Luís Mário Braz dos Santos

PERFORMANCE SENSOR FOR CMOS MEMORY CELLS

SENSOR DE PERFORMANCE PARA CÉLULAS DE MEMÓRIA CMOS

Instituto Superior de Engenharia

2020

Luís Mário Braz dos Santos

PERFORMANCE SENSOR FOR CMOS MEMORY CELLS

SENSOR DE PERFORMANCE PARA CÉLULAS DE MEMÓRIA CMOS

Mestre em Engenharia Elétrica e Eletrónica

Especialização em Tecnologias de Informação e Telecomunicações

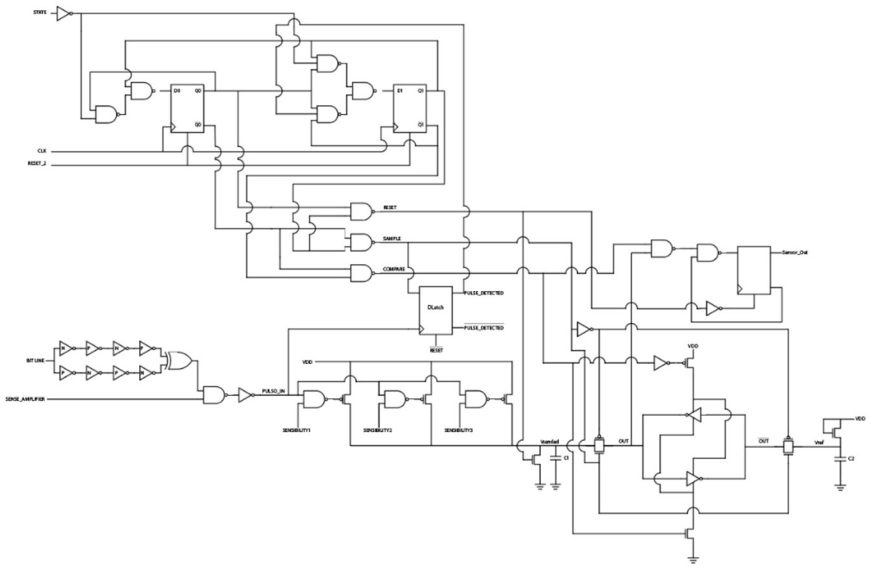
Trabalho efetuado sob orientação de: Professor Doutor Jorge Filipe Leal Costa Semião

Instituto Superior de Engenharia

2020

UAlg
UNIVERSIDADE DO ALGARVE

UNIVERSIDADE DO ALGARVE
INSTITUTO SUPERIOR DE ENGENHARIA



PERFORMANCE SENSOR FOR CMOS MEMORY CELLS

SENSOR DE PERFORMANCE PARA CÉLULAS DE MEMÓRIA CMOS

Luís Mário Braz dos Santos

Dissertação para a obtenção do grau de Mestre em
Engenharia Eléctrica e Eletrónica
Especialização em Tecnologias de Informação e Telecomunicações

Orientador: Professor Doutor Jorge Filipe Leal Costa Semião

Outubro de 2020

Title: Performance Sensor for CMOS Memory Cells

Authorship: Luís Mário Braz dos Santos

I hereby declare to be the author of this original and unique work. Authors and references in use are properly cited in the text and are all listed in the reference section.

_____

Luís Mário Braz dos Santos

ACKNOWLEDGMENTS

First of all, the greatest acknowledgment goes to my thesis advisor Professor Jorge Semião, for without his tireless guidance and motivation this stage of my academic life would have been much more complicated to overcome and surpass. His teachings and knowledge in this area made it possible to advance in a faster and more coherent way in the research and elaboration of the thesis that is now presented.

I also would like to thank my family for the support and encouragement to overcome this stage in my academic education.

Faro, October 9th, 2020

Luís Mário Braz dos Santos

RESUMO

Vivemos hoje em dia tempos em que quase tudo tem um pequeno componente eletrónico e por sua vez esse componente precisa de uma memória para guardar as suas instruções. Dentro dos vários tipos de memórias, as *Complementary Metal Oxide Semiconductor* (CMOS) são as que mais utilização têm nos circuitos integrados e, com o avançar da tecnologia a ficar cada vez com uma escala mais reduzida, faz com que os problemas de performance e fiabilidade sejam uma constante. Efeitos como o BTI (*Bias Thermal Instability*), TDDB (*Time Dependent Dielectric Breakdown*), HCI (*Hot Carrier Injection*), EM (*Electromigration*), ao longo do tempo vão deteriorando os parâmetros físicos dos transístores de efeito de campo (MOSFET), mudando as suas propriedades elétricas.

Associado ao efeito de BTI podemos ter o efeito PBTI (*Positive* BTI), que afeta mais os transístores NMOS, e o efeito NBTI (*Negative* BTI), que afeta mais os transístores PMOS. Se para nanotecnologias até 32 nanómetros o efeito NBTI é dominante, para tecnologias mais baixas os 2 efeitos são igualmente importantes. Porém, existem ainda outras variações no desempenho que podem colocar em causa o bom funcionamento dos circuitos, como as variações de processo (P), tensão (V) e temperatura (T), ou considerando todas estas variações, e de uma forma genérica, PVTA (*Process, Voltage, Temperature and Aging*).

Tendo como base as células de memória de acesso aleatório (RAM, *Random Access Memory*), em particular as memórias estáticas (SRAM, *Static Random Access Memory*) e dinâmicas (DRAM, *Dynamic Random Access Memory*) que possuem tempos de leitura e escrita precisos, estas ficam bastante expostas ao envelhecimento dos seus componentes e, consecutivamente, acontece um decréscimo na sua performance, resultando em transições mais lentas, que por sua vez fará com que existam leituras e escritas mais lentas e poderão ocorrer erros nessas leituras e escritas . Para além destes fenómenos, temos também o facto de a margem de sinal ruido (SNM - *Static Noise Margin*) diminuir, fazendo com que a fiabilidade da memória seja colocada em causa.

O envelhecimento das memórias CMOS traduz-se, portanto, na ocorrência de erros nas memórias ao longo do tempo, o que é indesejável, especialmente em sistemas críticos onde a ocorrência de um erro ou uma falha na memória pode significar por em risco sistemas de elevada importância e fundamentais (por exemplo, em sistemas de segurança, um erro pode desencadear um conjunto de ações não desejadas).

Anteriormente já foram apresentadas algumas soluções para esta monitorização dos erros de uma memória, disponíveis na literatura, como é o caso do sensor de envelhecimento embebido no circuito OCAS (*On-Chip Aging Sensor*), que permite detetar envelhecimento numa SRAM provocado pelo envelhecimento por NBTI. Contudo este sensor demonstra algumas limitações, pois apenas se aplica a um conjunto de células SRAM conectadas a uma *bit line*, não sendo aplicado individualmente a outras células de memória como uma DRAM e não contemplando o efeito PBTI. Outra solução apresentada anteriormente é o Sensor de Envelhecimento para Células de Memória CMOS que demonstra alguma evolução em relação ao sensor OCAS. Contudo, ainda tem limitações, como é o caso de estar bastante dependente do sincronismo com a memória e não permitir qualquer tipo de calibração do sistema ao longo do seu funcionamento.

O trabalho apresentado nesta dissertação resolve muitos dos problemas existentes nos trabalhos anteriores. Isto é, apresenta-se um sensor de performance para memórias capaz de reconhecer quando é que a memória pode estar na eminência de falhar, devido a fatores que afetam o desempenho da memória nas operações de escrita e leitura. Ou seja, sinaliza de forma preditiva as falhas.

Este sensor está dividido em três grandes partes, como a seguir se descreve. O *Transistion Detector* é uma delas, que funciona como um "conversor" das transições na *bit line* da memória para o sensor, criando pulsos de duração proporcional à duração da transição na *bit line*, sendo que uma transição rápida resulta em pulsos curtos e uma transição lenta resulta em pulsos longos. Esta parte do circuito apresenta 2 tipos de configurações para o caso de ser aplicado numa SRAM, sendo que uma das configurações é para as memórias SRAM inicializadas a VDD, e a segunda configuração para memórias SRAM inicializadas a VDD/2. É também apresentada uma terceira configuração para o caso de o detetor ser aplicado numa DRAM. O funcionamento do detetor de transições está baseado num conjunto de inversores desequilibrados (ou seja, com capacidades de condução diferentes entre o transístor N e P no inversor), criando assim inversores do tipo N (com o transístor N mais condutivo que o P) e inversores do tipo P (com o transístor P mais condutivo que o N) que respondem de forma diferente às transições de 1 para 0 e vice-versa. Estas diferenças serão cruciais para a criação do pulso final que entrará no *Pulse Detetor*. Este segundo bloco do sensor é responsável por carregar um condensador com uma tensão proporcional ao tempo que a *bit line* levou a transitar. É nesta parte que se apresenta uma caraterística nova e importante, quando comparado com as soluções já existentes, que é a capacidade do sensor poder ser calibrado. Para isso, é utilizado um conjunto de transístores para carregar o condensador durante o

impulso gerado no detetor de transições, que permitem aumentar ou diminuir a resistência de carga do condensador, ficando este com mais ou menos tensão (a tensão proporcional ao tempo da transição da *bit line*) a ser usada na Comparação seguinte. O terceiro grande bloco deste sensor é resumidamente um bloco comparador, que compara a tensão guardada no condensador com uma tensão de referência disponível no sensor e definida durante o projeto. Este comparador tem a função de identificar qual destas 2 tensões é a mais alta (a do condensador, que é proporcional ao tempo de transição da *bit line*, ou a tensão de referência) e fazer com a mesma seja "disparada" para VDD, sendo que a tensão mais baixa será colocada a VSS. Desta forma é sinalizado se a transição que está a ser avaliada deve ser considerada um erro ou não.

Para controlar todo o processo, o sensor tem na sua base de funcionamento um controlador (uma máquina de estados finita composta por 3 estados). O primeiro estado do controlador é o estado de *Reset,* que faz com que todos os pontos do circuito estejam com as tenções necessárias ao início de funcionamento do mesmo. O segundo estado é o *Sample*, que fica a aguardar uma transição na *bit line* para ser validada pelo sensor e fazer com que o mesmo avance para o terceiro estado, que é o de *Compare*, onde ativa o comparador do sensor e coloca no exterior o resultado dessa comparação. Assim, se for detetado uma transição demasiado lenta na *bit line*, que é um sinal de erro, o mesmo será sinalizado para o exterior activando o sinal de saída. Caso o sensor não detete nenhum erro nas transições, o sinal de saída não é activado.

O sensor tem a capacidade de funcionar em modo *on-line*, ou seja, não é preciso desligar o circuito de memória do seu funcionamento normal para poder ser testado. Para além disso, pode ainda ser utilizado internamente na memória, como sensor local (monitorizando as células reais de memória), ou externamente, como sensor global, caso seja colocado a monitorizar uma célula de memória fictícia.

PALAVRAS-CHAVE: Sensor de Performance, NBTI, PBTI, Memórias CMOS, SRAM, DRAM.

ABSTRACT

Within the several types of memories, the Complementary Metal Oxide Semiconductor (CMOS) are the most used in the integrated circuits and, as technology advances and becomes increasingly smaller in scale, it makes performance and reliability a constant problem. Effects such as BTI (Bias Thermal Instability), the positive (PBTI - Positive BTI) and the negative (NBTI - Negative BTI), TDDB (Time Dependent Dielectric Breakdown), HCI (Hot Carrier Injection), EM (Electromigration), etc., are aging effects that contribute to a cumulatively degradation of the transistors. Moreover, other parametric variations may also jeopardize the proper functioning of circuits and contribute to reduce circuits' performance, such as process variations (P), power-supply voltage variations (V) and temperature variations (T), or considering all these variations, and in a generic way, PVTA (Process, Voltage, Temperature and Aging).

The Sensor proposed in this paper aims to signalize these problems so that the user knows when the memory operation may be compromised. The sensor is made up of three important parts, the Transition Detector, the Pulse Detector and the Comparator, creating a sensor that converts bit line transition created in a memory operation (read or write) into a pulse and a voltage, that can be compared with a reference voltage available in the sensor. If the reference voltage is higher than the voltage proportional to the bit line transition time, the sensor output is not activated; but if the bit line transition time is high enough to generate a voltage higher than the reference voltage in the sensor, the sensor output signalizes a predictive error, denoting that the memory performance is in a critical state that may lead to an error if corrective measures are not taken.

One important feature in this sensor topology is that it can be calibrated during operation, by controlling sensor's sensibility to the bit line transition. Another important feature is that it can be applied locally, to monitor the online operation of the memory, or globally, by monitoring a dummy memory in pre-defined conditions. Moreover, it can be applied to SRAM or DRAM, being the first online sensor available for DRAM memories.

KEYWORDS: Performance Sensor, NBTI, PBTI, CMOS Memories, SRAM, DRAM.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BTI	Bias Temperature Instability
CLK	Clock
CMOS	Complementary Metal-Oxide Semiconductor
DRAM	Dynamic Random-Access Memory
HCI	Hot Carrier Injection
HW	Hardware
IC	Integrated Circuit
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
NBTI	Negative Bias Temperature Instability
NMOS	N-type Metal-Oxide Semiconductor
NMOSFET	N-type Metal-Oxide Semiconductor Field-Effect Transistor
PBTI	Positive Bias Temperature Instability
PMOS	P-type Metal-Oxide Semiconductor
PMOSFET	P-type Metal-Oxide Semiconductor Field-Effect Transistor
PVT	Process, power-supply Voltage and Temperature
PVTA	Process, power-supply Voltage ,Temperature and Aging
Si	Silicon (chemical symbol)
SW	Software
SRAM	Static Random-Access Memory
T	Temperature
V	Power-supply Voltage
Vth	Threshold Voltage

1. INTRODUCTION

The name Internet of Things (IoT) has been around for several years. This name is increasingly present in our daily lives with the evolution of wireless technologies [1]. In general, IoT refers to the networking of everyday objects, often equipped with artificial intelligence [2]. The idea behind this concept is the possibility that several objects can interact with each other [1].

The IoT is fueling the fourth industrial revolution, bringing significant benefits and being able to connect people, processes and data [4][5]. The possibility of interconnecting a huge number of intelligent hardware/software (hw/sw) systems, with a large increase in local artificial intelligence, is opening new avenues of research and innovative IoT applications with very diverse uses, from smart cities [6] up to health systems [7], automotive applications [8], aerospace, and so on.

1.1. HARDWARE CHALLENGES IN IOT

One of the major concerns that exists today is the ability to manage the energy of IoT devices [11][12]. A wide variety of IoT sensors, often powered with batteries, require high energy efficiency. This requires the use of ultra-low power microcontrollers and low power memories. Because of this, a key variable is the minimum voltage value of the power supply (V_{DD}) which can guarantee safe data retention and data access (read/write operations). Using a flexible Power Management Unit (PMU), it can be rewarding to use Dynamic Voltage and Frequency Scaling (DVFS) techniques to power-up memory arrays with the minimum V_{DD} value during memory access and data retention.

There is also the problem that large networks of IoT devices are extremely expensive, and are expected to remain in operation for a long period of time. And the problem gets worse if we consider that semiconductor aging can cause unacceptable degradation in the device, causing errors during product's lifetime. These issues pave the way for considering IoT manufacturing with features for embedded monitoring components, similarly to the way chip designers implement Design for Testability (DfT) techniques. This non-mission-related functionality should allow monitoring locally the operation of the device, during product's

lifetime, and may trigger warnings or corrective decisions, in order to ensure safe operation (considering a level of security appropriate to the IoT application). Moreover, it can also be used to run DVFS on microcontrollers and memory banks, allowing for considerable energy savings.

Systems-on-a-Chip (SoCs) and other integrated circuits are composed by nanoscale devices that are crammed in a very limited silicon area, presenting reliability issues and new challenges. CMOS circuits' performance is sensitive to parametric variations, such as Process, power-supply Voltage and Temperature (PVT) [13], as well as aging effects (PVT and Aging – PVTA). CMOS circuits' aging degradation is mainly caused by the following effects: Bias Temperature Instability (BTI), Hot-Carrier Injection (HCI), Electromigration (EM) and Time Dependent Dielectric Breakdown (TDDB) [14]. The most relevant aging effect is the BTI, namely the Negative Bias Temperature Instability (NBTI), which affects PMOS MOSFET transistors, resulting in a gradual increase of their absolute threshold voltages over time (|VthP|). As high-k dielectrics started to be employed from the sub-32nm technologies [15], BTI also significantly affects NMOS transistors – Positive Bias Temperature Instability (PBTI), resulting in a rise of their threshold voltages, VthN. These effects degrade digital circuits' performance over time, increasing the variability in CMOS circuits. Performance degradation decrease the switching speed, eroding time margins and leading to potential delay faults, and eventually chip failures.

As today's SoC face an increasing need to store more and more information produced, SRAMs (Static Random Access Memories) occupy most of the SoC silicon area, being currently about 90% of the SoC density [16]. Therefore, the robustness of SRAM is considered crucial to guarantee the reliability of these SoCs during the product's useful life [16]. In addition, the trend is that this predominance of the *Si* memory area over the *Si* logical area will continue to grow in the following years. Consequently, semiconductor memory has become the main responsible for the general SoC area and also for the active and leakage power in embedded systems and, therefore, in the hardware part of IoT devices.

One of the main problems in the design of an SRAM cell is stability. The stability of cells is basically their ability to maintain correct operation in the presence of noise signals, thus guaranteeing the correct read, write and hold operations. Therefore, it determines the sensitivity of the memory to parametric variations, induced in the manufacturing process and/or during operational conditions. Static noise margins (SNMs) are widely used as a stability criterion [17]. However, some authors argue that dynamic noise margins are also

important [18]. However, due to variations in PVTA (and knowing that aging is a cumulative process), degradation in memory performance and stability can occur.

1.2. MOTIVATION

In the past, significant research has been carried out and a set of performance sensors for digital logic were proposed, either in the design style of a cell library in custom SoCs or in an FPGA programmable fabric (see, for example, [19] [20] [22]). However, research on performance sensors for semiconductor memories has been much more limited, so far. We recognize that there is a lot of previous research dealing with aging sensors for SRAM cells, and especially focused on the BTI effect. These are attempts to increase reliability in the operation of SRAM. However, they do not simultaneously consider variations in PVT and aging.

Therefore, the previous works deal mainly with the detection of aging in SRAMs, but it lacks a simple generic sensor to deal with the performance and, simultaneously, PVTA variations in the memories. In addition, the previous work does not deal with the development of SRAM sensors for operation with extremely low energy consumption, a mandatory requirement for many IoT applications. The common problem is that all possible circuit variations (i.e. PVTA variations) cumulatively affect the performance and behavior of the circuit and, due to their cumulative effect, may be responsible for the occurrence of errors, compromising safe operation of the IoT. Therefore, it is important to develop a sensor that can be aware of all these time variations, that is, a performance sensor.

In fact, research on performance sensors for digital synchronous logic is much more advanced when compared to its memory counterparts. As an example, the Scout Flip-Flop sensor [23] [19] acts as a performance sensor for tolerance and predictive detection of delay faults in synchronous digital circuits (ASIC). However, SRAM sensors that can identify abnormal time response, regardless of their origin, are still limited. In fact, as far as the authors know, there are only two previous works on performance sensors [21] [3], which are an initial attempt by the authors to develop a performance sensor for SRAM. Unfortunately, the sensor architecture, proposed in [21], is complex and leads to a significant area overhead, and the performance of the sensor is limited in the presence of reduced VDD voltages. The sensor proposed in [3], although it solves part of the referred problems, it still has implementation

problems that prohibit its use in a real memory circuit. In addition, for DRAM circuits, as far as the authors know, there are no solutions available for performance sensors.

1.3. OBJECTIVES

The main purpose of this work is to develop a new online SRAM and/or DRAM sensor for IoT applications. We call it SCOUT (performance Sensor for toleranCe and predictive detectiOn of delay-faUlTs) for memories, or simply Scout memory sensor. The performance sensor, compatible with PVTA variations, should allow detecting delay degradation when accessing CMOS memory cells, namely in read/write operations. The sensor must be connected to the memory bit line, to monitor transitions that occurred in these signals during these read/write operations. It should monitor any online performance variation with a very low performance overhead and reasonable area overhead.

Aging and/or performance monitoring should be achieved by detecting slow transitions due to a reduction in performance caused by variations in PVTA (or any other effect that cause a degradation in the operation delays) in memory cells or in the memory circuit (as in the sense amplifier, also connected to the bit lines). In addition, the degradation of the transition time in the bit line can be carried out on purpose, to restrict energy consumption. Therefore, the possibility of using the sensor to adjust a Dynamic Voltage and Frequency Scaling methodology should be analyzed, signaling to the PMU the lowest VDD value that can be used to correctly execute memory access, within a user-defined safety margin. Also, the sensor architecture must be designed to ensure the correct operation of the sensor under these lower VDD values.

1.4. CONTEXT OF THE RESEARCH WORK

The research and development (R&D) work for this master's thesis was carried out at the Instituto Superior de Engenharia (ISE), University of Algarve (UAlg), in close collaboration with the Embedded Electronic Systems team at INESC-ID in Lisbon.

This work is part of the degree program in Master in Electrical and Electronic Engineering with specialization in Information and Communication Technologies.

1.5. THESIS OUTLINE

This thesis is organized in the following way.

Chapter 2 discusses the signal degradation problem, and its consequences in the performance of a memory, focusing on the NBTI, PBTI effects and PVTA variations. This chapter will also approach the architecture and functioning of a typical SRAM and DRAM CMOS memories, explaining its organization in terms of components and structure, as well as the particularities of a DRAM and a SRAM memory. It will also address the SNM (static noise margin) measure, and how this value may reflect the performance of a memory. Moreover, it also analyzes the state-of-the-art on performance sensors for memories, referring to the OCAS sensor (an aging sensor) and the Performance Sensor for SRAM.

Chapter 3 will present the Scout memory sensor, which details the new contributions presented in this dissertation thesis regarding performance sensors for memories. The new developed sensor is presented, displaying all its architecture, its components, and all the circuitry needed. Also in this chapter, the layouts are presented for the main blocks of the sensor.

In chapter 5, simulations results are presented for the entire sensor operation.

Finally, chapter 6 will summarize and conclude all the work carried out in the previously mentioned chapters, identifying the future work.

2. PRELIMINARY STUDIES

2.1. PERFORMANCE DEGRADATION EFFECTS IN CMOS

To be able to predict an error in the operation of a memory is a task that depends on several factors, which must be studied and understood, so that the implementation of a device that has the capacity to detect them can be adjusted.

The circuits we use today have a life span that can vary significantly, depending on their use. Circuit's aging is not an occasional and random deterioration that can damage components, nor is it caused by the incorrect use of the circuit, putting it under a stress that can irreversibly damage a component [26]. The degradation caused aging effects is a generalized change in the functioning of all components, in particular the integrated circuit transistors. This degradation worsens over time, reaching values that are no longer accepted and that jeopardize the correct functioning of the system, especially if other parametric variations pile up, like PVT variations (or in a general way, PVTA variations). With the constant reduction in the size of the components, as stated by Moore's Law, new challenges arise in this area, as the components, being smaller and smaller, become more sensitive to external interference. In addition to this phenomenon, there is also the fact that more and more of these components work with very low voltages. It means, then, that with evolution, circuits become more and more sensitive to phenomena that can jeopardize their correct functioning, and this phenomenon is enhanced in nano technologies [28].

In older CMOS technologies, the phenomena of circuit aging degradations are also present, but as explained earlier, these technologies are less susceptible to aging phenomena, because with larger dimensions, the voltages are also bigger and, thus, increasing their robustness and layout area. It can be said that circuits in older technologies do not age, comparatively with the smaller ones.

The main effects that cause a critical performance degradation in CMOS integrated circuits are: Process, power supply Voltage, Temperature and Aging effects (PVTA), with the negative and the positive BTI effects (NBTI and PBTI) as the main aging degradation effects.

2.1.1. AGING AND BTI EFFECT

As mentioned in Chapter 1, CMOS circuit's aging degradation is mainly caused by the following effects: Bias Temperature Instability (BTI), Hot-Carrier Injection (HCI), Electromigration (EM) and Time Dependent Dielectric Breakdown (TDDB) [14]. Considering these aging effects, the most relevant one is the BTI effect, namely the Bias Temperature Instability effect. This effect is caused by stressing both PMOS and NMOS transistors, as a result from, respectively, the Negative Bias Temperature Instability (NBTI), which affects P-type MOSFET transistors, and the Positive Bias Temperature Instability (PBTI), which affects N-type MOSFET transistors. This sub-section details some aspects of these two BTI effects.

NBTI Effect

The NBTI Phenomenon is a temperature-accelerated degradation in PMOS transistors when they are negatively polarized (with Vgs<0V) and at high temperatures. This phenomenon was identified in the mid-1960s and became particularly important for technologies below 130nm [29].

This phenomenon occurs since these operating conditions cause the appearance of new electron donors in the covalence band of the transistor channel, where two influential directions are reflected: 1) a reduction in the thickness of the oxide resulting from the increase in its electric field, where there is no corresponding reduction in the supply voltage, and 2) use of the gate dielectric to reduce the effect of leakage currents [29]. Measuring the degradation of a PMOS transistor due to the NBTI effect relies on several factors (the temperature profiles, the amount of time elapsed in use, the voltage that the transistor has already sustained and its workload, which makes it possible to determine the amount of time that the PMOS transistor is on), being the threshold voltage the most important parameter, in order to monitor the effect [27].

The degradation from the NBTI phenomenon results mainly in the wear of *Si-H* connections at the *Si* dielectric interface, resulting in the diffusion of hydrogen types in dielectric and *poly-Si* (see Figure 2.1). Therefore, several different versions of the reaction-diffusion model (R-D) were used to interpret the NBTI degradation [29].

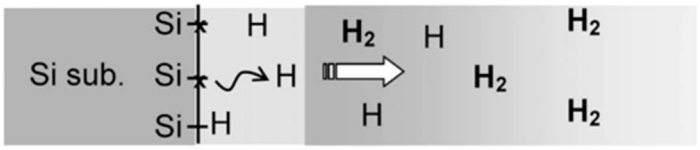


Figure 2.1 - Schematic Representation of the dissociation of the Si-H connection and resulting diffusion of types of hydrogen (H and H2) through the dielectric and poli-Si during NBTI stress [29]

During the application at the PMOS transistor gate of a continuous negative polarization, it decreases its performance over time. When removing this polarization, the transistor will recover more easily, cancelling some of the generated interface traps, thus partially recovering its threshold voltage. In Figure 2.2 one can see the generation of traps for continuous stress and for periodic stress and relaxation. It is also possible to see the recovery of the number of traps generated in the relaxation period [30].

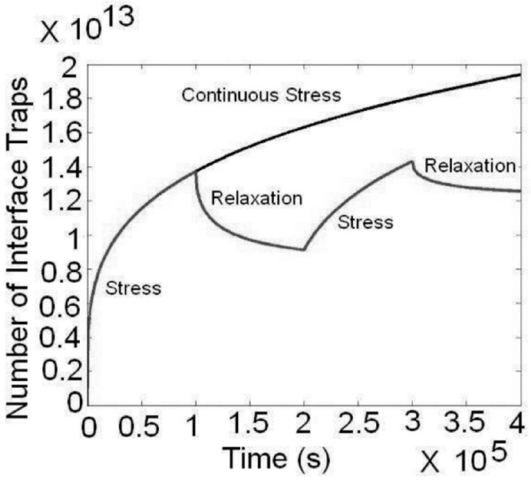


Figure 2.2 - Generation of trap in periodic stress and relaxation against continuous stress [30].

PBTI Effect

After the PBTI phenomenon in the MOSFET was studied, it was found that its influence on transistors can be just as significant as the negative polarization temperature instability (NBTI). While the NBTI includes both Interface state generation and positive charge formation in the gate oxide, the PBTI exists only in the form of donor-like interface state generation [31].

