.7	10.7	10.9	14.6	15.2	15.2	15.2
	WLi(V)	9.52	9.52	9.56	9.64	13.5	14.5	14.7	14.7
	Vrefi(V)	8.86	8.86	8.89	9.02	12.2	12.8	12.8	12.8
	BLj(V)	0.18	0.18	0.18	0.18	0	0	00	0
	Vfg1(V)	1.59	1.59	1.58	1.55	-0.715	-1.25	-1.28	-1.28
	VTH(V)	-1.04	-1.04	-1.04	-1.04	1.06	2.53	2.65	2.66
	VTH(V) of cell(i,j+1)	-1.02	-1.02	-1.02	-1.02	1.02	2.64	2.98	2.98
	VTH(V) of cell(i-1,j)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
READ Op.4	WLi(V)	2.04	2.07	2.3	2.96	3.26	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	0.969	0.969	0.966	0.96	0.957	0.955	0.955	1
	Iblj(A)	1.83E-5	1.86E-5	2.03E-5	2.34E-5	0	0	0	0
	Ibl of cell(i,j+1)	-5.8E-4	-5.7E-4	-4.6E-4	-1.4E-4	-2E-5	-3E-6	0	0
	Ibl of cell(i-1,j)	0	0	0	0	0	0	0	0

Table 3.5: Electrical simulation results for Df6 injection

For example, with Df6 = 10 Ω, the electrical model gives a VTH of about -1.04V. The VT values of cell(i,j+1) and cell(i-1,j) remain close to the ones obtained after Op.1.

Op.4: Df6 impact during a Read operation

We have seen that the expected VTH can never be reached by cell(i,j) for values of Df6 lower than 100kΩ and thus the core cell keeps its previous state. Then, when a voltage close to 0.7v is applied on the control gate of cell(i,j), a current close to 20µA, corresponding to the low threshold voltage of the core cell, is observed. This phenomenon occurs when Df6 is less than 10kΩ and corresponds to a VT of about -1.04v on cell(i,j). However, note that Df6 also induces a drop of the WL_i net voltage limiting the current delivered by the core cell but not enough efficient to inhibit it. When Df6 is less than 10kΩ, a logic ‘0’ is read instead of an expected logic ‘1’ on cell(i,j). Concerning cell(i,j+1) and cell(i-1,j), the presence of Df6 does not induce a faulty behavior and a logic ‘1’ is always read on these cells even if cell(i,j+1) has a low VT. In that case, the current that passes through the defect masks the current of the core cell.

Conclusion and fault modeling

As for Df5, a functional fault modeling can be performed with the help of electrical data provided by our FloTOx model. The presence of Df6 does not disturb the write operation on cell(i,j). The following read gives a logic ‘0’ that corresponds to a low VT. However, due to a drop on the high voltage generation, Df6 prevents the erase operation, so that the cell remains with a low VT. The read operation on the erased cell will provide a logic ‘0’ instead of a logic ‘1’. This faulty behavior corresponds to a transition fault (from a logic ‘0’ to ‘1’) as the cell(i,j) remains at a low VT after a write operation.

As mentioned above, the behavior of cell(i,j+1) is impacted by the presence of Df6 especially when a write operation in performed on cell(i,j). In that case and for 100 kΩ < Df6 < 400kΩ, the VT of cell(i,j+1) changes from VTH to VTL. From a functional point of view, a write on a cell (aggressor) involves the switch of another cell (victim) to a logic ‘0’. This is modeled like a State Coupling Fault; SCF(0,0). Finally, cell(i-1,j) is not impacted by Df6.

Df7- Analysis

In this subsection, the impact of defect Df7 (resistive short between WL_{i-1} and BL_j) on the behavior of cell(i,j) and its neighbors cell(i,j+1) and cell(i-1,j) is analyzed. Electrical measurements provided by our model are reported in Table 3.6.

Op.1: Df7 impact during a Write operation

The presence of Df7 during a write operation induces a high drop voltage on the Vm generation due to the potential between WL_{i-1} and BL_j. The unselected word line WL_{i-1} is pulled down to the ground whereas the bit line voltage BL_j is close to the Vm potential. The current through the defect is important and the charge pump device cannot deliver such amount of current. That involves the VT of the cell(i,j) does not reach its expected value VTL and remains quasi unchanged at VTH.

Node levels during Opi		Df7 value (Ω)							
		10	100	1E+3	1E+4	1E+5	1E+6	1E+7	∞
WRITE Op.1	Vm(V)	2.18	2.22	2.62	6	14.4	15.2	15.2	15.2
	WLi(V)	2.15	2.19	2.58	5.85	13.9	14.7	14.7	14.7
	Vrefi(V)	0.292	0.293	0.296	0.316	0.349	0.351	0.351	0.351
	BLj(V)	0.49	0.521	0.843	3.74	11.4	12.4	12.5	12.5
	Vfg1(V)	-1.31	-1.31	-1.31	-1.31	0.917	1.8	1.86	1.86
	VTL(V)	2.66	2.66	2.66	2.66	-1.39E-2	-1.09	-1.16	-1.16
	VTH(V) of cell(i,j+1)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
	VTH(V) of cell(i-1,j)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
READ Op.2	WLi(V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	0.51	0.522	0.62	0.877	0.959	0.954	0.954	1
	Iblj(A)	2.9E-4	2.9E-4	2.3E-4	7.6E-5	2.6E-5	2.8E-5	2.8E-5	2.8E-5
	Ibl of cell(i,j+1)	0	0	0	0	0	0	0	0
	Ibl of cell(i-1,j)	-6E-4	-5.8E-4	-4.8E-4	-1.6E-4	2.2E-5	-2.2E-6	-2.3E-7	0
ERASE Op.3	Vm(V)	15.2	15.2	15.2	15.2	15.2	15.2	15.2	15.2
	WLi(V)	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7
	Vrefi(V)	12.8	12.8	12.8	12.8	12.8	12.8	12.8	12.8
	BLj(V)	0	0	0	0	0	0	0	0
	Vfg1(V)	-1.55	-1.55	-1.55	-1.55	-1.28	-1.28	-1.28	-1.28
	VTH(V)	3	3	3	3	2.66	2.66	2.66	2.66
	VTH(V) of cell(i,j+1)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
	VTH(V) of cell(i-1,j)	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
READ Op.4	WLi(V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	0.51	0.522	0.62	0.877	0.959	0.954	0.954	1
	Iblj(A)	3E-4	2.9E-4	2.3E-4	7.65E-5	9.7E-6	1E-8	1E-7	0
	Ibl of cell(i,j+1)	0	0	0	0	0	0	0	0
	Ibl of cell(i-1,j)	-6E-4	-5.8E-4	-4.8E-4	-1.6E-4	-2.2E-5	-2.2E-6	-2.3E-7	0

Table 3.6: Electrical simulation results for Df7 injection

In Table 3.6, we see that this phenomenon occurs for Df7 less than 100kΩ. For example, when Df7 = 10kΩ, the VT value of cell(i,j) is about +2.66v as the one obtained after an erase operation. We can also observe in Table 3.6 that cell(i,j+1) and cell(i-1,j) are not impacted by the presence of Df7. These cells keep their VT values (VTH).

Op.2: Df7 impact during a Read operation

After a write operation (Op.1), we have seen that for defect Df7 less than 100kΩ the VT value of the cell(i,j) remains in an erased state and thus the core cell is not able to provide any current during the read operation. However, due to the presence of this defect, a defective current between WL_{i-1} and BL_j appears and masks the bad writing effect induced by Df7. From a logical point of view, a logic '0' is read even if the cell has a high VT. The bad writing of cell(i,j) is masked. Concerning cell(i,j+1) and cell(i-1,j), the expected logic '1' is read.

Op.3: Df7 impact during an Erase operation

During the erase operation, no problem occurs on the three cells. Only a small over-erase can be observed on cell(i,j) for Df7 size less than 100kΩ.

Op.4: Df7 impact during a Read operation

As explained above for Op.2, a defective current due to Df7 can be measured through the bit line BLj and even if the cell(i,j) state is erased with a VTH value close to +2.66v. The sense amplifier interprets this current (defective current through the defect) as a logic '0' on the cell(i,j). Df7 has no impact during Op.4 on cell(i,j+1) and cell(i-1,j) as a logic '1' is read.

Conclusion and fault modeling

As previously, a functional fault modeling based on electrical data provided by our model has to be performed. The presence of Df7 only impacts the behavior of cell(i,j). This defect prevents the write operation on cell(i,j) but, even if the cell has a high VT after the write operation, the sense amplifier provides a logic '0' during the read. In the same way, the erase operation is performed correctly, but the read after the erase operation gives a logic '0'. Thus, Df7 behaves like a Stuck-At '0' Fault (SAF0).

3.3.3 RESISTIVE SHORT BETWEEN TWO POLY-SILICON LAYERS

In this part, the resistive defects corresponding to Df8 and Df9 (short between two poly-silicon layers) described in Section 3.1.3 is detailed. These defects are represented in Figure 3.13.

Note that these two defects require only a faulty behavior analysis of the core cell belonging to the word line WL$_i$. Topologically, the defect impacts only these core cells.

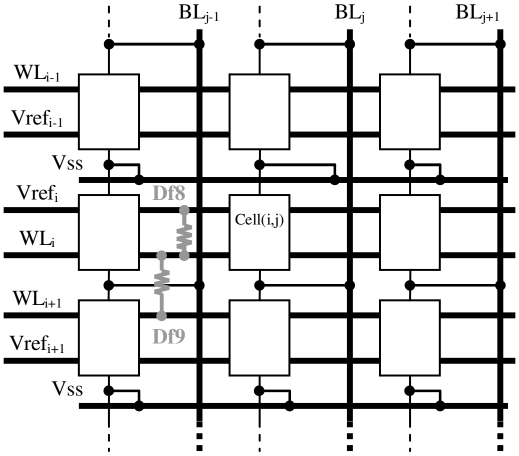


Figure 3.13: Resistive defects configuration between two poly-silicon layers

Df8- Analysis

In this part, the impact of defect Df8 (resistive short between WL$_i$ and Vref$_i$) on the behavior of cell(i,j) and its neighbors cell(i,j+1) is analyzed. Electrical measurements provided by our model are reported in Table 3.7.

Op.1: Df8 impact during a Write operation

The presence of Df8 during a write operation induces an increase of Vref$_i$. In fact, due to this resistive short, Vref$_i$ is loaded by the potential applied on WL$_i$. This effect begins earlier with defect size close to 1MΩ. Due to the increase of Vref$_i$, the write operation is not performed and cell(i,j) remains erased with a VTH. Moreover this defect also induces an over-erase on all core cells belonging to WL$_i$. Table 3.7 shows that the VT values of cell(i,j) and cell(i,j+1) achieve until +4.84v.

Op.2: Df8 impact during a Read operation

Due to the presence of Df8 we have seen that the write operation is inhibited even for high defect values. Thus, during the read operation, a logic ‘1’ is observed instead of an expected logic ‘0’.

Op.3: Df8 impact during an Erase operation

During the erase operation, no problem occurs on core cells sharing WL$_i$. Only a small over-erase can be observed on all the core cells belonging to WL_i for Df8 size less than 1MΩ.

Op.4: Df8 impact during a Read operation

During this operation, the presence of Df8 does not disturb the read-out content. The expected logic ‘1’ is read in all core cells sharing WL_i.

Node levels during Opi		Df8 value (Ω)							
		10	100	1E+3	1E+4	1E+5	1E+6	1E+7	∞
WRITE Op.1	Vm(V)	2.18	2.22	2.62	6	14.4	15.2	15.2	15.2
	WLi(V)	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7
	Vrefi(V)	14.7	14.7	14.7	14.7	14.7	14.7	14.6	0.351
	BLj(V)	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5
	Vfg1(V)	-3.07	-3.07	-3.07	-3.07	-3.07	-3.07	-1.1	1.86
	VTL(V)	4.84	4.84	4.84	4.84	4.84	4.84	-1.16	-1.16
	VTH(V) of cell(i,j+1)	4.84	4.84	4.84	4.84	4.84	4.84	4.03	2.66
READ Op.2	WLi(V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Vrefi(V)	3.29	3.29	3.29	3.29	3.29	2.67	0.756	0.7
	BLj(V)	1	1	1	1	1	1	1	1
	Iblj(A)	0	0	0	0	0	0	0	2.8E-5
	Ibl of cell(i,j+1)	0	0	0	0	0	0	0	0
ERASE Op.3	Vm(V)	15.2	15.2	15.2	15.2	15.2	15.2	15.2	15.2
	WLi(V)	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7
	Vrefi(V)	14.7	14.7	14.7	14.7	14.7	14.6	12.8	12.8
	BLj(V)	0	0	0	0	0	0	0	0
	Vfg1(V)	-3.32	-3.32	-3.32	-3.32	-3.32	-3.26	-3.05	-1.28
	VTH(V)	4.82	4.82	4.82	4.82	4.82	4.82	4.82	2.66
	VTH(V) of cell(i,j+1)	4.84	4.84	4.84	4.84	4.84	4.84	4.84	2.66
READ Op.4	WLi(V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Vrefi(V)	3.29	3.29	3.29	3.29	3.29	2.67	0.756	0.7
	BLj(V)	1	1	1	1	1	1	1	1
	Iblj(A)	0	0	0	0	0	0	0	0
	Ibl of cell(i,j+1)	0	0	0	0	0	0	0	0

Table 3.7: Electrical simulation results for Df8 injection

Conclusion and fault modeling

To conclude the Df8 study and to analyze this defect from a functional point of view, the presence of Df8 can be interpreted as a Stuck-At ‘1’ Fault (SAF1). This fault model occurs when Df8 has a size lower than 10MΩ.

Df9- Analysis

In this part, the impact of defect Df9 (resistive short between WL_i and WL_{i+1}) on the behavior of cell(i,j) and its neighbor cell(i,j+1)) is analyzed. Electrical measurements provided by our model are reported in Table 3.8.

Op.1: Df9 impact during a Write operation

The presence of Df9 during a write operation induces a high voltage drop on the Vm generation due to the potential between WL_i and WL_{i+1}. Indeed, the word line WL_{i+1} is unselected and its potential is pull down to the ground (a strong 0v) whereas the selected word line WL_i is pull up to the Vm potential. The current passing through the defect is significant and the charge pump device cannot deliver such amount of current. Note that before this *Write operation*, an erase has been performed and the same problem has been observed. The high voltage drop related to Df9 involves an impact on the VT of the cell(i,j). VT does not reach its expected value VTL and remains quasi unchanged to the virgin value. In Table 3.8, we see that this phenomenon occurs for Df9 less than 100kΩ. We can also observe in Table 3.8 that cell(i,j+1) is impacted by the presence of Df9 because the erase operation due to the defect current passing through Df9 is not correctly acted. To generalize, cells belonging to WL_i have their VT window (VTW) considerably reduced due to the bad erase and write operations induced by Df9.

Op.2: Df9 impact during a Read operation

After a write operation (Op.1), for Df9 less than 100kΩ, the VT value of the cell(i,j) remains close to its virgin state and the core cell is not able to provide any current during the read operation. Thus, a logic '1'is read instead of an expected '0' for a Df9 size less than 100kΩ. This also true for the other cells belonging to WL_i in which a logic '1' is always read but note that for these cells this is the expected value.

Op.3: Df9 impact during an Erase operation

During the erase operation, the same problem as for Op.1 occurs and impacts the VT values of the core cells sharing WL_i. All the VT potentials of these cells are close to their virgin values due to the inhibition of the erase operation induced by the presence of Df9. This problem appears when Df9 begins to be less than 100kΩ.

Op.4: Df9 impact during a Read operation

As explained above for the previous operations, Df9 inhibits write and erase and thus the cell remain virgin. The limit of this faulty behavior is when Df9 is less than 100kΩ. Due to this problem, a logic ‘1’ is always read on these core cells whatever the operation acted on them.

Node levels during Opi		Df9 value (Ω)							
		10	100	1E+3	1E+4	1E+5	1E+6	1E+7	∞
WRITE Op.1	Vm(V)	2.18	2.22	2.62	6	14.4	15.2	15.2	15.2
	WLi(V)	2.15	2.19	2.58	5.85	13.9	14.7	14.7	14.7
	Vrefi(V)	0.292	0.293	0.296	0.316	0.349	0.351	0.351	0.351
	BLj(V)	0	0	0.2	3.55	11.6	12.4	12.5	12.5
	Vfg1(V)	0.31	0.31	0.31	0.31	0.51	1.8	1.86	1.86
	VTL(V)	0.73	0.73	0.73	0.73	0.49	-1.09	-1.16	-1.16
	VTH(V) of cell(i,j+1)	0.79	0.79	0.79	0.79	2.66	2.66	2.66	2.66
READ Op.2	WLi(V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	0.954	0.954	0.954	0.954	0.954	0.954	0.954	1
	Iblj(A)	0	0	0	0	6E-6	2.8E-5	2.8E-5	2.8E-5
	Ibl of cell(i,j+1)	0	0	0	0	0	0	0	0
ERASE Op.3	Vm(V)	2.18	2.22	2.62	6	14.4	15.2	15.2	15.2
	WLi(V)	2.15	2.19	2.58	5.85	13.9	14.7	14.7	14.7
	Vrefi(V)	0	0	0.5	3.8	11.8	12.8	12.8	12.8
	BLj(V)	0	0	0	0	0	0	0	0
	Vfg1(V)	0.31	0.31	0.31	0.31	-0.09	-0.95	-1.28	-1.28
	VTH(V)	0.73	0.73	0.73	0.73	1.2	2.26	2.66	2.66
	VTH(V) of cell(i,j+1)	0.79	0.79	0.79	0.79	2.66	2.66	2.66	2.66
READ Op.4	WLi(V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Vrefi(V)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
	BLj(V)	0.954	0.954	0.954	0.954	0.954	0.954	0.954	1
	Iblj(A)	0	0	0	0	0	0	0	0
	Ibl of cell(i,j+1)	0	0	0	0	0	0	0	0

Table 3.8: Electrical simulation results for Df9 injection

Conclusion and fault modeling

Modeling the faulty behavior induced by Df9 is very easy. When Df9 is less than 100kΩ, a Stuck-At ‘1’ Fault (SAF1) is observed whereas for defects above this limit the core cells are fault free.

3.4 SUMMARY AND DISCUSSION ON TEST SOLUTIONS

In the previous investigations, all actual defects that may induce a faulty behavior of the eFlash core cell have been analyzed. From these different failure mechanisms, a functional fault modeling has been performed. Table 3.9 summaries the different fault models associated to defects and targeted cells. This allows to show in few lines the result of the previous analysis on each failure mechanism and defect configuration. In this table, the defect sizes from which the faulty behaviors occur are also reported except for defects Df1 to Df4 in which this is not applicable (NA) because they are hard defects. In fact, Df1 to Df4 are either a pure open (∞Ω) or a pure short (0Ω). Moreover, in Table 3.9 the cases containing a dashed line represent the fault free-cells in presence of a defect. Note that the fault models are represented with respect to the operations acted on the different cells.

Defect	**Cell(i,j)**		**Cell(i,j+1)**		**Cell(i-1,j)**	
	Defect size	Fault model	Defect size	Fault model	Defect size	Fault model
Df1	NA	SAF1	--	--	NA	SAF1
Df2	NA	SCF(0,0)	NA	SCF(0,0)	--	--
Df3	NA	SAF1	--	--	NA	SAF1
Df4	NA	SCF(0,0)	NA	SCF(0,0)	--	--
Df5	< 1 MΩ	SAF1	--	--	< 100 kΩ	SAF0
Df6	< 10 kΩ	TF (0 to 1)	100kΩ < … < 400kΩ	SCF(0,0)	--	--
Df7	< 100 kΩ	SAF0	--	--	--	--
Df8	< 10 MΩ	SAF1	--	--	--	--
Df9	< 100 kΩ	SAF1	--	--	--	--
Df11	NA	SAF1	--	--	--	--

Table 3.9: Related fault models

By looking at this table, the obtained functional fault models correspond to the memory testing literature; SAF0, SAF1, TF and more complex coupling fault SCF(0,0). Note that the defect Df10 and the coupling phenomenon described in Section 3.1 are not presented in Table 3.9 because they require a more accurate analysis that will be performed in Chapter 4.

From Table 3.9 results, the patterns or strategies currently used to test Flash memories, like these presented in Chapter 2 (the 7-patterns and 5-steps test sequences), are enough efficient to detect the established fault models. However, the 7-patterns and 5-steps test sequences present few weaknesses (complexity, testing time…) and a test alternative is proposed in Chapter 5.

CHAPTER 4: TESTING OXIDE THICKNESS VARIATION OF THE TUNNEL WINDOW

This chapter is dedicated to the detailed analysis of a defect that corresponds to the defectiveness of the tunnel window oxide. We show that this defective oxide thickness impacts erase or/and write operations as well as retention and reliability of FloTOx eFlash memories.

As these problems do not affect the functional behavior of the FloTOx core cell, they require a specific test approach to be detected. The proposed solution to detect such defectiveness is based on the coupling phenomenon existing between two bit lines. First, the origin of the coupling phenomenon existing between bit lines is detailed and the influence of its resulting voltage disturbance on a core cell state is analyzed. Finally this voltage disturbance is shown as non aggressive for a defect free core cell whereas this voltage is very useful to detect an oxide thickness variation in a FloTOx tunnel window.

4.1 ORIGIN OF THE COUPLING PHENOMENON

Thanks to its structural specificities and to the presence of a SEL-transistor before the FG-transistor, an eFlash memory built with FloTOx core cells may be affected by only one disturb mechanism. This disturbance is due to the bit line coupling between a targeted cell and a victim cell sharing the same word line. The aggressive over-scalability of eFlash technology enables two adjacent bit lines (from a layout point of view) to create a non-negligible coupling capacitance (C1 in Figure 4.1). This coupling capacitance can create a capacitive divider bridge with the equivalent bit line capacitance due to the bit line path under the PWELL block (C2).

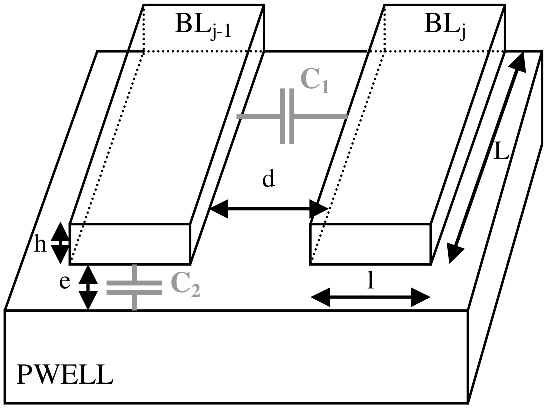


Figure 4.1: Equivalent coupling capacitance

From an analytical point of view these two capacitance values can be expressed with their geometric characteristics:

- $C_1 = \varepsilon_r \times (L \times h) / d$ (1)
- $C_2 = \varepsilon_r \times (L \times l) / e$ (2)

In expressions (1) and (2), there are the technology dependencies caused by the design rules but also the parameter L representing the eFlash memory density. Indeed, values of C1 and C2 depend on the number of FloTOx core cells placed in the memory array. In Figure 4.2, the equivalent scheme of the capacitive divider bridge between two adjacent FloTOx core cells sharing the same word line is represented.

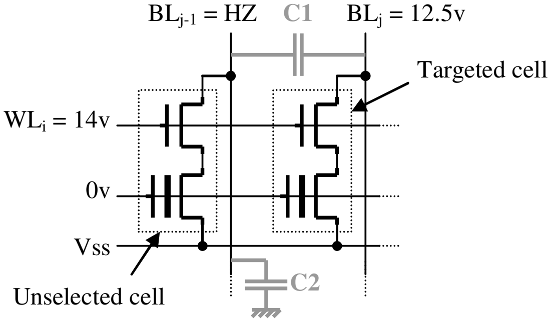


Figure 4.2: Disturb failure due to bit line coupling

Due to this coupling effect, the high voltage applied on the BL_j node for a write operation involves an undesired increase of the potential of node BL_{j-1}. With advanced technology rules (<150nm), the ratio between the coupling capacitance C1 and the bit line capacitance C2 increases drastically as C1 is in the same order of magnitude than C2. To illustrate the previous statement, an eFlash memory of 1Mbits in the 150nm technology has a bit line coupling capacitance close to 110fF and an equivalent bit line capacitance around 200fF.

The resulting voltage on the unselected bit line increases to reach a value that can cause a cell disturbance. Some measurements have been carried out on the 150nm eFlash technology with a variable memory density and different patterns written in the memory. These measurements show that the unselected victim bit line can reach a voltage close to 7V.

4.1.1 Influence of the Disturb Voltage

We have shown previously that a non negligible disturb voltage (about 7V) is present on the unselected victim bit line. As the SEL-transistor of the victim cell is 'on', the disturb voltage is directly applied on the drain node of the FG-transistor. The presence of this important voltage creates an electric field between the floating gate and the drain diffusion (BN+) of the FG-transistor. To have a well understanding of this phenomenon, the Fowler-Nordheim current equation acting the programming operation in a FloTOx core cell must be considered:

- $I_{FN} = A \times \alpha \times E_{ox}{}^{2} \times exp\left(-\beta / E_{ox}\right)$ (3)

with:

A = Tunnel window area

α = Fowler Nordheim constant

β = Fowler Nordheim constant

E_{ox} = Oxide electric field

Moreover, the threshold voltage value of a FloTOx core cell depends on the charge quantity stored in its floating gate. This quantity is given by the integration of Equation (3) during the write time Tp:

$VT_{cell} = VT_0 - (Q_{FG} / Cc)$ (3)

with:

$Q_{FG} = Q_{FG0} + \int^{Tp} I_{FN} * dt$

VT_0= Initial threshold voltage of the FloTOx core cell

C_c = ONO (Oxide Nitride Oxide) capacitance

In Equation (3), we see that the oxide electric field Eox takes an important part in the Fowler-Nordheim current generation and we know that this electric field directly depends on the voltage between the drain (BN+) and the floating gate (FG) node. With the help of Equations (3) and (4), a relation between the threshold voltage variation of an erased cell (ΔVTH) under a voltage disturbance and its exposition duration can be established. Thus, we know that a voltage disturbance can occur on the bit line node of an unselected core cell due to a coupling effect between two bit lines. From a theoretical point of view, the core cell is designed to avoid a VT changing under some considerations, i.e. a minimal electric field is required to shift the threshold voltage of the cell. Normally, the voltage threshold of a defect free FloTOx core cell can not be impacted by a voltage disturbance. However, in presence of a defective tunnel oxide thickness, a large variation of this voltage is observed. This is detailed in the next section.

4.1.2 Fault Modeling of the Oxide Thickness Variation of The Tunnel Window

Based on the previous statements, we have analyzed the impact of a voltage disturbance on the bit line node of an unselected cell affected by an oxide thickness variation for a nominal programming time, Tp = 2ms. Table 4.1 summarizes the results obtained. The first column gives the Tox variation and the following ones give the threshold voltage variation from a nominal value (ΔVTH) for different disturb voltages. The nominal threshold voltage value of an erased core cell is assumed to be close to +2.5V.

With the help of Table 4.1, we see the main impact of the disturb voltage when the tunnel window oxide thickness is less than its nominal value (about 75Å). Indeed, even if there is an important Tox variation (-ΔTox), an impact on the VTH of an erased cell is observed when the disturb voltage is equal or higher than 7V. Note that this voltage value is realistic and has already been measured on an unselected bit line in a memory designed in 150nm technology. In such a case, the VT value of the erased cell can shift from a logic '1' (VTH) to a logic '0' (VTL) when the VT variation is close to 2V (gray part in Table 4.1).

		Disturb Voltage					
		5v	6v	7v	8v	9v	10v
Tunnel Oxide Thickness (Å)	54	0.168	1.145	2.285	3.429	4.572	5.716
	57	0.047	0.774	1.901	3.044	4.188	5.332
	60	0.011	0.437	1.519	2.660	3.804	4.948
	63	0.003	0.191	1.141	2.277	3.420	4.564
	66	0.001	0.065	0.777	1.895	3.037	4.181
	69	0.000	0.019	0.453	1.515	2.655	3.798
	72	0.000	0.005	0.214	1.141	2.272	3.415
	75	0.000	0.001	0.083	0.784	1.891	3.033

Table 4.1: ΔVTH of a disturbed FloTOx core cell

Thanks to this analysis the functional model related to these two defective mechanisms is easy to obtain. As there are the notions of victim cell and aggressor cell, we are facing a coupling phenomenon. The faulty behavior due to the disturb mechanism can thus be modeled as a coupling fault. From a logic point of view, this is a shift from logic ‘1’ to logic ‘0’ of the aggressor cell that induces a shift from a logic ‘1’ to a logic ‘0’ on the victim cell. Nevertheless, this shift of the victim cell is not irreversible because an erase of the victim cell is still possible. Thus, the fault model related to this faulty behavior is a State Coupling Fault – SCF(0,0) [VAN98a].

This fault model is well adapted to this complex phenomenon. However, if the topological context of the disturb occurrence is considered, some restrictions have to be given to the model. Indeed, the SCF(0,0) does not correspond, like in CMOS memories, to all possible fault combinations between the matrix’s core cells. In the eFlash environment, the SCF(0,0) only occurs between core cells sharing the same word line. Thus, we will note this model as SCF(0,0)*, which is a restriction of the standard fault model proposed for RAM memories.

4.2 Testing Solution for an Oxide Thickness Variation

We have seen previously that a defective oxide thickness can be detected by using the disturb voltage due to the bit line coupling. This particular defective mechanism is modeled as a SCF(0,0)* restricted to a word line from a topological point of view.

In this section the test patterns to apply to the memory in order to detect this particular SCF(0,0)* is first defined. Next, we show that these two patterns have to be applied following a particular programming mode that uses a parallel approach.

4.2.1 Sensitization Sequence

From an exhaustive point of view, to detect SCF(0,0)* on the same word line, all possible combinations of such fault involving two core cells have to be considered. As this fault only occurs between core cells of the same word line, the pattern applied to a given word line can be applied with the help of a parallel configuration to the others word lines. This particular operating mode is called CCWP for *Concurrent Chip Write Pattern*.

In Figure 4.3, the pattern sequence that detects the SCF(0,0)* is applied to an eFlash word line containing 4bits. First of all, the whole page is erased in one time “1111”, this is called the initialization phase. Then, the first step is to write a logic '0' into the first core cell whereas the others remain to a logic '1'. During this step, the first core cell is considered as a possible aggressor and the other core cells as victims. These two phases are repeated until all possible aggressors are sensitized.

1	1	1	1	Initialization Phase
0	1	1	1	1st Step + Page Read
1	1	1	1	Re-Initialization
1	0	1	1	2nd Step + Page Read
1	1	1	1	Re-Initialization

1	1	1	0	Last Step + Page Read

Figure 4.3: SCF(0,0)* exhaustive test approach

In the literature, this approach is called *Galloping Pattern* [VAN98a] and its main detractor states in the large number of patterns to generate during its application.

For this example the eFlash is considered with a page granularity and that the write / erase operations of a page take the same time (2ms). Moreover, to simplify this illustration the read time is considered as negligible compared to programming operations. The number of operations to perform in our example is 4 page write and 4 page erase. Even if we use a parallel programming approach in which we write or erase a data of a page in one time in the whole memory chip with a duration close to a simple page programming, this test approach is time consuming. This is due to its complexity in O(2n) with n being the number of bit lines in the memory chip.

To define a good sequence with a minimized number of patterns, a better way will be to consider the electrical behavior and the origin of the disturb mechanism modeled by a SCF(0,0)*. As one of the origins of SCF(0,0)* is a coupling voltage from an adjacent core cell of a victim cell, the Figure 3.15 can be used and the coupling voltage can be expressed by the equation:

- $\Delta V_{BLj-1} = \Delta V_{BLj} \times [C_1 / (C_1 + C_2)]$ (5)

With the help of Equation (5) a pattern that increases the coupling voltage contribution by a superposition principle can be developed. Now let us consider an example of electrical configuration represented in the Figure 4.4.

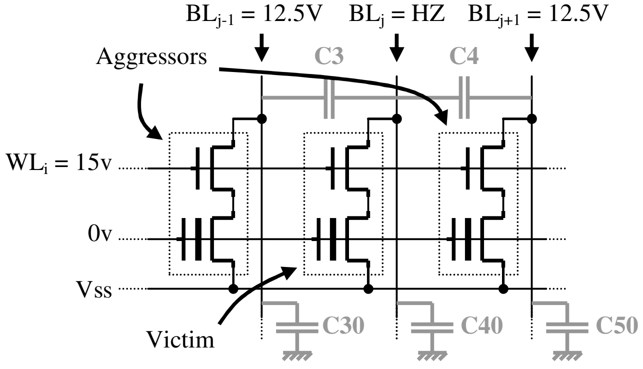


Figure 4.4: Extended coupling equivalent scheme

If we consider the victim cell on BL_j in the center of Figure 4.4 and surrounded by aggressor cells that are going to be written, the coupling voltage contribution to BL_j can be increased To illustrate this assumption we can write the equivalent capacitive divider bridge expression to the selected bit line voltage variation ΔVBL_j in applying the superposition theorem:

- $$\Delta V_{BLj} = \Delta V_{BLj-1} \times \frac{C3}{C3 + C40} + \Delta V_{BLj+1} \times \frac{C4}{C4 + C40} \qquad (6)$$

With the help of all previous statements and considerations, two test patterns are sufficient to sensitize all $SCF(0,0)^*$; Column Bars (CB) and its complement (CBI). These two patterns consist in write two complementary data in two adjacent columns namely bit lines in a FloTOx eFlash memory. CB and CBI are defined as follows:

Pattern 1 (CB): 10101010…1010

Pattern 2 (CBI):01010101…0101

Pattern 1 allows the sensitization of all $SCF(0,0)^*$ of core cells belonging the first, third, fifth … bit lines. In the same way, Pattern 2 sensitizes the faults of core cells belonging the second, fourth, sixth … bit lines. Some electrical simulations have been performed to evaluate the disturb voltage of victim cells due to the application of the two proposed patterns. These simulations have been carried out thanks to our SPICE-like model describing the electrical behavior of a FloTOx core cell presented previously and the help of capacitance values measured on the 150nm eFlash technology. The results obtained show that the disturb voltage reached on victim bit lines is about 6.5V. With the help of data presented in Table 4.1, we can conclude that this disturb voltage is not enough important to detect little variations of the oxide thickness as the victim cell does not flip ($\Delta VTH < 2V$). Based on this statement, the next section presents a parallel programming approach which allows to increase the coupling phenomenon and thus the disturb voltage on victim bit lines.

4.2.2 PARALLEL PROGRAMMING APPROACH

In the previous paragraph, we have seen that a sequence composed of CB and CBI patterns is a good way to sensitized SCF(0,0)*. The remaining problem is the resulting disturb voltage which is not enough important to detect little variations of the oxide thickness. In this section, a parallel programming approach is shown as a great solution to significantly increase the disturb voltage of victim bit lines.

Let us consider the electrical structure of a FloTOx core cell array represented in Figure 4.5 with the active equivalent capacitances.

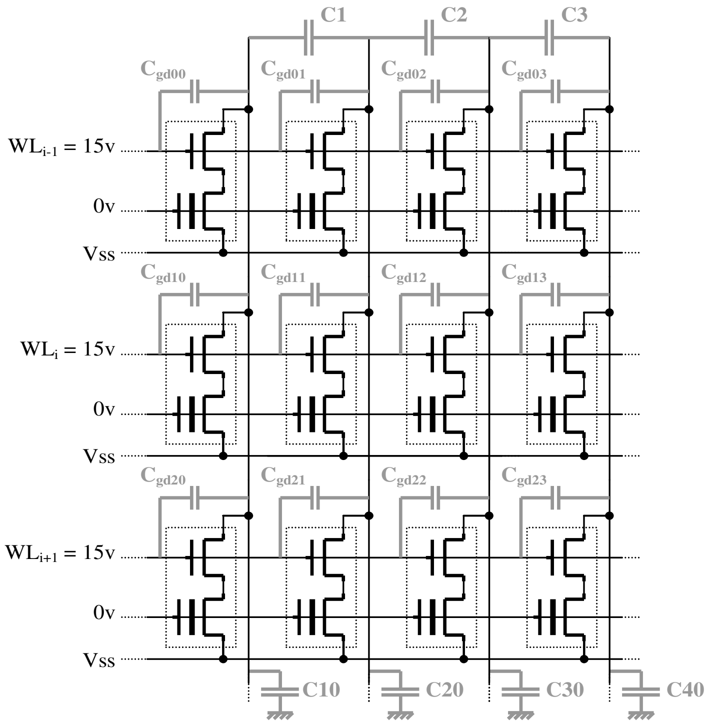


Figure 4.5: Equivalent core cells array in parallel selection

Using a parallel approach selecting a large amount of word lines in one time, the overlap capacitance gate-to-drain (Cgd in Figure 4.5) of each SEL-transistor in the core cell must be considered. In fact, due to the parallel selection, these capacitances can contribute to the increase of each coupling voltage. In the 150nm eFlash technology, the Cgd capacitance can be close to $9E^{-17}$F and becomes predominant when the number of word lines in the memory reaches 1024. In this case, the equivalent parallel capacitance will be close to 100fF that corresponds to the magnitude of the bit line coupling capacitance. Then, the equivalent coupling capacitance (C3 and C4 in Equation (6)) increases, thus inducing a much more aggressive disturb voltage on the victim cell.

As previously, electrical measurements have been performed. An eFlash memory composed of 1024 pages (word lines) with 64 words of 32 bits has been considered. Table 4.2 reports the bit line voltages (B0 to B31) with respect to the pattern written (CB and CBI) using the parallel programming approach.

	VBL (V)							
	B0	B1	B2	B3	…	B29	B30	B31
CB	**6.92**	12.5	**8.15**	12.5	…	12.5	**8.15**	12.5
CBI	12.5	**8.15**	12.5	**8.15**	…	**8.15**	12.5	**6.92**

Table 4.2: Bit line voltages using the parallel programming approach

First, all core cells of a word line are sensitized by a disturb voltage (bold values in the table) which is high enough (about 8.15V) to sensitize little defective oxide thickness. Moreover, we can also observe that the boarder bit lines (B0 and B31) are always less sensitized (disturb voltage is about 6.92V) than the others. This is because the boarder bit lines are subjected to the coupling contribution of only one aggressor. This is a limitation of the proposed method that can be solved by increasing the programming time and thus the impact of the disturb voltage on the victim core cells. To conclude, the CB and CBI patterns applied in a parallel programming mode are efficient to detect little oxide thickness variations of the tunnel window.

4.3 CONCLUSION

The impact of a defective tunnel window oxide (thickness variation) on the FloTOx behavior has been analyzed. The resulting faulty behavior of such a defect has been modeled as a State Coupling Fault (SCF(0,0)*). Based on the voltage disturbance caused by the bit line coupling phenomenon, a test solution has been proposed. This solution consists in two patterns column bars: CB and CBI. To be enough efficient to detect small oxide thickness variations, the increase of the disturb voltage on victim bit lines is required. A solution has been proposed to increase this voltage by using a parallel programming approach during the CB and CBI patterns application. This new solution achieves a detection of oxide thickness variation close to 12Ǻ. We can imagine applying this proposed method in a characterization context. Indeed, different programming modes can be defined in which the memory blocks have different size. That will allow to modify the bit line coupling and thus to module the victim bit line potential. This new characterization method can allow us to establish statistical data on the tunnel window oxide thickness variation in a FloTOx eFlash array without the using of costly measurement devices.

CHAPTER 5: TESTING FOR ADDRESS DECODER FAULTS IN EFLASH MEMORIES

Experiments on eFlash memories have shown that the most impacting faults from testing time considerations are the Address decoder Faults (AFs). In the literature, only few solutions exist to test such faults in eFlash and they are not optimal.

In this Chapter, the basis of Address decoder Faults (AFs) occurring in eFlash are introduced. By exploiting the memory specificities, a new methodology to test such faults reducing the testing time while preserving the same coverage rate compared to existing methods is presented. With the help of a home made fault simulator, we will show how the proposed methodology can detect almost all faults resulting from the defect analysis carried out in Chapter 3.

5.1 INTRODUCTION TO ADDRESS DECODER FAULTS (AFS) IN EFLASH

Many studies dealing with AF testing for RAM memories exist and are based on March test algorithms [VAN98a]. AF testing for eFlash memories is much more critical as existing March algorithms are not applicable in this context for test time reason, i.e. the slow programming time of an eFlash. Practically, AF testing is generally done by performing a diagonal of ‘0’ in the core cell array [SHA97]. This test pattern, called Diagonal 0 pattern, detects all AFs but its application time remains very high, e.g. about 4s for a 1024 pages eFlash.

Consequently, important efforts have to be done to find efficient AF test solutions that alleviate this test time problem. Beforehand, in this section some general basics of AFs in eFlash memories are presented.