The PBTI phenomenon is most visible in NMOS transistors and in technologies of less than 32nm, causing a degradation of the transistor's threshold voltage and also its instability over time. These effects are of greater visibility the smaller the technology used in the transistor construction is, and the transistors are subject to a positive polarization stress on their gate [27].

The PBTI phenomenon is caused by the trapping of electrons in the high-k layer, and this happens due to holes resulting from the lack of oxygen in the separation layer of the materials [32]. Two causes are pointed out for this, the filling of holes by electrons that already existed and the creation of new holes, each one of these causes happens depending on the several voltage levels present [27][32].

2.1.2. OPERATION-INDUCED VARIATIONS AND CUMULATIVE EFFECTS

Process (P) variations and physical defects depend on the semiconductor technology, and on the manufacturing process. The use of newer technologies and fabrication processes that are not so mature as older technologies are, contributes to the existence of deviations in several parameters when a chip is manufactured. Parameters like channel length, channel thickness, number of dopant atoms, silicon oxide thickness, etc., have small deviations along the chip and differently in every transistor and circuit. These variations, random or not, contribute to produce unique circuits with unique timing profiles. This means that each circuit has a unique timing response for every function or task. These variations are static, has they are introduced during the manufacturing process, but they change the delay map of a circuit from what was supposed to be manufactured, and change statically circuit delays, creating deviations from what were the initially expected circuit delays.

During circuit lifetime, circuit operation and environmental conditions also influence circuit timing delays. Aging, as explained in the previous section, changes the timing map of a circuit, creating unique changes in circuit delays. For instance, paths that initially are not critical may become critical during circuit lifetime, according to the operation imposed to the circuit (and its aging condition). These changes produced in the circuit are small physical changes that continuously alter circuit delays during its lifetime. Moreover, these changes are cumulative, as aging process continues to occur, which imposes a unique delay map evolution in the circuit along its lifetime and according to its own environmental conditions and circuit operation.

Different from aging effects and process variations, there are parameters that influence temporarily and sparsely in time circuit timings. Two important parameters are power-supply voltage (V) and temperature (T) variations. VT variations induce transient timing variations, not only in the "functional part" of the system (combinational and sequential logic, memory, test hardware, etc.), but also in the supporting physical infrastructure, like the power grid and the clock distribution network. Normal circuit operation and operation-dependent power consumption and signal switching creates, not only thermal maps along the circuit, with hot and cold spots, but also power-supply voltage fluctuations along the power-supply distribution grid, both variable in time, and space, creating different VDD and T values. Static power consumption associated with leakage is also strongly dependent on temperature variations. These effects, affect circuit timings, delays and, consequently, circuit performance.

Moreover, regardless the origin of the cause and the effect, all effects that can affect delays in a circuit can pile-up and their cumulative effect will change enormously the timing response of the circuit, and consequently change circuit performance, eventually introducing delay faults.

Therefore, the most important parameters that can influence circuit performance are Process, power supply Voltage, Temperature and Aging (PVTA) variations. Decreasing the size of the chips has numerous advantages, but it also has its disadvantages. Due to their small size, they are more exposed, both to outside interference, as well to changes caused by the operation itself, as it was mentioned.

2.2. CMOS MEMORIES

2.2.1. ARCHITECTURE AND OPERATION OF CMOS MEMORIES

Any system in today's market must have a memory in order to work, even if only the memory is used during its operation. These memories may vary according to type and access speed. If we focus on computers, it is possible to distinguish the memories into two types: main storage and secondary storage. The main storage is thus called since it is a memory which the processor may directly address and for which the computer cannot function without. Secondary storage is the mass storage memory, for permanent data storage (hard drives, optical drives such as CDs, DVDs and Blu-Rays, floppy disk drives, etc.).

Coming back to the main storage, it usually provides a bridge to the external storage, but its main purpose is to contain the necessary information for the processor in a specific moment. It is usually faster and it is where all the program instructions are executed, since it has a much higher speed of access to the information, and is a random access memory, where the time needed to read or write the information does not depend on its physical location [33]. This category includes RAM (Random-Access Memory), which is a volatile semiconductor memory (i.e., it loses the information when the supply is turned off), with random access (i.e., the individual memory words are directly accessed, using a hardware implemented addressing logic), and it may be static (SRAM) or dynamic (DRAM). Also in this category, there are the ROM memories (Read Only Memory, non-volatile, read-only memory) and records. These types of memories are also characterized by their high performance and have been widely used in recent years [27]. Dynamic memories (DRAM) differ from the static ones (SRAM) because they need to be constantly refreshed, so that their memory contents are not lost and are, therefore, slower on average than static ones (which do not need refreshment).

In this paper, the focus will be on the random-access SRAM and DRAM memories.

Memory organization

On a memory chip, the data addressed is allocated in groups of 4 to 16 bits. If we take as an example a 64-Mbit memory, and assuming that all bits are addressable individually organized in words of 64M x 1 bit, this memory needs an address with 26bit, since 2^{26} = 67.108.864 = 64M. However, if the memory contains 16M x 4-bit words, only 24 bits are needed because 2^{24} = 16,777,216 = 16M [33]. Let's consider the first case, where all the bits are addressable, and that we build an array of cells of 2M rows and 2N columns, for a total storage capacity of 2^{M+N}. If a square array is considered with 1 Mbit of possible addresses, it will be an array with 1024 rows and 1024 columns. Each space in the array will be connected to a 2^{M} line named word line and a 2^{N} line named bit line, as it can be seen in Figure 2.3[33].

Each memory space may be triggered, both for reading and writing, by addressing its word line and bit line. This access is initiated by activating a line decoder present in Figure 2.3 (combining logic circuit that decodes a digital word), which allows selecting the specific place in the memory to be accessed. In the specific case of a DRAM, this memory address has stored very small voltage values, an average of 0.1V to 0.2V, since given the proximity and the large amount of memory spaces, the voltage values have to be very small. However, this

signal will be connected to the Sense Amplifier, which will amplify the voltage values to values 0 or VDD. This value will then be connected to the column decoder which will then select the column to be specifically accessed [33].

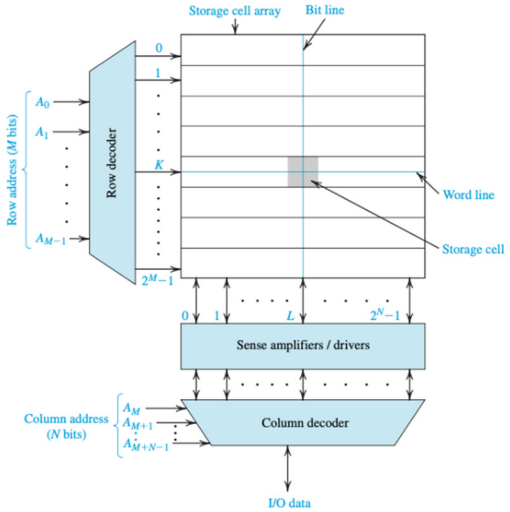


Figure 2.3 - Memory organized in an array with 2^M lines x 2^N columns [33].

DRAM

DRAM cells have been tested and studied over the years to achieve the best possible implementation. On Figure 2.4 it is possible to see the standard DRAM memory cell. This cell is made up with one single NMOS transistor, called an access transistor, and a C_s capacitor that will store the voltage value which one wants to keep, and which needs the periodic refreshment characteristic of DRAM memories. The cell illustrated on Figure 2.4 is known as one-transistor cell, in which the transistor gate is connected to the Word line and the drain to the bit line, so that when active it is able to charge the C_s capacitor, which is loaded to VDD if the purpose is to save a 1, or to the VSS if it is to save a 0. With DRAM, only one bit line is used instead of the SRAM, which always use two, one being the complement to the other [33].

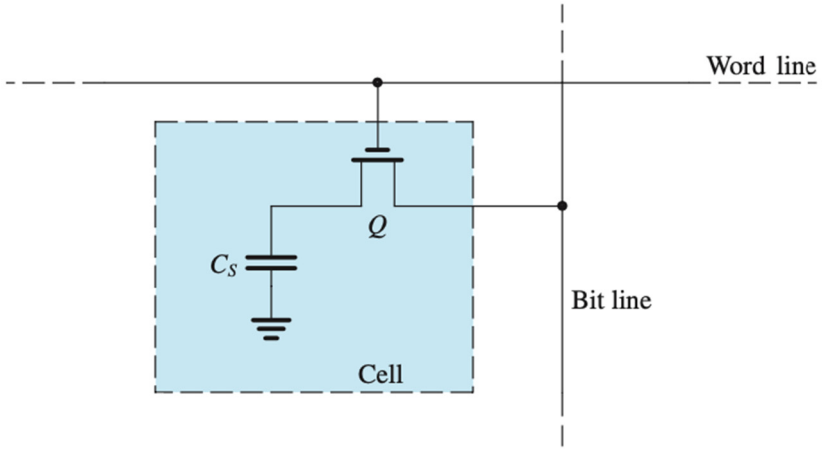


Figure 2.4 - One-transistor cell of a DRAM memory [33].

For a better understanding of the reading and writing process in a DRAM memory, let us begin by assuming that we wish to write the logical value 1 in the memory (VDD). This means that the bit line and Word line are VDD, putting the Q transistor conducting and in turn loading the C_s capacitor. The Q transistor will be conducting until the C_s capacitor reaches the VDD-V_{th} voltage, for we must assume the voltage drop that the transistor causes. This small voltage drop provoked by the transistor's V_{th}, creates a problem in the memory, since the value stored will not be VDD as mentioned previously. The solution found to overcome this situation, was to increase the supply voltage to a value similar to VDD + V_{th}, so that the capacitor can actually store the VDD voltage.

The biggest difference, as mentioned before, between an SRAM memory and a DRAM memory is due to the fact that this C_S capacitor discharges over time, making it necessary to refresh the memory periodically, so that the stored value is kept. During this update, the value saved is read and rewritten, but with the maximum voltage concerning the bit stored in the C_S capacitor (VDD or VSS). This update will happen every 5ms to 10ms [33].

Taking a closer look at the DRAM memory and analyzing its components (Figure 2.5), the line decoder, as in the SRAM, will select the specific line from the memory one wishes to read, increasing its voltage to activate the reading, and making all the transistors to become conductive and letting the values stored in the capacitors pass to the bit line. It should also be noted that every time the C_S capacitor is connected to the bit line, it is as if we were also putting in parallel a C_B capacitor (the capacitance of the bit line) that by default is much larger than C_S. If instead of writing one wishes to read (Figure 2.6), the bit line will be previously loaded with the value of VDD/2, which will then be added to the value that already exists in the Cs

capacitor, making it save a 1 or a 0 in memory, being these voltage variations in the order of +/- 90mV and knowing that the charges in C_S and C_B will be shared causing a voltage slightly above or below the VDD/2 value if a 1 or a 0 is read, respectively [33]. All the changes taking place in the bit line when reading a memory, are then amplified in the sense amplifier, ensuring that the value read from the memory, in its output, has the correct 1 and 0 voltage values (VDD and VSS).

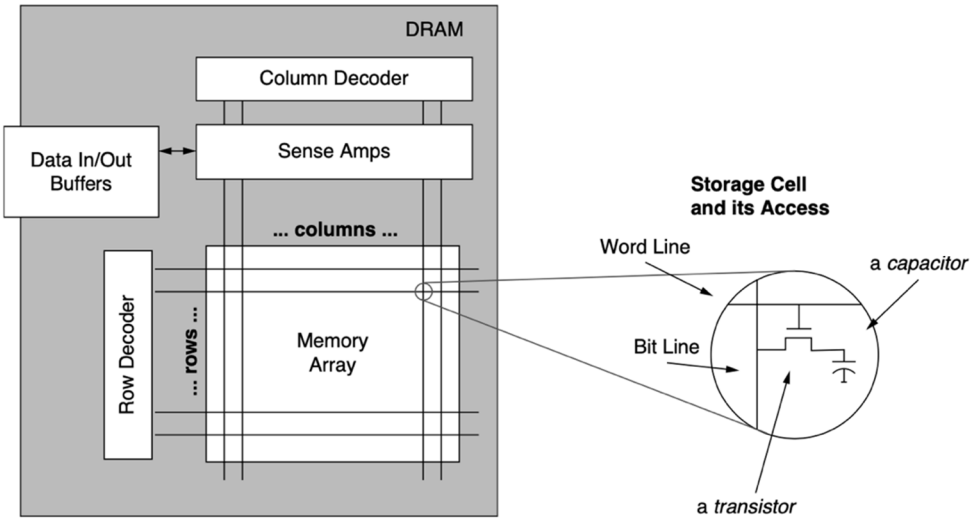


Figure 2.5 - Schematic of a DRAM [39].

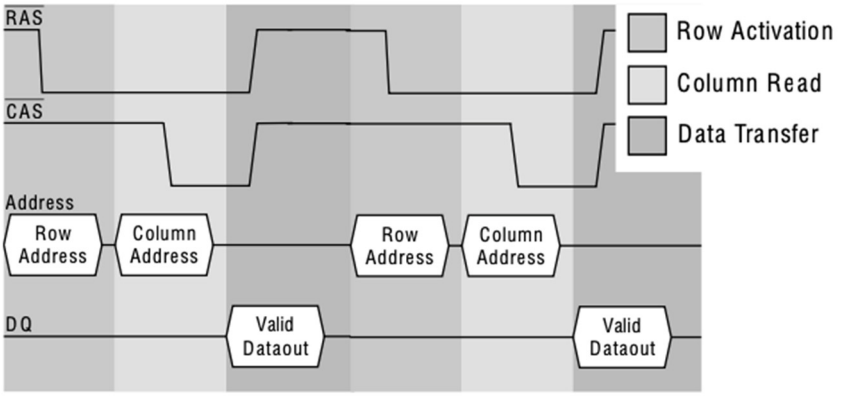


Figure 2.6 - Reading of a DRAM memory[39].

In the case of writing in memory, it is in everything similar to reading, however it happens in the opposite direction, being that if the objective is to write the value 1 in memory, the bit line is loaded (in this case the CB) to VDD and which through the column decoder,

unlocks the transistor where one wishes to write in the memory and the Cs capacitor of that memory location is loaded with VDD [33].

Although every time there is a memory reading or writing, the entire column that will not be changed is updated by refreshing it, it is also necessary to refresh the entire memory, usually every 5ms to 10ms, depending on the type of memory that is being used. This refreshment is carried out in burst mode, which means that it is executed line by line. During the refreshment, the memory will be offline for reading or writing, i.e., the memory units are only available 98% of the time.

SRAM

Let us focus now on static memories using CMOS technology, the SRAMs. The model of a memory like this may be observed in Figure 2.7, which is composed of a flip-flop that has two cross coupling inverters and two access transistors, Q5 and Q6. These are active when the Word Line (W) is connected to the VDD, connecting the cell to the bit line B and the bit line $\bar{B}$. In this implementation it is possible to verify in Figure 2.7 that both bit lines are used to increase the credibility of the results in the end. This circuit is typically known as a six-transistor cell or 6T.

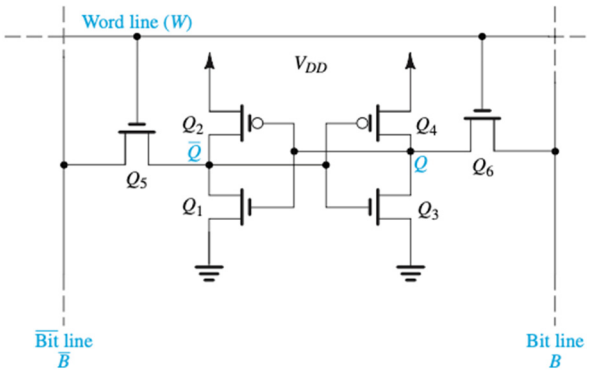


Figure 2.7 - SRAM Memory with CMOS technology [33].

Sense Amplifiers

When implementing a memory, there are several important blocks for its operation and correct implementation, in addition to the way a SRAM or DRAM works. In this case we will

observe the Sense Amplifier, which has a critical role in the functioning of memories. These are essential regarding the DRAM and provide a better performance considering the SRAM. There are several ways of implementing this part of the circuit, we shall consider first the differential sense amplifier, which allows the direct implementation in the SRAM using bit line B and $\bar{B}$, and in DRAMs it can be used using a "dummy-cell" as an aid. The sense amplifier's purpose will be to amplify the small differences between the bit line B and $\bar{B}$, variations that can be in the order of 20mv to 500mv, depending on the type of memory and the design of the memory cell in use. At the sense amplifier output there will always be a signal that will be either 0V or VDD, without fluctuations. In the sense amplifier which we will observe, there is the peculiarity of having the same terminals for input and output.

Considering Figure 2.8, it is possible to see that the Sense Amplifier is made up of two interconnected CMOS inverters, one consisting of the two transistors Q1 and Q2 and the other of the transistors Q3 and Q4. The other two transistors observed, Q5 and Q6, act as switches that connect the Sense Amplifier to VDD and VSS, having its gate connected to ϕ_s and when this is 0v the Sense Amplifier is turned off. In turn, if ϕ_s is VDD, it turns on the Sense Amplifier and it starts working. This operating method allows the Sense Amplifier to only be active when needed and, therefore, to economize some energy, since there is a circuit like this in every column of the memory and this represents huge energy savings [33].

As it was previously mentioned, the Sense amplifier presented has its bidirectional bit line connection terminals, being used the same terminal for input and output signals. This circuit stage is signaled in Figure 2.8 in sections x and y, connected to the bit line B and $\bar{B}$. This point can be simultaneously input and output of the amplifier since it will reinforce the signal that it is reading from the bit lines. This means that when a value in bit line B that is greater than the value in bit line $\bar{B}$ is entered in Sense amplifier, the Sense amplifier will trigger the value of bit line B to VDD and the value of bit line $\bar{B}$ to VSS, forcing these values to be updated in the respective bit lines so that they can be read or written to memory again. Regarding a DRAM, this also happens during the refreshment [33].

In Figure 2.8 it is possible to observe a cell, which may represent a DRAM or SRAM circuit. It is also possible to see two more circuits which are complements of the Sense amplifier, the precharge and equalization circuits. Here, when the ϕ_p changes to VDD before a reading operation, the Q8 and Q9 transistors will start to conduct, preloading in the bit line B and bit line $\bar{B}$ of VDD/2. The Q7 transistor's purpose is to accelerate the preload process of

VDD/2 in the two bit lines, being essential that the two voltages are exactly the same in order not to induce the Sense Amplifier in error at the beginning of its operation [33].

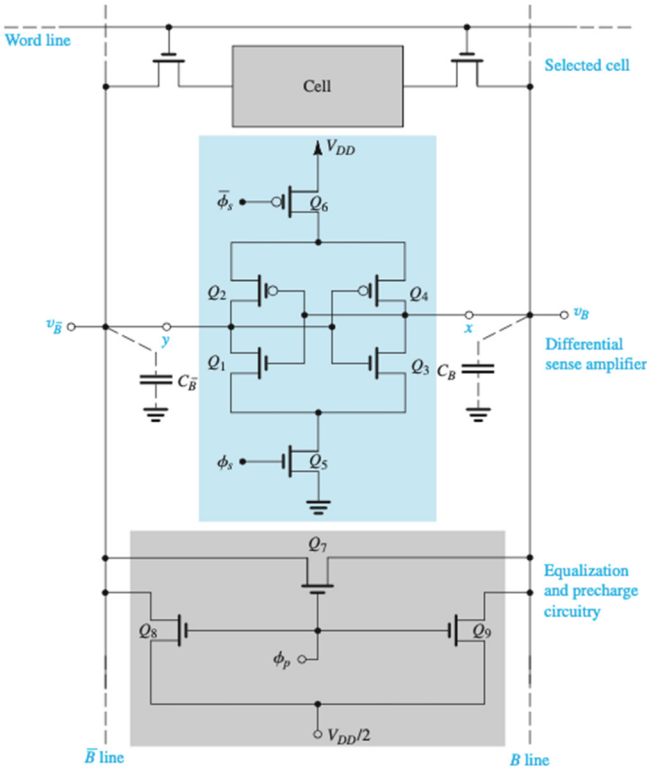


Figure 2.8 - Differential Sense Amplifier connected to the bit line [33].

Observing the rhythm of actions that happen in a reading instruction, we have:

1. By placing the ϕ_P to VDD, the precharge circuit is activated which will cause the bit line B and bit line $\bar{B}$ to be loaded to VDD/2 and the equalization circuit to ensure that the values loaded on both sides are exactly the same. Afterwards, the ϕ_P will change to VSS, leaving the two bit lines with the VDD/2 value, ready to receive the values stored in memory.
2. Word line will change to VDD, causing the values stored in the memory cell to change to the bit lines, thus changing the value which was there. If a 1 is read in the memory this means that the voltage in the bit line B is higher that the voltage in bit line $\bar{B}$. If a 0 is read, the voltage values are the opposite
3. As soon as a difference is detected on both bit lines, the Sense amplifier does its job, ensuring that one of the bit lines is at 0v and the other at VDD. This operation is clearly visible in Figure 2.9.

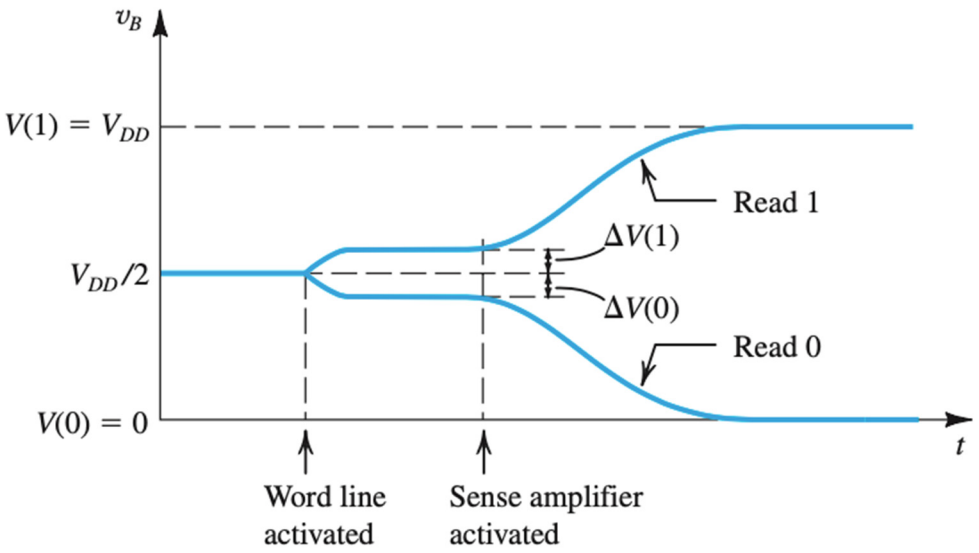


Figure 2.9 - Voltage in bit lines [33].

Differential DRAM operation

The main goal is that the values stored in the memory are real and have no mistakes nor interpretation issues, which could induce the results into wrong information. The Sense Amplifier explained previously aims to create two signals, one inverse of the other, in order to ensure that the information is transmitted in a reinforced way, thus firmly rejecting possible external interferences. The solution that is illustrated in Figure 2.10 has existed for many years but is still efficient. Basically, each bit line is divided in two similar parts. Each part is connected to the memory cells in that side of the division which are, in turn, connected to an extra cell named dummy cell, with a capacitor equal to the capacitor of the cells, Cd=Cs. When a bit line from the left side is selected to carry out a memory reading, the dummy cell from the right side will also be selected at the same time through the ϕ_D control and vice-versa. Therefore, if the left part is being used, the right part is working as a complement and vice versa [33].

The circuit starts by loading both sides of the memory with VDD/2 through the precharge and the equalization and the C_D dummy cell's capacitors. After selecting the bit line from one of the sides, a dummy cell is selected from the opposite side, causing the pre-loaded voltages to be changed in the active part of the circuit which, in turn, while passing in the Sense Amplifier, will cause its complement to be placed in the second half of the memory. What is expected at the end is to have one side of the circuit at VDD and the other at 0v.

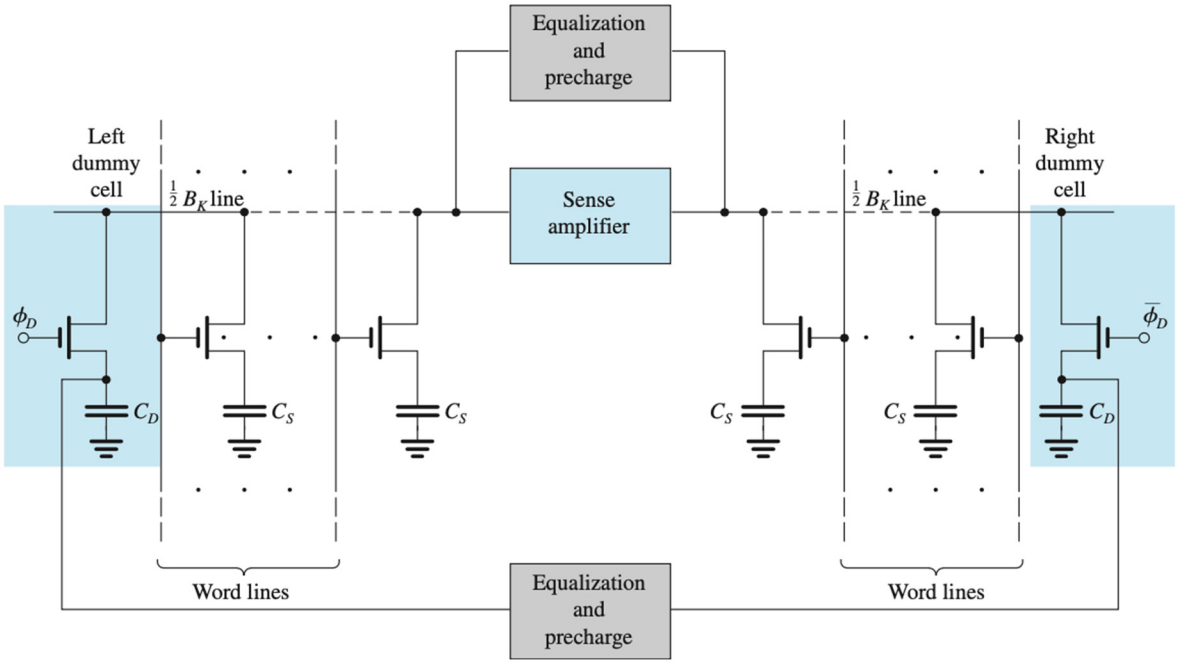


Figure 2.10 - Differential cell of a DRAM memory [33].

Alternative to the Precharge and Equalization circuit

Based on the same philosophy but using a different method to prepare the voltages in the circuit, in order to receive information from the memories, there this a different approach for the precharge and equalization circuit, this time loading the bit lines B and $\bar{B}$ not in VDD/2 but in VDD. It is possible to observe the solution in Figure 2.11, which when we have ϕ_P to VDD, it causes the bit lines to be loaded with a predefined value (in (a) this value is VDD and in (b) this value is VDD-Vt). The Q7 transistor is working as an equalizer of the voltages in the two bit lines.

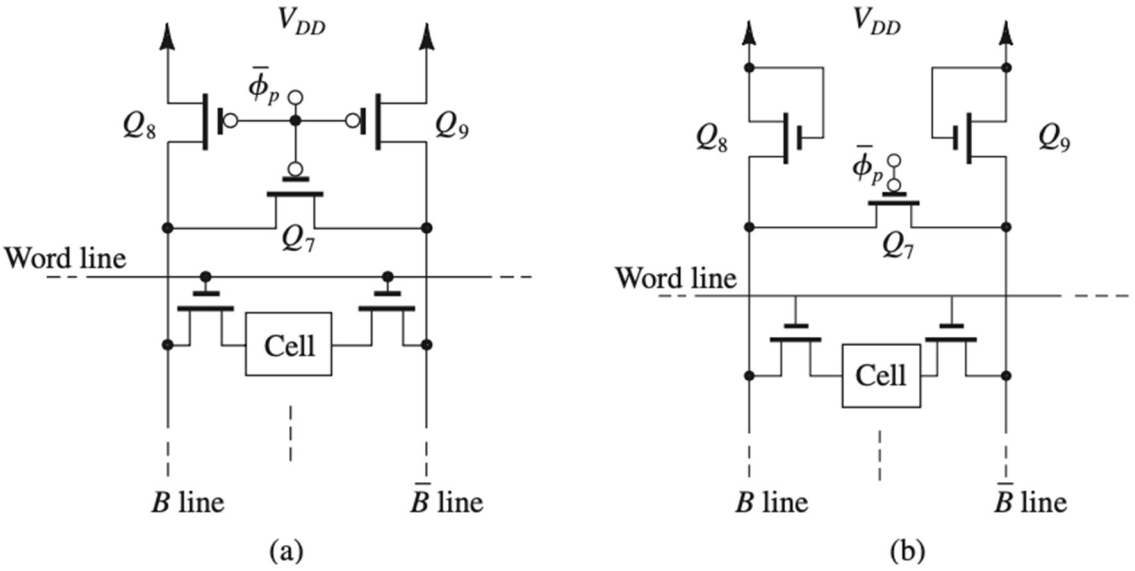


Figure 2.11 - Alternatives to the Precharging circuit. (a) bit lines loaded to VDD. (b) bit lines loaded to VDD-V_t [33]

Alternative to the Sense Amplifier

As an alternative to the Sense Amplifier mentioned above, there is the differential MOS amplifier. In Figure 2.12 it is possible to see this solution suggested as an alternative to the Sense amplifier. Q1 and Q2 transistors are controlled by the voltages of the corresponding bit lines, passing through them an I current from the transistor Q5. Transistors Q3 and Q4 form a current mirror that behaves like the charging circuit of transistors Q1 and Q2. Because it is a differential amplifier, it considerably facilitates its efficiency as a sensorial amplifier, rejecting noise signals and external interferences, amplifying only the small differences between the two bit lines with values resulting from the reading of a memory cell, where in a normal operation all its transistors are working in their saturation zone. In Figure 2.12(b) it is possible to see the amplifier in balance, with $V_B = V_{\bar{B}} = V_{DD} - V_t$ and the I current being equally divided by the transistors Q1 and Q2.

Since the moving currents are all the same, I/2, there is no current flowing to the capacitor C [33].

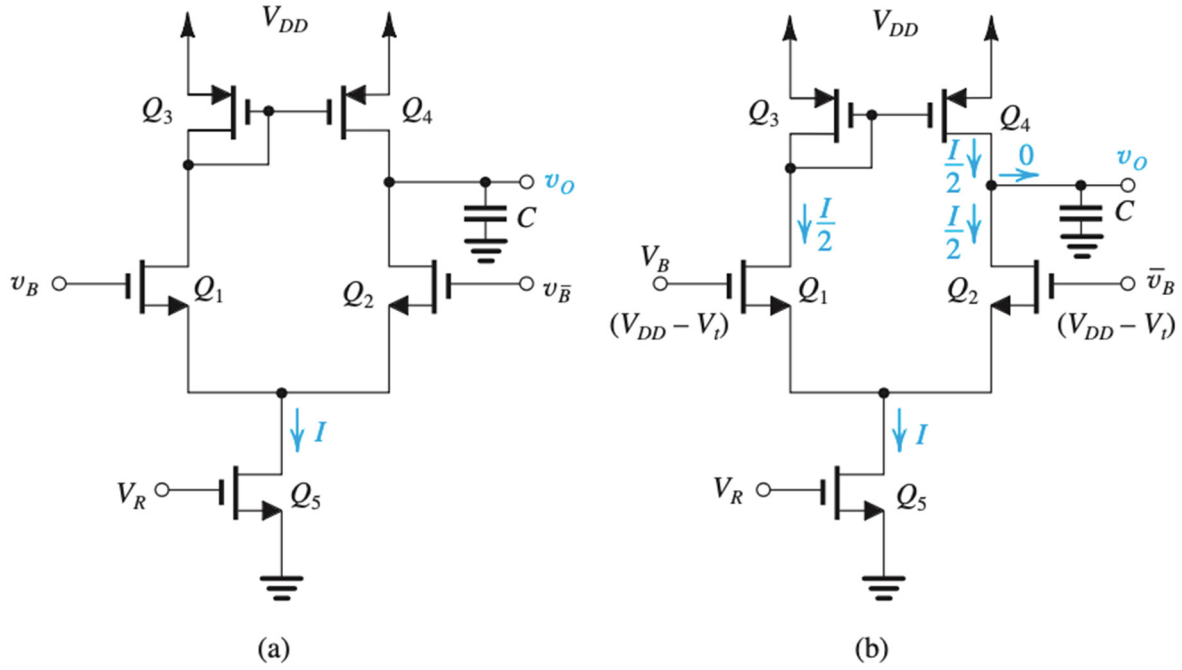


Figure 2.12 - Differential MOS amplifier [33]

Row-Address Decoder

A memory like the ones we've been seeing, are a synonym of a complex cluster of spaces where information can be stored. It is no use to have that information stored if later we are not able to access it or even store it in the place we want without jeopardizing the remaining information that may be stored around the memory location we want to use.

This is where the significance of having a component in our memory circuit capable of finding and selecting the exact memory location we want to work with comes in. The row-address decoder is able to select one of the 2^M word lines available marked by M-bit address. Considering the example of there being 3 bits to identify the memory place to be used, bit A_0, A_1 and A_2, we have a case of M=3, therefore 8 possibilities of address combinations (2^3=8), thus having eight word lines, W_0, W_1 ... W_7 [33]. The first line which may be selected is word line W_0, that will be activated when A_0=A_1=A_2=0, being possible to translate this into a Boolean function

$$W_0 = \overline{A_0}\,\overline{A_1}\,\overline{A_2} = \overline{A_0 + A_1 + A_2}$$

We can see that the word line W_0 can be selected by AND with its entries all denied or by a NOR.

With this example, it is possible to conclude that the row-address decoder may be built with eight NOR of three inputs connected to the decoder input. Each of these NOR will have in its entry a correct combination of bits to activate each of the available word lines [33].

Figure 2.13 illustrates a way to implement the row-address decoder with three input bits. The input NOR functions are built through an array with the various available bits which will activate the transistors that are building the resulting word. Afterwards, the final line transistor is activated through the ϕ_P that allows the reading of the memory address to be inquired. Since we only activate the transistor of the line we specifically want to read, this allows us that, during the pre-load of the row address, we can activate all the transistors, loading all the lines with its binary sequence [33].

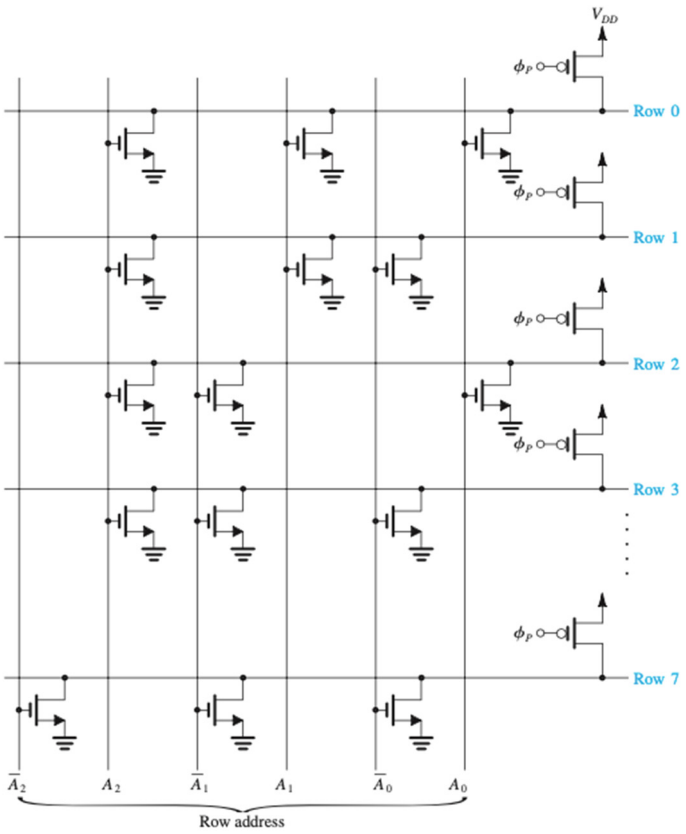


Figure 2.13 - Row-address Decoder made up of a NOR array [33]

Column-address decoder

Previously we have analyzed how the decoder of a line to access a specific memory place would be. Now we shall consider a column decoder with 2^N bits that connects the input

and output bit line from memory. For this it is necessary to use a multiplexer which may be built through the use of the pass-transistor logic as it may be seen in Figure 2.14. In this case the bit lines are all connected to the input and output line, with a transistor controlled by the decoder.

We can also optimize the performance of the decoder if instead of placing transistors to connect the bit lines, we place transmission gates, however in this case we must have complementary signals in the output [33].

In Figure 2.15 it is also possible to observe another way of implementing the column decoder, this time with a less amount of transistors in order to implement this solution. However, by removing transistors, we will damage the operating speed of the circuit, which will be slower than the solution presented previously. This solution is typically known as a tree decoder, which is composed of passing transistors.

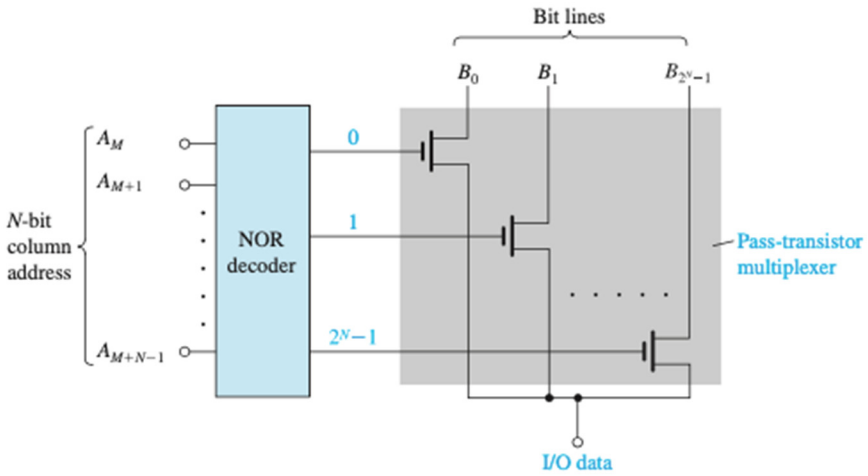


Figure 2.14 - Column decoder using the NOR and pass-transistor multiplexer combination [33].

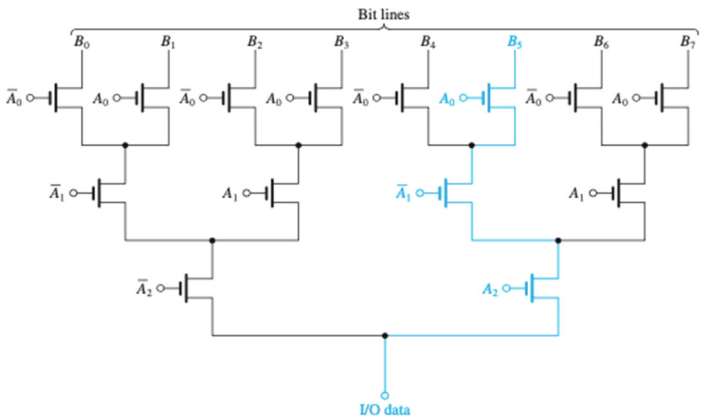


Figure 2.15 - Colunm Decoder in tree shape [33].

2.3. SNM - STATIC NOISE MARGIN

In the last years there has been a need to investigate the ultra-low-power operation and, consequently, the need to reduce the power-supply voltages of memory circuits for sub-threshold voltage values, meaning that these power-supply are below the threshold voltages of the transistors [34].

The energy reduction present in the memory brings out a problem in this type of circuits, which has always been present, but which at higher voltages makes it not so visible. By reducing the power-supply voltages, and consequently the energy, the noises introduced from outside assume values similar to those of the power-supply circuit, making it possible to start misleading the data stored in memory, drastically reducing the Static Noise Margin (SNM) [34].

2.3.1. ANALYSIS

The devices that are used nowadays need a great storage capacity to store information and, at the same time, need to save as much energy as possible, in order to maintain a minimum power consumption (making batteries profitable, for example). Moreover, systems on a chip (SoCs) and other integrated circuits, are a cluster of devices placed in several layers, some on top of others, with dimensions in the nano scale. This causes the power-supply lines to cross several layers and allowing to produce additional interference and noise [35]. In SRAM memories, these interferences are more significant when subthreshold voltage levels are used, causing the leakage currents and the energy in the memory to be very small. Consequently, the SNM is decreased.

When the memory cell is storing information, the word line is at 0 V (low), causing the NMOS access transistors to be inactive. In order to correctly maintain the received data, the memory cell must have its sense amplifier working, to trigger the values to extreme high or low voltages, being here that the SNM can drastically influence the values in the memory, since when the noise mixes with the signal, it can cause wrong values to be read or stored [34].

The Static Noise Margin (SNM) can be defined as the minimum DC noise voltage present at each of the cell storage nodes necessary to flip the state of the cell. This means that

the higher this margin is, the more reliable the values stored in the memory are, being the SNM responsible for measuring a "safety margin" for the memory circuit.

Figure 2.16 illustrates a 6T SRAM memory with the possibility of simulating the noise through the V_N voltage sources. Voltage sources are placed in the internal state nodes of the memory cell, in order to test the stability of the memory, and whenever the V_N values change, the stability of the cell will undergo changes. The cell coupling inverters remain in a bi-stable state, and their output nodes have the data stored in the cell. With the increase in the voltage of the V_N sources, in turn the increase of the noise, will cause the stability of the memory cell to decrease due to the fluctuations of voltages in the nodes. The purpose of the SNM is to measure the possible levels of noise for the correct functioning of the memory, i.e., measure the ability that these coupling inverters have to maintain the correct value in the outputs even in the presence of noise [35].

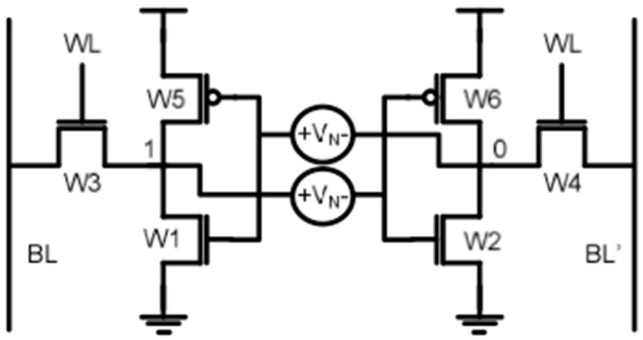


Figure 2.16 - Diagram of a 6T memory with noise voltage sources for SNM measurement [35].

Figure 2.17 presents a way of representing the SNM with a diagram of a memory cell with recorded data. In this figure, it is possible to see the voltage transfer characteristic (VTC) of the inverter 2, to the inverter 1, from Figure 2.16. The two-lobe curve represented is called the butterfly curve and represents the possibility of determining the SNM, being defined by the length of the side of the largest square that can be placed inside the lobes of the butterfly curve.

Taking the example where we have a transition from 0 to 1, this causes the inverter 1 to move down, as we can see in the image, and the VTC of inverter 2 to move to the right, moving the two curves and finding themselves only on two occasions [34].

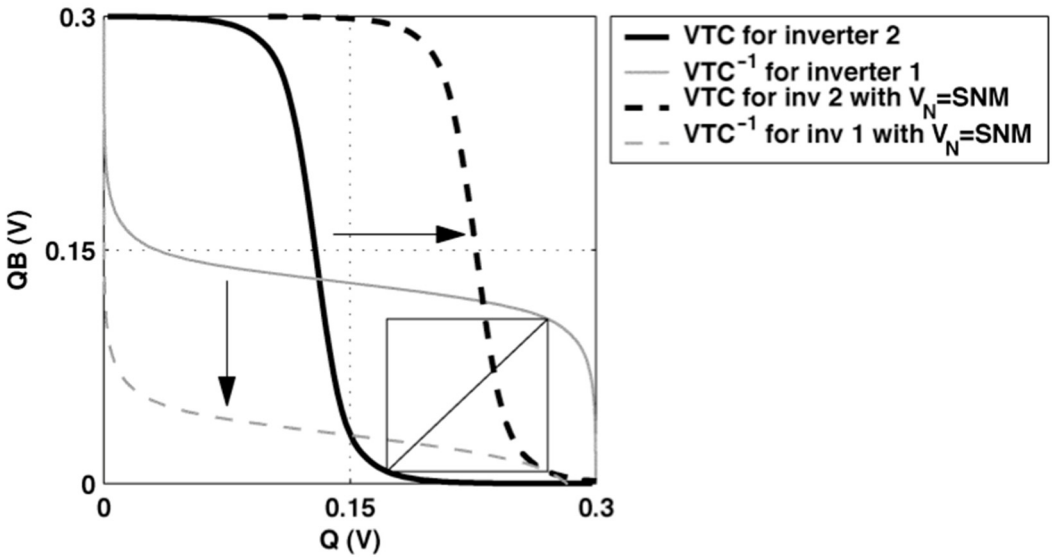


Figure 2.17 Diagram from the stability of a memory with the influence of the SNM [34].

2.3.2. SNM IN A 6T SRAM CELL

Taking as an example a 6T SRAM cell, to explore SNM and how it can influence the correct functioning of a memory, we will observe several processes which can be degraded by this situation.

In order to find the margin of the SNM needed in order to ensure the correct functioning of a memory cell, it is necessary to observe the VTCs of the coupling inverters on Figure 2.16. Taking the values obtained by one of the inverters and inverting that sign, making y = x, we form the butterfly curves mentioned above. The SNM value is the side of the largest square that can be placed inside the "eye" formed by the two butterfly curves, as we can see in Figure 2.17.

The values for the SNM may be obtained in the memory operations, such as reading or writing, data retention, among others, which will be covered below.

Data retention or hold noise margin

When the wordlines are not active, the isolated cell must retain its data at the outputs of the coupled inverters. In Figure 2.18 it is possible to see in (a) the equivalent circuit during data retention of a SRAM, and in (b) the circuit used to make the measurement of the SNM

during data retention, where a DC variable voltage is applied at node V1, measuring the output of node V2 [35].

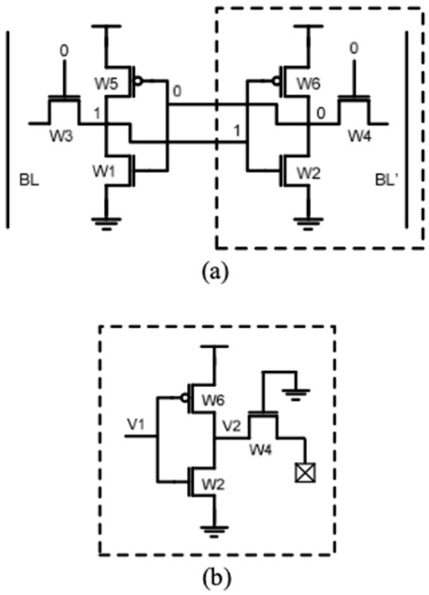


Figure 2.18 - (a) Equivalent circuit during data retention in a SRAM. (b) Circuit to measure the retained noise margin [35].

Read noise margin

As mentioned above, before a reading operation there is a need to carry out a pre-loading of the bit line in the cell to "prepare it" for the reading, which will executed. This makes this process more sensitive and susceptible to noise, making the cell more vulnerable when it is accessed during the reading operation, requiring greater attention to prevent the memory cell from changing its state during the reading cycle based on wrong data, making a destructive reading [35].

In Figure 2.19 it is possible to see the equivalent circuit during a reading operation, where the worst SNM is obtained in this process. At the beginning of the reading process, the bit lines are pre-loaded to VDD, then activating the word line in order to access the information. If the value to be stored is a 0, the bit line is passed to VDD and then this voltage is recorded in memory. On Figure 2.19(b) we observe the SNM measuring circuit during the reading, where one of the drives is connected to VDD to simulate a reading operation, and then a DC voltage variation is applied to node V1, then measuring VTC at node V2 [35].

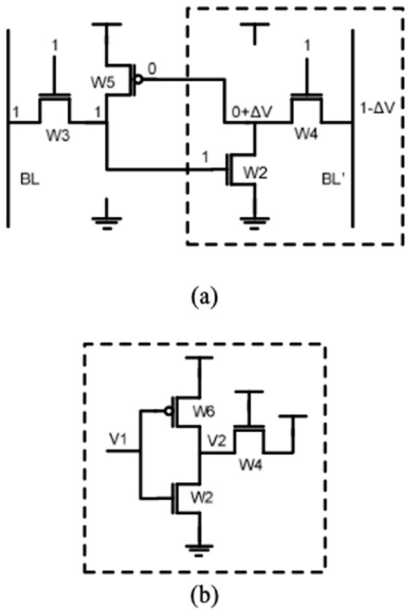


Figure 2.19 - (a) Equivalent circuit during a reading operation. (b) circuit used to measure the SNM in this process [35].

Write noise margin

When we approach a memory writing operation, the situation is different from the previous one, since the SNM in the data recording is defined as the minimum voltage in the bit line necessary to reverse the status of a previously worked memory cell. When a writing operation is activated, the data to be stored in the memory is placed in the bit lines and then the corresponding word lines are activated to access the memory locations desired to change [35].

Looking at Figure 2.20 and assuming that one wants to record in memory the value 0 and there is previously a 1 recorded in this memory location, this causes the coupling inverters to have complementary data at their ends, being their VTC's measured using different circuits as in Figure 2.20(b). The circuit on the right is the same as the one used for measuring the reading margin, while the circuit on the left represents the inverter with the value 1 at the output, having its bit line connected to VSS to simulate the recording of a 0, then applying a DC voltage variation at node V2, measuring the output voltage at node V1.

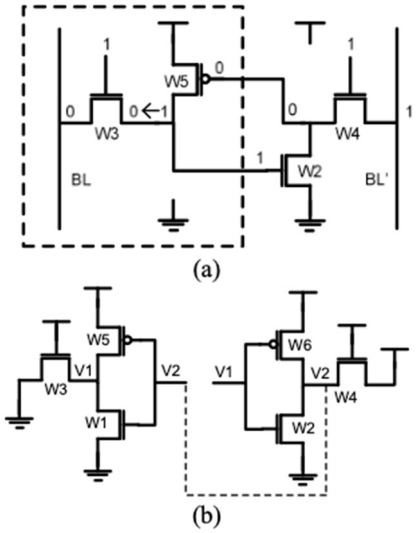


Figure 2.20 - (a) Equivalent circuit during a writing operation. (b) circuit to measure the SNM [35].

Effect of cell ratio and Pull-up ratio

In the operation of a SRAM, 6T cell are affected in the reading by the cell ratio and in the recording of data by the pull-up ratio. With this, it is possible to see that the SNM will also be affected, as it is dependent on these ratios.

The cell ratio is obtained by the proportion of the NMOS transistor sizes of the inverters, and the NMOS access transistor size. In turn, the Pull-up ratio is the division of the pull-down transistor size, which has the purpose of downloading the bit line preloaded to VDD for VSS, by the access transistor to the memory unit, the access transistor [35]. Observing Figure 2.19(a) it is possible to see the pull-down transistor identified by W2 and the access transistor identified by W4.

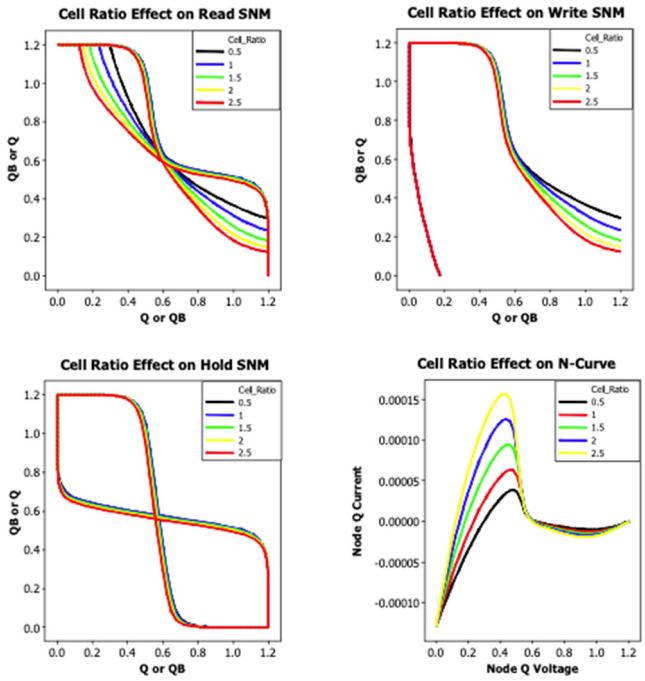


Figure 2.21 - Effect of the cell ratio on (a) Read Noise Margin, (b) Write Noise Margin, (c) Hold Noise Margin and (d) Noise curve [35].

Observing Figure 2.21, it is possible to see the effects of the SNM in the several operations: reading, writing, holding and N-curve. To obtain these premises of the SNM effect, different cell proportions were used, with values of 0.5, 1, 1.5, 2 and 2.5. It is possible to confirm what was mentioned before, that this effect is more visible in the reading operation, as seen on Figure 2.21(a), increasing cell's proportion, and in the cases of writing and holding, they do not have a significant change with the increase of the proportion.

Effect of the supply voltage and temperature

The supply voltage is directly associated with SNM problems. The goal is to have circuits that are as small as possible and also consume as little energy as possible. By analyzing Figure 2.22, it is possible to see the influence that the supply voltage has in the several functioning processes of a memory, where it has varied between 0,2V and 1,6V making it possible to confirm with the diagrams, that the noise margin is proportional to the supply voltage. This is due to the fact that with the decrease of the supply voltage, the noise in the components will increase, leaving the memory circuit more unstable during its operation. The supply voltage also defines the values that we will obtain at the output of the memory unit

making the noise even more harmful, by reaching values similar to those of the output signal [35].

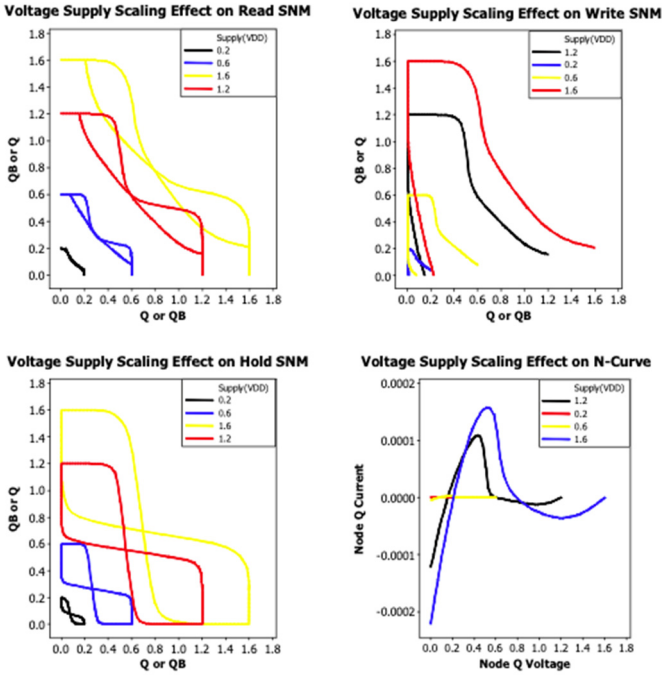


Figure 2.22 - Effect of the supply voltage in the functions of (a) Read Noise Margin, (b) Write Noise Margin, (c) Hold Noise Margin and (d) Noise curve [35].

Regarding the influence of temperature variations in the SNM, it is possible to see in Figure 2.23 that it is almost insignificant when compared with the supply voltage. In Figure 2.23 are represented results for -40ºC, 27ºC, 100ºC e 125ºC. As the temperature increases, the static noise margin decreases, in turn also decreasing the current that is needed to reverse the state of the cell, meaning that the memory cell becomes less stable with the increase in temperature.