5.1.1 EFLASH DECODING CIRCUITRY

Like almost all memories, the address decoding is performed by using an equivalent tree of pass-gates as represented in Figure 5.1. In this figure, an addressing path D defined from values (100) of the address bus (A0A1A2) through a tree of pass-gates is shown.

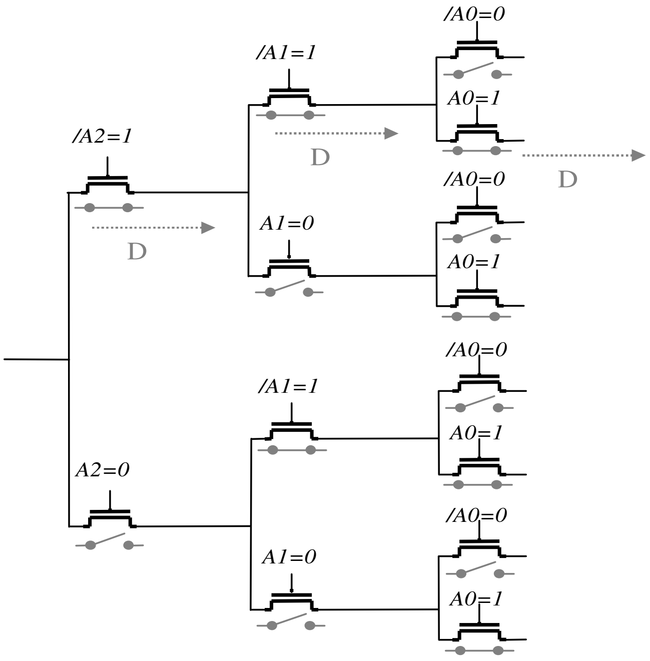


Figure 5.1: Tree of pass-gates for the address decoding logic in an eFlash memory

Exploiting the structural methods to address memory core cell, the eFlash memories has different two different operating modes: the user mode and the test mode. In the user mode, an eFlash has two possible programming operations. The first one is the Functional Write (FW) operation that allows writing a data into a word. This FW operation is composed of an Auto Erase (AE) of the selected page containing the word to be written (without losing the content of the others words) followed by the write operation of the desired data. The time needed for this FW operation is 4ms for an eFlash designed in the 150nm technology.

The second possible programming operation is the Page Write mode (PW). The eFlash has a page register allowing the user to program a whole page with the same duration as a FW operation. Also in this mode, the AE operation is preliminary performed.

In the test mode, there are three possible programming operations. The two first operations allow erasing (all 1') or writing (all 0') the eFlash in one step and in 10ms: Chip Erase (CE) and Chip Write (CW). The third programming operation, called Concurrent Chip Write Pattern (CCWP), allows writing a particular pattern in different selected memory addresses. To explain this programming operation, let us consider again the address decoding logic presented in Figure 5.5. This address decoder has 3-bit addresses and thus allows selecting 8 word lines. To select one word line at a time, the corresponding address is applied on the Ai address-bits and the complemented address is applied on the /Ai address-bits, thus resulting in a unique address selection. When the CCWP operation is used, more than one word line has to be selected. This can easily be done by applying a certain address on the Ai address-bits and a different address on the /Ai address-bits. For example, let us assume that the CCWP operation has to write the same page on WL0, WL2, WL4 and WL6. To do that, 110 must be applied on A2, A1 and A0 respectively and 111 must be applied on /A2, /A1 and /A0 respectively. In summary, for a single page selection, the Ai and /Ai bits are complemented bits. When the CCWP operation is used, the value of Ai and /Ai bits depends on which word lines has to be selected.

Moreover, note that whatever the operating mode (user and test), the read operation in a 150nm eFlash memory takes around 20/25ns irrespective of the word or memory size and remains negligible compared to programming operations.

5.1.2 TYPES OF AFS IN AN EFLASH

In this part, we define the possible AF combinations in an eFlash. To ease the explanations, only two addresses and their two corresponding memory cells are considered. This representation can be applied for the word line decoders as well as for the bit line decoders. Addi and Addj corresponds to memory cells Ci and Cj respectively. Addi and Addj are addresses of an entire word line or bit line. In the same way, memory cells Ci and Cj can be considered as stand alone memory cells or as a set of memory cells sharing the same word line or bit line addressed with Addi and Addj.

As presented in [VAN98a], functional faults within the address decoders may result in the four following subtypes of AFs:

subtype1: with a certain address, no cell will be accessed.

subtype2: there is no address with which a cell can be accessed, the cell is never accessed.

subtype3: with a certain address, multiple cells are accessed simultaneously.

subtype4: a certain cell can be accessed with multiple addresses.

Note that AFs must be at least the combination of two subtypes from the above list. From these different subtypes, we can then classify the AFs into two families. These two families are described in Figure 5.2 and Figure 5.3.

Figure 5.2 illustrates the first AF family in which a cell can be accessed by a maximum of one address. This AF family is referred to as Single Access AFs (SA_AFs). Figure 5.3 illustrates the second family of AFs in which a cell can be accessed by more than one address. This second family is referred to as Multiple Access AFs (MA_AFs).

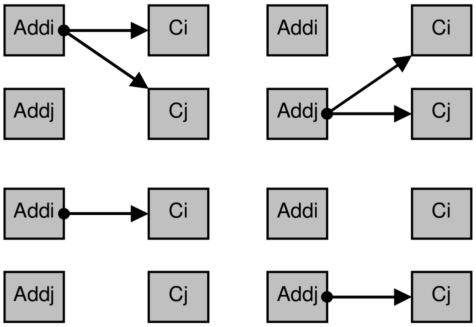


Figure 5.2: Single Access AFs (SA_AFs)

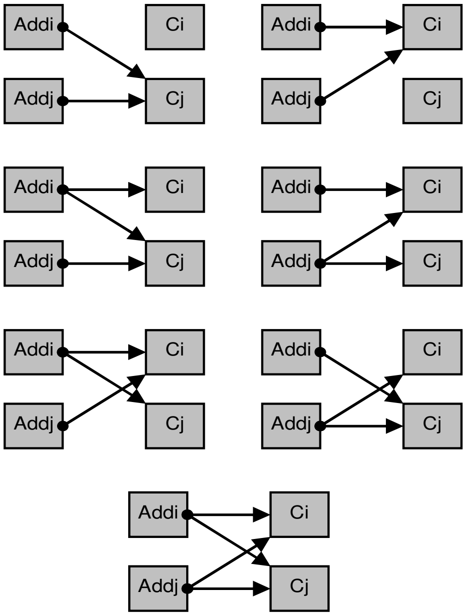


Figure 5.3: Multiple Access AFs (MA_AFs)

5.1.3 Basic Operations For AFs Detection

As discussed in Chapter 1, an erase operation performed on the FloTOx core cell changes the threshold voltage of the FG-transistor to a high VT (VTH), which is then interpreted by the sense amplifier as a logic ‘1’ (no current passes through the bit line during the read operation). During the write operation, the threshold voltage of the sense transistor changes to a low VT (VTL) interpreted as a logic ‘0’ by the sense amplifier (a current passes through the bit line during the read operation). From these electrical behaviors, some remarks can be made that will be used to define the AF detection sequence:

Remark 1: A read operation performs on a virgin FloTOx core cell (a cell not erased and not written) provides a logic ‘1’ as no current passes through the cell.

Remark 2: If an address does not access any cell, the data read is a logic ‘1’ as the sense amplifier does not measure any current.

Remark 3: If an address selects two core cells, which are connected to the same bit line and containing opposite data (‘0’ and ‘1’), the data read is a logic ‘0’. In fact, in that case, the sense amplifier measures the current passing through the cell having the logic ‘0’. The same behavior occurs if an address selects two core cells which are connected to the same word line and containing opposite data (‘0’ and ‘1’). This assumption is due to eFlash specificities.

Detection of SA_AFs

The detection of SA_AFs can be performed using a global pattern approach. The solution consists in writing all the cells in one time (chip write: CW) and then read the expected value ‘0’. This basic detection of SA_AFs is described in Figure 5.4.

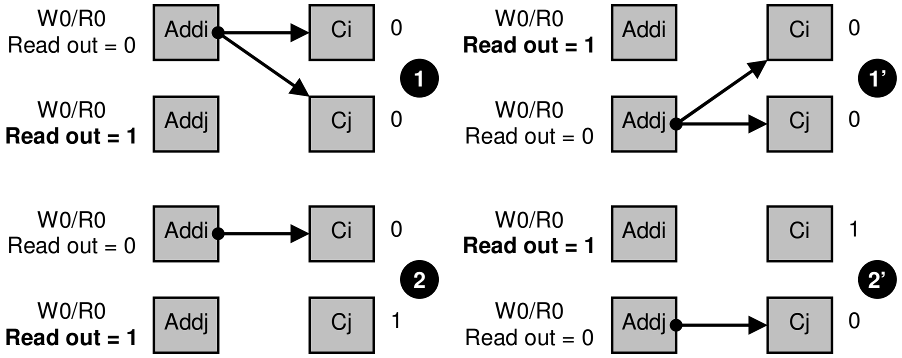


Figure 5.4: Basic detection of SA_AFs

In the first example, SA_AFs #1, a write operation is performed in one time on Addi and Addj. Cells Ci and Cj contain a logic ‘0’. During the read sequence, Addi gives a logic ‘0’ whereas Addj gives a logic ‘1’. The fault is sensitized and observed. For the dual representation SA_AF #1’, the same phenomenon occurs as for SA_AF #1. For the second examples, SA_AFs #2 and #2’, the same behavior occurs but this time the content of the unselected cell is always stuck-at logic ‘1’. Thus, this simple detection sequence (CW and Read) allows testing all SA_AFs.

Detection of MA_AFs

Detecting MA_AFs is harder than detecting SA_AFs and cannot be obtained with global patterns only, i.e. CE and/or CW patterns. To find the optimal sensitization sequence of MA_AFs, we have processed in an exhaustive way by applying all possible patterns to the addresses Addi and Addj. This method was performed for each MA_AFs case depicted in Figure 5.2. In Table 5.1, this exhaustive method is applied to the MA_AFs #3. The two first columns describe the operations acted at the addresses Addj for t=t_0 and Addi for t=t_{0+1}. The next two columns represent the data contained in Ci and Cj respectively. And the last two columns are the results after a read operation performed on Addj and Addi.

Sequence Number	**Programming operation**		**State of Ci at t_i**		**State of Cj at t_i**		**Result from a read operation**	
	Addj at t_0	Addi at t_{0+1}	t_0	t_{0+1}	t_0	t_{0+1}	Addj at t_{0+2}	Addi at t_{0+3}
0	Φ	Φ	*virgin*	*virgin*	*virgin*	*virgin*	‘1’	‘1’
1	Erase (‘1’)	Erase (‘1’)	*virgin*	*virgin*	‘1’	‘1’	‘1’	‘1’
2	**Write (‘0’)**	**Erase (‘1’)**	*virgin*	*virgin*	‘0’	‘1’	**‘1’**	**‘1’**
3	**Erase (‘1’)**	**Write (‘0’)**	*virgin*	*virgin*	‘1’	‘0’	**‘0’**	**‘0’**
4	Write (‘0’)	Write (‘0’)	*virgin*	*virgin*	‘0’	‘0’	‘0’	‘0’

Table 5.1: Exhaustive method to find the MA_AFs #3 sensitization sequence

With the help of Table 5.1 the sensitization sequences detecting the MA_AFs #3 can be deduced. The two possible sensitization sequences (2 and 3) are written in bold in the table. After the sequence 2, a logic ‘0’ is expected on Addj and a logic ‘1’ on Addi but caused to the MA_AF a logic ‘1’ is read on Addj and a logic ‘1’ on Addi. During the sequence 3, a logic ‘1’ is expected on Addj and a logic ‘0’ on Addi but caused to the MA_AF a logic ‘0’ is read on Addj and Addi. Thus, these two sequences detect the MA_AFs #3.

This exhaustive method is also applied for the other cases of MA_AFs and the cross checking of the resulting truth tables associated to each MA_AFs case (#3 to #6) provides a sensitization sequence. From this exhaustive method, the basic sensitization sequence found is composed of two steps, the first step performs an erase operation (‘1’) at the two addresses Addi and Addj, and the second step performs a write ‘0’ on Addi. Figure 5.5 presents the application of the detection sequence to all MA_AFs cases. However, note that the opposite sequence corresponding to the sequence 3 in Table 5.1 could be performed to achieve the same sensitization.

For the MA_AFs #3, during the read operation of Addi and Addj, the content of cell Cj is read two times, a logic ‘0’ instead of the expected ‘0’ on Cj and ‘1’ on Ci. For the MA_AFs #3’, as for the MA_AFs #3 the content of the same cell (Ci) which is fixed to a logic ‘0’ is read two times.

The MA_AFs #4’ example is more complex than the previous ones because this time, Addj selects two cells (Ci and Cj) containing opposite data. According to Remark 3, the data read with

Addj is a logic ‘0’.

For the MA_AFs #5, Addj addresses cell Ci containing a logic ‘0’. During the read operation, a logic ‘0’ is read for both Addi and Addj. MA_AF #5’ behaves as MA_AFs #4’ because Addj selects two cells containing opposite data. Finally MA_AFs #6 gives two logic ‘0’ during the read operation.

In each case the expected values were ‘0’ for Ci and ‘1’ for Cj but due to the MA_AFs, a logic ‘0’ is always read on both Addi and Addj addresses.

Thus, there is a detection sequence to test complex AFs such as MA_AFs. This sequence is more complex than for the SA_AFs but remains enough simple to apply in Flash memories.

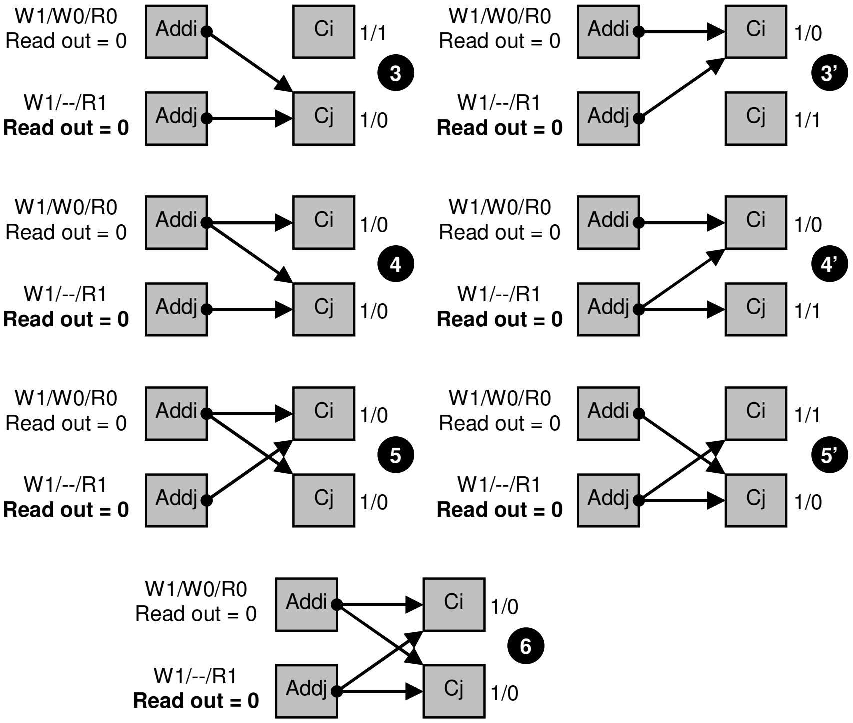


Figure 5.5: Basic sensitization of MA_AFs

5.1.4 SIMILARITIES BETWEEN AFS IN EFLASH AND RAM

Based on the previous AF analysis, the optimization of the global test flow of an eFlash memory is possible by exploiting some similarities existing between SA_AFs / MA_AFs and classical RAM fault models such as Stuck-At and Coupling Fault models.

For example, the faults in the SA_AFs family can be compared to Stuck-At Faults ‘1’ (SAF1) because the faulty address gives always a logic ‘1’.

For the MA_AFs family, the faulty behavior induced by each fault operates as a coupling fault. In fact, for the MA_AFs #3 and #3', one cell is selected by two different addresses, such that writing at a certain address induces a write on the other one. The same behavior occurs for the erase operation. In that case, a cell can be the victim and the other the aggressor or vice-versa. From the RAM test literature, MA_AFs #3 and #3' behave as State Coupling Faults, denoted as SCF<0,0> and SCF<1,1>. To test such fault, the two addresses must be written with two opposite data. This statement is shown by the Table 5.1 in which two sensitization sequences are possible.

Finally, for the MA_AFs #4, #4', #5, #5' and #6 a complex faulty behavior like a coupling phenomenon also occurs. In this case, as a write or an erase at any address induces the same operation on its coupled neighborhood, the equivalent fault model is also a State Coupling Fault: SCF<0,0> and SCF<1,1>. However, due to the context the fault model SCF proposed must be restricted to the decoding plan considered. In the WLi or BLj address decoder plan, only the SCF <0,0> and <1,1> involving respectively word line or bit line addresses must be tested. In the following, the fault model restriction will be called SCF*.

By exploiting these similarities, an efficient test strategy can be developed to detect not only all AFs but also almost all other faults that may affect an eFlash.

5.2 Concurrent Approach For Testing Address Decoder Faults

In this section, an efficient strategy to test AFs is proposed using a minimal number of programming operations to reduce testing time. Note that this described test strategy is based on the concurrent programming capability available in almost all stand-alone or embedded Flash memories. Nevertheless, if this capability is not built in, a low overhead logic (DfT) can be added to the Flash in order to get it. Moreover, by using the similarities between AFs in eFlash and RAM, we want to optimize the global eFlash flow to test all fault models from the actual list of faults established in Chapter 3. The following subsections present the proposed test strategy for AFs testing, first in the word line decoder and next in the bit line decoder.

5.2.1 AFs Testing in the WL$_I$ Decoder

For detecting SA_AFs, we have previously given the test sequence consisting in a full Chip Write (CW) followed by a read of the entire eFlash array. This sequence is able to detect all possible SA_AFs equivalent to the SAF1 model. Such a test sequence can easily be performed as an eFlash has a special programming mode to act the same operation (erase and write) in one time on all core cells. For a 150nm eFlash technology built with FloTOx core cells, this special mode allows erasing or writing the whole memory in 10ms. Thus, the testing time of SA_AFs along the word line decoder will take 10ms for the programming operations and less than two milliseconds for the read operations. Note that the read operation depends on the eFlash memory size but remains always negligible in comparison with the programming time.

The test strategy is more complex for detecting MA_AFs. From the previous section, the MA_AFs has been shown as equivalent to coupling faults (SCFs*). Coupling fault testing has to consider all combinations involving two cells (aggressor cells and victim cells). For MA_AFs testing, all possible conflicts between two addresses which correspond to all possible address couples have to be considered. Let us assume that the eFlash has M word lines. The total number NB of faulty address couples can be deduced as:

- $$NB = M \times \frac{M - 1}{2} \qquad (1)$$

Note that Equation (1) does not give the total number of possible MA_AFs but rather the number of possible faulty address couples. For example, let us consider a set of four distinct addresses (A0, A1, A2 and A3) for which the six possible faulty address couples are as follows:

A0 → A1 and A0 → A2 and A0 → A3

A1 → A2 and A1 → A3

A2 → A3

If an eFlash memory of 1024 pages is considered, NB reaches 523776 possible faulty address couples. From this very high number of possible faulty address couples, it is clear that the basic sensitization sequence to test MA_AFs is not applicable as it would take around 2100s.

A first solution to detect MA_AFs should consist in using a March test algorithm able to detect coupling faults. However, as mentioned previously, this class of solution is not applicable as March tests are not compatible with eFlash testing. The main solution generally used to detect AFs in an eFlash consists in using the Diagonal 0 pattern [SHA97]. The goal of this pattern is to write a diagonal of 0's in the array. Figure 5.6 presents an example of a Diagonal 0 pattern applied on a 16x16 eFlash array.

```
0111111111111111
1011111111111111
1101111111111111
1110111111111111
1111011111111111
…
1111111111111101
1111111111111110
```

Figure 5.6: Diagonal 0 pattern for a 16x16 eFlash

Let us consider again the example of a 1024 pages eFlash. The resulting number of write operations used to perform the Diagonal 0 pattern will be 1024. This approach reduces considerably the test time of MA_AFs although it still remains significant, i.e. about 4s in this example.

It is defined as follows:

- $T_{\mathrm{Diag0}} = \mathrm{CE} + \mathrm{PW} \times \mathrm{NbPage}$ (2)

Note that in Equation (2), the time required for the read operation is not taken into account as it is considered as negligible compared to the programming time.

The test strategy proposed in this paper allows detecting all AFs in the word line decoder in a time much lower compared to the Diagonal 0 pattern. To do that, the March approach used to test Coupling Faults in word oriented memories [VAN98b is combined with the basic sensitization sequence presented previously for detecting MA_AFs. The resulting test strategy looks like the method proposed in [KAU74] [ABR90] to detect open and short defects that may affect chip wire interconnections using the IEEE1149.1 standard.

The first pattern of the proposed strategy is a succession of ‘0’ and ‘1’ with a distance of δ = 1 between them. This pattern is equivalent to a checkerboard pattern. Here this pattern is represented on a set of eight different addresses:

A0	A1	A2	A3	A4	A5	A6	A7
0	1	0	1	0	1	0	1

If the address A0 and the seven address couples possible with this address are considered, we see that our first pattern is able to detect half of all MA_AFs combinations involving this address. Generally, with address A0 as a reference, the possible MA_AFs involving A0 and the addresses located at a distance of (k+1)*δ are tested with k ∈ {0, 2, 4 , …} and (k+1)*δ < number of addresses. In the previous example, the addresses coupled with A0 are A1, A3, A5 and A7.

The detection of the other address couples is done in the same way. The second pattern is obtained by setting δ = 2:

A0	A1	A2	A3	A4	A5	A6	A7
0	0	1	1	0	0	1	1

With this second pattern, the possible MA_AFs involving A0 and addresses located at a distance of (k+1)*δ with δ = 2 from A0 are tested.

The possible faulty address couples are A0/A2 and A0/A6. With the same method, a third pattern is built by setting δ = 4:

A0	A1	A2	A3	A4	A5	A6	A7
0	0	0	0	1	1	1	1

This third pattern allows the detection of the remaining faulty address couple A0/A4. Here too, the distance between A0 and A4 is (k+1)*δ with δ = 4.

With the three previous patterns, all MA_AFs involving A0 are detected. Moreover, due to the periodicity of each pattern all the other possible faulty address couples corresponding to the others MA_AFs are tested. To illustrate this statement let us consider the state diagram depicted in Figure 5.7 that represents the possible MA_AFs between four distinct addresses and the sensitizing diagram depicted in Figure 5.8 in which our sensitization sequence is applied.

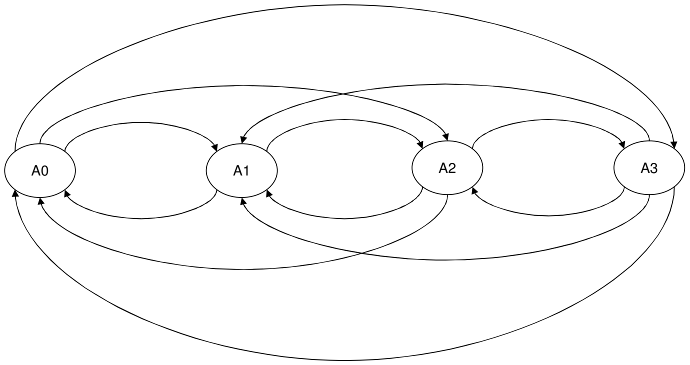


Figure 5.7: State diagram of possible MA_AFs involving four addresses

In Figure 5.7, each arc represents the possible MA_AFs. The address at the arc beginning corresponds to the aggressor address and the arc end corresponds to the victim address.

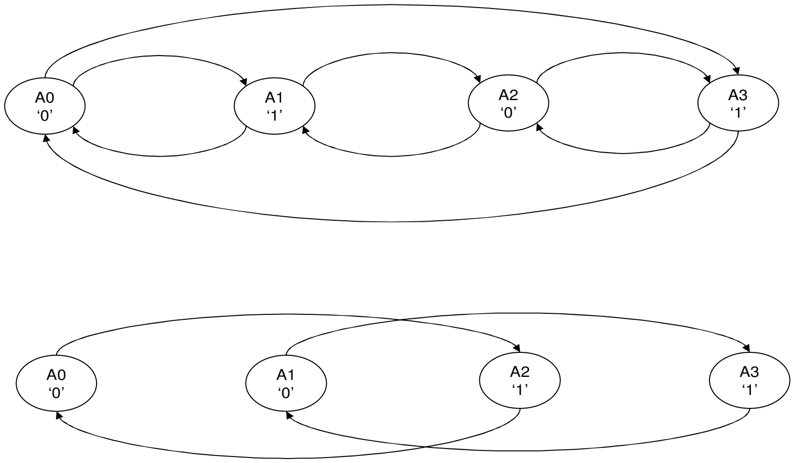


Figure 5.8: Sensitizing diagram of possible MA_AFs involving four addresses

In Figure 5.8, the first diagram corresponds to the first step of our proposed sequence, δ = 1. This first pattern tests 8/12 possible MA_AFs and the second one build with δ = 2 tests the 4 remaining MA_AFs combinations. Thus, with the help of Figure 5.8 we confirm that our proposed test sequence tests all combinations of MA_AFs involving four addresses because each possible arc is sensitized. This test sequence can be easily extended to N different addresses.

The global sequence presents a modulo 2 periodicity as the logic '0' and the logic '1' duration is multiplied by 2 between two consecutive patterns. The limit number of patterns to generate with the help of the relation can be calculated:

- $\log_2 B$ (3)

where B is the total number of addresses.

In our example involving eight addresses, the number of patterns to generate is 3. The main advantage of the proposed sequence is the logarithmic relationship defining the number of patterns to generate. In fact, if an eFlash memory of 1024 word lines is considered, the proposed sequence will contain 10 patterns only. These 10 patterns can be applied using the special programming mode of the eFlash (concurrent chip write pattern: CCWP) resulting in 10 programming operations compared to 1024 programming operations in case of the Diagonal 0 pattern.

In order to illustrate the proposed test strategy, let us consider an eFlash consisting of 8 word lines and 4 bit lines. From Equation (3), three patterns are needed to detect all MA_AFs. These three patterns are applied using the CCWP (concurrent chip write pattern). This specific eFlash programming mode requires a chip erase (CE) before each pattern programming as presented in

Figure 5.9. From this example, we can see that for applying the three initial test patterns required to detect all MA_AFs in the word line decoder, six programming operations are finally applied.

	CE	CCWP.1	CE	CCWP.2	CE	CCWP.3
WL_0	1111	0000	1111	0000	1111	0000
WL_1	1111	1111	1111	0000	1111	0000
WL_2	1111	0000	1111	1111	1111	0000
WL_3	1111	1111	1111	1111	1111	0000
WL_4	1111	0000	1111	0000	1111	1111
WL_5	1111	1111	1111	0000	1111	1111
WL_6	1111	0000	1111	1111	1111	1111
WL_7	1111	1111	1111	1111	1111	1111

Figure 5.9: MA_AFs testing in the word line decoder

5.2.2 AFs Testing in the BL_J Decoder

The problem of AF detection in the bit line decoder is equivalent to the AF detection in the word line decoder. First, detecting SA_AFs is done in the same way as previously, i.e. a chip write (CW) followed by a read operation. Secondly, detecting MA_AFs is performed with the same type of patterns than for MA_AFs in the word line decoder. For example, Figure 5.10 presents the patterns in the case of a 8-bit line eFlash. For simplicity, Figure 5.10 only represents the page corresponding to WL0.

	BL_0	BL_1	BL_2	BL_3	BL_4	BL_5	BL_6	BL_7
CE	1	1	1	1	1	1	1	1
CCWP.1	0	1	0	1	0	1	0	1
CE	1	1	1	1	1	1	1	1
CCWP.2	0	0	1	1	0	0	1	1
CE	1	1	1	1	1	1	1	1
CCWP.3	0	0	0	0	1	1	1	1

Figure 5.10: MA_AFs testing in the bit line decoder

As for detecting MA_AFs in the word line decoder, the number of patterns to apply is given by the following relation:

- $\log_2 W$ (4)

where W is the total number of bit lines. Moreover, as for testing MA_AFs in the word line decoder, a chip erase (CE) between each CCWP pattern is required.

5.2.3 AFs Testing Summary

In this chapter, a test strategy to detect all AFs that may affect the address decoders is proposed. In this section, the complete approach is summarized on a 4x4 (B=4 and W=4) eFlash memory. Figure 5.11 represents the sequence needed to test AFs in the word line decoders. A CW is first applied on the array followed by a read operation to detect all SA_AFs. Then, the four patterns allowing to detect the MA_AFs are applied.

CW		CE	CCWP.1		CE	CCWP.2	
0000		1111	0000		1111	0000	
0000		1111	1111		1111	0000	
0000		1111	0000		1111	1111	
0000		1111	1111		1111	1111	
	↑ Read			↑ Read			↑ Read

Figure 5.11: Word lines AFs testing

In the same way, Figure 5.12 presents the test sequence to detect AFs in the bit line decoder.

CE	CCWP.1		CE	CCWP.2	
1111	0101		1111	0011	
1111	0101		1111	0011	
1111	0101		1111	0011	
1111	0101		1111	0011	
		↑ Read			↑ Read

Figure 5.12: Bit lines AFs testing

This sequence allows the detection of all MA_AFs in the bit line decoder. Note that the detection of SA_AFs in the bit line decoder is already done during the word lines AFs testing by the CW operation in the previous sequence presented Figure 5.11.

The global test time of the proposed test strategy can be expressed as follow:

- $T = CW + (CE + CCWP) \times (\log_2 B + \log_2 W)$ (5)

in which CW, CE and CCWP each take 10ms. Once again, in Equation (5) the time required for the read operation is not considered as it is negligible compared to the programming time.

Let us now compare the resulting test time of the proposed test strategy with that of the Diagonal 0 pattern. Considering a 1024x1024 eFlash array (B = 1024 and W = 1024), our test strategy will take around 410ms compared to 4s with the Diagonal 0 pattern. Our strategy reduces **by a factor of 10** the resulting test time without degradation of the AF coverage.

5.3 Optimization of the AFs Testing Strategy

Up to now, we have seen how to test AFs by considering separately the word line and the bit line address decoders. In this section we show how these two approaches can be combined to further reduce the resulting test time.

Let us first deal with the problem of MA_AFs testing. Thanks to the modulo 2 periodicity of the proposed test strategy, a compaction of the two previous sequences (Figures 5.11 and 5.12) is possible. Considering the basic MA_AFs detection sequence, two opposite data have to be written in two different word lines or bit lines. Let us consider the 8x8 eFlash example presented in Figure 5.13. In this sequence, each pattern (CCWP.1 to CCWP.3) is preceded by a CE and followed by a read operation.

CE	CCWP.1	CE	CCWP.2	CE	CCWP.3	
1111 1111	0101 0101	1111 1111	0011 0011	1111 1111	0000 1111	
1111 1111	1111 1111	1111 1111	0011 0011	1111 1111	0000 1111	
1111 1111	0101 0101	1111 1111	1111 1111	1111 1111	0000 1111	
1111 1111	1111 1111	1111 1111	1111 1111	1111 1111	0000 1111	
1111 1111	0101 0101	1111 1111	0011 0011	1111 1111	1111 1111	
1111 1111	1111 1111	1111 1111	0011 0011	1111 1111	1111 1111	
1111 1111	0101 0101	1111 1111	1111 1111	1111 1111	1111 1111	
1111 1111	1111 1111	1111 1111	1111 1111	1111 1111	1111 1111	
	↑ Read		↑ Read			↑ Read

Figure 5.13: Compaction of the proposed test strategy for MA_AFs detection in 8x8 eFlash

So, with the help of such a compaction technique, the resulting number of programming operations is reduced compared to the initial MA_AFs test strategy presented in the previous section. However, when applying this compaction technique, the eFlash topology (square or rectangle) must be considered. In case of a square array, the compaction has already been shown in Figure 5.13 for a 8x8 eFlash. The problem is different in case of a rectangle array. Let us consider a 4x8 eFlash example presented in Figure 5.14. This time, the array is composed of 8 bit lines and 4 word lines. In such case, detection of MA_AFs in the word line decoder requires 2 patterns ($\log_2 4 = 2$) while 3 patterns ($\log_2 8 = 3$) are needed for the bit line decoder. As more patterns are required for the bit line decoder, the last pattern is applied to the whole memory (see CCWP.3 in Figure 5.14).

CE	CCWP.1	CE	CCWP.2	CE	CCWP.3	
1111 1111	0101 0101	1111 1111	0011 0011	1111 1111	0000 1111	
1111 1111	1111 1111	1111 1111	0011 0011	1111 1111	0000 1111	
1111 1111	0101 0101	1111 1111	1111 1111	1111 1111	0000 1111	
1111 1111	1111 1111	1111 1111	1111 1111	1111 1111	0000 1111	
	↑ Read		↑ Read			↑ Read

Figure 5.14: Compaction of the proposed test strategy for MA_AFs detection in a 4x8 eFlash

In summary, for an eFlash using a rectangle array, the total number of patterns needed to detect all MA_AFs in both decoders is defined by the widest decoder.

Then, the global test time of the compacted test strategy can be defined as follow:

- $T_{AF_{op}} = CW + (CE + CCWP) \times \max(\log_2 B, \log_2 W)$ (6)

in which, once again, the time required for the read operation is not considered.

Let us now compare the resulting test time of the proposed compacted AFs test strategy with that of the Diagonal 0 pattern. Considering again a 1024x1024 eFlash array (B = 1024 and W = 1024), the compacted AFs test strategy will take around 210ms compared to 4s for the Diagonal 0 pattern. Our strategy reduces **by a factor of 20** the resulting test time without degradation of the AF coverage.

5.3.1 AFs Testing Time Evaluation

In this section, the interest of using the proposed strategy for testing AFs in a global eFlash test flow is shown. Let us first start by a summary of our previous work on the eFlash array. Studies presented in Chapter 3 have conducted to a fault list that covers actual defects occurring in the array of FloTOx core cells. As a main result, the following list of faults that have to be tested in order to guarantee the eFlash array functionality is given:

- Stuck-at-faults
- Transition faults
- State coupling faults
- To this fault list, the AFs (SAF1, SCF*<0,0> and SCF*<1,1>) are obviously added.

A global test flow generally used to test eFlash is called 5-steps test sequence and has been described in Chapter 2. This test sequence is composed of basic patterns as CE (Chip Erase), CW (Chip Write), CKB (Checkerboard), CKBI (Checkerboard inverse) and Diag0 (Diagonal 0 pattern). Considering the previous fault list, we have to check if this global test flow is able to detect all the faults. To do that, the fault simulation tool presented in Chapter 2 is used to evaluate the efficiency of the global test flow. The resulting fault coverage is reported in the second column of Table 5.2. From these data, the global test flow is able to detect 100 % of SAFs, TFs, AFs but only 55% of SCF. The last line in Table 5.2 reports the resulting test time for three eFlash sizes (1, 2 and 4Mbits eFlash). Once again, the time required for the read operations is negligible compared to programming times.

With the help of our fault simulation tool, the efficiency of our proposed test strategy in both its initial version and its improved version with compaction has been evaluated. Results are reported in Column 3 and Column 4 respectively. A first comment is that both AFs test strategies detect all SAFs, TFs and AFs. The AFs test strategy in its compacted version offers the better test time but it does not detect all SCF. Only the initial AFs test strategy allows to detect SCF with a good coverage rate close to 83%.

These comparisons show the interest of the proposed test strategy in detecting all AFs as well as all faults that may affect the eFlash array. Moreover, the test time of the proposed approach increases only logarithmically with the size of the memory array and not linearly as for a conventional global test flow. From the test time data shown in Table 5.3, the initial AFs test strategy reduces the test time by a factor of 9.8, 18.6 and 34 for a 1 Mbits, 2 Mbits and 4 Mbits eFlash respectively.

		Global test flow	**AFs test strategy**	**Compacted AFs test strategy**
SAF		100 %	100 %	100 %
TF		100 %	100 %	100 %
AF		100 %	100 %	100 %
SCF		43 %	83 %	25 %
Test time	1 Mb eFlash	~ 4.1s	~ 420ms	~ 220ms
	2 Mb eFlash	~ 8.2s	~ 440ms	~ 240ms
	4 Mb eFlash	~ 16.4s	~ 480ms	~ 280ms

Table 5.2: Test sequence evaluations

5.4 CONCLUSION

In this Chapter, we have addressed AFs testing in eFlash memories. These faults have been studied and classified in two families, SA_AFs and MA_AFs, according to their faulty behaviors. Based on this classification, a test strategy that reduces drastically the resulting test time, compared to the Diagonal 0 pattern [SHA97] currently used, has been proposed.

This new AFs test strategy has been compared with a global flow generally used to test eFlash. From these comparisons, the proposed solution is the only one that covers all fault models that may affect an eFlash, i.e. all SAFs, TFs, AFs, and SCF are tested. Moreover, our solution drastically reduces the test time compared to the '5'-Pattern Sequence generally used to test eFlash; by a factor of 34 for a 4Mbits eFlash.

General Conclusion

Different types of memories can be embedded in a SoC such as SRAM, DRAM, EEPROM and eFlash. The increased use of portable electronic devices such as mobile phones and digital camera produces a high demand for Flash memories.

We have seen that the mainstream operation of an eFlash memory is based on the floating gate transistor concept on which charges can be stored and removed. Two typical mechanisms to transfer electric charges from and into the floating gate transistor are used: Hot Carrier Injection (HCI) and the Fowler-Nordheim (FN) tunneling effect. Due to several advantages such as low power operation and very good endurance, the FN tunneling effect is the most often used in an eFlash memory. Moreover, as the FN tunneling effect allows a very low current consumption during the programming operation, this mechanism offers good opportunities to decrease the programming time by implementing concurrent programming modes. As a result, FN tunneling is extensively used for both erase and write operations in embedded Flash (eFlash) memories.

One of the main problems in the design of eFlash memories lies in the use of high electric field during the FN tunneling effect. This high electric field can disturb an eFlash core cell or affect its reliability. The second problem of such memories lies in their high integration density associated to their particular manufacturing process steps (floating gate transistor concept) that makes the eFlash memory more and more prone to inter or intra core cell defects. Thus, the production of efficient eFlash memory tests is a big issue.

The first step to develop efficient test solutions for eFlash memories is the analysis of test solutions used for other memories. The solutions currently used to test RAM memories have been first analyzed. These solutions mainly focus on the detection of a set of functional fault models owing to linear algorithms called March tests. Regarding the description of each possible fault models associated to RAM, we have shown that this set of tested faults is not always realistic in an eFlash context and must be redefined. In addition, due to the eFlash slow programming time, eFlash memories can not be tested by a March algorithm. In the literature, test solutions other than March test solutions can be found. These test solutions called the 7-pattern test sequence or the 5-steps basic sequence have been analyzed. By using our home made functional fault simulator, we have analyzed the robustness of the 5-steps basic sequence from a fault coverage point of view. The result from our fault simulator shows that the 5-step basic sequence presents some limitations in the detection of certain faults like State Coupling Faults.

To conclude on the previous statements, two main aspects have to be considered during eFlash testing, namely eFlash technology specificities and the slow programming time. The eFlash technology is important in the fault modeling process whereas the slow programming time has to be considered for test sequences or algorithms development.

This is the reason why after this first analysis of existing test solutions, we have analyzed a standard FloTOx core cell structure used to build an eFlash memory array. This first step was to carry out an exhaustive analysis of actual defects that may occur in the FloTOx core cell and in the memory array. Defects extracted from the 150nm eFlash technology were opens and bridges (pure or resistive), capacitive coupling but also defects related to the floating gate transistor such as bad oxide in the tunnel window. For each defect, their impact on the eFlash behavior have been analyzed. Certain defects have required the development of a FloTOx electrical model to perform simulations in order to analyze precisely the faulty behavior that they induce. At the end of this exhaustive analysis, a list of functional fault models has been proposed. These fault models have some similarities with RAM fault models with some restrictions and modifications.

Considering the limitations of existing test solutions, new test methods have been developed for eFlash and the existing ones have been enhanced.

A first method used to test the oxide thickness variation of the tunnel window has been proposed. This test solution is based on the eFlash main feature that allows mass programming operations of core cells but also on the coupling phenomenon existing between two adjacent core cells in the array. This test solution is based on a unique pattern to program a whole eFlash memory chip in which we alternate ‘0’ and ‘1’ on each column. Note that after our modeling process coming from the first failure mechanism analysis, the oxide thickness variation of the tunnel window was modeled as a State Coupling Fault (SCF) and that the 5-basic step sequence most often used to test eFlash is not able to test such fault. Now thanks to our test solution all actual SCFs are tested keeping a reduced test time (a few ms).