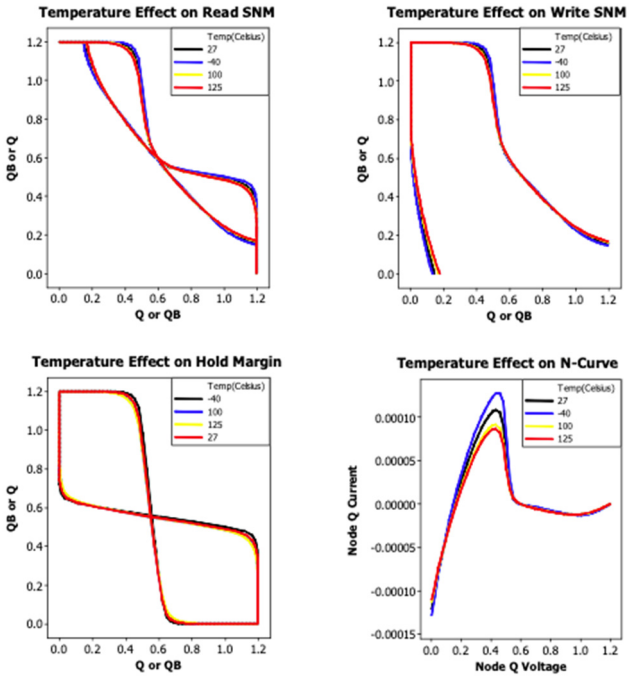


Figure 2.23 – Effect of the temperature on the (a) Read Noise Margin, (b) Write Noise Margin, (c) Hold Noise Margin and (d) Noise curve [35].

2.4. STATE OF THE ART ON PERFORMANCE SENSORS

Today, developing devices that are robust, in respect to their function, is a great challenge. As mentioned before, the smaller the manufacturing scale of chips, the greater are the challenges to make them robust, reliable and insensitive to outside influences that jeopardize their performance.

Making a robust equipment is halfway to its success in the market, because no one wishes to purchase equipment that has a very short durability or that only works in a set of limited conditions. Regarding memories, monitoring their operation using sensors is a way of making them more robust and reliable, improving the systems where they are used.

In this section of the document, we will address some previous work that aimed to mitigate these negative influences on memory performance, making them more reliable, regardless of the working conditions in which they are operating.

Unfortunately, there are not many published works on sensors for monitoring the performance of a memory cell. The OCAS sensor [16], being an aging sensor, measures the performance of the memory regarding this type of degradation. Another work already done on performance sensors for memories, this one to measure the performance in SRAM memories,

is the work presented in [37]. However, this work has some limitations, since its operation is very dependent on a stable clock signal and on the internal delays implemented by hardware, not being able to change its sensibility and calibrate the sensor after it has been implemented. Regarding DRAM memory cells, as far as authors knowledge, there are no previous works on sensors to measure performance, regardless the effects that cause the performance variations. However, for application in sequential circuits, there are examples of sensors of this type, such as the Scout Flip-flop [38].

One of the purposes of the present research work is, precisely, to create a sensor with the same functionalities as the SCOUT, but for SRAM and DRAM memories.

This way, this section will present with further detail the 2 works already developed for memories, [16] and [37].

2.4.1. ON-CHIP AGING SENSOR (OCAS)

Component degradation is usually associated with two phenomena, Bias Temperature Instability (BTI) and Hot Carrier Injection (HCI), causing a deviation of the component limit voltage over time. Of these two main effects, the BTI is the one that has the greatest influence in CMOS circuits' aging. It can be in the form of a PBTI (positive BTI), that affects NMOS transistors, and NBTI (negative BTI), that affects PMOS transistors, the latter being the one that creates the most wear in SRAM memories.

A solution to detect the aging degradation state of a memory, knowing whether it is aging to such an extent that it jeopardizes the correct manipulation of data, is OCAS (On-Chip Aging Sensor), which makes it possible to analyze the results created by the NBTI in the SRAM memories during their entire functioning. This solution proposes that a sensor of this kind be placed in each column of memory data, performing off-line tests periodically, in order to monitor the operating condition of the memory [36].

By observing the block diagram on Figure 2.24, it is possible to see the connection from OCAS to the SRAM column, with the test mode being controlled by the TT1 transistor that connects the real VDD and the virtual VDD'.

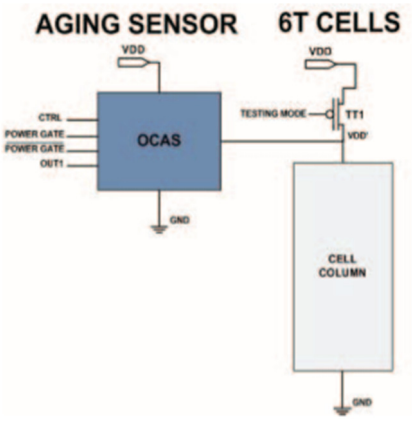


Figure 2.24 - Block Diagram of an OCAS [36].

Analyzing Figure 2.25, it is possible to see that the OCAS is controlled by the TPG and TNG transistors, which have the Power-Gating as a control signal of the entire OCAS, being able to turn on and off the previously mentioned transistors. This signal mainly serves to enable the test circuit to be switched off during normal memory operation, thus reducing the aging of the test circuit, and avoiding the sensor to lose its ability to detect the aging of the circuit to which it is attached.

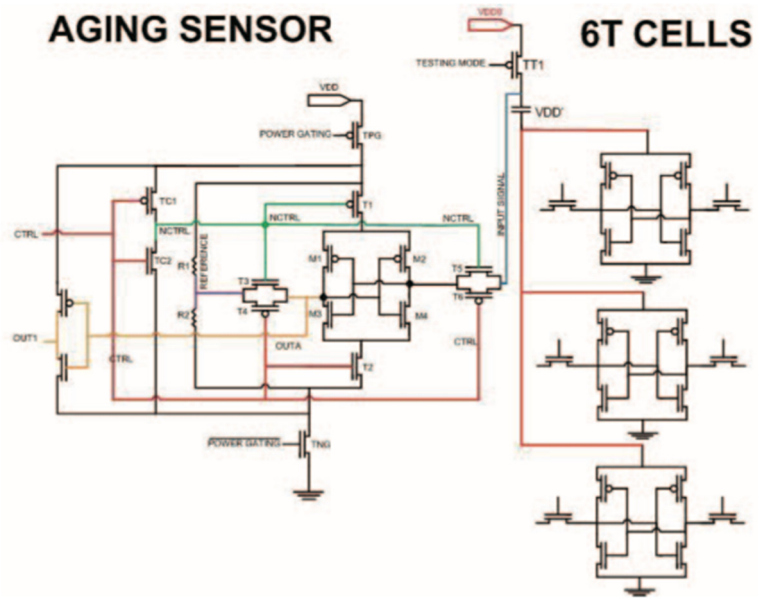


Figure 2.25 - Diagram of an OCAS [36].

In order to activate the test mode, it is necessary to activate the OCAS through the Power-Gating signal, which in turn connects the TPG and TNG transistors, while the TT1 transistor is simultaneously inactive, since it is an offline test, and the memory unit is not

running during the test period. During the test, a specific memory unit, which one intends to use, is activated through a writing operation, making a comparison between the VDD value and a value previously defined in OCAS at the end of the operation. If this value is within what is defined and accepted by the sensor, it will place in its OUT1 output the logical value “0”, which indicates that everything is working within the stipulated, or puts the logical value "1", reporting an error in the memory unit, which is not in a condition to continue working. In turn, the CTRL signal is placed with the logical value “0” during the entire pre-loading stage of the test mode, causing the TC1, T3, T4, T5 and T6 transistors to change to bu turned on, so that the signals which will be verified can pass to the voltage comparator formed by the transistors M1 and M4.

When the sensor moves to the phase of evaluation and comparison of the stored signals, the CTRL signal is defined with the logical value of "1", this causes the transistors TC1, T3 and T6 to be disabled, and transistors TC2, T1 and T2 to be switched on, allowing the input signal to be evaluated by M1 and M4.

Briefly, to know the aging condition of a SRAM memory, the following steps are performed:

1. Chose and select the place of the memory to be verified and read its information;
2. Turn the test mode signal on the previously selected column from "0" to "1";
3. Set the CTRL signal to "0" (Precharge mode) and write the opposite signal to the one read during phase 1;
4. Change the CTRL signal to “1” (Evaluation Mode) and observe the output of the OCAS to know if any error was detected.

In order to guarantee that the test circuit is working correctly, there is a small complementary circuit whose purpose is to carry out that verification, executing an auto-test to the sensor before it evaluates the memory cells. This circuit is composed of two resistors (R3 and R4 in series), in the same configuration as resistors R1 and R2, but connected to the drain of transistors M2 and M4. The voltage produced here is the same as the one from the VDD node after a sequence of two writing operations in an aged cell that was activated during the test mode, so that when the OCAS test mode is activated, the OUT1 output has the logical value of "1", which indicates the occurrence of an error [36].

OCAS displays some disadvantages, namely the fact that, in order to carry out a test, it needs to set the memory offline, resulting in periods of time where it will not be operational for the user. It also requires all memory units to have a chip of this type, causing an overhead

in the size of the memory. This solution also does not allow its use in DRAM memories and does not contemplate the PBTI effect in the memories where it is applied.

2.4.2. PERFORMANCE SENSOR FOR SRAM

The sensor presented in [37] is an on-line sensor to monitor the aging and performance in SRM memory cells implemented in CMOS, which allows the detection of time degradation in the access to SRAM memory cells, in their reading and writing operations. This sensor is connected to the memory bit lines, so that it is able to detect when there is a change in the bit line logic value. The purpose of this sensor will be to detect slow transitions of the bit line signal, thus signaling the aging of the memory if the transitions are not done in a predefined time (hardware defined). This monitoring and signaling is done in "real-time", i.e., not being necessary to turn off the memory unit to be tested, as in the previously presented solution.

With the existence of PVTA variations, the physical properties of the memory and the operation are affected, endangering the good performance of transistors, particularly their response time. The degradation of operating conditions by PVTA effects, or any other effect that affects the performance of a memory, ultimately causes the switching time between low and high binary states to be increased, leading to slower transitions. Having said this, by monitoring the switching time of signal transitions in a memory allows us to test its aging over time, or also to test the degradation of the operation, caused by PVTA effects or by any other effect (although the PVTA effects are the most important).

This sensor is composed of two large blocks, according to the following figure, and it is connected to a bit line. Monitoring the switching time of the bit line makes it possible to have the information of how fast the memory is when performing the read and write operations, thus monitoring its performance.

Figure 2.26 - Blocks diagram of an aging and performance sensor of a SRAM [37].

By observing Figure 2.26, it is possible to see that the sensor consists of a transition detector, which will be explained in detail below. It is also possible to see the pulse detector, which has the purpose of sensing the pulse that comes from the transition detector. The pulse,

has a duration proportional to the transition time occurred in the memory signals. The pulse detector, generates a signal proportional to the pulse duration (which in turn is proportional to the memory signal transition time), and decides if the value is above or below a predefined value. In case of being higher, it signals out an error in the memory, since the transition time was slower than the hardware pre-defined time considered as acceptable.

Transition Detector

The Transition Detector consists of a set of inverters. As it may be seen in Figure 2.27, the Transition Detector is made up of two paths with 4 inverters each. Each path has *n*-type inverters (inverters with the *n* transistor being highly conductive, when compared with the *p* transistor) and two *p*-type inverters (inverters with the *p* transistor being highly conductive, when compared with the *n* transistor), placed alternately along each path. The difference from both paths is the placement order of the type of inverters used, since in the upper path the first inverter is *n*-type and in the lower path the first inverter is *p*-type. This results in one path being always faster than the other (one is faster for a low-to-high transition, while the other is faster for the high-to-low transition), which leads to a fraction in time where the output of both paths is different. This fraction of time will vary according with the amount of aging in the circuit, and since the outputs of the inverters are connected to an XOR gate, it generates a pulse at its output whenever the inputs are different. Therefore, the generated pulse has a time duration equivalent to the time when the signals are different in the inverter paths, which is also proportional to transition time of the input signal. Moreover, this pulse changes proportionally with the aging state of the components, or with a PVTA variation.

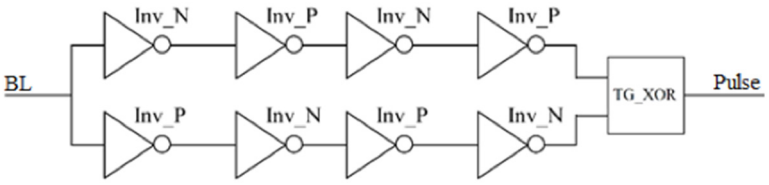


Figure 2.27 - Transition Detector [37].

Pulse Detector

The Pulse Detector's purpose is to verify the stability of the system and detect when the duration of the pulses generated in the Transition Detector are higher than a specific time defined by the clock signal of the system.

By observing Figure 2.28 it is possible to see that the Pulse Detector consists of a delay element, an inverter and a stability-checker. The delay element is, basically, a buffer that provides a delay to the input signal. The stability-checker is used to detect transitions in the delayed pulses, but only during the "0" state of the clock. Therefore, the clock signal is used as a timing reference to detect abnormal delays in the pulses generated by the Transition Detector. If a pulse is generated in the Transition Detector during the active state of the clock, but if its duration and the propagation delay of the delay element makes the pulse to last until the clock signal switches to "0", it causes a transition in the delayed input of the stability-checker and the Pulse Detector will signalize an error at the output. This means that the transition occurred in the bitline is slower than expected, denoting an unsafe condition for the memory, or an unsafe PVTA variation, or an unsafe performance reduction in the memory operation.

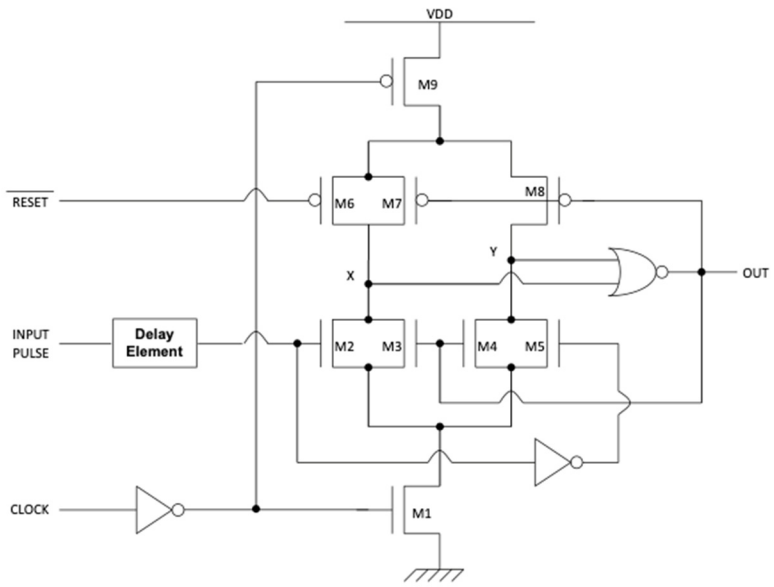


Figure 2.28 - Pulse Detector's implementation[37].

Having said this, there are always two parameters which may be controlled to verify the existence or absence of errors:

1- The width of the pulses generated by the Transition Detector;
2- The delay entered in the Pulse Detector by the Delay Element.

In Figure 2.29 we see a summary of the operation of the sensor for the detection of slow transitions in the bitlines. The grey/dashed areas represent the possible change in the signals according to the transition speed of the bitline signals' transition in the memory, indicating a degradation in the system's performance.

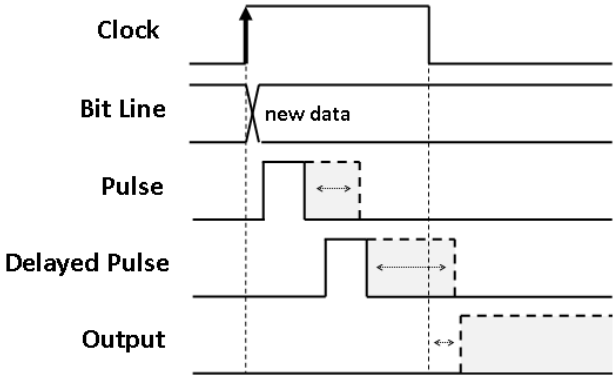


Figure 2.29 - Pulse Detector with the Stability-checker [37].

In a more recent publication [38], this same sensor solution was improved, resulting in a new Pulse Detector, based on the functionality of a NOR gate, as it can be seen in Figure 2.30. The underlying idea is to use the NOR functionality to detect when simultaneously two signals are at low state. Considering that signals in the memory are generated in the rising edge of the clock, the pulses in the pulse detector's input will occur also during the high state of the clock. Hence, considering that a small duration pulse will be produced for a fresh memory cell (denoted here as a cell with no PVTA degradation) and that a large duration pulse will be produced when a performance error occurs, the error detection should occur during the low state of the clock. Using the NOR functionality, the input pulse must be inverted, i.e., active low. Besides, to allow a better control of the error/non-error pulse durations, the authors included a delay element in the input signal (PULSE) to postpone pulse activation.

In comparison with what was proposed in [37], this new solution for the Pulse Detector from [38] has three main advantages: complexity, reliability and power. The new architecture is less complex than the previous one, reducing from 17 to 11 transistors and generating a smaller sensor area. Regarding reliability, the solution in [37] has an intrinsic delay which becomes prohibitive when VDD is reduced, making the solution improper to work with DVFS.

On the contrary, the solution in [38] is much more stable and reliable when working at reduced power-supply voltages (or even sub-threshold voltages), because of the simpler behavior based on the NOR gate. Regarding power, the work in [37] uses dynamic CMOS logic, which imposes constant switches of signals in every clock cycle. This behavior imposes higher dynamic power dissipation when compared with the classic CMOS NOR gate behavior of the pulse detector from [38]. Moreover, there is also a small difference on the RESET signal, which in [37] is active LOW and in [38] is active HIGH.

The basic idea of this second version of the pulse detector (from [38]) is to use the main clock signal as a fixed reference to detect abnormal delays in the pulses generated by the transition detector. In the memory (and in a common digital circuit), all the control signals and all the instructions are synchronously generated with the main clock. Therefore, considering that pulses in the transition detector are generated during the active state of the clock, if pulse duration and the propagation delay of the Delay Element makes the delayed pulse to reach the NOR during the low state of the clock, an error signal will be generated.

Hence, by design, two parameters control the delays in the sensor and the error/non-error decision: (1) the sensitivity of the transition detector (controlled by the number of inverters used in the two paths) and thus, the width of the pulses generated in the transition detector; and (2) the time delay introduced in the pulse detector's Delay Element. During the design stage, spice simulations allow to determine the slowest delay for which the error signal is not produced, for a specific maximum frequency and PVTA variation. If a higher PVTA variation exists, or if frequency is raised, and error signal is produced at sensor's output. Note that this error signal should indicate, with a certain safety margin, that the performance of the circuit is on the eminence of a delay-fault, and does not indicate a real error in a read/write operation. This error signal should allow corrective actions to take place and avoid real errors. Thus, by design the sensor is hardwired programmed to a maximum clock frequency and a maximum PVTA degradation.

The NOR-based pulse detector implementation (as shown in Figure 2.30), uses 4 transistors to implement a CMOS NOR logic gate (M1, M2, M4 and M5), controlled by a clock signal (CLOCK) and delayed pulses (Delayed Pulse). The inverter and transistors M3 and M6 ensure, in case of a detection, that the output signal (OUT) remains active until a reset occur (this allows to exempt the use of a latch to keep the sensor active in case of an error). The reset signal (RESET) controls M7 transistor operation, and reinitiates all the circuit for a new detection [38].

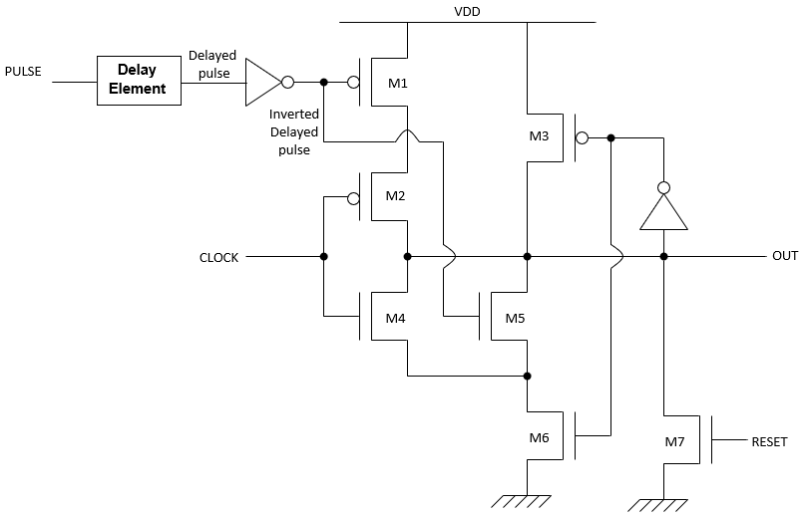


Figure 2.30 - Pulse Detector's new implementation [38]

In Figure 2.31 we can see the shaded areas where a slow transition happens in the *bit line*, resulting in a bigger pulse in the Transition Detector which, in turn will be signaled as an error if it is too big.

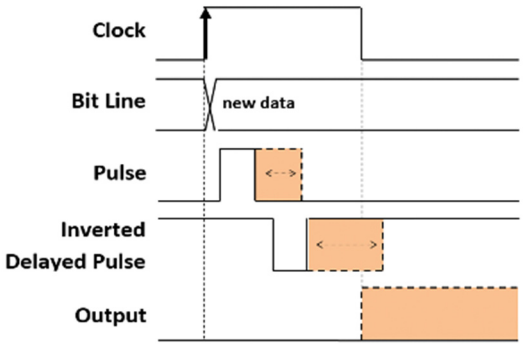


Figure 2.31 – New Pulse Detector with the Stability-checker [38]

This Performance Sensor has the disadvantage of needing to be synchronous with the memory, depending on the ascending and descending flanks of synchronous signals, thus imposing limitations on its operation. One other functionality not implemented is the possibility of the user to be able to calibrate or change sensor's sensitivity, which would make it more versatile and suited to the desired application. This sensor also doesn't allow its applicability on a DRAM memory.

3. SCOUT MEMORY SENSOR

One of the main problems when designing an SRAM cell is to maintain its stability. This stability is basically the cell's capacity of working correctly in the presence of noise signals, thus ensuring the correct reading, writing and recording operations. The static noise margins (SNMs) are frequently used as a stability criterion [17]. However, some authors defend that the dynamic noise margins are also important [18]. Nonetheless, due to the PVTA variations (and also knowing that aging is a cumulative process), a degradation may happen in the performance and stability of the memory.

Process, power-supply voltage, temperature and aging (PVTA) variations are main conditioning factors in the design of technology nowadays [24]. With process variations, the transistor's attributes (such as gate length, oxide thickness, diffusion depth and Vth variation, mainly due to random dopant fluctuation) differ from their nominal value. Temperature variations are mainly due to the switching activity, dependent on the work load, short-circuit and leakage currents of the transistors. Power-supply voltage variation (voltage drop) is mainly a result of changes in the power-supply grid, dependent on the workload, and on the power-supply wires, which can be model by a resistance and an inductance that affect the voltage drop. Transistor's aging (for example, NBTI, PBTI) increases the absolute value of the threshold voltage, which is the effect of the degradation of transistors' conductivity. In resume, PVTA variations have a significant impact in electronic circuits, both for logic or memories. For example, from the layout's perspective, if there is a transistor on a logic gate that is already weakened, the accumulated charge can generate a voltage failure. The failure will spread through the circuit resulting in an error, if stored on a flip-flop [25]. This process may be categorized in three different stages: failure generation, failure spread and failure block [24].

Given the PVTA variation problems in memories, these will naturally be reflected in the processing of data that are in bit lines. Therefore, as we want our sensor to be able to monitor the information from all the processing problems which may exist at the moment, we use the bit lines as the place to monitor memory performance. The problems that the PVTA variations places in the memory components are, ultimately, reflected in the speed with which transition occur in the bit lines, which means that the slower these transitions are, the greater the probability of an error occurrence, or it means that an error is in the eminence to occur, and our sensor will be able to detect it. Working online, the sensor will monitor bit line

transitions and analyze its delays to detect performance degradations in the memory, which reflects a PVTA variation and may cause an error. The purpose is to avoid error occurrence by detecting errors predictively, or predicting the eminence of an error. Similarly to the performance sensor for synchronous circuits presented in [38], as the focus of our work is to online monitor SRAM and/or DRAM circuits, we call it the memory sensor version for the Scout sensor, or simply the scout memory sensor (the name *scout* stands for "performance Sensor for toleranCe and predictive detectiOn of delay-faUlTs").

3.1. SENSOR ARCHITECTURE

The sensor which will be presented in this chapter has the purpose to alert the user for performance variation problems. It will not alert which problem happened, or which PVTA variation occurred (or any other), but it will give a warning that something is affecting the performance of the memory and, with this, it triggers an alert.

The *scout memory sensor* has a controller block, implemented with a Finite State Machine (FSM) with 3 states, *Reset*, *Sample* and *Compare*, to define and control all the operation. The first state ensures that the sensor's start-up conditions are always the same as when a new test is carried out, thus resetting specific components in the circuit. The *Sample* state is the state in which the sensor is waiting for a bit line switch that can be analyzed by the sensor, and the state machine can remain in this state while the *Pulse_detected* control variable remains at 0. When this variable changes to 1, this means that a new pulse has been detected and the state machine changes to the following state, the *Compare*. In this *Compare* state the sensor will compare the signal generate by the bit line transition with the sensor's reference value, in order to be able to indicate if the transition is safe or unsafe (in the latter, outputting an error).

The scout memory sensor consists of 3 parts: the Transition Detector, the Pulse Detector and the Comparator. The Transition Detector is responsible for receiving the bit line transition and transforming the duration of this transition into a pulse with a proportional width, i.e., a fast bit line transition will create a pulse with a short duration, while a slow bit line transition will create a pulse with a large duration. The structure of the Transition Detector varies depending on the type of memory and on the type of initialization value used in the memory bit lines. In this work, 3 models and architectures are being presented: a model for an

SRAM memory initialized to VDD; a model for SRAM memory initialized to VDD/2; and a model for DRAM memory initialized to VDD/2.

The Pulse Detector is responsible for sensing the voltage that will charge the capacitor used in the *Compare* state, and storing the voltage to be compared. The Pulse Detector presented has the improved feature of being able to be change its sensibility online, or its sensitivity can be tuned online during circuit's lifetime. This happens because it consists of 3 transistors that can be activated separately or in combination, and because of their different sizes, the sensor can be adjusted or calibrated to several different sensitivity levels, being more or less sensitive to the signals that are being analyzed.

The Comparator is responsible for comparing the signal coming from the Pulse Detector with the reference value of the sensor, being the latter the threshold which decides between considering a transition as a success or as an error for being slower.

In Figure 3.1 it is possible to observe how each signal flows between the sensor and the memory cell. It is also possible to perceive the connections and signals from each block inside the sensor, seeing which signals pass between the Transition Detector, the Pulse Detector, the Comparator and the Controller (the state machine).