The second test method developed was dedicated to the Address decoder Faults (AFs) detection. Once again, a test solution saving test time owing to the concurrent programming modes available in a FloTOx eFlash memory array has been developed. This test solution is based on existing solutions used to test buses in a complex chip like SoC or intra core cell faults in a word-oriented RAM. Moreover, with the help of our eFlash fault simulator we have shown that this solution is able to test all actual fault models occurring in FloTOx eFlash and established thanks to our previous failure mechanism analysis. This solution reduces the testing time to a factor of 34 for a 4Mbits eFlash in comparison with the 5-basic steps sequence.

The next target of our study will be the test of faults that may affect other parts of the eFlash memories, such as the sense amplifiers, the level shifters and the high voltage generation block. Some investigations will be carried out in the field of faults occurring during read operations and affecting the dynamic behavior of eFlash memories.

Scientific Contributions

U.S PATENT

O. Ginez, B. Godard and J.-M. Daga
"Method and System for Providing a Nonvolatile Content Addressable Memory using a single FloTOx Element"
US Patent, USPTO-11650104, January 2007

JOURNALS

[JETTA08] **O. Ginez, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, J.-M. Daga,**
"Electrical Simulation Model of the 2T-FLOTOX Core cell for Defect Injection and Faulty Behavior Prediction in eFlash"
Accepted for publication in JETTA: Journal of Electronics Testing: Theory and Applications, Elsevier Publishers.

PUBLICATIONS IN INTERNATIONAL CONFERENCES PROCEEDINGS

[VTS06] **O. Ginez, J.-M. Daga, M. Combe, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel**
"An Overview of Failure Mechanisms in Embedded Flash Memories"
24th IEEE VLSI Test Symposium, Berkeley, USA, May 2006, pp. 108-113.

[DTIS06] **O. Ginez, J.-M. Daga, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel**
"Embedded Flash Testing: Overview and Perspectives"
1st IEEE International Conference on Design and Test of Integrated Systems, Tunis, Tunisia, September 2006, pp. 86-92.

[VTS07] **O. Ginez, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, J.-M. Daga**
"Retention and Reliability Problems in Embedded Flash Memories: Analysis and Test of Defective 2T-FLOTOX Tunnel Window"
25th IEEE VLSI Test Symposium, Berkeley, USA, May 2007, pp. 47-52

[ETS07] **O. Ginez, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, J.-M. Daga**

"Electrical Simulation Model of the 2T-FLOTOX Core cell for Defect Injection and Faulty Behavior Prediction in eFlash"

12th IEEE European Test Symposium, Freiburg, Germany, May 2007, pp. 77-84

[ITC07] **O. Ginez, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, J.-M. Daga**

"A Concurrent Approach for Testing Address Decoder Faults in eFlash Memories"

To appear in Proc. of IEEE International Test Conference, Santa Clara, USA, October 2007.

PUBLICATIONS IN NATIONAL CONFERENCES PROCEEDINGS (FRANCE)

[JNRDM07] **O. Ginez, J-M. Daga, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel**

"Test des Mémoires Flash Embarquées : Analyse de la perturbation entre cellules FloTOx voisines durant une phase de programmation"

Proc. JNRDM : Journées Nationales du Réseau Doctoral de Microélectronique, Lille, France, 14-16 Mai 2007

INTERNATIONAL SEMINARS

[SETS05] **O. Ginez, J-M. Daga, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel**

"New Test Methodologies for Embedded and Stacked Flash Memories"

South European Test Symposium (SETS), Pitztal, Austria 2005

References

[ABR90] Abramovici M., Breuer M.A., Friedman A.D.
"Digital Systems Testing and Testable Design"
Computer Science Press, 1990.

[ALA01] Al-Ars Z., Van de Goor A.J.
"Static and Dynamic Behavior of Memory Cell Array Opens and Shorts in Embedded DRAMs"
Proc. Design, Automation and Test in Europe, 2001, pp. 496-503.

[BOR05] Borri S., Hage-Hassan M., Dilillo L., Girard P., Pravossoudovitch S., Virazel A.
"Dynamic Fault Models for Embedded-SRAMs: analysis and Test"
JETTA'05: Journal of Electronic Testing: Theory and Applications, Kluwer Academic Publishers, 2005, Vol. 21, pp. 169-178.

[BRO98] Brown W. D., Brewer J.E.
"Non-Volatile Semiconductor Memory Technology"
IEEE Press, New York, 1998.

[CHA87] Chao C. C.,White M. H.
"Characterization of Charge Injection and Trapping in Scaled SONOS/MNOS Memory Devices"
Solid-State. Electronics, 1987, Vol. 30, pp. 307.

[CHE77] Chen P. C.
"Threshold-Alterable Si-Gate MOS Devices"
IEEE Trans. On Elec. Devices, May 1977, Vol. 24, n°5, pp. 584.

[DEL87] Dellin T. A., Mc Whorter P. J.
"Scaling of MONOS non-volatile memory transistors"
Proc. Electrochemical Society, 1987, Vol. 87, n°10, pp. 3.

[DEK90] Dekker R.
"A Realistic Fault Model and Test Algorithms for Static Random Access Memories"
IEEE Trans. On Computers, June 1990, Vol. 9, n°6, pp. 567 - 572.

[DIC76] Dickson J. F.
"On-Chip High-Voltage Generation in NMOS Integrated Circuits Using an Improved Voltage Multiplier Technique"
IEEE Journal of Solid-State Circuits, June 1976, Vol. 11, n° 3, pp.374-378.

[DIL05] Dilillo L.
"Test de Fautes Dynamiques dans les Mémoires SRAM"
PhD Thesis, France, Montpellier, May 2005.

[GIL02] Gill M., Lowrey T., Park J.
"Ovonic Unified Memory-A High-performance Non-volatile Memory Technology for Stand Alone Memory and Embedded Applications"
Proc. ISSCC: Solid-State Circuits Conference, 2002, pp. 158.

[HAR78] Harari E., Schmitz L., Troutman B., Wang S.
"A 256 bit Non-Volatile Static RAM"
IEEE ISSCC: Solid-State Circuits Conference, 1978, pp. 108.

[KAU74] Kautz W.H.
"Testing of Faults in Wiring Interconnects"
IEEE Trans. On Computers, April 1974, Vol. 23, n° 4, pp. 358-363.

[LAI01] Lai S., Lowrey T.
"OUM-A 180 nm Nonvolatile Memory Cell Element Technology For Stand Alone and Embedded Applications"
IEEE IEDM: International Electron Device Meeting, 2001, pp. 365.1-365.4

[MAE89] Maes H.E., Groeseneken G., Lebon H., Witters J.
"Trends in Semiconductor Memories",
Microelectronics Journal, 1989, Vol. 20, pp. 9.

[MAR82] Marinescu M.
"Simple and Efficient Algorithms for Functional RAM Testing"
Proc. IEEE ITC: International Test Conference, 1982, pp.236-239.

[MOH01] Mohammad M., Saluja K.
"Flash Memory Disturbances: Modeling and Test"
Proc. IEEE VLSI Test Symposium, 2001, pp. 218-224.

[MOH03a] Mohammad M., Saluja K.
"Simulating Disturb Faults in Flash Memories Using SPICE Compatible Electrical Model"
IEEE Trans. On Electron Devices, November 2003, Vol. 50, n° 11, pp. 2286-2291.

[MOH03b] Mohammad M., Saluja K.
"Electrical Model for Program Disturb Faults in Non-Volatile Memories"
Proc. International Conference on VLSI Design, 2003, pp. 217-222.

[NIG98] Niggemeyer D., Redeker M., Otterstedt J.
"Integration of Non-classical Faults in Standard March Tests"
Proc. International Workshop on Memory Technology, Design and Testing, 1998.

[PAR98] Park Y.-B., Schroder D.K.
"Degradation of Thin Tunnel Gate Oxide under Constant Fowler-Nordheim Current Stress for Flash EEPROM"
IEEE Trans. On Electron Devices, June 1998, Vol. 45, n° 6, pp. 1361-1368.

[POR02] Portal J.M. & al.
"Floating gate EEPROM Cell Model Based on MOS Model 9"
IEEE ISCAS: International Symposium on Circuits and Systems, 2002.

[ROS77] Rössler B., Müller R.
"Electrically Erasable and Reprogrammable Read-Only Memory using the n-channel SIMOS One-Transistor Cell"
IEEE Trans. On Electron. Devices, 1977.

[SDC91] **"**IEEE Standard Definitions and Characterization of Floating gate Semiconductor Arrays"
IEEE 1005-1998, Revision of the IEEE std. 1005-1991.

[SHA97] Sharma A.K.
"Semiconductor Memories: Technology, Testing and Reliability"
IEEE Press, Piscataway, 1997.

[SIA03] Semiconductor Industry Association (SIA)
"International Technology Roadmap for Semiconductors (ITRS)"
2003, http://www.sia-online.org/

[SIA06] Semiconductor Industry Association (SIA)
“International Technology Roadmap for Semiconductors (ITRS)”
2006, http://www.sia-online.org/

[SUK81] Suk D.S., Reddy S.M.
“A March Test for Functional Faults in Semiconductor Random Access Memories”
IEEE Trans. On Computers, 1981, Vol. 30, pp982-985.

[VAN96] Van de Goor A.J.
“March LR: A test for Realistic Linked Faults”
Proc. IEEE: VLSI Test Symposium, 1996, pp. 272 - 280.

[VAN98a] Van de Goor A.J.
“Testing Semiconductor Memories, Theory and Practice”
COMTEX Publishing, 1998.

[VAN98b] Van de Goor A.J., Tlili I.B.S.
“March Tests for Word-Oriented Memories”
Proc. DATE: Design Automation and Test in Europe, Paris, 1998, pp. 501-509.

[VAN00] Van de Goor A.J., Al-Ars Z.
“Functional Memory Faults: A Formal Notation and a Taxonomy”
Proc. IEEE VLSI Test Symposium, May 2000, pp. 281-289.

[YAM91] Yamada N., Ohno E., Nishiuchi K., Akahira N., Takao M.
“Rapidphase Transitions of GeTe-Sb2Te3 Pseudobinary Amorphous Thin Films for an Optical Disk Memory”
Journal of Applied Physics, 1991, Vol. 69, n° 5, pp. 2849-2857.

[YAR82] Yaron G., Prasad S., Ebel M., Leong B.
“A 16k EEPROM Employing New Array Architecture and Designed-in Reliability Features”
IEEE Journal of Solid State Circuit, 1982, Vol. 17, n°5, pp. 833.

[YEH02] Yeh J.C., Wu C.F., Cheng K.L., Chou Y.F., Huang C.T., Wu C.W.
“Flash Memory Built-in Self-Test Using March-like Algorithms”
Proc. IEEE DELTA: Int. Workshop on Electronic Design, Test, and Applications, Christchurch, January 2002, pp. 137–141.

List of Figures

List of Tables

UNIVERSITE MONTPELLIER II
SCIENCES ET TECHNIQUES DU LANGUEDOC

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE MONTPELLIER II

Spécialité : Microélectronique.

Formation Doctorale : Systèmes Automatiques et Microélectroniques.

Ecole Doctorale : Information, Structures et Systèmes.

présentée et soutenue publiquement

par

Olivier GINEZ

le 29 Novembre 2007

Modélisation de Fautes et Test des Mémoires Flash

JURY

M. Dominique Dallet, Professeur, IMS, ENSEIRB, Bordeaux	Rapporteur
M. Jean-Michel Portal, Maître de Conférence, L2MP, Polytech' Marseille	Rapporteur
M. Jean-Michel Daga, Sté ATMEL, Rousset	Examinateur
M. Arnaud Virazel, Maître de Conférence, Université Montpellier II – LIRMM	Examinateur
M. Joan Figueras, Professeur, Politecnica de Catalunya, Barcelone	Membre Invité
M. Patrick Girard, Directeur de Recherche CNRS, LIRMM	Directeur de Thèse
M. Serge Pravossoudovitch, Professeur, Université Montpellier II – LIRMM	Co-Directeur de Thèse

1. Introduction aux mémoires Flash embarquées

Un grand nombre de mémoires, comme les SRAM, DRAM, EEPROM et les Flash, peuvent être embarquées dans les systèmes sur puce (SoC). La constante augmentation des systèmes électroniques portables, comme les appareils photo numériques ou les téléphones mobiles, accroît fortement l'utilisation de mémoires Flash embarquées (eFlash). La mémoire eFlash est de type non-volatile, elle est effaçable et programmable électriquement. Sa faible consommation ainsi que sa grande densité d'intégration font que la mémoire eFlash est très populaire pour les applications embarquées. D'un point de vue fonctionnel, la mémoire eFlash comprend une matrice de cellules mémoires, des décodeurs d'adresse ainsi que des amplificateurs de lecture. De plus, la mémoire eFlash dispose d'un bloc particulier appelé HVG pour *High Voltage Generator*, permettant de générer une tension élevée (>15v) que l'on utilise durant les phases d'écriture et d'effacement. Les mémoires eFlash sont construites à partir d'un transistor à double grille, une grille de contrôle et une grille flottante servant au stockage des charges. Il existe deux principes physiques pour faire transiter les charges dans les transistors à double grille et donc de les programmer, le mécanisme dit d'injection d'électrons chauds (HEI) et le mécanisme dit d'effet tunnel Fowler-Nordheim. Dans notre étude, nous considérerons tout particulièrement les cellules mémoires eFlash de type FloTOx (Floating gate Tunnel Oxide) composées d'un transistor à double grille et d'un transistor de sélection (Figure 1.1). Pour programmer les cellules eFlash de type FloTOx, le mécanisme le plus couramment utilisé est l'effet tunnel Fowler-Nordheim (FN). L'utilisation cet effet tunnel FN permet de réduire la consommation durant l'effacement ou l'écriture du point mémoire eFlash mais aussi d'en accroître l'endurance. De plus, le faible courant utilisé durant la programmation par effet tunnel FN permet de réduire les temps d'écriture et d'effacement en implémentant des modes de programmation parallèle. Trois opérations peuvent être réalisées sur les cellules eFlash FloTOx, l'effacement, l'écriture et la lecture. La phase d'effacement consiste à injecter des électrons dans la grille flottante (FG) du transistor à double grille grâce à l'application d'une haute tension sur différents nœuds de la cellule mémoire. Pour injecter des charges dans la FG, il faut appliquer une haute tension sur le nœud Vrefi du transistor à double grille alors que son drain doit être fixé à la masse (Figure 1.1). Durant l'effacement, la cellule mémoire est dans un état 'on' et permet de fixer le nœud BLj au même potentiel que Vss (0v). Il est important de noter que l'opération d'effacement est activée parallèlement sur toutes les cellules mémoires d'une même page. A la fin d'un effacement, des électrons sont stockés dans la grille flottante (FG) du transistor double grille et sa tension de seuil VT en est augmentée, soit VTH sur la Figure 1.2. D'un point de vue purement fonctionnel, une tension de seuil haute VTH correspond à un état logique '1' de la cellule mémoire eFlash.

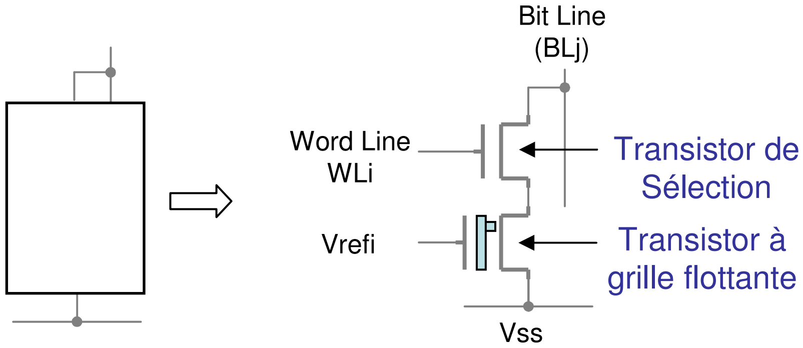


Figure 1.1: Schéma d'une cellule eFlash de type FloTOx.

L'opération d'écriture a pour but de retirer les charges stockées dans la grille flottante après l'opération d'effacement en fixant le nœud Vrefi à 0v tout en maintenant BLj à une haute tension. Durant cette opération d'écriture, les électrons sont retirés de la grille flottante et la tension de seuil VT du transistor double grille devient très faible VTL sur la Figure 1.2.

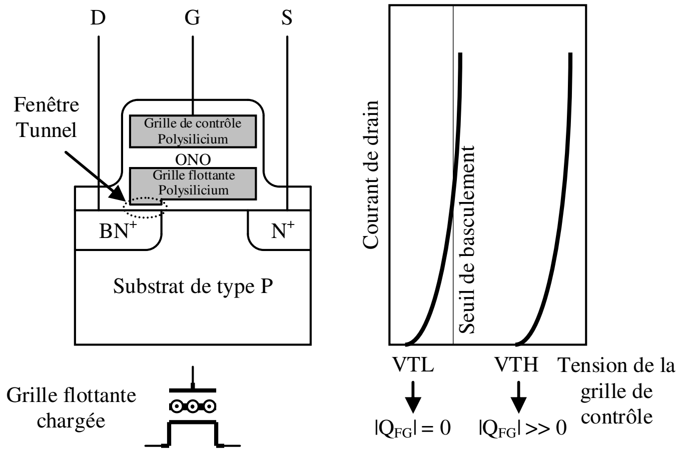


Figure 1.2: Illustration du concept de grille flottante.

D'un point de vue fonctionnel, une tension de seuil VTL correspond à un niveau logique '0'. La fenêtre définie par la différence entre VTL et VTH est appelée marge de lecture. Il est important de noter que l'opération d'écriture, contrairement à l'effacement, se fait avec une granularité bit c'est-à-dire cellule par cellule. En mode lecture, la cellule Flash est polarisée pour ne fournir aucun courant à travers le nœud BLj lorsqu'elle est dans un état logique '1' et qu'elle en fournisse lorsqu'elle est dans un état logique '0'. Ce courant de cellule est détecté à l'aide d'un amplificateur de lecture conçu à cet effet. La faible vitesse de programmation est une des principales caractéristiques des mémoires Flash embarquées. Cette particularité est liée au mécanisme de programmation par effet tunnel FN qui implique de faible quantité de courant durant l'écriture ou l'effacement de la mémoire, de l'ordre de 10^{-12}-10^{-11}A.

2. Analyse des défauts *durs* et des défauts *résistifs* dans les mémoires Flash embarquées

La première partie de ce travail de thèse a été essentiellement l'étude et l'analyse des défauts et mécanismes de défaillance susceptibles de survenir dans les mémoires Flash embarquées (eFlash) conçues avec des cellules FloTOx. Cette analyse a permis de donner une liste de modèles de faute liée aux comportements fautifs induits par la présence de ces défauts dans une matrice mémoire eFlash.

Environnement de notre étude

Tout d'abord il est important de noter que les défauts et mécanismes de défaillances traités dans notre étude sont tirés de données silicium extraites d'une technologie eFlash 150nm et mises à notre disposition par un partenaire industriel. La Figure 2.1 décrit la matrice mémoire eFlash dont nous nous sommes servis pendant toute la durée de notre analyse. Il s'agit d'une matrice mémoire eFlash de 3*3 bits en technologie FloTOx. Les cellules FloTOx sont arrangées en parallèle, il s'agit donc d'une structure de type NOR.

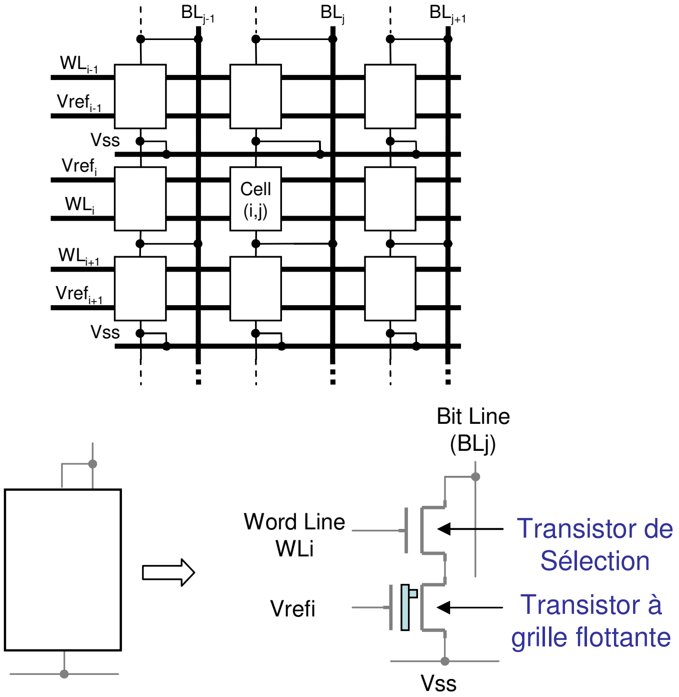


Figure 2.1: Matrice eFlash 3*3bits à base de cellules FloTOx.

Sur la Figure 2.1, la cellule(i,j) est la cellule correspondante a la *word line* WL$_i$ et à la *bit line* BL$_j$. Notons que la cellule(i,j) est la seule cellule sur laquelle seront effectuées des opérations d'écriture ('0') et de lecture. Lors de notre étude les potentiels appliqués à la cellule durant les phases d'effacement, d'écriture et de lecture sont présentés dans le Tableau 2.1.

	Effacement	Ecriture	Lecture
BLj	0v	12v	1v
BLx	0v	HZ	HZ
WLi	14v	14v	3.3v
WLy	0v	0v	0v
Vss	0v	1.2v	0v
Vrefi	12v	0v	0.7v
Vrefy	HZ	HZ	HZ

Table 2.1: Potentiels appliqués à la cellule FloTOx durant l'étude

Concernant les défauts et mécanismes de défaillance étudiés, une liste les classant suivant deux familles a été proposée. Voici cette liste:

- **Défauts durs:** courts-circuits et circuits ouverts dans la matrice.
- **Défauts résistifs:** courts-circuits et circuits ouverts résistifs dans la matrice.

Les défauts durs

Sur notre matrice eFlash 3*3 bits de départ, nous avons injecté des courts-circuits et des circuits ouverts en respectant leur probabilité d'occurrence et la topologie layout de la matrice mémoire. Ceci est représenté par la Figure 2.2.

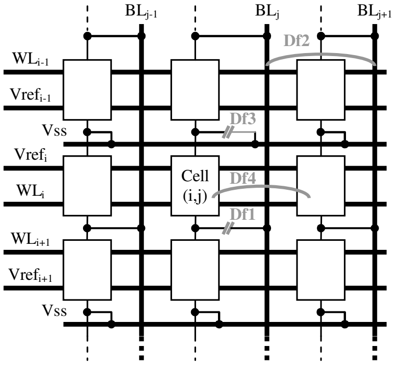


Figure 2.2: Défauts durs dans une matrice eFlash 3*3 bits.

Df1 – Contact ouvert sur la *bit line*

D'un point de vue layout, deux cellules adjacentes partagent le même contact. Si ce contact est ouvert ($R = \infty\Omega$), le drain de chacun des deux transistors de sélection est déconnecté de la *bit line* . A cause de ce défaut, l'opération d'effacement reste possible mais malheureusement l'opération d'écriture ne peut pas être effectuée. De plus, pendant la phase de lecture, quelque soit le contenu de la cellule, aucun courant ne pourra être délivré par la cellule à travers la *bit line* et donc un état logique '1' sera toujours lu dans les cellules (i,j) et (i+1,j). D'un point de vu fonctionnel ce comportement s'apparente à une faute de collage à 1 (CA1).

Df2 – Court circuit entre 2 *bit lines* (au niveau 1 de métal)

Un court circuit entre deux *bit lines* ne peut être sensibilisé que lors des phases d'écriture sur une cellule appartenant à l'une ou l'autre des *bit lines* affectées par le défaut. En effet, durant l'opération d'écriture de la cellule (i,j), la *bit line* cible BL_j (voir Tableau 2.1) est fixée à 12volts et les autres *bit lines* sont dans un état haute impédance. En présence de Df2, ce potentiel 12volts est aussi appliqué à la *bit line* voisine défectueuse et cela implique une écriture à '0' de la cellule(i,j+1) se trouvant sur la même ligne sélectionnée que la cellule(i,j).

Du au fait que ce défaut implique la notion de cellule agresseur et cellule victime, la modélisation du comportement fautif induit par Df2 peut se faire par une faute de couplage, *FC*. Plus précisément, il s'agit d'une faute de couplage d 'état, *FCE*. Comme ce comportement est visible uniquement durant la phase d'écriture d'un état logique '0', le modèle de faute exact est le couplage d'état FCE(0,0). Rappelez-vous qu'un effacement se fait simultanément sur une page entière et de ce fait le comportement fautif inverse du à Df2 pendant les phases d'effacement n'est pas possible.

Df3- Contact de source ouvert

Deux cellules mémoires (i,j) et (i-1,j) partagent le même contact de source Vss. En présence de ce défaut les opérations d'effacement et d'écriture sont encore possibles. En revanche, lors des phases de lecture aucun courant ne peut être lue par l'amplificateur de lecture à cause de ce défaut. Les deux cellules affectées par ce défaut sont toujours interprétées comme stockant un état logique '1' et le comportement fautif du défaut Df3 peut être modélisé par une faute de collage à '1' (CA1).

Df4- Court circuit au niveau de la grille flottante (au premier niveau de poly silicium)

A cause du faible espace qu'il y a entre les transistors à grille flottante placés sur une même page, des courts circuits entre deux grilles flottantes peuvent être observés. Ce défaut de court circuit a pour conséquence d'impliquer le même état sur deux cellules mémoire appartenant à la même page. Le modèle de faute relatif à ce défaut est le même que celui utilisé pour modéliser le défaut Df2, il s'agit d'un FCE(0,0). Une fois encore la faute de couplage d'état FCE(1,1) n'est pas possible grâce au propriété d'effacement parallèle de la mémoire.

Les défauts résistifs

Dans cette partie nous analysons le comportement fautif induit par la présence de courts circuits résistifs dans notre matrice mémoire eFlash 3*3 bits. D'un point de vue layout ce type de défaut peut survenir sur les mêmes couches de métallisation (poly/poly, *i.e.* court circuit résistif entre une *word line* et une ligne Vrefi); cela correspond aux défauts Df8 et Df9, ou bien un court circuit résistif peut survenir entre deux niveaux layout différents (métal/polysilicium, *i.e.* court circuit résistif entre une *word line* et une *bit line*); ceux sont les défauts Df5, Df6 et Df7. Il est important de noter que dans notre analyse, chaque défaut implique une ligne (*bit line* ou *word line*) appartenant à notre cellule cible, à savoir la cellule(i,j).

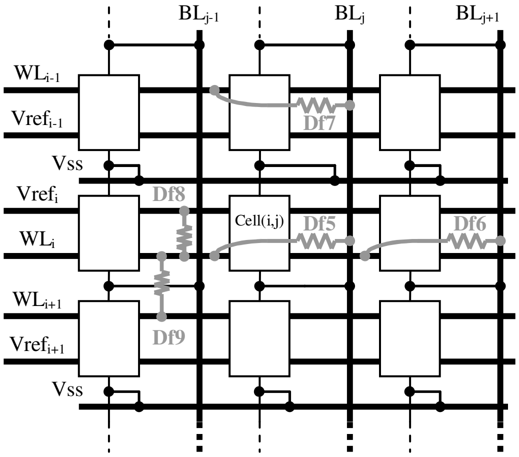


Figure 2.3: Courts circuits résistifs dans une matrice eFlash 3*3 bits.

Pour mener notre étude nous avons donc injecté cinq défauts (Figure 2.3):

- **Df5:** court circuit résistif entre WL_i et BL_j. Ce défaut concerne la cellule(i,j).
- **Df6:** court circuit résistif entre WL_i et BL_x. Ce défaut concerne une cellule(i,x) partageant la même *word line* que la cellule(i,j).
- **Df7:** court circuit résistif entre WL_y et BL_j. Ce défaut concerne une cellule(y,i) partageant la même *bit line* que la cellule(i,j).
- **Df8:** court circuit résistif entre WL_i et $Vref_i$. Ce défaut concerne toutes les cellules partageant la même *word line* et la même ligne Vref (ligne de contrôle).
- **Df9:** court circuit résistif entre WL_i et WL_{i+1}. Comme pour Df8, ce défaut concerne toutes les cellules partageant la même *word line*.

Basé sur cette configuration d'injection de défauts, nous avons mené une analyse qualitative des mécanismes de défaillance associés à ces défauts de courts circuits résistifs sur notre matrice eFlash 3*3 bits. Cette analyse qualitative a été menée suivant les trois opérations:

- Effacement de la *word line* WL_i
- Ecriture de la cellule(i,j)
- Lecture de la cellule(i,j)

Effacement de la *word line* WL_i

Pendant l'opération d'effacement, 14v est appliqué sur WL_i et 0v sur BL_j. En présence de Df5, une forte impulsion de courant peut apparaître et impliquer une chute de tension sur la haute tension de programmation (HVG). Par conséquent, l'opération d'effacement n'est pas correctement effectuée sur la cellule (i,j) car beaucoup moins de charges peuvent être injectées dans la grille flottante du transistor double grille. Le VTH de la cellule(i,j) est affecté. Pour un défaut Df5 de 500Ω, la haute tension de programmation HVG diminue d'environ 1v ce qui implique une diminution du VTH de la cellule(i,j) de 860mV. Le même comportement survient en présence de Df6 car ce défaut produit aussi un court circuit résistif entre la haute tension HVG et la masse (niveau 0volt). En ce qui concerne Df7 et Df8, ils ne présentent aucun impact sur le comportement de la mémoire. Concernant Df9, une chute de la haute tension de programmation peut une fois encore survenir durant la sélection de la page à programmer (WL_i). Toutefois il est à noter que ce comportement n'apparaît que pour certaines valeurs de Df9.

Ecriture de la cellule(i,j)

Pendant la phase d'écriture de la cellule(i,j), la présence de Df5 n'a pas d'impact sur l'état de la cellule(i,j) car la haute tension est appliquée simultanément sur les lignes WL_i et BL_j. En présence de Df6, la tension présente sur WL_i peut être transmise sur le nœud BL_{j+1}. Suivant la taille de Df6, le niveau de tension de BL_{j+1} peut être suffisamment importante pour entraîner une opération d'écriture sur la cellule(i,j+1) mais sans impacter le contenu de la cellule(i,j). Df7 implique le même comportement que Df5 durant une phase d'effacement. En effet, Df7 opère un court circuit résistif entre la haute tension (HVG) et la masse, ce qui entraîne une chute de cette haute tension. Par conséquent, la valeur du VTL de la cellule cell(i,j) est impactée. Le défaut Df8 peut perturber l'opération d'écriture de la cellule(i,j) parce que la présence de ce défaut entraîne la montée de la ligne $Vref_i$ vers une haute tension et donc réduit considérablement le champs électrique entre la grille flottante et le drain de la cellule(i,j). Enfin, Df9 induit comme pour l'opération précédente, une chute sur la haute tension et par conséquent l'opération d'écriture de la cellule(i,j) n'est pas bien effectuée.

Lecture de la cellule(i,j)

En présence de Df5, le noeud BL_j se charge à cause du courant induit par ce défaut. A cause de ce courant de charge à travers Df5 et suivant sa valeur, un un masquage du courant fourni par la cellule(i,j) peut être observé si celle-ci contient un état logique '0'. De ce fait, aucun courant n'est mesuré par l'amplificateur de lecture. Notez que ce comportement fautif survient pour des valeurs de Df5 en dessous de 100kΩ. Concernant Df6, ce défaut induit une augmentation du potentiel de WL_i à cause du court circuit que ce défaut créée entre le nœud WL_i et BL_{j+1}. Le défaut Df7 implique un comportement fautif de la cellule(i,j) lorsque celle-ci contient un état logique ‘1’. Dans ce cas aucun courant ne doit être mesuré par l’amplificateur de lecture (lecture d’un ‘1’) mais comme la présence de Df7 créée un court circuit entre le nœud BL_j et la masse, un état logique ‘0’ est tout le temps lu en lieu et place d’un état logique ‘1’ attendu. Concernant les défauts Df8 et Df9, aucun impact sur le comportement fonctionnel de la mémoire n’est à observer.

Modélisation de faute

Dans les investigations précédentes, nous avons analysé tous les défauts réalistes appartenant aux deux familles de défauts possibles pouvant induire un comportement fautif sur une mémoire eFlash conçues avec des cellules de type FloTOx. Dans le Tableau 3.9, nous avons attribué à chaque défaut précédemment étudié un modèle de faute fonctionnelle et la limite de validité de ce modèle en fonction de la taille du défaut concerné.

Défaut	**Cellule(i,j)**		**Cellule(i,j+1)**		**Cellule(i-1,j)**	
	Taille du défaut	Modèle de faute	Taille du défaut	Modèle de faute	Taille du défaut	Modèle de faute
Df1	NA	CA1	NA		NA	CA1
Df2	NA	FCE(0,0)	NA	FCE(0,0)	NA	
Df3	NA	CA1	NA		NA	CA1
Df4	NA	FCE(0,0)	NA	FCE(0,0)	NA	
Df5	< 1 MΩ	CA1	-	-	< 100 kΩ	CA0
Df6	< 10 kΩ	FT (0 à 1)	100kΩ < … < 400kΩ	FCE(0,0)	-	-
Df7	< 100 kΩ	CA0	-	-	-	-
Df8	< 10 MΩ	CA1				
Df9	< 100 kΩ	CA1				

Tableau 2.1: Modèles de faute associés aux défauts analysés.

Il faut toutefois remarquer que les défauts Df1 à Df4 ne peuvent être associé à une limite de validité car ces défauts appartiennent à la famille des défauts durs et ne peuvent prendre que deux valeurs extrêmes de résistivité 0Ω et ∞Ω.Grâce à ce tableau nous pouvons constaté que tous les modèles de faute proposés sont tirés de la littérature relatives au test des mémoires RAM : collage à 0 (CA0), collage à 1 (CA1), faute de transition (FT) et des fautes plus complexes comme les fautes de couplage d’état FCE(0,0). Grâce à cette liste de fautes il est aisé de trouver une stratégie de test pour les détecter. Dans le chapitre qui suit une analyse des problèmes relatifs aux deux autres familles de défaut est proposée.

3. Détection de la variation d'épaisseur d'oxyde dans la fenêtre tunnel

La présence d'une variation d'épaisseur d'oxyde au niveau de la fenêtre tunnel d'un transistor à grille flottante peut impacter la rétention et la fiabilité d'une cellule mémoire FloTOx. Comme ces problèmes n'affectent pas directement le comportement des cellules FloTOx mais en affecte plutôt la fiabilité, il faut toutefois trouver une solution pour les détecter. Dans cette partie nous démontrons comment détecter cette variation d'épaisseur d'oxyde grâce aux phénomènes de couplage inter *bit lines*. Nous détaillerons l'origine du couplage entre *bit lines* et ensuite nous expliquerons comment utiliser ce couplage non agressif pour une cellule saine, pour détecter une variation d'épaisseur d'oxyde de la fenêtre tunnel.

Origine du couplage entre deux *bit lines*

A cause du faible espace autorisé par le noeud technologie entre deux *bit lines* adjacentes, nous pouvons observer une capacité de couplage non négligeable entre deux *bit lines* (C1 sur la Figure 3.1). Un pont diviseur capacitif (Figure 3.1) peut alors apparaître à cause de cette capacité de couplage mais aussi de la capacité équivalente (capacité C2 sur la Figure 3.1) à la bit line faite en métal et se trouvant au dessus d'un puit dopé P.

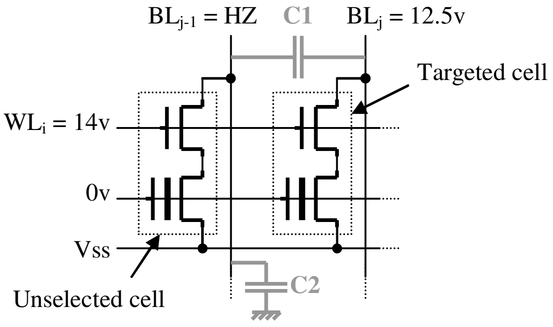


Figure 3.1: Couplage inter *bit lines*

A cause de ce couplage, la haute tension appliqué sur BL_j pendant une phase d'écriture induit une montée non désirée du potentiel BL_{j-1} de la cellule couplée voisine.

Pour illustrer ceci des mesures électriques sur des mémoires Flash en 150nm ont été menées et nous avons pu observer que la tension sur la bit line de la cellule non sélectionnée pouvait atteindre 7volts.

Utilisation du couplage inter *bit lines* dans la détection d’une variation d’oxyde tunnel

Nous savons, grâce une bonne caractérisation des mémoires FloTOx en 150nm, que cette tension de perturbation du au phénomène de couplage inter *bit lines* n’impacte pas les cellules saine. Par contre que se passe t-il lorsque nous en présence de cellule présentant une variation de leur épaisseur d’oxyde tunnel ?

Pour répondre à cette question nous avons mené une étude sur la variation de la tension de seuil haute (VTH) d’une cellule FloTOx victime. Cette cellule est soumise à une tension de perturbation pendant 2ms (Tp) et à une variation d’épaisseur d’oxyde de sa fenêtre tunnel. Le Tableau 3.1 résume les résultats ainsi obtenus. La première colonne donne la variation d’épaisseur d’oxyde tunnel, les colonnes suivantes donnent les variations de VTH (ΔVTH) depuis une valeur nominale (VTH=+2,5v) et ce pour différentes valeurs de tension de perturbation.

		Tension de perturbation					
		5v	6v	7v	8v	9v	10v
Epaisseur d’oxyde tunnel (Å)	54	0.168	1.145	2.285	3.429	4.572	5.716
	57	0.047	0.774	1.901	3.044	4.188	5.332
	60	0.011	0.437	1.519	2.660	3.804	4.948
	63	0.003	0.191	1.141	2.277	3.420	4.564
	66	0.001	0.065	0.777	1.895	3.037	4.181
	69	0.000	0.019	0.453	1.515	2.655	3.798
	72	0.000	0.005	0.214	1.141	2.272	3.415
	75	0.000	0.001	0.083	0.784	1.891	3.033

Tableau 4.1: Variation ΔVTH d’une cellule perturbée présentant une variation d’oxyde tunnel

Grâce au Tableau 3.1, nous pouvons voir le principal impact d’une tension de perturbation quand la fenêtre d’oxyde tunnel d’une cellule FloTOx présente une variation. En effet, si l’on reprend la valeur de tension de perturbation de 7volts, mesurée sur une eFlash en 150nm, lorsque l’épaisseur d’oxyde tunnel de la cellule FloTOx subit une variation de 21Å par rapport à sa valeur nominale, on observe une variation de sa tension de seuil VTH de 2,285volts. Cette variation de tension de seuil de la cellule suffit à la faire basculer d’un état logique ‘1’ (VTH) vers un état logique ‘0’ (VTL).

Grâce à cette analyse nous pouvons proposer un modèle de faute de ce phénomène de perturbation inter *bit lines* (couplage) associé à une variation d’épaisseur d’oxyde tunnel de la cellule FloTOx. Comme il y a la notion de cellule victime et cellule agresseur, nous sommes en présence d'une faute de couplage. Pour être plus précis dans le travail de modélisation, il s'agit d'une faute de couplage d'état, FCE (0,0).

Solution de test proposée

Nous avons vu précédemment que la tension de perturbation issue du couplage inter *bit line* pouvait être une solution de sensibilisation de la variation d'épaisseur d'oxyde tunnel sur une cellule FloTOx. Grâce à la Figure 3.1, nous pouvons exprimer cette fameuse tension de perturbation :

- $\Delta V_{BLj-1} = \Delta V_{BLj} \times [C_1 / (C_1 + C_2)]$

A l'aide de cette expression, nous pouvons facilement envisager d'augmenter la contribution du couplage capacitif grâce qu théorème de superposition. Considérons la Figure 3.2 ci-dessous:

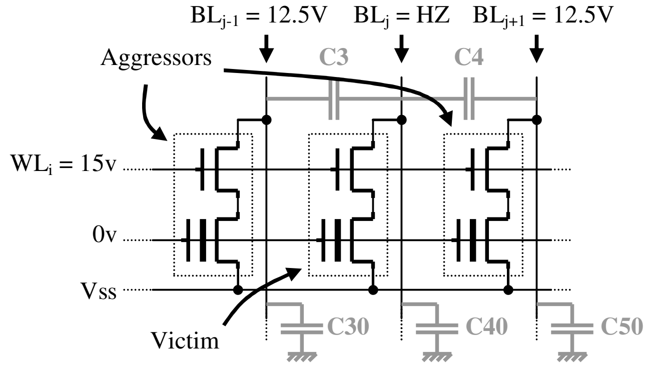


Figure 3.2: Superposition de deux tensions de perturbation

Si une cellule victime est entourée par deux cellules perturbatrices (agresseurs) sur lesquelles une opération d'écriture d'un '0' est en cours, nous pouvons écrire l'expression régissant la superposition des deux contributions de perturbation :

- $\Delta V_{BLj} = \Delta V_{BLj-1} \times \frac{C3}{C3 + C40} + \Delta V_{BLj+1} \times \frac{C4}{C4 + C40}$

Grâce à cette expression nous pouvons en déduire le motif permettant d'atteindre la tension de perturbation maximale par couplage capacitif. Il s'agit des motifs échiquier (Checkerboard=CKB) et échiquier inverse (Checkerboard Inverse=CKBI):

CKB: 10101010…1010

CKBI: 01010101…0101

Des simulations électriques ont été menées pour valider cette solution mais malheureusement les tensions de perturbation atteintes avec ces deux motifs ne sont pas suffisantes pour pouvoir détecter des variations d'oxyde tunnel dans une cellule FloTOx. En effet les tensions de perturbation atteintes en technologie eFlash 150nm à l'aide des motifs CKB et CKBI sont de l'ordre de 6volts. Aux vues des valeurs inscrites dans le Tableau 3.1, cela n'est pas suffisant pour détecter une variation d'épaisseur d'oxyde tunnel.