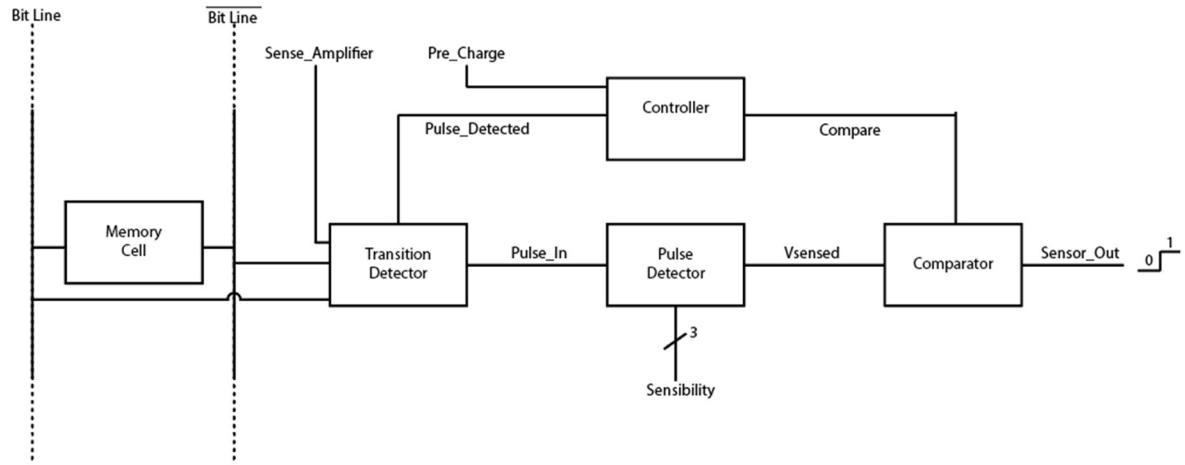


Figure 3.1 - Performance sensor block diagram

Looking at the previous figure in more detail, it can be seen the Sense_Amplifier signal, which is the signal used to activate the sense amplifier and is used here in the Transition Detector to indicate when there is a memory write or read, being a trigger to activate the sensor. Connecting the Transition Detector and the Pulse Detector we have only the pulse signal with the generated pulse. In the Pulse Detector we also have the bus to control the sensitivity, here

implemented with 3 lines, the Sensibility control signals. Connecting the Pulse Detector and the Comparator we have the signal of a pulse to be "analyzed" in the Comparator. From the latter we have the Sensor_Out signal that is active at 1 if an error is detected. The Controller receives a signal from the Transition Detector, the Pulse_Detected, that indicates when a pulse has entered, so that the state machine can advance to the next state and activates the Compare signal for the Comparator block, so that it starts the comparison between the sensed voltage Vsensed and the reference voltage.

3.1.1. TRANSITION DETECTOR

When a memory is being read/write, there are changes in its bit lines signals. The purpose of the Transition Detector is to detect those signal transitions in the bit lines, in order to be able to generate a pulsed signal, with the pulse width being proportional to the delay of the bit line signal transition, which will be used in the reminding parts of the sensor. According to the memory performance status, the transition speed of the bit lines from low-to-high and from high-to-low has a tendency of becoming slower when the memory ages, or generally when a PVTA degradation occurs.

In normal sensor operation, the transition detector will detect all the transitions that occur in a *bit line*. However, we can identify transitions if we have a pre-charge operation, to prepare the reading/writing operations, the read/write operations, or even a refresh operation in case of the DRAM. Moreover, it is also important that the transition detector block can observe transitions in the bit lines when they are initialized to different initial values (for instance, SRAM can be initialized to VDD or VDD/2, and DRAM can be initialized at VDD/2). Therefore, this block has several implementations, according to the memory type.

The advantage presented in this work is that, regardless of the implementation type of the Transition Detector chosen, the circuit onwards will always be the same.

Transition Detector for Initialization at VDD

The VDD initialization of a memory is the typical case of an SRAM, having its bit lines initialized at VDD, and the complemented bit line at VSS, meaning that Transition Detector will detect the transitions from VDD to VSS in bit Line.

The presented structure for the transition detector is based on the concept described in Hugo Santos' thesis [37], for the same transistor detection block. As it was described in section 2.4.2, the transistor detector block has a set of *p*-type and *n*-type inverters, placed alternately along two paths, as we can see in Figure 3.2 (in the upper path the first inverter is type N, and in the lower other path is type P). Note that a *p*-type inverter is here defined as an inverter with unbalanced *n* and *p* transistors, and with the *p* transistor with a higher conductivity when compared with the *n* transistor, which means that its output will rise quickly when compared to its fall transition. In the same way, an *n*-type inverter has a more conductive *n* transistor, when compared with the *p* transistor, meaning that when a signal switches in the bit line from 0 to 1 (this will put on the *n* transistor of the inverter), *n*-type inverters will process that signal faster than *p*-type inverters (if a 1 to 0 transition occurs in the bit line, the opposite will occur). Consequently, as the different inverters are placed alternately along the paths, one path is always faster than the other. To create these unbalanced inverters, and comparing their implementation with a balanced inverter (with similar conductive transistors), the highly conductive transistor has a channel that is 5 times bigger than the minimum size used in the balanced inverter. In turn, these 2 inverter paths are connected to an XOR gate, which works by outputting a logical value "1" when the input signals are different, and the logical value "0" when the input values are equal. Since we have one of the paths faster than the other (one is faster for the low-to-high transition, while the other is faster for the high-to-low transition), this will cause the XOR gate to have a period of time in which the input lines are with a different logic value, causing the output of this XOR gate to be activated during a period of time proportional to the bit line transition delay.

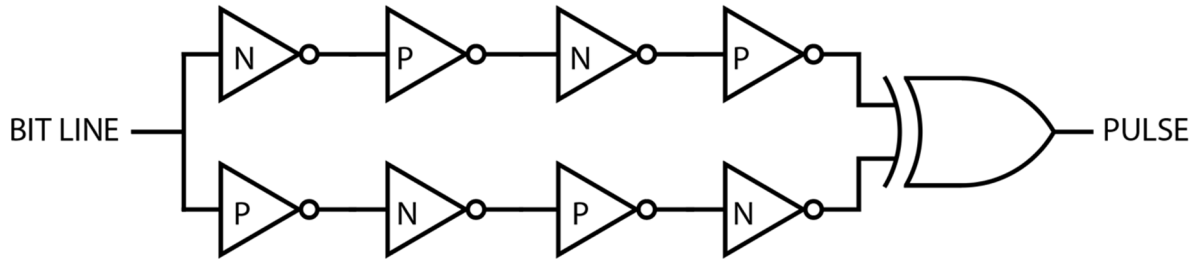


Figure 3.2 - Transition Detector in a bit Line.

For the construction of the Transistor Detector, the transistors used correspond to what is described in Table 1.

Path	Inverter	Nmos	Pmos	L	V_{th_n}	V_{t_p}
1	Xinv1	5*WNmin	WPmin	65n	0.423V	-0.365V
	Xinv2	WNmin	5*WPmin			
	Xinv3	5*WNmin	WPmin			
	Xinv4	WNmin	5*WPmin			
2	Xinv1	WNmin	5*WPmin			
	Xinv2	5*WNmin	WPmin			
	Xinv3	WNmin	5*WPmin			
	Xinv4	5*WNmin	WPmin			

Table 1 - Transistor Detector – Size of the Transistors

Since the main purpose is to detect the transitions on a SRAM with a pre-charge to VDD, it is necessary to monitor the transitions that occur in the bit line and also in the complementary bit line. Note that, because both bit lines are pre-charged to VDD, this means that in a read/write operation only one will switch, because only the high-to-low transition will occur. For example, if a 0 is read, the bit line signal will switch from 1 to 0, while the complementary bit line will remain in the pre-charged 1 value (on the contrary, if a 1 is read, the bit line will remain in the pre-charged 1 value, while the complementary bit line will switch to 0). This makes it necessary to change the circuit from Figure 3.2 and create a new implementation for the transition detector, presented in Figure 3.3, which allows us to detect all the existing transitions in the circuit. Now, a read/write operation is monitored whether a 0 or a 1 is being read/write, as both bit lines are monitored.

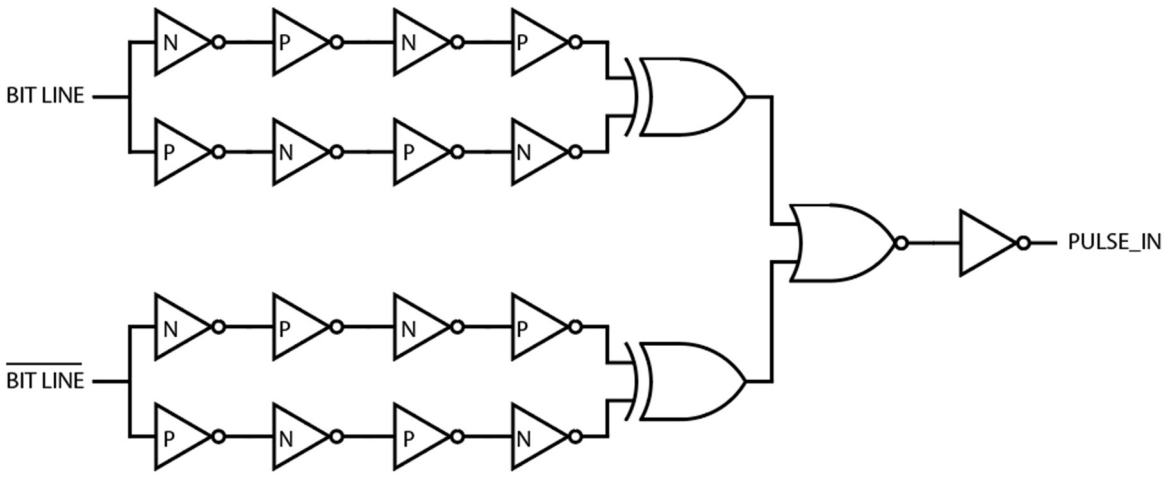


Figure 3.3 - Transition Detector for VDD initialization.

In order to carry out the necessary verification tests of the proper functioning of the Transistor Detector, the circuit was implemented in HSPICE:

```
***Inverters in the bit line
Xp11_inv BL p11_inv vss! vdd! INVN0
Xn12_inv p11_inv n12_inv vss! vdd! INVP0
Xp13_inv n12_inv p13_inv vss! vdd! INVN0
Xn14_inv p13_inv n14_inv vss! vdd! INVP0
Xn21_inv BL n21_inv vss! vdd! INVP0
Xp22_inv n21_inv p22_inv vss! vdd! INVN0
Xn23_inv p22_inv n23_inv vss! vdd! INVP0
Xp24_inv n23_inv p24_inv vss! vdd! INVN0
Xxor_signal n14_inv p24_inv Pulso_in1 vss! vdd! XOR20
***Inverters in the complementary bit line
Xp11_inv2 BLB p11_inv2 vss! vdd! INVN0
Xn12_inv2 p11_inv2 n12_inv2 vss! vdd! INVP0
Xp13_inv2 n12_inv2 p13_inv2 vss! vdd! INVN0
Xn14_inv2 p13_inv2 n14_inv2 vss! vdd! INVP0
Xn21_inv2 BLB n21_inv2 vss! vdd! INVP0
Xp22_inv2 n21_inv2 p22_inv2 vss! vdd! INVN0
Xn23_inv2 p22_inv2 n23_inv2 vss! vdd! INVP0
Xp24_inv2 n23_inv2 p24_inv2 vss! vdd! INVN0
Xxor_signa2 n14_inv2 p24_inv2 Pulso_in2 vss! vdd! XOR20
***final XOR to create the pulse
Xor_pulso Pulso_in1 Pulso_in2 Pulso_in_inv vss! vdd! NOR20
Xor_inv Pulso_in_inv Pulso_in vss! vdd! INV0
```

To test this implementation, several simulations were made in order to demonstrate that the Transition Detector was working as it was intended.

The first test simulates two different transition delays in the bit line, to demonstrate that the Transition Detector would correspond in the same way. This means that, the higher transition times in the bit line produce longer pulses generated by the Transition Detector, as we can see in Figure 3.4. In this case there is a variation in the transition time of 200p seconds, which shows a visible increase in the pulse generated in the Transistor Detector.

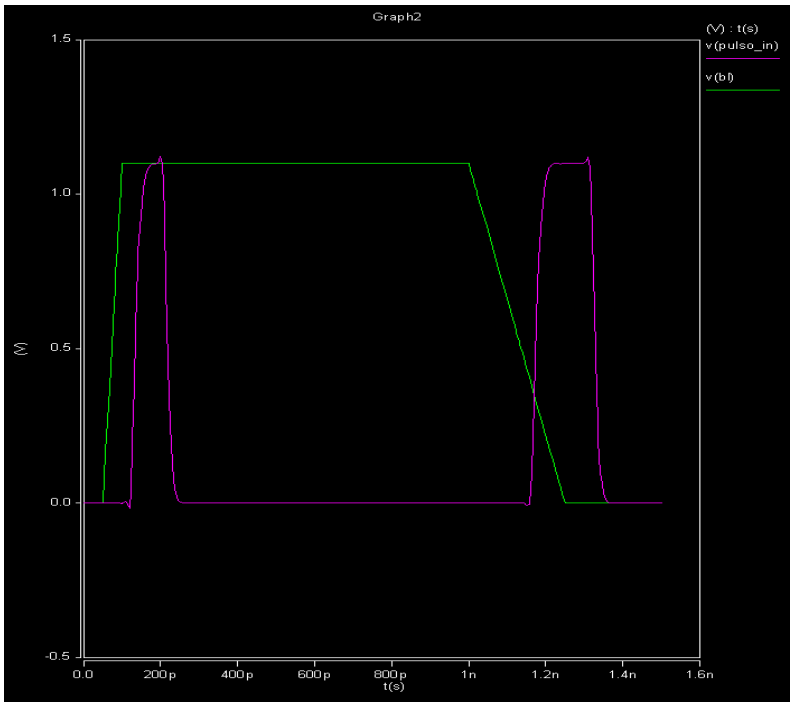


Figure 3.4 - Variation in the bit line VS Pulse generated by the Transition Detector.

Simulating again, but with several switching delays in the bit line, in Figure 3.5 it is possible to see once again the generated pulses with the a width proportional to the switching delay of the bit line.

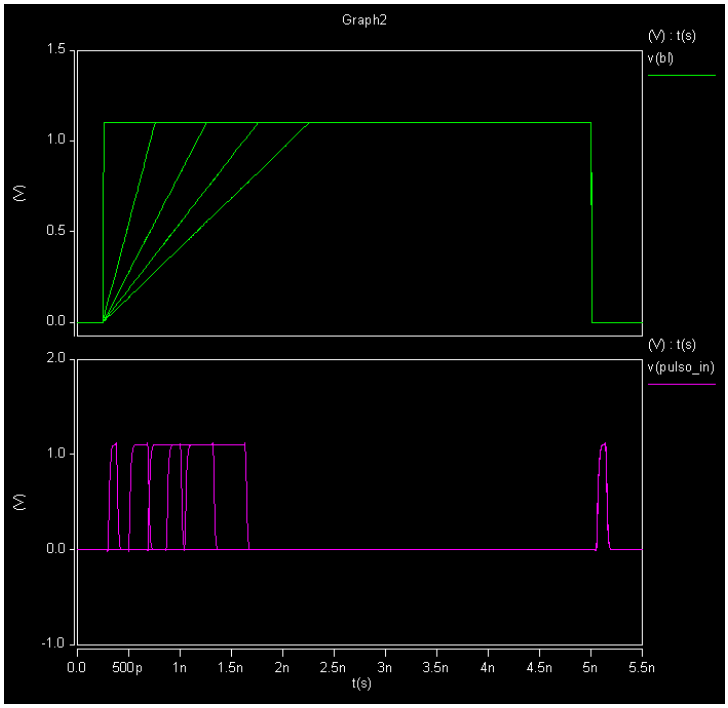


Figure 3.5 - Transition Detector - bit line duration varying.

These differences in the transition delay of the bit line, in the previously presented cases, were imposed by simulation, which means that we directly changed its transition time. Since the purpose is to demonstrate that the performance sensor is sensitive to the working conditions, let us test its sensitivity to temperate variations. In Figure 3.6 it is possible to see the variations of the pulse generated by the Transition Detector when there is a change in the operating temperature of the circuit.

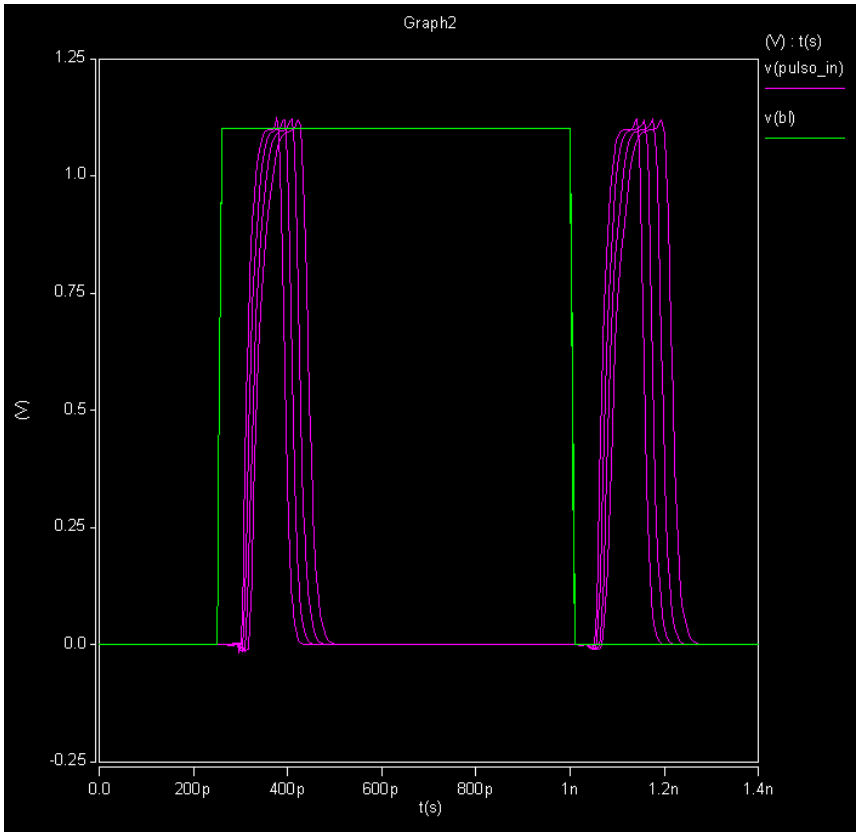


Figure 3.6 - Transition Detector – Temperature variation of the circuit.

In Figure 3.6 it was simulated changes in the temperature from 27ºC to 100ºC, with an increase of 20ºC in each new simulation. It is easy to notice the increase of the pulse width as the temperature is also increasing, thus showing that the sensor is sensitive to temperature changes.

There is also the possibility of a supply voltage variation, thus jeopardizing the performance of the memory being monitored. The next test is done by simulating two fast transitions in the bit line but at different power-supply voltage values.

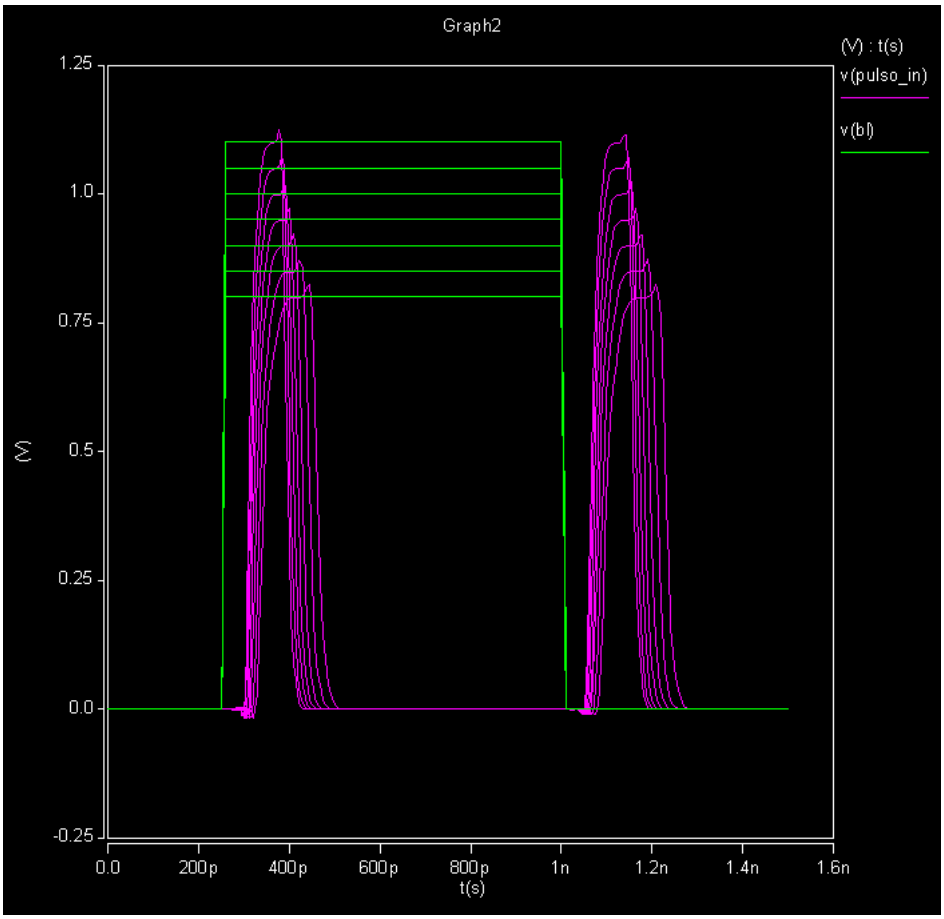


Figure 3.7 - Transition Detector – Supply Variation.

As it can be seen in Figure 3.7, the variation in the supply voltage of the circuit also influences the size of the generated pulse, and as the power-supply voltage decreases, the generated pulses are longer in time.

For better understanding how the Transition Detector operates, it is possible to see in Table 2 the differences in the generated pulse width, regarding the working conditions which may affect its performance (in this case, for different values of power-supply voltage, temperature and transistors' threshold voltage, which models different aging conditions). In the first line there is the nominal working conditions for VDD, Temperature and aging, which originates a pulse with 0.0897 ns. In the second line we see a change in the temperature of 100 degrees, which causes the pulse to increase its duration to 0.134 ns. In the third line what changed was the supply voltage of the circuit, which was changed to 0.8V, causing the generated pulse to be of 0.132 ns. The fourth line shows an aging degradation condition, by increasing 10% in the absolute value of transistors' threshold voltage (*p* and *n* type transistors), to emulate aging effects, which generated a pulse with 0.0962 ns. Finally, it was tested the sum of all possible problems, with the variation of temperature, supply voltage and transistors'

threshold voltage, making the generated pulse to be the largest of all, as expected, with 0.2333ns.

Temperature	VDD	Vth_N	Vth_P	Value
27º	1,1 V	0,423 V	-0,365 V	0,0897ns
100º	1,1 V	0,423 V	-0,365 V	0,134ns
27º	0,8 V	0,423 V	-0,365 V	0,132ns
27º	1,1 V	0,4653 V	-0,4015 V	0,0962ns
100º	0,8 V	0,4653 V	-0,4015 V	0,2333ns

Table 2 - Transition Detector – Final pulse size

Transition Detector –VDD/2 Initialization

Some memory implementations use in the bit lines a different pre-charge value, i.e, the initialization of the bit lines, prior to a read/write operation, is made with a value that is not VDD. For instance, in a DRAM memory, the bit lines' initialization is made to VDD/2, making the transition to VDD or VSS powered by the sense amplifier, to be faster and with smaller amplitude in volts. This is a constraint that may require a different transition detector implementation, and this is needed for DRAM, but also for SRAM initialization at VDD/2.

Given this problem, there was a need to create a different Transition Detector implementation, capable of detecting these smaller and faster transitions. Its functioning is based on the previous one, with the difference that we need to have access to the Sense_Amplifier signal from the sense amplifier (the signal that enables the sense amplifier and triggers the transition from a near VDD/2 voltage value in the bit line, to a VDD or VSS final value), which together with the set of inverters connected to a NAND gate, is able to create a pulse with a duration proportional with the transition time that occurred in the bit line.

Leta us consider first the application for a DRAM. In this case, this Transition Detector implementation allows us to connect only to one of the bit lines, as a DRAM cell is only connected to one bit line, leaving the complementary bit line with an independent sensor (or without a sensor, in case of only specific pre-defined bit lines are monitored).

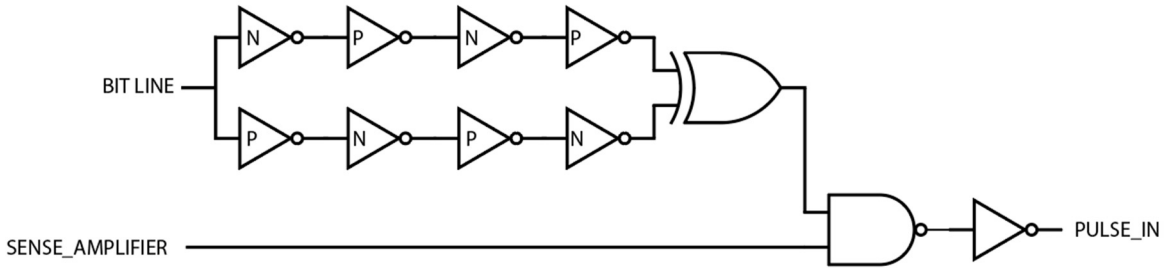


Figure 3.8 - Transition Detector initialized in VDD/2.

As we can see in Figure 3.8, part of the Transition Detector is similar to the previous one, but now with an additional input signal, Sense_Amplifier, to control the transitions that are being detected, as the pre-charge of the bit lines could generate higher pulses, jeopardizing sensor operation.

For the implementation of the Transistor Detector, specifically for the unbalanced inverters, the size of the transistors are the same as those used for the previous transistor detector implementation, and are detailed in Table 2.

In order to carry out the necessary verification tests of the proper functioning of the Transistor Detector, the same was implemented in HSPICE:

```
***Inverters in the bit line
Xp11_inv BL p11_inv vss! vdd! INVP0
Xn12_inv p11_inv n12_inv vss! vdd! INVN0
Xp13_inv n12_inv p13_inv vss! vdd! INVP0
Xn14_inv p13_inv n14_inv vss! vdd! INVN0
Xn21_inv BL n21_inv vss! vdd! INVN0
Xp22_inv n21_inv p22_inv vss! vdd! INVP0
Xn23_inv p22_inv n23_inv vss! vdd! INVN0
Xp24_inv n23_inv p24_inv vss! vdd! INVP0
Xxor_signal n14_inv p24_inv Pulso_in1 vss! vdd! XOR20
Xnand_pulso Pulso_in1 WL Pulso_in2 vss! vdd! NAND20
Xinv_pulso Pulso_in2 Pulso_in vss! vdd! INVN0
```

With some small changes, this Transition Detector implementation can also be used in SRAM memories that are initialized to VDD/2, if we want to monitor both bit lines, as it is possible to depict in Figure 3.9. In this case we added a new NAND gate at the end, because it allows to include both pulses from both bit lines in the sensor.

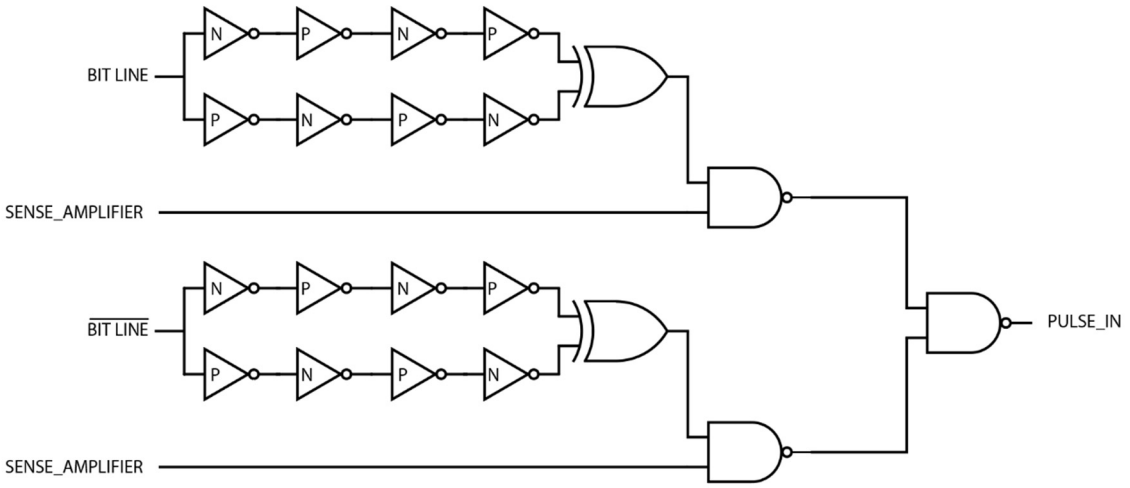


Figure 3.9 - Transition detector for SRAM initialized in VDD/2.