Pour palier à cette limitation, une solution envisagée est l'utilisation d'un mode de programmation parallèle dans lequel toutes les *word lines* de la mémoire eFlash sont sélectionnées en même temps. En effet grâce à ce mode de programmation parallèle, la tension de perturbation due au couplage capacitif peut être augmentée grâce la présence de la capacité de recouvrement C_{gd} présente sur le transistor de sélection de la cellule mémoire FloTOx. Ceci est représenté par la Figure 3.3 ci-dessous :

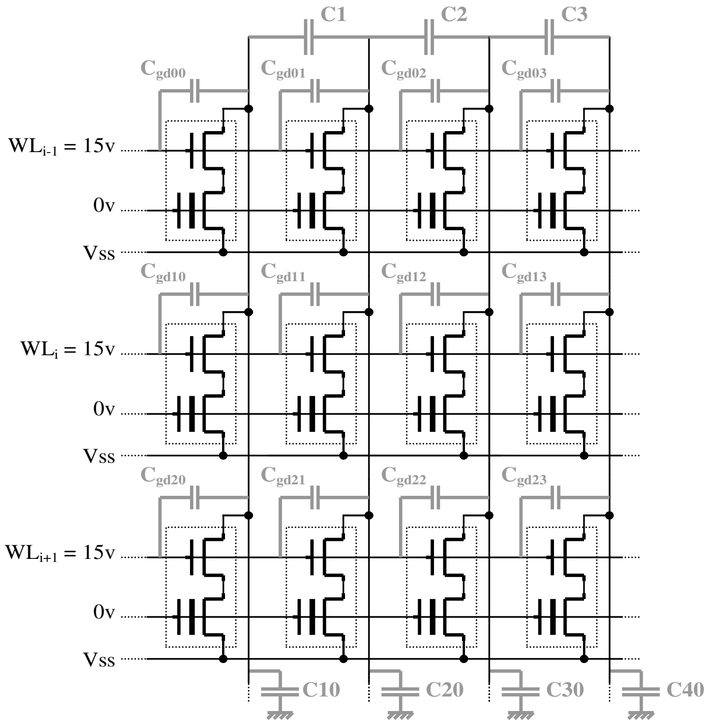


Figure 3.3: Capacités de couplage mises en jeu lors d'une programmation en mode parallèle

En technologie eFlash 150nm cette capacité C_{gd} est proche de $9E^{-17}$F et devient donc prépondérante dans la contribution à la tension de perturbation lorsqu'une mémoire eFlash dispose de plus de 1024 *word lines*. La capacité équivalente de toutes ces capacités C_{gd} en parallèle est donc proche de la valeur de la capacité de couplage entre deux *bit lines* (100fF). De ce fait le pont diviseur capacitif est augmenté et la tension de perturbation devient plus importante. De la même façon que précédemment nous avons mesuré la tension de perturbation due au couplage capacitif inter *bit lines* lorsque nous programmons des motifs CKB et CKBI en parallèle dans toute la mémoire et ce pour une technologie eFlash 150nm. Notez que la mémoire eFlash utilisées était composée de 1024 pages (*word lines*) de 64 mots de 32 bits. Le Tableau 3.2 présente les résultats ainsi obtenus.

	VBL (V)							
	B0	B1	B2	B3	…	B29	B30	B31
CKB	**6.92**	12.5	**8.15**	12.5	…	12.5	**8.15**	12.5
CKBI	12.5	**8.15**	12.5	**8.15**	…	**8.15**	12.5	**6.92**

Table 3.2: Mesure de la tension de bit line dans un mots de 32 bits en mode de sélection parallèle

Dans ce tableau nous pouvons observer que les cellules agressées par le motif CKB ou bien CKBI subissent une tension de perturbation de l'ordre de 8,15volts, ce qui est suffisamment important pour détecter de faible variation d'oxyde si l'on se réfère au Tableau 3.1 précédent. Les variations d'oxyde tunnel détectées par ce mode de programmation de motif CKB et CKBI commence à partir de 12Å. En revanche il est important de noter que notre méthode de détection est limitée car au niveau des bits de bord de la mémoire le potentiel de perturbation n'atteint que 6,92volts car ces bits de bord ne subissent que la tension de perturbation due à une seule cellule agresseur voisine. Excepté les problèmes liés aux bits de bord, notre solution de test reste robuste pour les autres bits de la matrice afin de détecter les possibles variation d'oxyde tunnel survenant dans le transistor à grille flottante.

4. Test des fautes de décodeur d'adresse (FAs) dans les mémoires eFlash

Dans la littérature, de nombreux travaux existent sur le test des fautes d'adressage (FAs) dans les mémoires RAM et les méthodes proposées sont pour la plupart basées sur les algorithmes de type March. Ces solutions ne sont pas applicables pour tester les FAs survenant dans les eFlash à cause de la lenteur de programmation de ces mémoires. La solution la plus couramment utilisée pour tester les FAs dans les eFlash est l'utilisation du motif *Diagonale '0'* qui consiste à programmer à '0' toute la diagonale de la matrice eFlash. Cependant cette solution reste coûteuse en terme de temps de test car pour appliquer le motif *Diagonale '0'* il faut opérer un grand nombre d'opérations de programmation dépendant du nombre de pages (*word lines*) qui constituent la mémoire eFlash.

Une approche concurrente pour tester les FAs dans les eFlash

Dans ce qui suit nous proposons une solution permettant de tester toutes les FAs possibles dans une matrice mémoire eFlash et ce basé sur ses propriétés de programmation parallèle.

En premier lieu, nous allons définir toutes les combinaisons de FAs impliquant des adresses et leurs cellules mémoires correspondantes. Pour réduire la complexité de la représentation, nous considérons seulement deux adresses et leurs cellules mémoires associées. Cette représentation peut être faite aussi bien pour la partie décodage ligne (word line) que pour la partie décodage colonne (bit line). Nous notons Addi et Addj les adresses qui correspondent respectivement aux cellules mémoires Ci et Cj. Nous considérons que Addi et Addj représente l'adresse d'une ligne ou d'une colonne complète de cellules mémoires eFlash. De la même façon, Ci et Cj peuvent être considérées chacune comme étant une cellule mémoire unique ou comme un ensemble de cellules partageant la même ligne ou la même colonne ayant pour adresse respectives Addi et Addj. Il faut savoir que les fautes dans les décodeurs d'adresse (FA) sont issues des quatre sous-types de fautes suivants:

Subtype1 : avec une certaine adresse, aucune cellule ne sera accédée.

Subtype2 : il n'y a pas d'adresse avec laquelle une cellule peut être accédée. Une certaine cellule n'est jamais accédée.

Subtype3 : avec une certaine adresse, plusieurs cellules sont accédées simultanément.

Subtype4 : une certaine cellule peut être accédée avec plusieurs adresses.

Les FAs doivent être au moins la combinaison de deux sous-types de la liste mentionnée. De ces différents sous-types, nous pouvons classifier les FAs selon deux familles décrites dans les Figure 4.1 et Figure 4.2. La Figure 4.1 illustre la première famille de FAs dans lequel une cellule peut être accédée par au maximum une adresse. Cette famille de FAs est dite à *Accès Simple* (FA_ASs). La Figure 4.2 illustre la deuxième famille de FAs dans lequel une cellule peut être accédée par plus d'une adresse. Cette deuxième famille est appelée FA à *Accès Multiple* (FA_AMs).

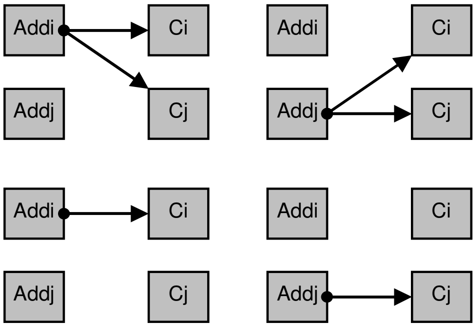


Figure 4.1: FAs à Accès Simple (FA_ASs)

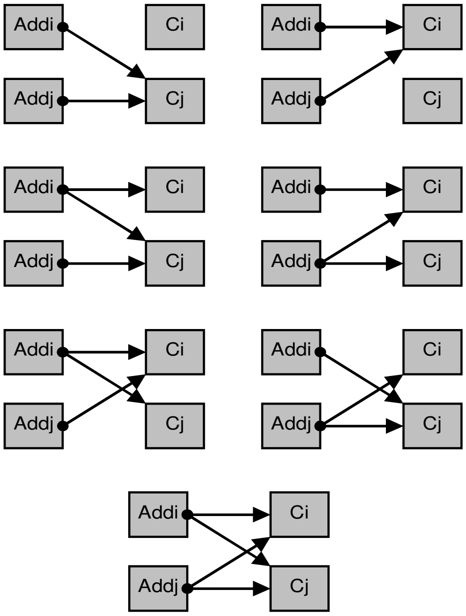


Figure 4.2: FAs à Accès Mulriple (FA_AMs)

Détection des FAs

Comme décrit dans le premier chapitre, l'effacement d'une la cellule FloTOx implique une tension de seuil haute (VTH) du transistor à grille flottante qui sera alors interprétée comme un '1' logique par l'amplificateur de lecture (aucun courant passe par la *bit line* pendant l'opération de lecture). Après l'opération d'écriture, la tension de seuil du transistor à grille flottante devient très basse (VTL) et est interprétée comme un état logique '0' par l'amplificateur de lecture (un courant passe par la *bit line* pendant l'opération de lecture). De ces comportements électriques, nous pouvons faire quelques remarques qui seront utilisées pour définir la séquence de détection d'une FA:

Remarque 1 : Une opération de lecture exécutée sur une FloTOx vierge (une cellule ni effacée et ni écrite) fournit un '1' logique en sortie de l'amplificateur de lecture car aucun courant n'est fourni par la cellule.

Remarque 2 : Si une adresse n'accède à aucune cellule, les données lues seront interprétées comme un état logique '1' car l'amplificateur de lecture ne mesure aucun courant à travers les cellules.

Remarque 3 : Si une adresse accède à deux cellules mémoires connectées à la même *bit line* et contenant des données opposées ('0' et '1'), la donnée lue à l'adresse en question sera un logique '0'. En fait, dans ce cas, l'amplificateur de lecture mesure le courant passant par la cellule contenant le '0'. Le même comportement survient si une adresse accède à deux cellules mémoires connectées à la même *word line* et contenant des données opposées ('0' et '1'). Ce comportement est du aux spécificités des mémoires eFlash.

Détection des FA_ASs

La détection de FA_ASs peut être faite à l'aide d'une méthode utilisant un motif de test global. La solution consiste à écrire un '0' logique dans toutes les cellules en même temps grâce au mode *CW* (Chip Write) et de lire la valeur prévue '0'. Ce principe de détection des FA_ASs est décrit dans la Figure 4.3.

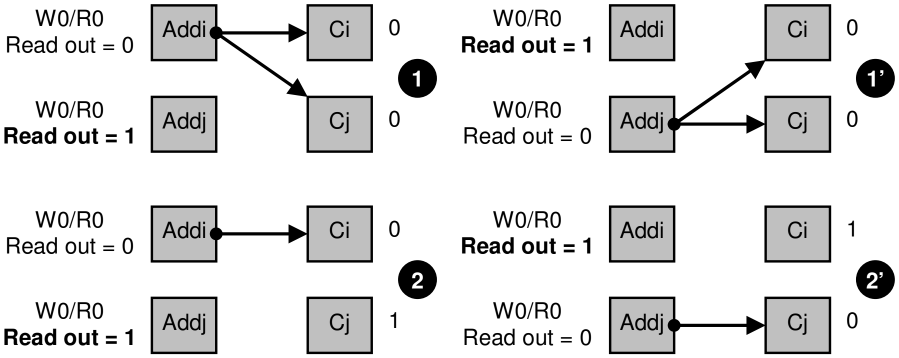


Figure 4.3: Séquence de détection des FA_ASs

Dans le premier exemple, FA_ASs #1, l'opération d'écriture d'un '0' est exécutée en une seule fois sur Addi et Addj. Les cellules Ci et Cj contiennent un logique '0'. Pendant la lecture, Addi donne un '0' logique tandis qu'Addj donne un logique '1'. La faute est sensibilisée et observée. Pour l'exemple FA_AS #1', le même comportement est observé. Pour le deuxième exemple, FA_ASs #2 et #2', le même comportement survient mais cette fois le contenu de la cellule non adressée est toujours un logique '1'. Ainsi, nous pouvons conclure que cette séquence de détection (Ecriture d'un '0' + Lecture de tous les '0') permet de sensibiliser tous les FA_ASs.

Détection des FA_AMs

La détection des FA_AMs est plus complexe que celle des FA_ASs. Pour détecter les FA_AMs, premièrement nous initialisons la matrice mémoire en exécutant le mode d'effacement parallèle (CE) ce qui initialise toutes cellules à un logique '1'. Ensuite, nous écrivons un '0' logique sur Addi seulement afin d'obtenir deux données opposées sur Addi et Addj. Finalement, une opération de lecture est exécutée. La Figure 4.4 décrit la séquence de détection des FA_AMs sur les adresses Addi et Addj. Pour le cas FA_AMs #3, pendant l'opération de lecture de Addi et Addj, nous lisons deux fois le contenu de la cellule Cj, un logique '0', au lieu de lire '0' sur Ci et '1' sur Cj. Pour le cas FA_AMs #3', une fois de plus nous lisons deux fois le contenu de la même cellule Ci (un logique '0'). Le cas FA_AMs #4' est l'exemple le plus complexe parce que cette fois, Addj sélectionne deux cellules (Ci et Cj) contenant des données opposées. Selon la Remarque 3, la donnée lue à l'adresse Addj sera un logique '0'. Pour le FA_AMs #5, Addj accède à la cellule Ci contenant un logique '0'. Pendant la phase de lecture, un logique '0' est aux adresses Addi et Addj. La faute FA_AM #5' se comporte comme la faute FA_AMs #4' parce qu'Addj sélectionne deux cellules contenant des données opposées. Finalement pendant la phase de lecture en présence de la faute FA_AMs #6, deux états logiques '0' seront lus aux adresses Addi et Addj. Dans chaque cas les valeurs prévues étaient '0' pour Ci et '1' pour Cj mais en raison des FA_AMs, un état logique '0' est toujours observé durant les phases de lecture des adresses Addi et Addj. Notre séquence proposée permet donc la sensibilisation de tous les cas de FA_AMs possibles.

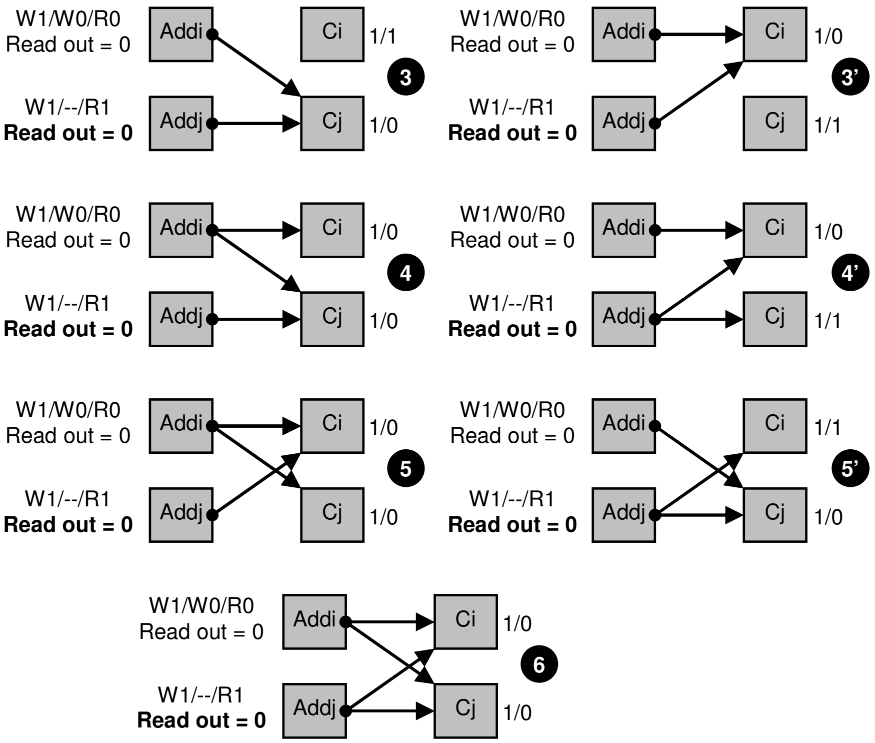


Figure 4.4: Séquence de détection des FA_AMs

Similarités entre FAs dans les eFlash et dans les RAM

Basé sur l'analyse précédente des FAs, nous pouvons essayer d'optimiser le flot de test global d'une mémoire d'eFlash en exploitant quelques similarités existantes entre FA_ASs/FA_AMs et les modèles de faute existants dans les mémoires RAM tel que les fautes de couplage FCs et les fautes de collages CAs. Par exemple, nous pouvons assimiler les fautes d'adressage de la famille FA_ASs aux fautes de collage simple à '1' (CA1) parce que l'adresse défectueuse donne toujours en lecture un état logique '1'.

Pour la famille FA_AMs, chaque comportement fautif résultant de ces modèles sont comparables à des fautes de couplage. En effet, pour le FA_AMs #3 et #3', une cellule est sélectionnée par deux adresses différentes si bien que l'écriture d'un '0' à une adresse implique la même écriture sur l'autre adresse. Ce comportement fautif est observé de la même façon durant l'opération d'effacement. Dans ce cas, une cellule peut être la victime et l'autre l'agresseur ou inversement. Dans la littérature associée au test de mémoire RAM, les fautes FA_AMs #3 et #3' se comportent comme des fautes de couplage d'état FCE, FCE<0,0> et FCE<1,1>. Pour les fautes FA_AMs #4, #4', #5 et #5', le comportement fautif peut être lui aussi associé à une faute de couplage. Dans ce cas, comme écrire ou effacer n'importe quelle adresse implique la même opération sur sa voisine couplée, il s'agit d'une faute de couplage par perturbation, FCP<w1,↑> et FCP<w0,↓>. En ce qui concerne les fautes relatives au modèle FA_AMs #6, le comportement fautif peut être aussi associé à une faute de couplage par perturbation, FCP<w1,↑> et FCP<w0,↓>.

En exploitant ces similarités, nous pouvons envisager de développer une stratégie de test efficace pour détecter non seulement tous les FAs mais aussi toutes les autres fautes qui peuvent affecter une mémoire eFlash.

Approche concurrente pour détecter les FAs

Dans cette partie, nous proposons une stratégie efficace pour tester tous les FAs possibles en utilisant un nombre minimal d'opérations de programmation afin de réduire le temps de test. Cette stratégie de test est basée sur la possibilité de programmer simultanément un grand nombre de cellules mémoires (blocs).

Test des FAs relatives aux décodeurs de *word lines* (WLi)

Pour détecter les FA_ASs, nous avons proposé précédemment une séquence de test qui consiste à écrire un '0' dans toutes les cellules de la mémoire eFlash (CW) suivi d'une lecture de tous ces '0'. Cette séquence peut détecter toutes les FA_ASs possibles. En technologie eFlash FloTOx 150nm, le mode spécial permettant d'effacer (CE) ou d'écrire la mémoire en une seule fois dure 10ms. Ainsi, le temps de test des FA_ASs dans le décodeur de WLi ne prendra que 10ms pour l'opération d'écriture d'un '0' dans toutes les cellules et moins de 2ms pour lire toutes les cellules. Notons que le temps de lecture est toujours négligé comparé au temps d'effacement ou d'écriture.

La stratégie de test est plus complexe pour détecter les FA_AMs. Dans la section précédente, nous avons vu que les FA_AMs étaient équivalentes aux fautes de couplage d'état FCE et de perturbation FCP. Nous savons que pour ce type de faute toutes les combinaisons impliquant deux cellules à des adresses différentes doivent être considérées. Pour le test des FA_AMs, nous avons donc à considérer tous les conflits possibles entre deux adresses (couplage). Supposons que la mémoire eFlash possède M *word lines* à tester. Nous pouvons donc en déduire le nombre total NB de couple d'adresses fautives possibles à tester:

- $NB = M \times \frac{M-1}{2}$ (7)

Considérons une série de quatre adresses distinctes (A0, A1, A2 et A3) pour lesquelles les six couples d’adresses fautives sont :

A0 → A1 et A0 → A2 et A0 → A3

A1 → A2 et A1 → A3

A2 → A3

Si nous considérons une mémoire eFlash de 1024 pages, nous avons 523776 couples d’adresses défectueuses possibles. Il est évident que nous ne pouvons pas appliquer la séquence de sensibilisation des FA_AMs proposée précédemment pour toutes ces combinaisons.

Une première solution pour détecter les FA_AMs pourrait être l'utilisation d'un algorithme de test de type MARCH fréquemment utilisé pour tester les mémoires RAM. Cependant, de nombreuses études ont démontrées que ce type d’algorithme n’était pas adapté à l’environnement eFlash à cause de la lenteur de programmation que présente ce type de mémoire. Cependant, il existe une solution pour pouvoir détecter les FAs dans les eFlash que de nombreux industriels utilisent. Il s’agit du motif *Diagonale de ‘0’*. Le but de ce motif de test est d'écrire une diagonale de ‘0’ dans la matrice mémoire. La Figure 4.5 est un exemple d'une diagonale de 0 appliquée à une mémoire eFlash de 16x16bits.

```
0111111111111111
1011111111111111
1101111111111111
1110111111111111
1111011111111111
…
1111111111111101
1111111111111110
```

Figure 4.5: Diagonale ‘0’ dans une matrice eFlash 16*16bits

Considérons encore l'exemple d'une eFlash possédant 1024 pages. Le nombre d’opérations d’écriture en mode page à réaliser pour avoir un motif diagonal de 0 dans cette mémoire sera 1024. Cette approche réduit considérablement le temps de test de FA_AMs bien qu'il reste toujours significatif. Si l’opération d’écriture d’une page dans la mémoire eFalsh dure 4ms, le temps total d’écriture du motif diagonale de ‘0’ sera d’environ de 4 secondes tout en négligeant comme précédemment le temps de lecture de chaque bit de la mémoire.

La stratégie de test que nous proposons ici, pour détecter toutes les combinaisons possibles de FAs dans le décodeur *word lines* réduit considérablement le temps de test en comparaison au motif diagonale de 0.

Pour faire cela, nous combinons le test MARCH utilisé pour détecter les fautes de couplage dans les mémoires RAM orientées mots avec la séquence de sensibilisation présentée dans la section précédente pour détecter les FA_AMs. La stratégie résultante possède de nombreuses similitudes avec la solution proposée pour détecter les défauts de circuits ouverts et courts circuits qui peuvent affecter les liaisons entre différents circuits utilisant la norme IEEE1149.1.

Le premier motif de la stratégie proposée est une succession de ‘0’ et ‘1’ avec une distance de 1 bit entre eux. Ce motif est équivalent à un motif damier (échiquier). Pour illustrer ce motif nous l’avons représenté sur une série de huit adresses distinctes :

A0	A1	A2	A3	A4	A5	A6	A7
0	1	0	1	0	1	0	1

Si nous considérons l'adresse A0 et les sept possibles adresses qui peuvent lui être couplées, nous voyons que notre premier motif peut détecter la moitié de toutes les combinaisons de FA_AMs possibles impliquant l’adresse A0. Avec l’adresse A0 comme référence, les FA_AMs possibles impliquant A0 et les adresses situées à une distance de $(k+1)*\delta$ sont testées avec $k \in \{0, 2, 4 , \ldots\}$ et $(k+1)*\delta <$ aux nombres total d’adresse. Dans l'exemple précédent, les adresses couplées avec A0 sont A1, A3, A5 et A7.

La détection des autres adresses couples avec A0 est faite de la même façon en utilisant un second motif permettant la détection d’autres couples où $\delta = 2$:

A0	A1	A2	A3	A4	A5	A6	A7
0	0	1	1	0	0	1	1

Avec ce second motif, nous sommes capables de détecter les FA_AMs affectant les adresses couples constituées de A0 avec les adresses situées à une distance $(k+1)*\delta$ avec $\delta = 2$ de A0. Les deux couples d’adresse possibles sont A0/A2 et A0/A6. Basé sur la même méthode nous pouvons construire un nouveau motif avec $\delta = 4$:

A0	A1	A2	A3	A4	A5	A6	A7
0	0	0	0	1	1	1	1

Ce troisième motif permet de détecter les FA_AMs affectant le couple d’adresse A0/A4. Ici aussi, la distance entre A0 et A4 est $(k+1)*\delta$ avec $\delta = 4$.

Avec les trois motifs de test précédents, nous détectons toutes les FA_AMs possibles impliquant A0. De plus, grâce à la périodicité de cette stratégie de test toutes les autres FA_AMs affectant les autres couples possibles d’adresse sont testées.

La séquence complète présente une périodicité 2 car la durée d'un état logique '0' ou d'un état logique '1' est multipliée par 2 entre chaque motif. Grâce à cette particularité, la limite de motifs à générer durant la séquence complète est donnée par l'opération $\log_2 B$ où B est le nombre total d'adresses à tester.

Dans l'exemple ci-dessus le nombre total de motifs à générer est de 3. L'avantage majeur d'une telle stratégie est la relation logarithmique liant le nombre de motifs à écrire dans la mémoire durant la séquence de test avec le nombre d'adresse possible à tester. Considérons un exemple concret de mémoire eFlash possédant 1024 *word lines*, donc 1024 adresses de *word lines* possibles, le nombre de motif à écrire durant notre séquence de test sera de 10. De plus notons que cette stratégie repose essentiellement sur la capacité de la mémoire eFlash à écrire un motif en une seul fois dans certains blocs de la mémoire (selon une sélection particulière) et ce en quelques millisecondes. En technologie eFlash 150nm, un tel mode que l'on nommera *CCWP* pour *Concurrent Chip Write Pattern*, ne dure que 10ms.

Test des FAs relatives au décodage des *bit lines* (BLj)

Le problème lié à la détection des FAs dans le décodeur de *bit lines* est équivalent à celui la détection de FAs dans le décodeur de *word lines*. Tout d'abord pour détecter les FA_ASs il faut procéder de la même façon que pour la détection de FA_ASs dans le décodeur de *word lines*, il faut écrire des '0' dans toute la mémoire en une seule fois grâce au mode d'écriture parallèle CW et ensuite venir lire tous les bits de la mémoire.

Pour détecter la deuxième famille de FAs, les FA_AMs, il faut suivre la même stratégie de test que celle employée pour tester les FA_AMs affectant les adresses relatives aux *word lines*. Cependant, pour calculer le nombre de motifs à utiliser pour tester les FA_AMs dans les décodeurs de *bit lines* il faut remplacer le nombre B donnant le nombre d'adresses de *word lines* par W donnant le nombre total d'adresses possibles de *bit lines* ($\log_2 W$).

Concaténation des deux séquences de test

Pour avoir une stratégie globale permettant de tester tous les FAs possibles, nous avons concaténé les deux solutions testant les FAs relatives au décodage d'adresse de *word lines* et de *bit lines*. Cette solution globale est décrite dans les Figures ci-dessous.

CW		CE	CCWP.1		CE	CCWP.2	
0000		1111	0000		1111	0000	
0000		1111	1111		1111	0000	
0000		1111	0000		1111	1111	
0000		1111	1111		1111	1111	
↑ Read			↑ Read			↑ Read	

Figure 4.6: Test des FAs dans les décodeurs de WLi

CE	CCWP.1		CE	CCWP.2	
1111	0101		1111	0011	
1111	0101		1111	0011	
1111	0101		1111	0011	
1111	0101		1111	0011	
	↑ Read			↑ Read	

Figure 4.7: Test des FAs dans les décodeurs de BLj

Dans cette stratégie de test globale il faut noter que l'opération CW permet de détecter toutes les FA_Ass possibles en une seule fois. Les opérations CCWP.i permettent de programmer les motifs de test périodiques, précédemment décrits, grâce au mode de programmation parallèle de la mémoire eFlash. Ce mode permet de programmer des motifs particuliers dans certains blocs de la mémoires et ce en une seule fois. Le temps total *T* de notre stratégie de test peut être calculé comme suit:

- $T = CW + (CE + CCWP) \times (\log_2 B + \log_2 W)$

Il est important de noter que sur des mémoires eFlash en technologie 150nm les modes de programmation CW, CE et CCWP durent 10ms chacun. De plus il faut souligner que le temps de lecture présent à chaque étape de notre stratégie de test n'est pas considéré vu son faible impact sur le temps de test global.

Si l'on considère une mémoire eFlash dont la matrice régulière fait 1024*1024bits, le temps de test des FAs grâce au motif diagonale sera d'environ 4s alors qu'avec la stratégie proposée dans cette thèse nous atteignons un temps de test d'environ 410ms. Notre stratégie réduit dans ce cas précis le temps de test des FAs par 10 sans en dégrader le taux de couverture.

Evaluation du test des FAs

Dans cette partie, l'évaluation du taux de couverture de cette nouvelle stratégie de détection des FAs sera effectuée pour de nouvelles fautes et comparée à une autre solution de test très utilisée en milieux industriels pour tester les mémoires eFlash. Grâce à nos précédents travaux sur la modélisation de faute nous avons pu établir une liste complète de modèles de faute réalistes associées aux mémoires eFlash.

Voici cette liste de fautes à tester si nous voulons assurer une parfaite fonctionnalité de notre mémoire eFlash :

- Fautes de collage à '1' et à '0', CA1 et CA0,
- Fautes de transition, FT,
- Fautes de couplage d'état, FCE, (FCE<0,0>, FCE<1,1>),
- Fautes de décodeur d'adresse FA.

La stratégie la plus utilisée par les industriels pour tester les mémoires eFlash est appelée la séquence '5' motifs. Cette séquence est composée des 5 motifs suivants : effacement de la mémoire à la valeur logique '1' (CE=Chip Erase), écriture de la mémoire à la valeur '0' (CW=Chip Write), écriture d'un motif damier (échiquier) dans la mémoire (CKB=Checkerboard), écriture du motif damier inverse (CKBI=Checkerboard inverse) et finalement écriture d'une diagonale de 0 (Diag0) dans la mémoire.

Pour nous permettre d'évaluer et de comparer cette séquence '5' motifs, nous avons développé un simulateur de fautes. Les taux de couverture issus de ce simulateur pour la séquence '5' motifs sont reportés dans la seconde colonne du Tableau 4.1. Cette colonne montre que la séquence '5' motifs est capable de tester 100 % des CA1s, CA0s, FTs et FAs. En revanche, cette séquence de test ne permet pas de tester toutes les fautes de couplage d'état (FCEs) mais uniquement 55% de celles-ci. L'évaluation du temps de test de cette séquence '5' motifs est donnée par la dernière ligne du Tableau 4.1 pour trois tailles de mémoire eFlash (1, 2 et 4Mbits eFlash). Cette évaluation est faite en négligeant comme dans tout ce qui précède le temps de lecture.

A l'aide de notre simulateur nous avons de la même façon évaluée la stratégie que nous proposons pour tester les FAs. Les résultats de cette évaluation sont reportés dans les colonnes 3 et 4 du Tableau 4.1. Comme pour la séquence '5' motifs, notre solution permet de tester 100% des CA1s, CA0s, FTs et FAs. De plus notre stratégie permet de tester plus de fautes de couplage d'état (FCE) que la séquence '5' motifs et ce pour un temps de test largement inférieur. En effet comme notre stratégie est basée essentiellement sur l'écriture de motifs de test dont le nombre évolue de façon logarithmique avec la taille de la mémoire, cela permet en comparaison avec la séquence '5' motifs de réduire le temps de test par 9.8, 18.6 et 34 pour une mémoire eFlash 1Mbits, 2Mbits et 4Mbits respectivement.

A partir de cette évaluation nous pouvons envisager d'utiliser cette solution pour tester les mémoires eFlash grâce à sa bonne couverture de fautes et à son temps d'application minime en comparaison avec la séquence '5' motifs utilisée dans l'industrie.

		Séquence '5' motifs	**Stratégie de test des FAs**
CA1s et CA0s		100 %	100 %
FTs		100 %	100 %
FAs		100 %	100 %
FCE		55 %	83 %
Temps de Test	1Mbits eFlash	~ 4.1s	~ 420ms
	2Mbits eFlash	~ 8.2s	~ 440ms
	4Mbits eFlash	~ 16.4s	~ 480ms

Table 4.1: Evaluation de la séquence '5' motifs et de la nouvelle séquence proposée

5. Conclusion

Différents types de mémoires peuvent être embarquées dans les systèmes sur puce (SoC) comme les SRAM, DRAM, EEPROM et eFlash (embeded Flash). La constante augmentation de l'utilisation d'appareils électroniques portables tels que les téléphones mobiles et les appareils photo numériques a créée ces dernières années une incroyable progression de l'utilisation des mémoires eFlash.

Du fait de leur technologie, un des principaux problèmes que l'on rencontre dans la conception de mémoires eFlash réside dans l'utilisation d'un fort champ électrique durant les phases de programmation. En effet, ce fort champ électrique peut parfois perturber les cellules et affecter leur fiabilité. Le second problème est la grande densité d'intégration de ces mémoires associée à leur complexe procédé de fabrication qui les rendent de plus en plus sensibles aux défauts intra ou inter cellules. Ces deux problèmes font que le test des mémoires eFlash après leur production devient primordial et requiert des solutions optimales.

La première étape pour développer des solutions de test efficaces est l'analyse des défauts réalistes qui peuvent survenir sur ce type de mémoire. Pour ceci nous avons choisi d'étudier une cellule mémoire de type FloTOx en technologie 150nm dans une matrice réduite (3*3bits). A partir de données provenant d'industriels fabricant ces mémoires eFlash en à base de cellule FloTOx 150nm, nous avons recensé les différents types de défauts pouvant survenir dans une telle matrice eFlash. Dans ces défauts il peut y avoir des courts circuits et des circuits ouverts, des couplages capacitifs entre les chemins de haute tension et des variations d'épaisseur d'oxyde dans la fenêtre tunnel du transistor à grille flottante de la cellule FloTOx.

Pour chaque défaut nous avons analysé leur impact sur le comportement de la cellule FloTOx défaillante et sur ces voisines dans la matrice. A la suite de quoi nous en avons proposé une liste de modèles de faute au niveau fonctionnel relatifs à chaque défaut.

Finalement, basées sur ces modèles de faute, des solutions de test permettant de réduire le temps test total tout en gardant un très bon taux de couverture ont été proposées et évaluées grâce à notre propre simulateur de faute.

La première solution proposée permet de tester les variations d'épaisseur d'oxyde dans la fenêtre tunnel du transistor à grille flottante. Cette solution est basée sur la possibilité dont dispose la eFlash à programmer un grand nombre de cellules mémoires en une seule fois. Cette solution de test fait aussi appel au phénomène de couplage qu'il existe entre les *bit lines* de deux cellules FloTOx voisines dans la matrice. Le motif de test utilisé dans cette solution est une alternance de '0' et '1' sur les différentes colonnes de la matrice mémoire.

La seconde solution de test proposée permet de tester les fautes de décodeur d'adresse, FAs. Une fois encore cette solution est basée essentiellement sur le mode de programmation parallèle dont dispose la eFlash. Elle a été développée à partir des méthodes qu'il existe pour tester les courts circuits résistifs entre les différents bus présents dans un système complexe tel qu'un SoC. Cette méthode ainsi proposée a été évaluée par notre simulateur de faute ce qui a permis de mettre en évidence sa grande capacité à détecter toutes les fautes appartenant à liste de fautes possibles dans les eFlash que nous avions établie au préalable mais aussi de réduire le temps de test d'un facteur 34 sur une eFlash 4Mbits comparée à la séquence '5' motifs couramment utilisée dans l'industrie pour tester les eFlash.

La prochaine étape de notre travail sera essentiellement axée sur l'étude des défauts survenant dans les circuits périphériques à la matrice eFlash mais aussi l'étude et la modélisation des défauts introduisant des fautes de type dynamiques.

Modélisation de Fautes et Test des Mémoires Flash

RESUME : Les mémoires non volatiles de type Flash sont aujourd'hui présentes dans un grand nombre de circuits intégrés conçus pour des applications électroniques portables et occupent une grande partie de leur surface. L'absence de défauts à l'intérieur de ces mémoires constitue donc un des éléments clés du rendement de production pour tous les fabricants de ce type d'applications. Cependant, la grande densité d'intégration et la complexité de leur procédé de fabrication rendent ces mémoires Flash de plus en plus sensibles aux défauts de fabrication. Pour mettre en évidence les défaillances qui altèrent la fonctionnalité de ces mémoires, des solutions de test efficaces et peu coûteuses doivent être mises en place Les solutions et algorithmes actuellement utilisés pour tester les mémoires RAM ne sont pas adaptés à l'environnement Flash à cause de la faible vitesse de programmation de celle-ci. De plus, les modèles de faute que l'on trouve dans la littérature et qui sont relatifs aux mémoires RAM ne sont pas forcément réalistes dans le cas des mémoires Flash. La première partie de cette thèse propose une analyse complète des défauts réalistes que l'on trouve dans ces mémoires et qui sont extraits de données silicium issue d'une technologie Flash 150nm. Cette analyse, basée sur l'injection de défauts dans une matrice réduite de mémoire Flash, a permis de mettre en exergue un grand nombre de comportements fautifs et de leur attribuer des modèles de faute fonctionnels. La suite de ce travail de thèse est consacrée à l'élaboration de nouvelles solutions de test permettant d'améliorer les stratégies existantes. Les solutions proposées sont construites en s'appuyant sur les spécificités de la mémoire Flash, comme par exemple sa faculté à programmer certains de ses blocs en une seule fois avec le même motif et en un temps de programmation réduit. Une évaluation de ces solutions est ensuite effectuée à l'aide d'un simulateur de faute que nous avons spécialement développé à cet effet. Cette évaluation montre l'efficacité des solutions de test proposées en termes de couverture de fautes et de temps de test. La validation sur une mémoire Flash de 4Mbits a montré un gain en temps de test considérable (d'un facteur 34) ainsi qu'une couverture de fautes accrues (notamment pour les fautes de couplage) par rapport à des solutions utilisées dans l'industrie.

MOTS CLEFS : Flash, FloTOx, Défauts, Mécanismes de défaillance, Modélisation de fautes, Test

Fault Modeling and Test of Flash Memories

ABSTRACT: Flash memories more and more occurs in complex integrated circuits designed for portable electronic devices and dominate the area of such circuits. The lack of defects within these memories is therefore one the key elements of the production yield for manufacturers of these types of applications. However, the high integration density and the complexity of the fabrication process make these Flash memories more and more prone to manufacturing defects. To exhibit the failures that affect the functionality of these memories, efficient and low cost test solutions must be proposed. The solutions and algorithms currently used to test RAM memories are not well adapted to test Flash memories due of the low programming time of such memories. Moreover, functional fault models proposed in the RAM testing literature are not always realistic in the case of Flash memories. The first part of this thesis proposes a complete analysis of actual defects extracted from silicon data extracted from a 150nm Flash technology. This analysis, based on a defect injection in a reduced Flash memory array, has allowed to exhibit a lot of faulty behaviors and to propose comprehensive fault models for all defects. The next part of this thesis focuses on the development of new and improved test solutions. The proposed solutions are based on Flash specificities like its concurrent programming mode allowing to program certain memory cell blocks in one time with the same pattern and with a reduced programming time. The evaluation of the proposed solutions is carried out with the help of a home made fault simulator. This evaluation has shown the efficiency of the proposed test solutions in terms of fault coverage and test time. The validation on a 4Mbits Flash memory has shown a considerable reduction in test time (by a factor of 34) as well as an improved fault coverage (especially for coupling faults) with respect to solutions currently used in industry.

KEYWORDS : FLASH memories, FloTOx, Defects, Failure mechanisms, Fault modeling, Test

Université de Montpellier II : Science et Techniques du Languedoc
LIRMM : Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
161 Rue Ada – 34392 Montpellier Cedex 5