Note that for an SRAM with initialization to VDD/2, both versions from Figure 3.8 and Figure 3.9 can be used, because it depends on the monitoring strategy defined for the bit lines and for the entire memory.

For this Transition Detector model, it is also possible to see the differences in the generated pulse width as the transitions in the bit line are slower or faster. In Figure 3.10 it is possible to observe the variation of the bit line transition time between 10ps and 1ns with increases of 0.2ns, which results in an increase in the duration of the generated pulses, which is proportional to the increase of the bit line transition time.

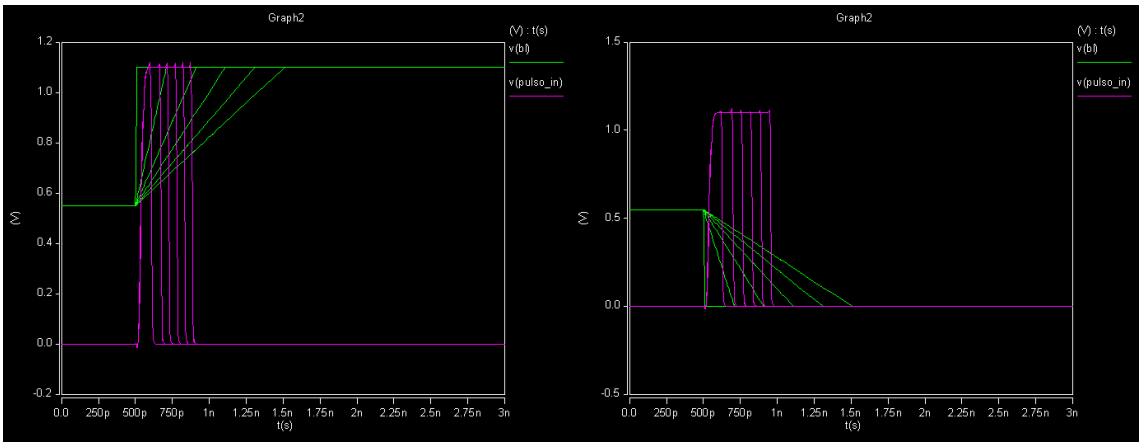


Figure 3.10 - Transition Detector - a) VDD/2 Variation to VDD; b)VDD/2 Variation to VSS.

Considering now a temperature variation of 27º and 100º with increases of 20º, the Transition Detector is still able to detect this temperature increase, reflecting it in the size of the pulse generated, as it may be seen on Figure 3.11.

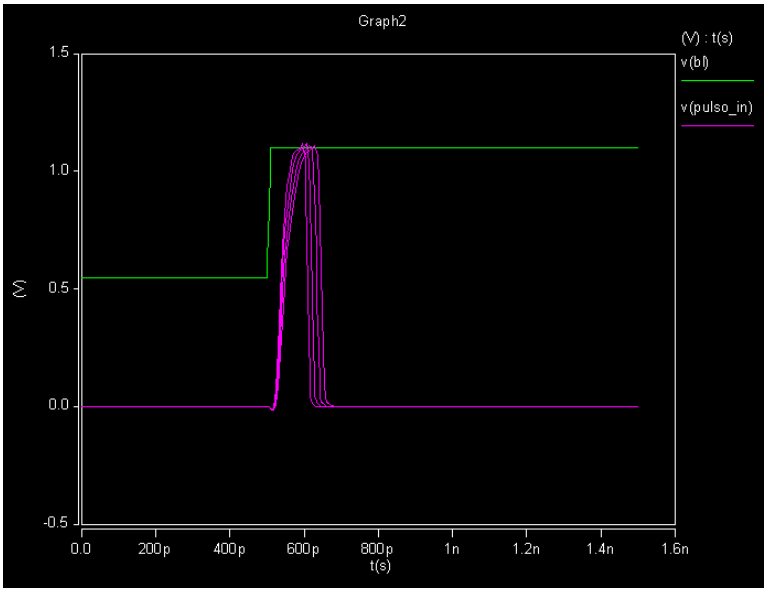


Figure 3.11 - Transistion Detetor - Variação da temperatura.

Moreover, as one would expect, by varying the supply voltage between 0.8V and 1.1V with a 0.1V increase, the generated pulse is also proportional to the supply voltage and reflects in its duration the supply voltage reduction value, making long pulses for the lower the supply voltage values, as it may be seen in Figure 3.12.

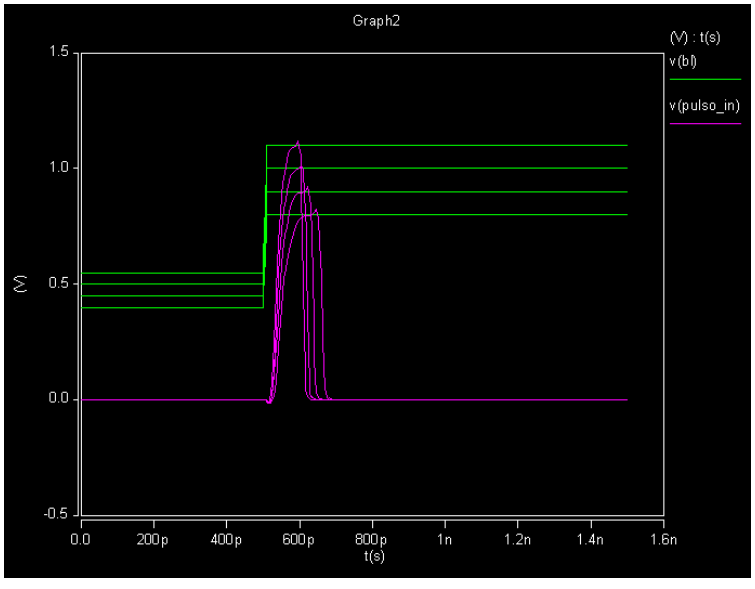


Figure 3.12 - Transition Detector – Supply Voltage variation.

3.1.2. PULSE DETETOR

The Pulse Detector is a crucial part of the performance sensor. The Transition Detector allows to detect the transitions that may exist in the bit line and transform them into a pulse to be used in the performance sensor. However, there is a need to have a Pulse Detector, which can receive the information that comes from the Transition Detector and convert it to a DC voltage value (Vsensed), proportional to the pulse width. In a simple manner, its operation is based on receiving the pulse from the Transition Detector and charge a capacitor with a voltage level proportional to the pulse width.

The Pulse Detector presented here (Figure 3.13) includes a new feature that all other Pulse Detectors and performance sensors in previous works did not provide: this is the possibility to change sensor's sensibility during online operation, which allows tuning the sensor according to the specific memory that is being monitored and its operating conditions, or to perform sensor calibration procedures during its life-span. To implement this feature, the Pulse Detector has 3 NAND gates controlling 3 PMOS transistors, which make it possible to change sensor's sensitivity by changing the current that will charge C1 capacitor. This is carried out through a 3-wire input signal, SENSIBILITY, which allows C1 capacitor to be charged quicker or slower, according with the required sensor's sensitivity, with 7 different sensibility levels (note that if more sensibility levels are required, more bits can be used, controlling additional NAND gates and PMOS transistors).

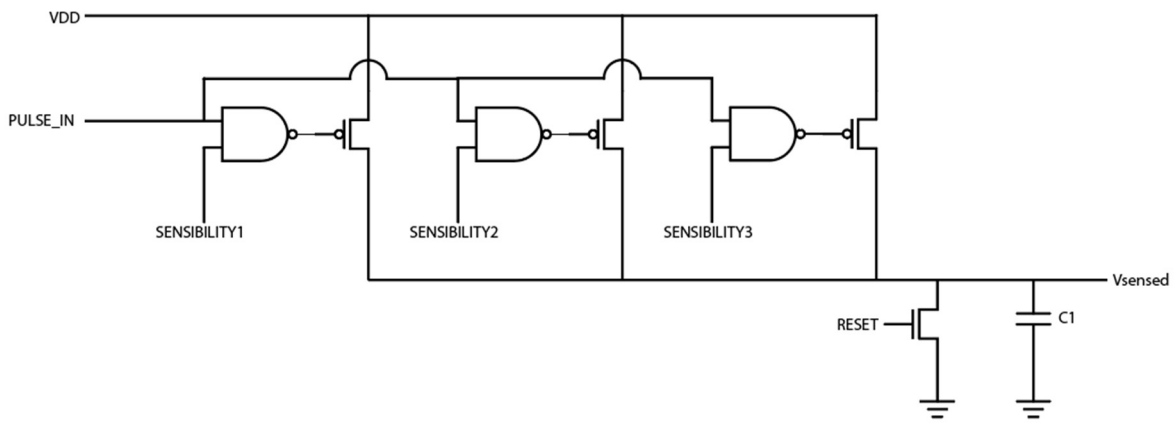


Figure 3.13 - Pulse Detetor.

As it may be seen in Figure 3.13, the Pulse Detector proposed for this sensor consists of 3 control sections. What changes in each of them is the minimum size of the transistor, which is being controlled by the NAND, as seen in Table 3.

Path	Size PMOS	L	V_{th_n}	V_{t_p}
SENSIBILITY1	WNmin	65n	0.423V	-0.365V
SENSIBILITY1	2*WNmin			
SENSIBILITY1	4*WNmin			

Table 3 - Pulse Detector – Size of the Transistors

With this configuration, it is possible to make several combinations to change sensor's sensitivity during its operation. Regarding the size of the transistors, the capacity of the capacitor will also be designed so that the sensitivity of the sensor is correctly tuned.

In order to carry out the necessary verification tests for the Pulse Detector, an HSPICE implementation was made:

```
****** Voltage Sources for controlling sensitivity ******
Vsensibility1 controlo1 vss! dc vdd
Vsensibility2 controlo2 vss! dc vdd
Vsensibility3 controlo3 vss! dc vdd
******* INPUT TRANSISTORS TO CHANGE CHARGING CURRENT******
Xnand1 Pulso_in controlo1 P1 vss! vdd! NAND20
XM11 N1 P1 vdd! vdd! PMOSFET w=WNmin
Xnand2 Pulso_in controlo2 P2 vss! vdd! NAND20
XM12 N1 P2 vdd! vdd! PMOSFET w='WNmin*2'
Xnand3 Pulso_in controlo3 P3 vss! vdd! NAND20
XM13 N1 P3 vdd! vdd! PMOSFET w='WNmin*4'
********** C1 CAPACITOR **********
CC1 N1 vss! 80ff
```

The voltage in capacitor C1 (Vsensed) will represent a sensed voltage obtained from the bit line transition, i.e., a DC voltage proportional to the bit line transition delay. This sensed voltage (Vsensed) stored in C1 should be compared with the reference voltage (Vref) also defined in the performance sensor. The sensed voltage must be calibrated for the "decision" to

be made when the sensor should consider a bit line transition as an error or as a success. There is also the possibility of calibrating the Pulse Detector through a change in the capacity of the C1 transistor, since changing its capacity it will also change its charged voltage. Therefore, sensor's sensibility could also be implemented by controlling the connection of several capacitors in parallel, but this approach would require higher silicon area, when compared with the used approach.

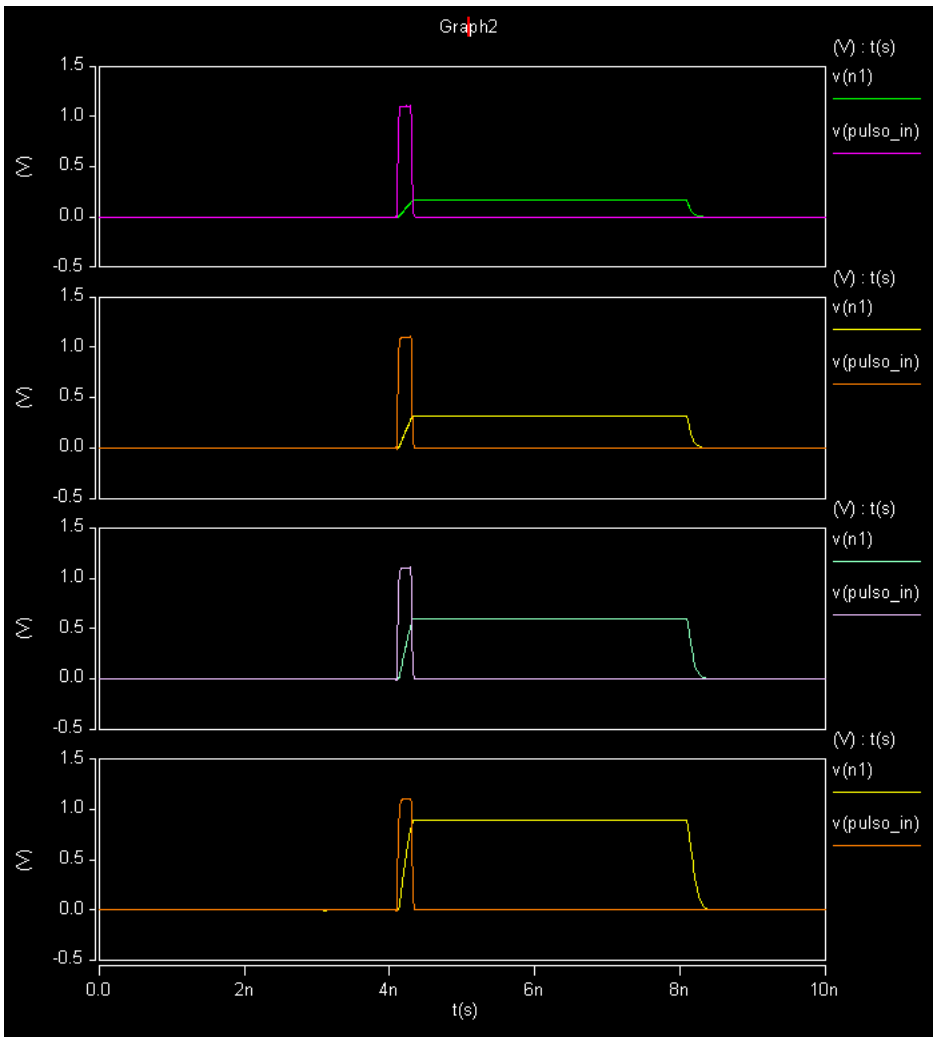


Figure 3.14 - Pulse Detector with several configurations.

By observing Figure 3.14, it is easy to understand how the proposed Pulse Detector works. The four simulations were carried out in the same conditions, only changing the transistors that charge the C1 capacitor (through the Sensibility 3-wire bus). The period used

was 2n seconds, the input pulse in the four simulations was exactly the same, as also the capacitor C1 value, with 80ff. Note that Vsensed signal is identified by node *n1*.

In the first diagram only Sensibility1 signal was activated which, in turn, caused the pulse_in to load the C1 capacitor only through the first transistor which has a size of WNmin. The second diagram, refers to the same simulation, however only Sensibility2 signal was active, causing the C1 capacitor to be loaded through the second transistor which has a 2*WNmin size. The third diagram changes from the two previous ones since it only has active the Sensibility3 signal, loading the C1 capacitor through the third transistor, which has a 4*WNmin size. In the last diagram it is possible to see a loading of the C1 capacitor using all 3 transistors. This way, it is easier to understand the positive aspects of this Pulse Detector, since it easily allows to change sensors sensitivity allowing to calibrate the sensed voltage that one intends to have for comparison, making it a much more versatile version when compared with the previously presented solutions.

3.1.3. REFERENCE VALUE FOR COMPARISON

It is extremely important to be able to create a reference value that will serve as a basis for comparison with the value stored in the Pulse Detector's C1 capacitor and, thus, be able to know whether the sensor is facing a performance error or not.

This reference circuit consists of a small circuit designed in order to, presumably, age more than the memory. This higher aging degradation will assure that the sensibility of the sensor is higher when aging conditions are worse. As it may be seen in Figure 3.14, the reference value placed in the output is controlled by the threshold voltage of the NMOS transistor (Vref = VDD-V_{TH}), which is always in a stress-mode condition and charging the C2 capacitor. The Vref output will have VDD-V_{TH}, thus causing our reference value to be slightly below VDD.

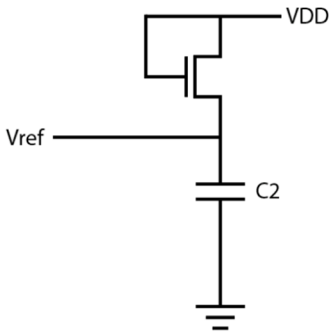


Figure 3.15 - Pulse Detector – Reference Value.

In order to carry out the necessary verification tests for the proper functioning of the reference value generator, the HSPICE simulation of the circuit was carried out:

```
XMH1 N2 vdd! vdd! vss! NMOSFET w='WNmin*2'
CH1 N2 vss! 100ff
```

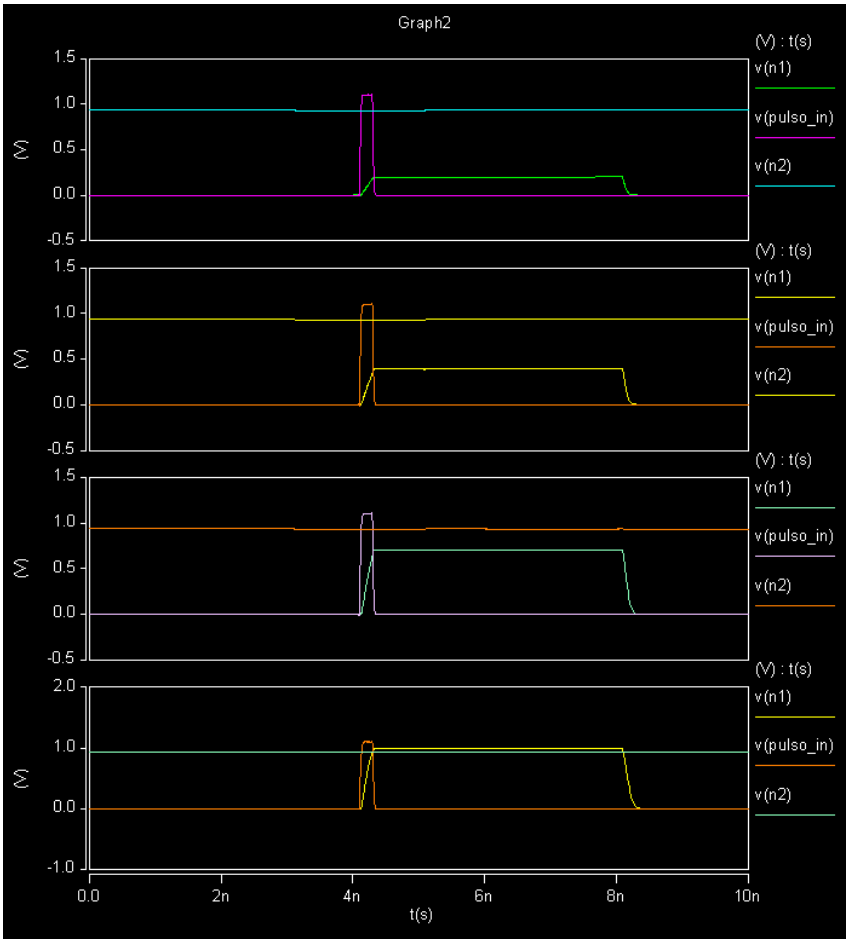


Figure 3.16 - Pulse Detector with reference value.

Similarly to what was seen in Figure 3.14, on Figure 3.16 there are the same signal configurations, but now with the addition of the reference voltage to the graphs and simulation, which is represented in node *n2*. Therefore, it is possible to see that in the first 3 simulations the sensor result would be considered as a successful writing or reading of the memory, since the pulse value generated by the transition in the bit line would generate a sensed voltage (Vsensed, identified by node *n1*) smaller than the reference value (Vref, identified by node *n2*). This means that it was a quick transition which in turn generated a shorter pulse.

However, in the fourth simulation it is possible to see that node *n1* is higher than the voltage in node *n2*, which means that the voltage stored in the C1 capacitor is higher than the reference voltage. This would trigger the alarm that a writing or reading predictive error occurred.

Since in the four simulations the only thing that was changed were the transistors that were selected from the Pulse Detector, having the input pulse always with the same dimension, it is possible to see that without having to change anything in the implementation of the sensor, and without having to make any change in the memory, it is possible to change sensor's sensibility and also to calibrate the Pulse Detector during its operation, so that it detects an error or not, or in this case, we decide from which value we want to consider an error or not.

3.1.4. SIGNAL COMPARATOR

To identify if an error signal should be generated in the sensor output, it is necessary to make a comparison between the sensed voltage and the reference voltage (the Vsensed signal and the Vref signal), as explained in the previous section. The Signal Comparator presented in Figure 3.17 has the purposed to amplify the differences between Vsensed and Vref voltages, triggering the values to the extreme Low / High voltages, VSS / VDD. This means that the signals Vsensed and Vref will be sampled through the through the transmission gates, activated by the *Sample* signal, and these sample voltages will be compared and amplified just like a sense amplifier does, when reading a memory cell. This is done by activating the *Compare* signal, causing the two cross-coupled inverters to be turned on. So, using the cross-coupled inverters to amplify the differences between Vsensed and Vref voltages that were sampled through the transmission gates, the amplifier works as a

comparator, by identifying the weakest and the strongest signal and forcing the weakest value to go to VSS and the strongest value to VDD, thus remaining at each other's extreme value and generating a result.

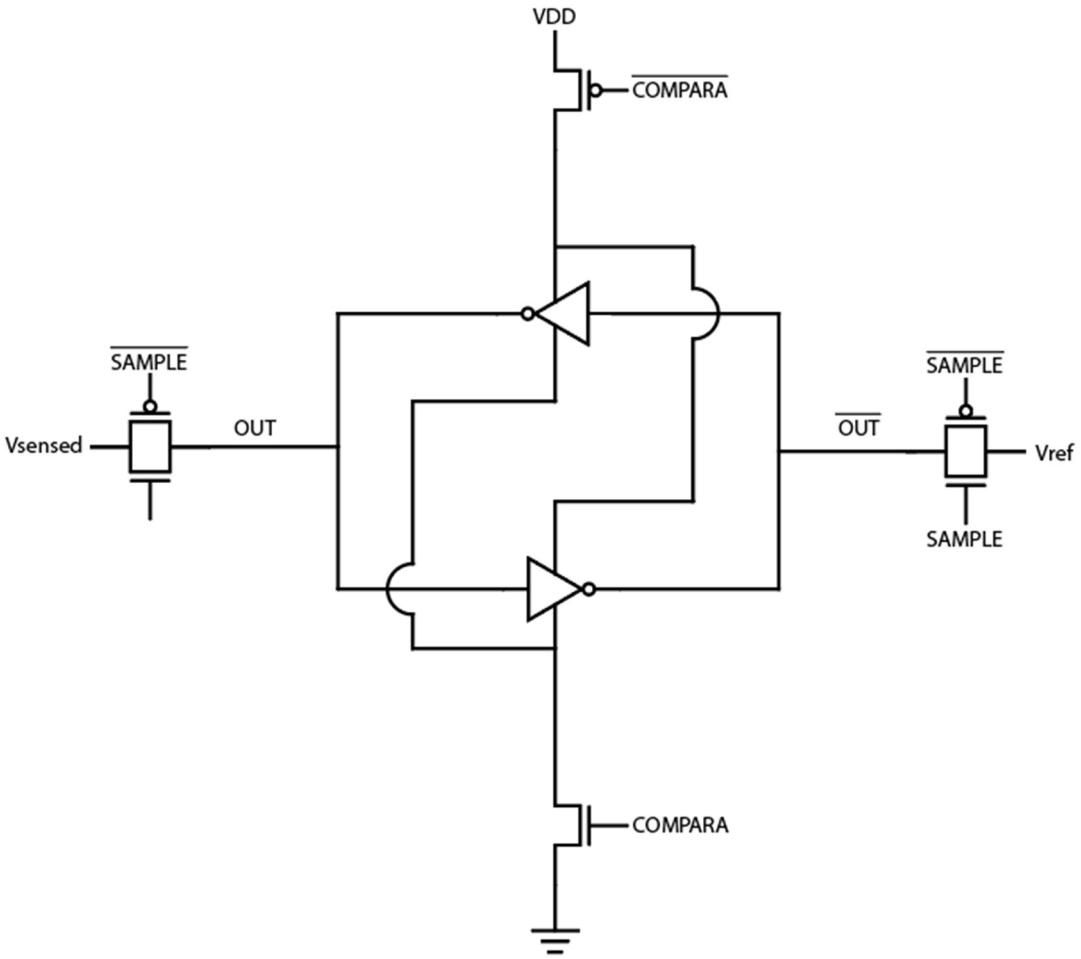


Figure 3.17 - Signal Comparator.

By analyzing the circuit, if in the OUT node, after the *Compare* signal has been active, there is a 0, this means that the transition in the bit line was quick and, therefore, the C1 capacitor ended up charging a low voltage, that after being compared with the reference voltage, made the comparator to output VSS, i.e. at 0 V, thus being signaled as a successful transition. On the contrary, if in the OUT node there is a 1, then it will be identified as a predictive error, or an unsafe bit line transition.

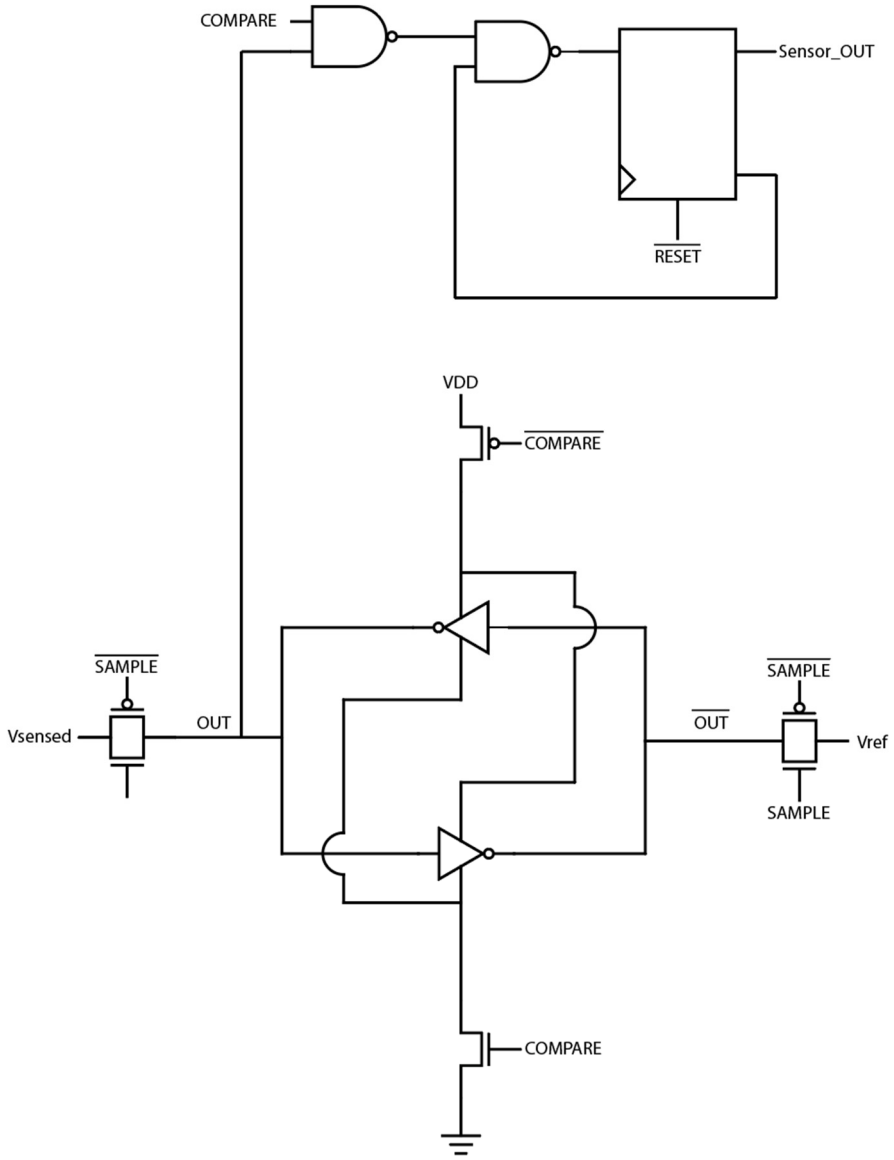


Figure 3.18 - Signal Comparator with flip-flop in the result.

In Figure 3.18 it is possible to see the circuitry added (a flip-flop and two NAND gates) in order to retain an active OUT signal at the sensor output port (Sensor_Out). This is important so that a predictive error detection occurrence can be interpreted in the following clock cycles, and corrective actions can take place. In this implementation, a predictive error detection will remain active at the output until a new sensor monitoring takes place, by resetting the sensor with the Reset signal.

In order to carry out the necessary verification tests for the proper functioning of the Signal Comparator, the HSPICE file was described:

```
**********  Inverters **********
      XM1c auxn Out OutN vss! NMOSFET
      XM2c auxp Out OutN vdd! PMOSFET
      XM3c auxn OutN Out vss! NMOSFET
      XM4c auxp OutN Out vdd! PMOSFET
      XM5c auxn COMPARA vss! vss! NMOSFET
      XM6c auxp COMPARA_N vdd! vdd! PMOSFET
**********  TRANSMISSION GATES **********
Xtgate1 SAMPLE SAMPLE_N N1 OUT Vss Vdd tgate_core
Xtgate2 SAMPLE SAMPLE_N N2 OUTN Vss Vdd tgate_core
********** Saida FLip Flop **********
Xnand_saida1 COMPARA OUT nand_saida1 vss! vdd! NAND20
Xnand_saida2 nand_saida1 Saida_N nand_saida2 vss! vdd! NAND20
XFF_saida nand_saida2 CLK RESET_N Saida Saida_N vss! vdd! DFC1
```

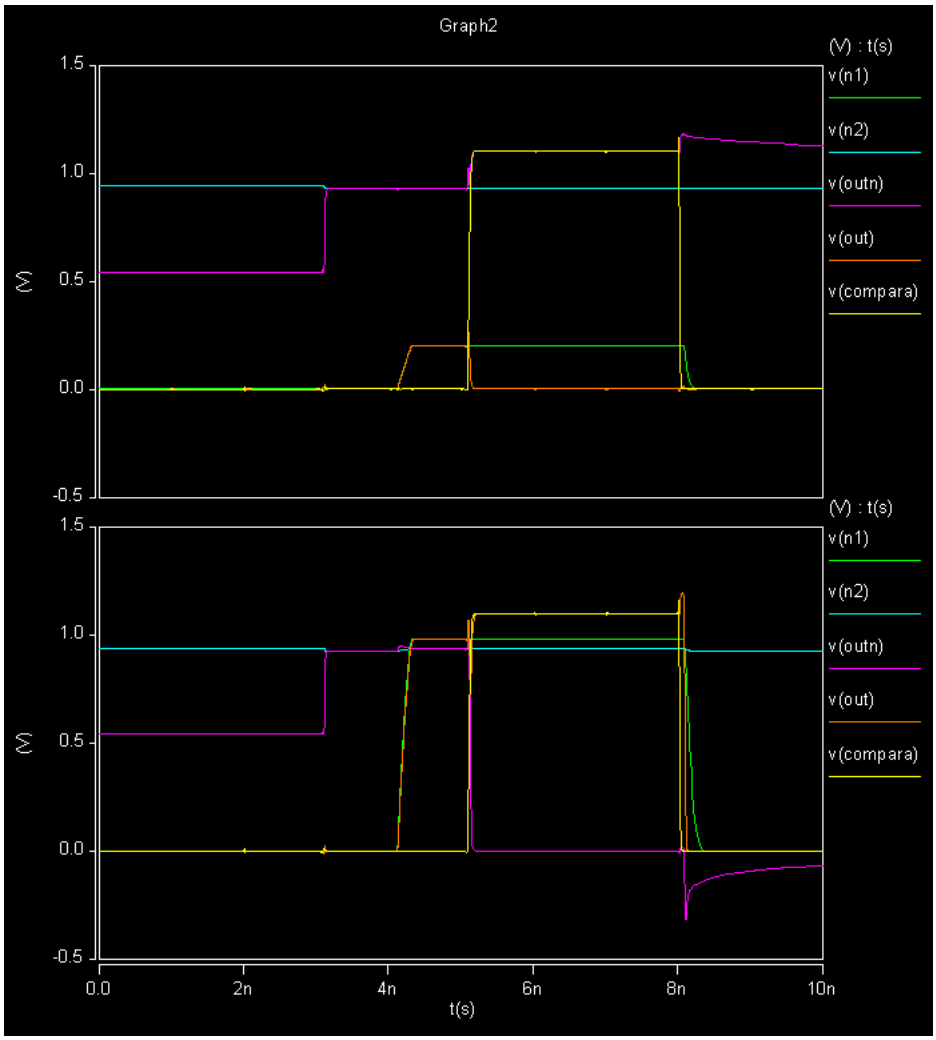


Figure 3.19 - Signal Comparator.

In Figure 3.19 it is possible to see the Signal Comparator carrying out two operations. The first graph shows a transition without error detection, while the second one shows a transition with an error detection.

By observing Figure 3.19, we are able to see in the first graph that the *n1* signal (the node with Vsensed signal) is the voltage stored in the C1 capacitor before the transmission gate, and the n2 signal is the reference voltage (the Vref signal). When the *Compare* signal is activated by the state machine (the controller), the transmission gates already were conducting and the comparator is powered-up by making the inverters to work. In the first diagram it is possible to see that when a *Compare* signal (Compara node on the graphs) is active, the value of *out* is triggered to VSS, while the *outn* (the complementary *out* signal) is triggered to VDD. This happens because the voltage in n1 is lower than in n2 and the inverters will put them at the end.

The same thing happens in the second graph, however in this case the value of n1 is higher thar n2, therefore, the *outn* signal will change to VSS and the out signal will change to VDD, causing the out signal to be interpreted as an error.

3.1.5. CONTROLLER AND SENSOR OPERATION

For the correct operation of the entire sensor, there was the need to implement a sensor controller, i.e., a Finite State Machine (FSM) which receives some input signals and generates the control signals for the remaining blocks, in order to control all sensor operation.

In Figure 3.20 it is possible to see the state diagram implemented in the FSM. It consists of 3 main states, *Reset*, *Sample* and *Compare*, as it will be explained ahead.

The *Reset* state's purpose is to reset the entire sensor operation to an initial state, ensuring that the values in the capacitors and main nodes are discharged and having the necessary values for the correct operation of the sensor, not misleading the measurements that will be made. This state is executed in a single clock cycle.

The *Sample* state purpose is to place the FSM waiting for a transition to happen in the bit line. When this transition occurs, it will generate a pulse and the FSM will change to the *Compare* state. To implement this behavior, an auxiliary variable holds the state machine in the *Sample* state until this pulse is detected (Pulse_detected = 0). When a new pulse is detected, the FSM will move on to the following state (Pulse_detected = 1).

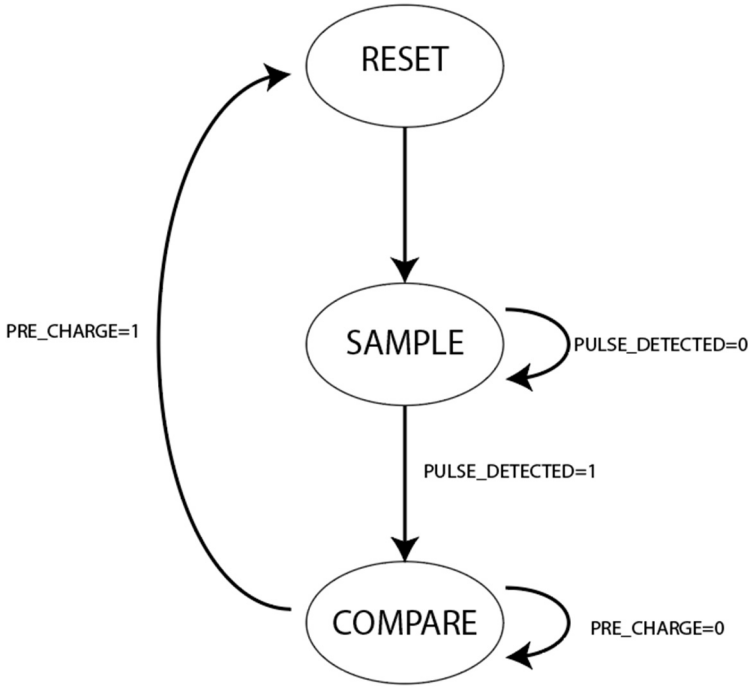


Figure 3.20 - State Machine of the Sense Performance.

The *Compare* state is where the comparator is activated, in order to compare the value stored in the C1 capacitor and the reference value, to make the decision whether the measured pulse is an error or a successful reading. This state activates the Compare signal, which enables the comparator to perform the comparison between Vsensed and Vref signals by powering the two cross-coupled inverters. The FSM will remain in this state, until a new Read/Write of the memory occurs. Therefore, prior to a Read/Write operation, there is a pre-charge signal to trigger the initialization of the bit lines, and this pre-charge signal indicates that a new Read/Write operation will take place. So, this signal is also used in the *Compare* state, to trigger a state change to the *Reset* state, to prepare the sensor for a new monitoring procedure. While there is no pre-charge signal activation, and consequently no new Read/Write operation, the FSM remains in the *Compare* state.

Figure 3.21 illustrates the implementation of the sensor's FSM using two D flip-flops in its structure. Figure 3.22 allows to see the signals coming from the flip-flops of the FSM, which indicate the state of the FSM, and the logic gates that generate the sensor's control signals from the state of the FSM.

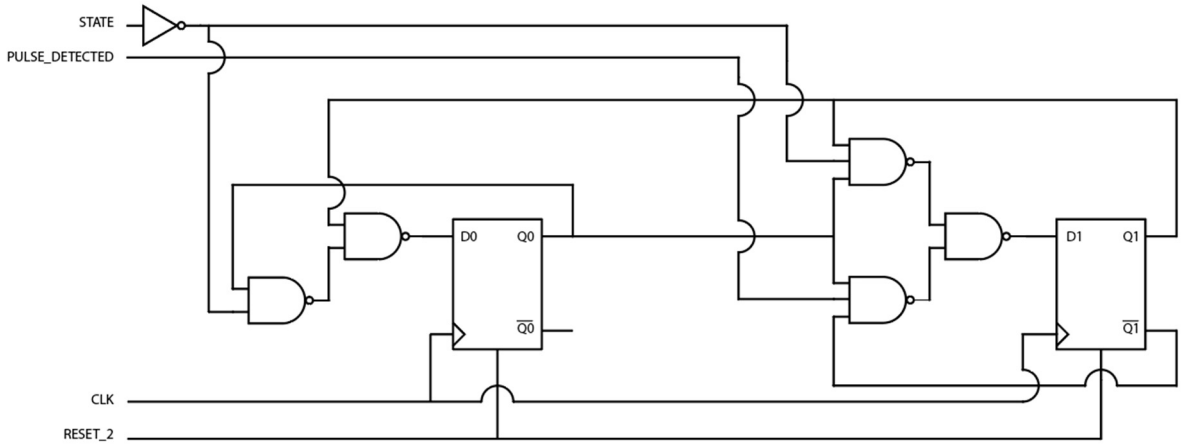


Figure 3.21 – Controller implementation.

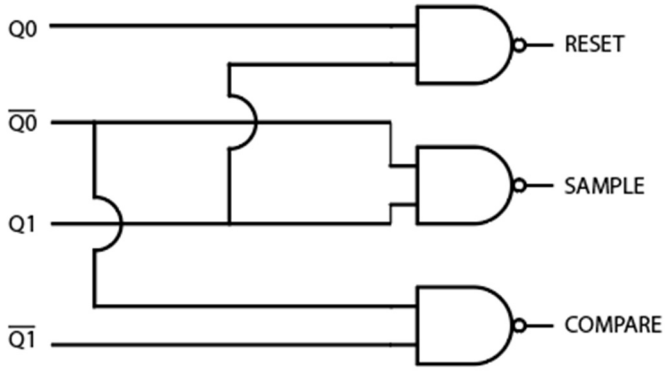


Figure 3.22 - Control Signals of the sensor.

Moreover, there are also one Latch to assist the operation of the FSM and help to generate the necessary control signals, which in this case are the auxiliary signals Pulse_detected and State. In Latch operation, whenever the clock is at 1, the latch is in transparent mode and the output will have what is on the input. When the clock is 0, the latch is in hold mode and the output will remain unchanged (even if the input is changed), i.e., with the last value that was updated in the last transparent mode.

In Figure 3.23 it is possible to see the Latch which controls the Pulse_detected signal of the FSM. We can see that the Latch's active-low reset is the complementary Reset of the state machine. This means that when there is a reset in the state machine, the latch will be reset and the output will be 0. Moreover, at the input there is the Sample signal, and every time a pulse is generated in the Transition Detector, it will work as the clock of the latch and causes the value of the sample signal to be placed at the output, making the state machine advance to the next state, the *Compare* state.

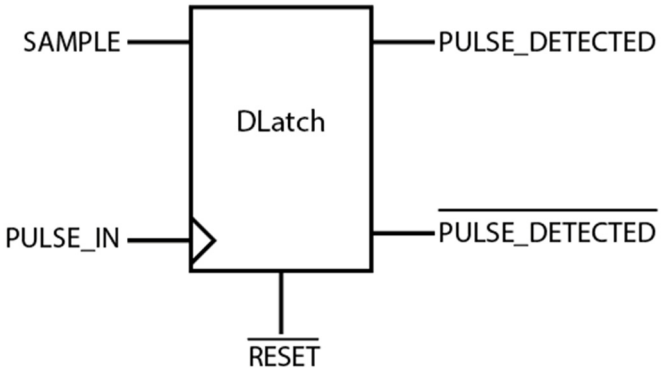


Figure 3.23 - Pulse Detection Latch.

In Figure 3.24 we can see all the signals from the Performance Sensor, and also the signals from the memory, so that the sensor operation can be analyzed with the memory Read/Write operation. It is also possible to analyze the operation of the controller FSM.

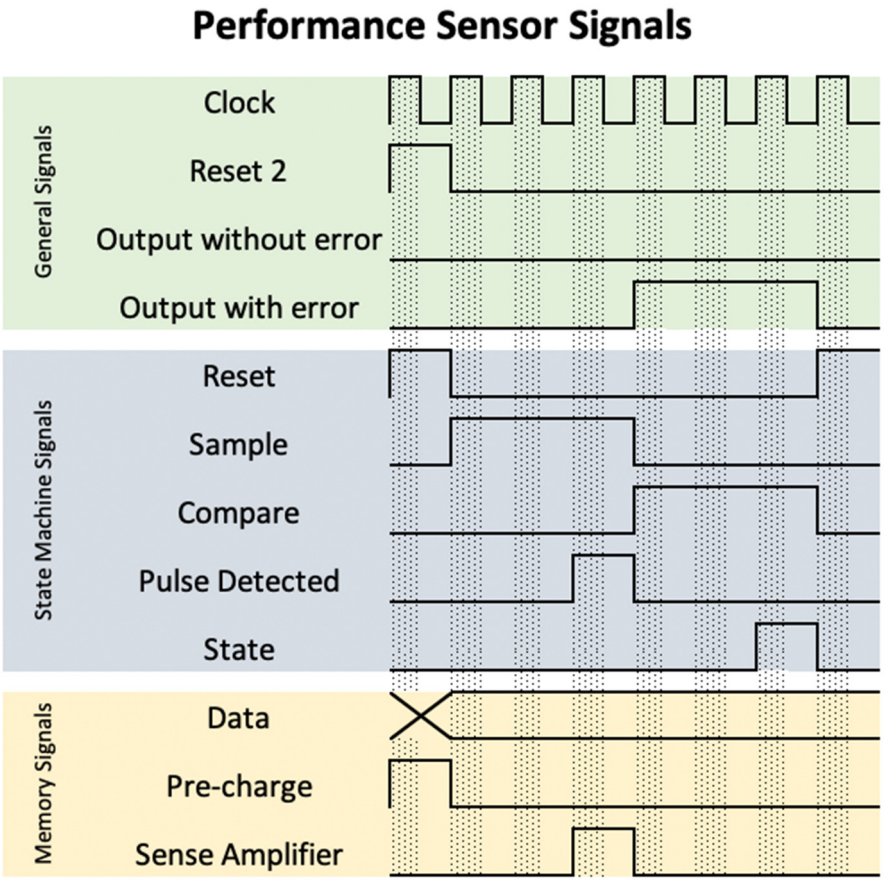


Figure 3.24 - All Signals of the Performance Sensor

3.1.6. COMPLETE CIRCUIT

In Figure 3.25 and in Figure 3.26 it is possible to see the complete circuit for the performance sensor for an SRAM. The first one is for a VDD initialization type, while the second one is for the VDD/2 initialization type.

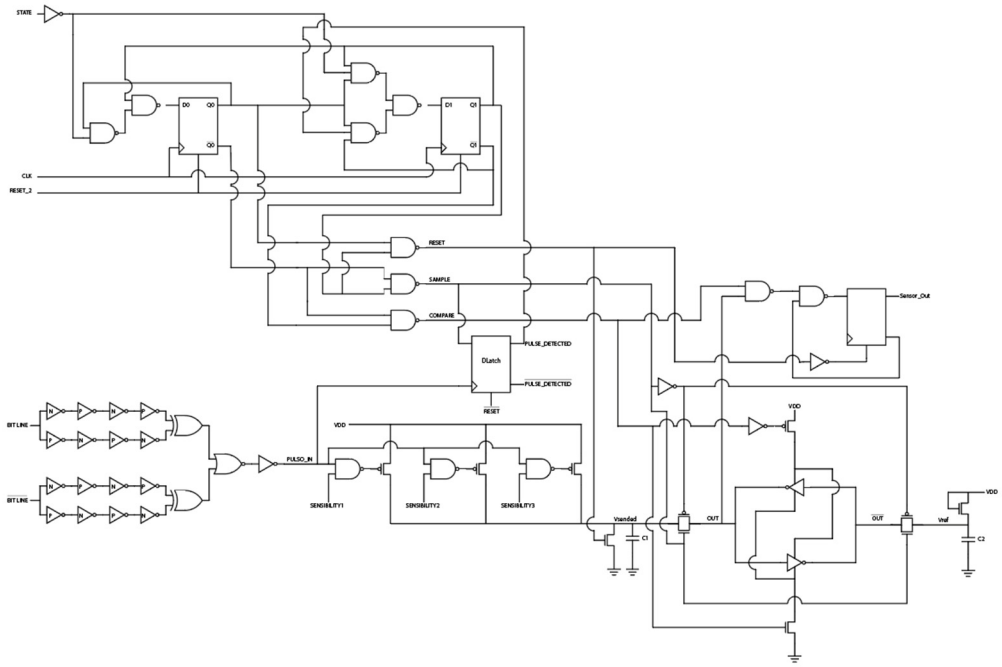


Figure 3.25 - Complete Circuit for VDD initialization in a SRAM.

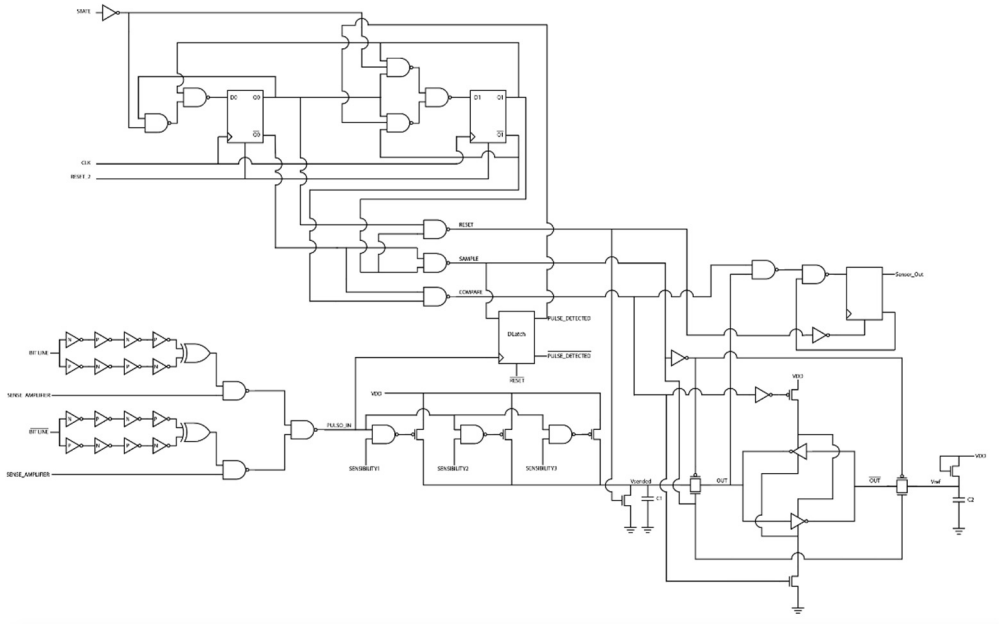


Figure 3.26 - Complete Circuit for VDD/2 initialization in a SRAM.

In Figure 3.27 we are able to depict the complete circuit for the performance sensor for a DRAM initialized at VDD/2.

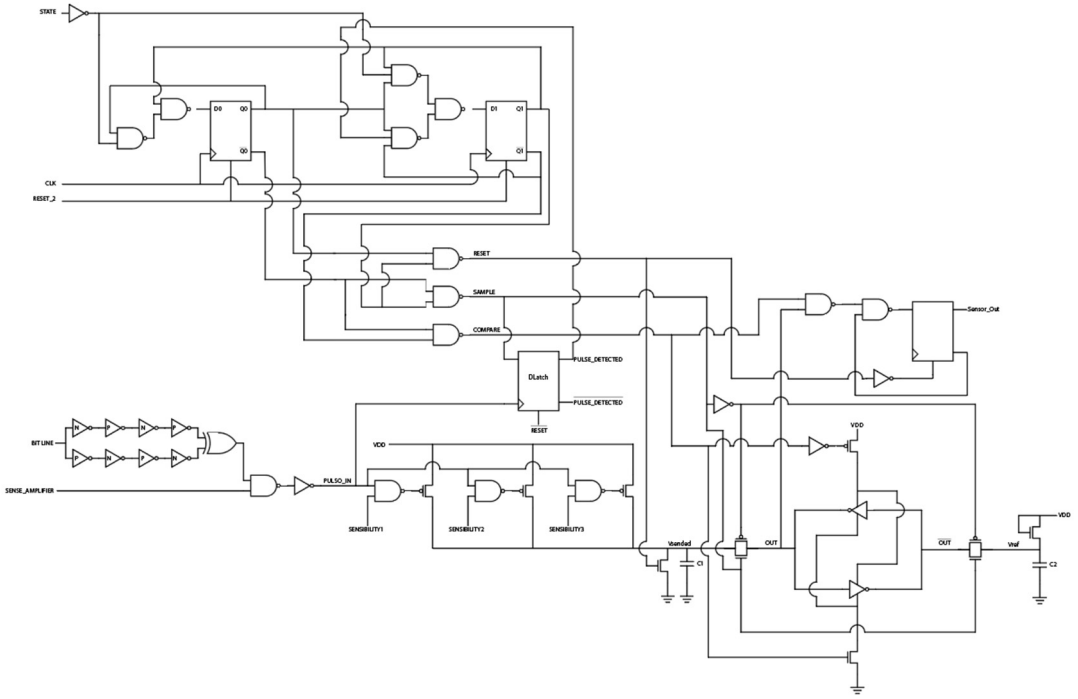


Figure 3.27 - Complete Circuit for VDD/2 initialization in a DRAM.

3.2. IMPLEMENTATION LAYOUTS

The purpose of this chapter is to present layout implementations for the previously described and tested circuits. Analyzing the layouts, it will be possible to understand the dimensions of the circuit and the different blocks, comparing their surface area.

The layouts here presented were designed in the Microwind software, using the 65nm CMOS technology.

3.2.1. TRANSITION DETECTOR LAYOUT

In this section the layout for the Transition Detector is presented. The Transition Detector has the specification of not having all the transistors with the same size, as it was

previously explained. This way, there are transistors N and transistors P with five times the minimum size, thus allowing high conductive passing gates (for specific conditions). Therefore, the PMOS type full-size transistors have W_{Pmin}=360nm, while the NMOS type transistors have W_{Nmin}=180nm, with the channel of the two transistors being 65nm.

Transition Detector Layout for VDD initialization

As mentioned before, two models of Transition Detector were tested. In this section, the model the for the SRAM, initialized at VDD, is presented. Figure 3.28 depicts its implementation.

The Transition Detector has:

- Width: 23,7 µm (678 lambda)
- Height: 3,2 µm (91 lambda)
- Surface area: 75,6 µm2

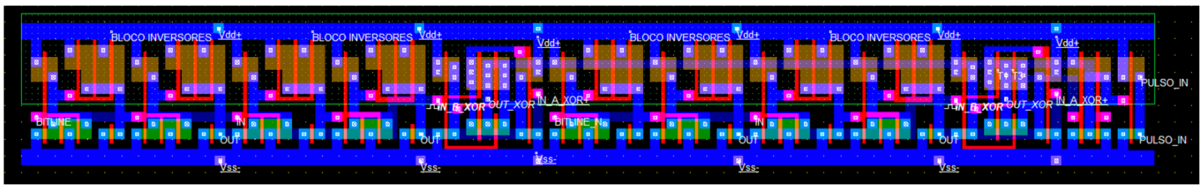


Figure 3.28 - Transition Detector Layout – VDD Initialization.

Transition Detector Layout for VDD/2 initialization

Considering the Transition Detector initialized at VDD/2, it is a simpler implementation, when compared with the previous Transistor Detector. In Figure 3.29 its layout implementation is presented.

The Transition Detector has:

- Width: 13,4 µm (382 lambda)
- Height: 3,2 µm (91 lambda)
- Surface area: 42,6 µm2

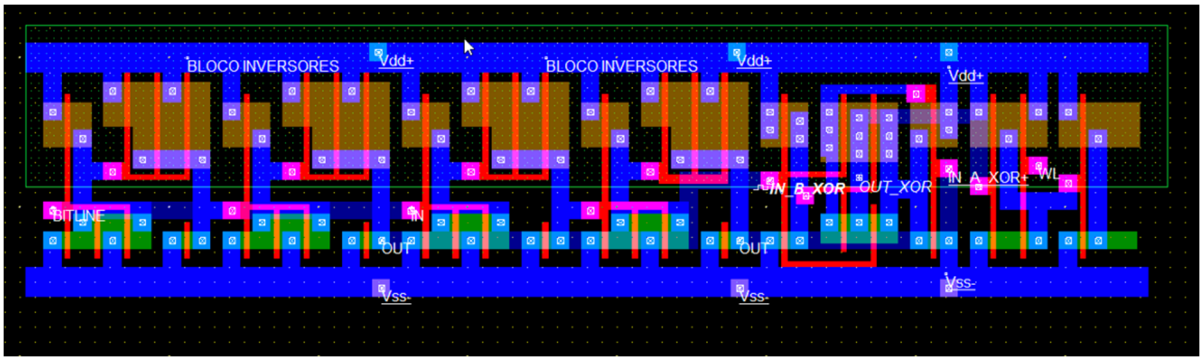


Figure 3.29 - Transition Detector - VDD/2 Initialization.

3.2.2. PULSE DETECTOR LAYOUT

In this section, the layout for the Pulse Detector is presented, depicted in Figure 3.30. The Pulse Detector has the specification of having 3 control signals (Sensibility signals), which may be used to calibrate the sensor, has shown in section 3.1.2. This way, there are PMOS transistors with W=360nm, NMOS transistors W=180nm and the and the channel of the two examples is 65nm. Also, in this part of the circuit there is a capacitor which was previously explained, the C1 capacitor.

The Pulse Detector has:

- Width: 6,0 µm (382 lambda)
- Height: 3,2 µm (91 lambda)
- Surface area: 19,2 µm2

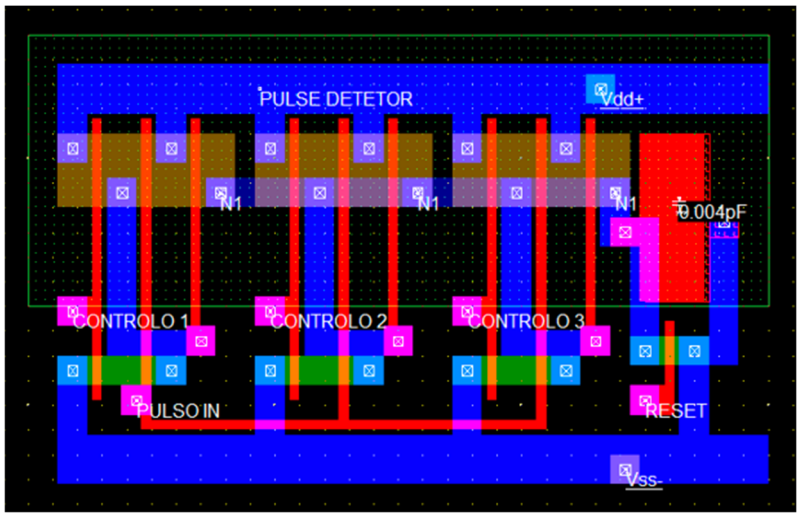


Figure 3.30 - Pulse Detetor Layout.

3.2.3. COMPARATOR LAYOUT

The Comparator is one of the most important parts of the Performance Sensor presented. In Figure 3.31 it is possible to see a layout implementation solution. This way, there are PMOS transistors with W=360nm, NMOS transistors W=180nm and the and the channel of the two examples is 65nm. Also in this part of the circuit there is a capacitor which was previously explained, the C1 capacitor.

The Comparator has:

- Width: 10,7 µm (382 lambda)
- Height: 3,2 µm (91 lambda)
- Surface area: 34,1 µm2

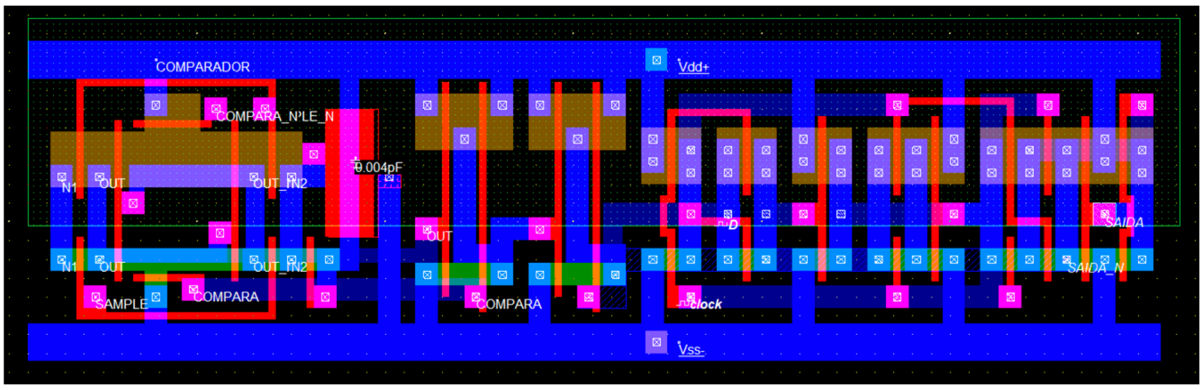


Figure 3.31 - Comparator Layout.

4. SIMULATION RESULTS

Given the specification of the sensor presented, there is a need to perform a set of tests to demonstrate the features, its reliability and its usefulness.

In the following sections, the performance sensor will be displayed in its several applications and working with different sensitivity levels.

4.1. SRAM PERFORMANCE SENSOR

SRAM memories have the particularity of being able to work with an initialization to VDD or to VDD/2. The different types of initialization and memory only differ in the Transition Detector block, while the remaining parts of the performance sensor are the same for the two types of SRAM memories here described.

4.1.1. PERFORMANCE SENSOR FOR VDD INITIALIZED SRAM

As shown in Figure 3.3, the Transition Detector for an SRAM with initialization to VDD consists of two set of double-path inverters, one set for the bit line and another one for the complementary bit line. The inverters are chosen so that *p*-type inverters and *n*-type inverters are placed alternately along the paths, to create two paths with different (opposing) delay characteristics, one being faster for a low-to-high transition, and the other to be faster for a high-to-low transistor. The delays introduced in these 2 paths are responsible for the pulse generation, that will allow the sensor to decide if the bit line transition (which its switching time is reflected in the generated pulse width) should be considered as an error or as a success.

The simulations were performed with a bit line parasitic capacity of 10ff, a sensor's C1 capacity of 40ff, temperature of 27º and a clock period of 2n, with the nominal VDD at 1.1V.

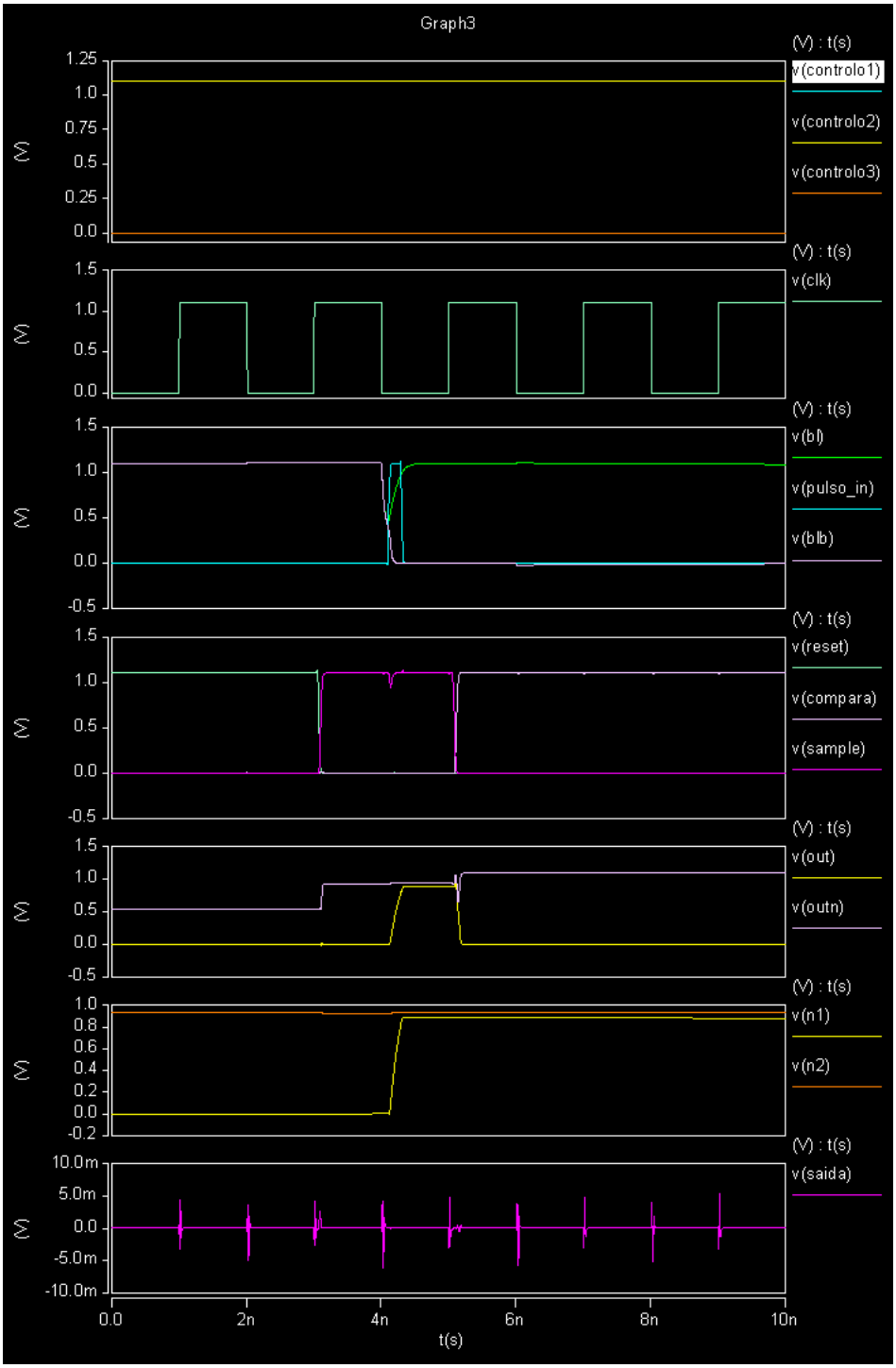


Figure 4.1 - Performance Sensor Simulation for SRAM initialization to VDD – Success.

In Figure 4.1 it is possible to observe a simulation of a successful transition. In this case we see the bit line initialized to VDD. The value to be stored in the memory is the logical value of 1, therefore the bit line remains at VDD while the complementary bit line switches to 0, causing a pulse to be created in the Pulse Detector, as it can be seen in the signal Pulse_in.

In this case, control 1 and control 2 are active (these are the node names for Sensibility1 and Sensibility2 signals), so that in the Pulse Detector the two smaller transistors are active.

The n1 node (representing the Vsensed signal) has the value coming from the Pulse Detector, and the n2 node (representing the Vref signal) has the reference value, as mentioned before. Looking now at the FSM's signals, when entering the Sample state, the value that was in *n1* passes to the out node, and the value that was in *n2* passes to the complementary out node. When the Compare signal (node compara in simulation) is activated, it causes the comparator to amplify the values in the out and complementary out (*outn* node in simulation) nodes to VSS and VDD limits. In this case, has the value in *n1* is lower than the value in *n2*, *out* is triggered to VSS and *outn* is triggered to VDD, making this a successful transition.

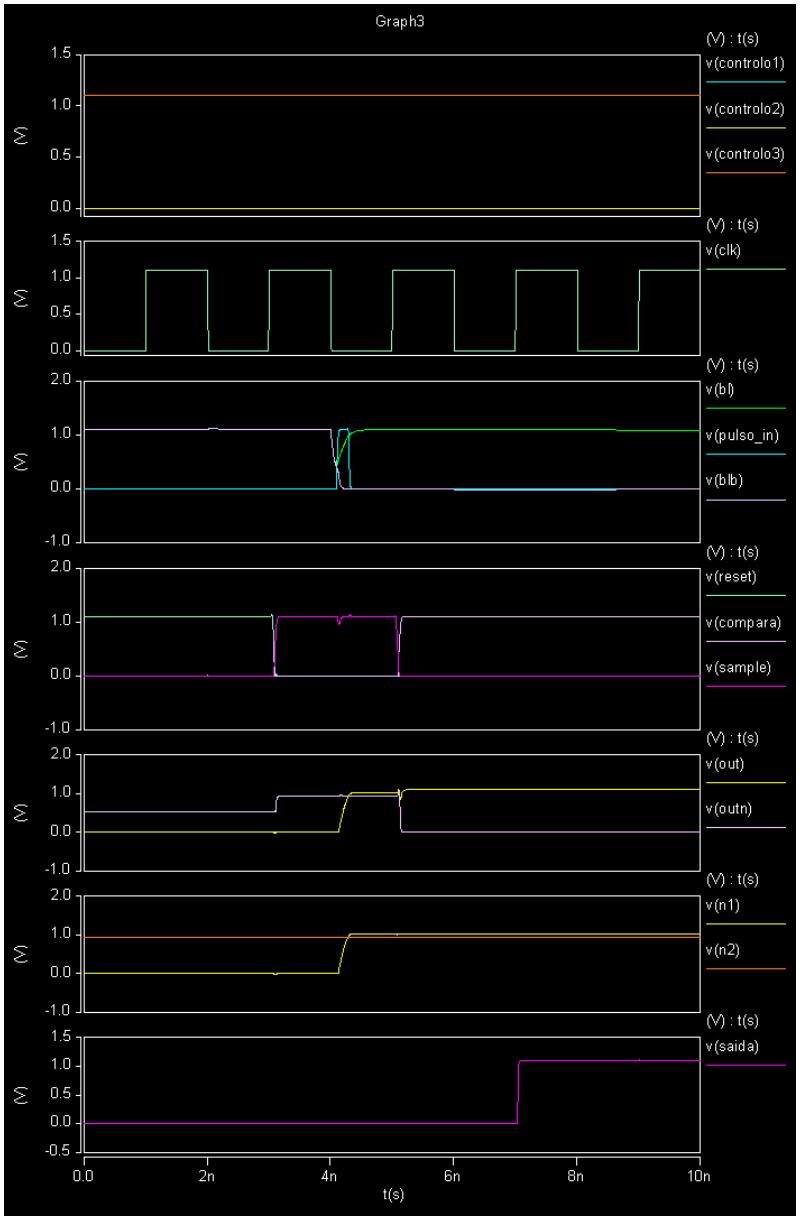


Figure 4.2 - Performance Sensor Simulation for SRAM initialization to VDD – Error.

Figure 4.2 illustrates a similar simulation as the one in Figure 4.1, but changing the sensibility of the Pulse Detector: control 1 and control 2 have been turned off, remaining only the control 3 (Sensibility3 signal) on. This small change allows charging the capacitor C1 with a higher current, causing the value in *n1* (Vsensed signal) to be above the value in n2 (Vref signal), therefore predicting an unsafe transition and outputting an error.

If in turn we activate control 1 and 2 simultaneously with control 3, the difference between the signal generated by the Pulse Detector in *n1* becomes even greater than the control signal in *n2*, as would be expected, and as can be seen in Figure 4.3.

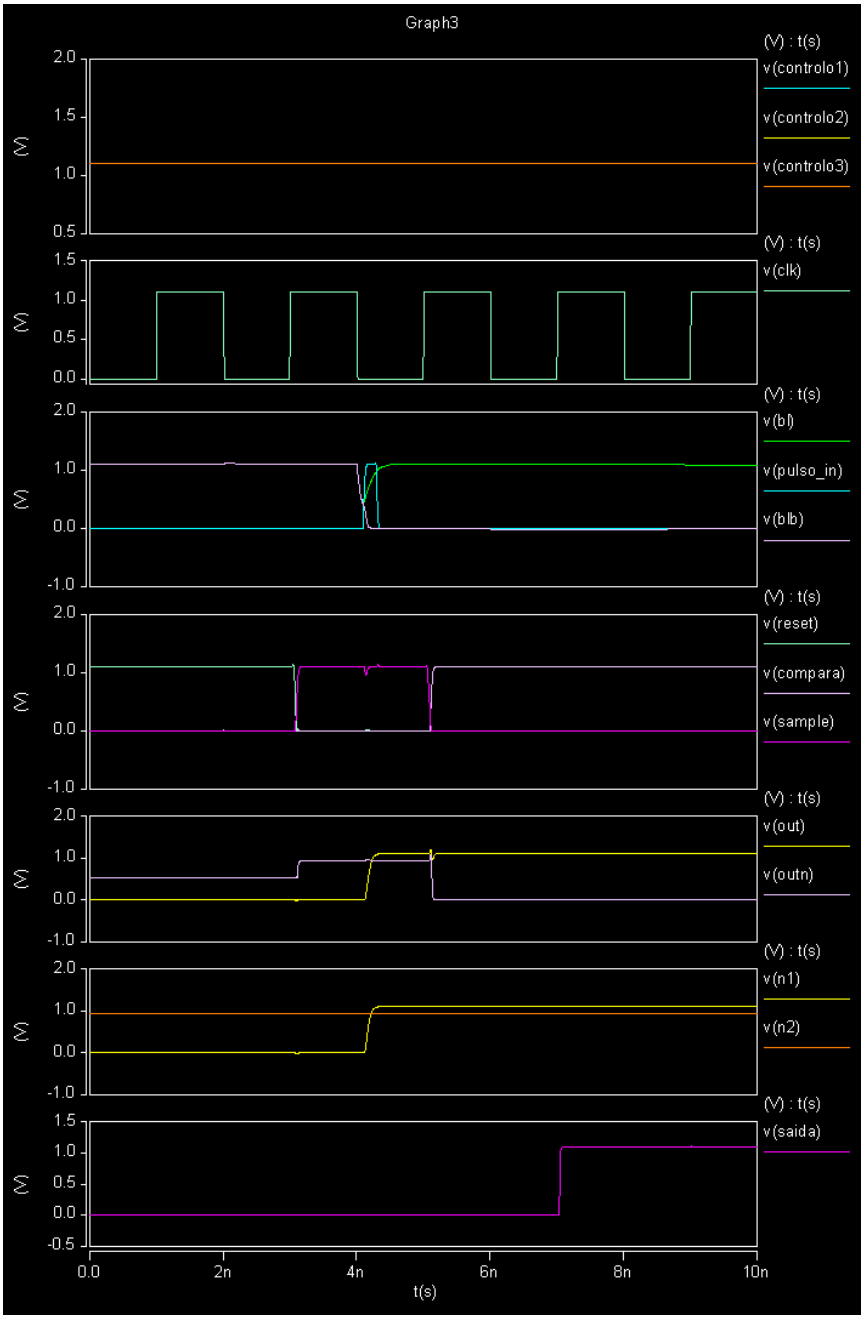


Figure 4.3 - Performance Sensor Simulation for SRAM initialization to VDD – Error with 3 controls.

Lastly, taking all the conditions simulated in Figure 4.1, which include a low sensibility of the sensor, the power-supply voltage was now changed to 0.8V and the transistors were aged by increasing their |Vth| in 10%. So, what was initially considered as a successful bit line transition, now is considered as an unsafe transition, since the bit line transition became slower due to the VDD reduction and aging (as it can be seen in Figure 4.4).

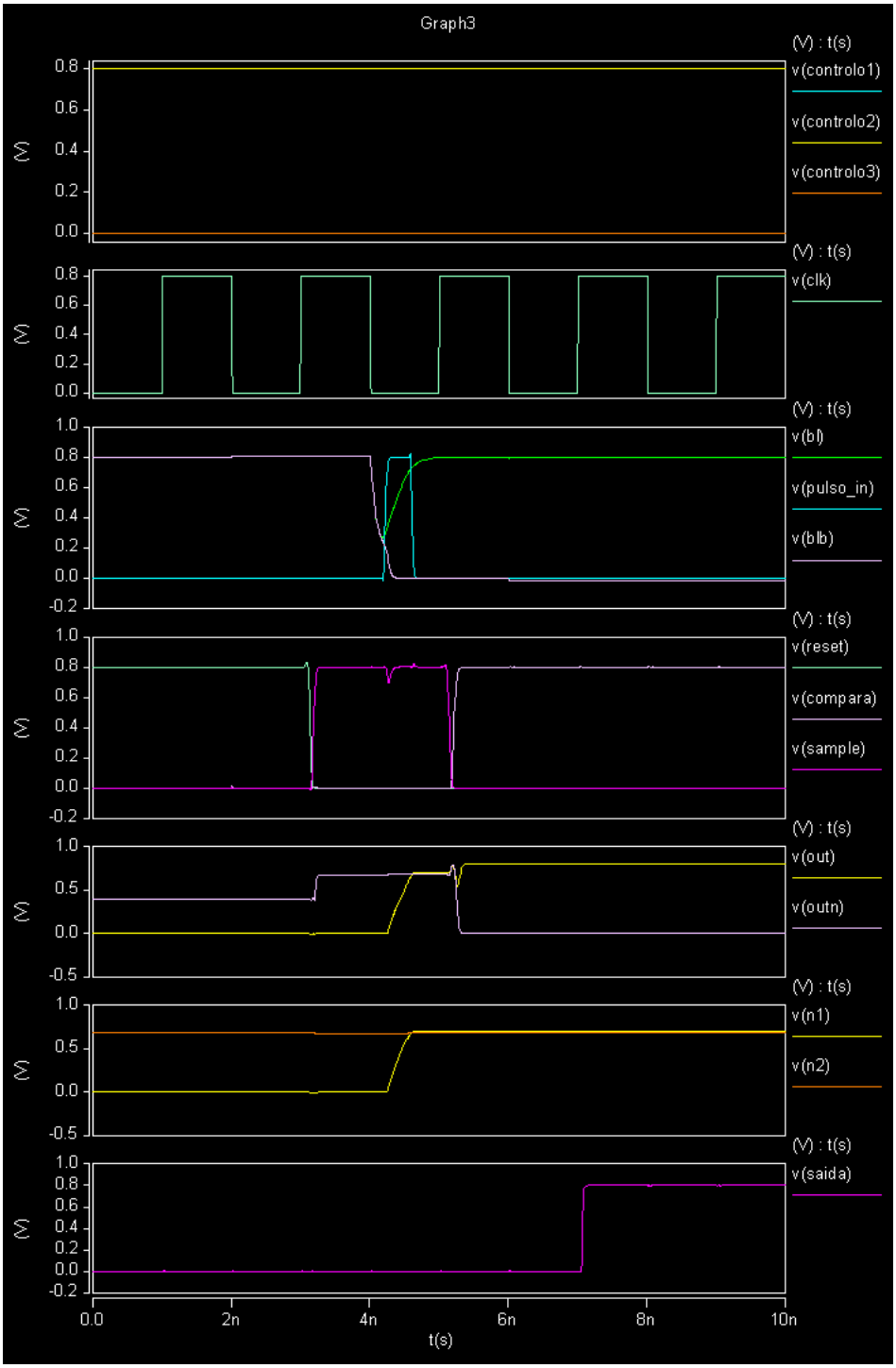


Figure 4.4 - Performance Sensor Simulation for SRAM initialization to VDD – Error with aging.

4.1.2. PERFORMANCE SENSOR FOR VDD/2 INITIALIZED SRAM

As mentioned in section 3.1.1, the Transition Detector for an SRAM initialized in VDD/2 (Figure 3.9) consists of two sets of inverters, one for the bit line and the other for the complementary bit line. As in the sensor for SRAM initialized to VDD, each set of inverters has 2 paths, with *n*-type and *p*-type inverters, placed alternatively in the paths, each path starting with one of the two inverter types. Both paths end-up in an XOR gate, and the propagation delays in both paths allows to create a pulse with a width proportional to the bit line transition. Note that each XOR gate is connected to a NAND gate, along with the Sense Amplifier signal of the SRAM memory, so that the transitions for the pre-charge of the bit lines to VDD/2 does not generate additional pulses.

The simulations were performed with a bit line parasitic capacity of 10ff, a sensor's C1 capacity of 40ff, temperature of 27º and a clock period of 2n, with the nominal VDD at 1.1V.

In Figure 4.5 it is possible to see the simulation of a successful transition in an SRAM memory initialized to VDD/2. In this case we see the bit line and the complementary bit line initialized to VDD/2, while the value to be stored in the memory is the logical value of 1. Therefore, when the Sense Amplifier signal (that activates the memory sense amplifier) is activated, the bit line and complementary bit line (nodes *bl* and *bln* in simulation) change from VDD/2 to, respectively, VDD and VSS voltages. Then, a pulse is created in the Pulse Detector (*pulso_in* node in simulation), and the voltages in *n1* (Vsensed) and *out* nodes are updated with a new value measured (a voltage proportional to the bit line transition time). In this case, as only the *control 1* node is active (Sensibility1 signal), which corresponds to a low sensibility on the bit line transition making Vsensed signal (*n1*) lower than Vref signal (*n2*), when the Compare (*compara* node in simulation) signal is activated, it causes the comparator to trigger the values in the out and complementary out nodes to the minimum and maximum values, respectively, i.e., VSS and VDD. So, this was a safe bit line transition.

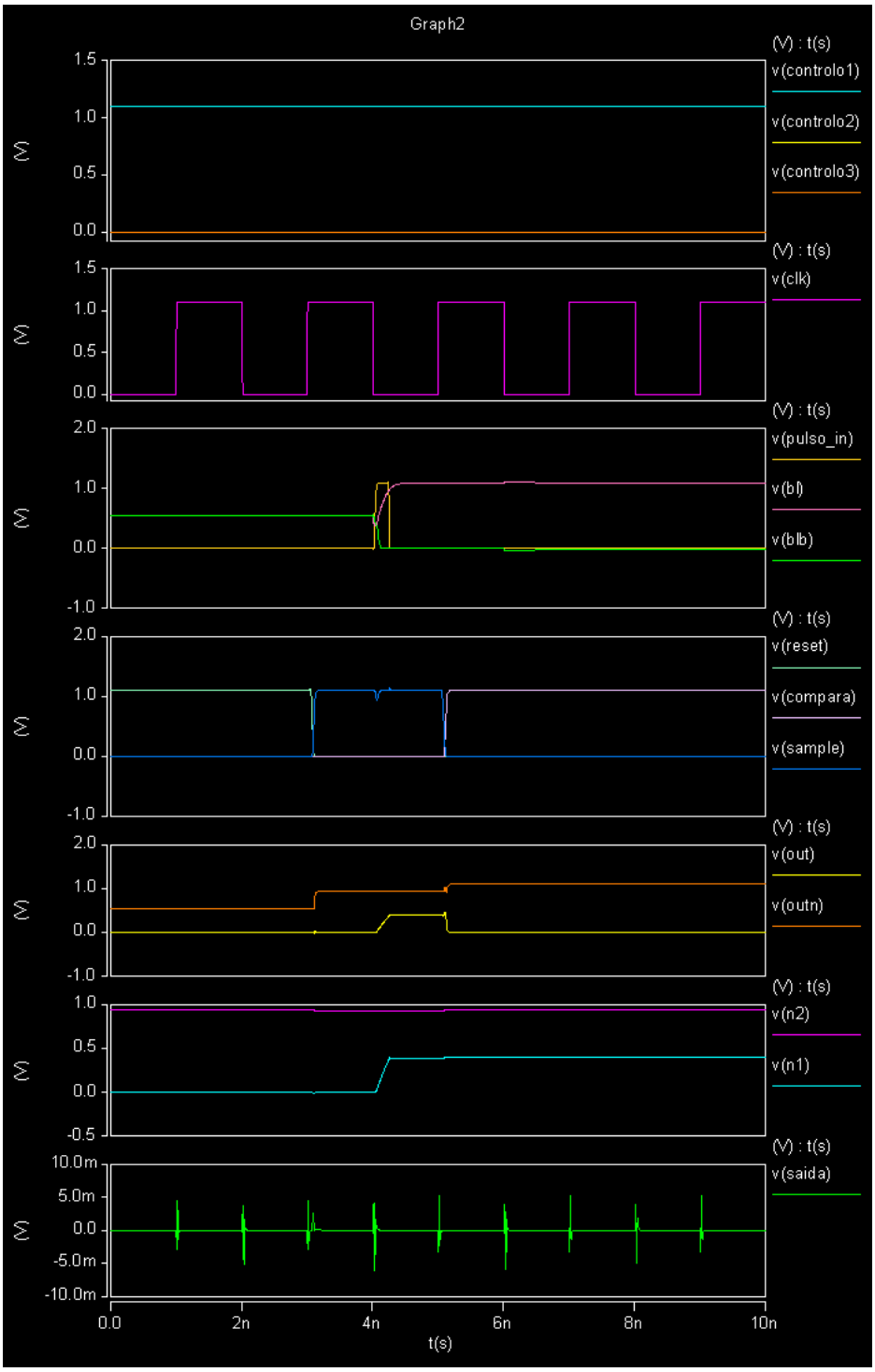


Figure 4.5 - Performance Sensor Simulation for SRAM initialization to VDD/2 – No error detected.

A similar simulation was again performed, but this time changing the sensibility of the sensor but activating not only control 1 node but also control 2 node (Sensibility2 signal). This sensibility change makes the same transition no longer to be considered as an unsafe transition, but rather an unsafe transition, generating an error (as it may be seen in Figure 4.6).

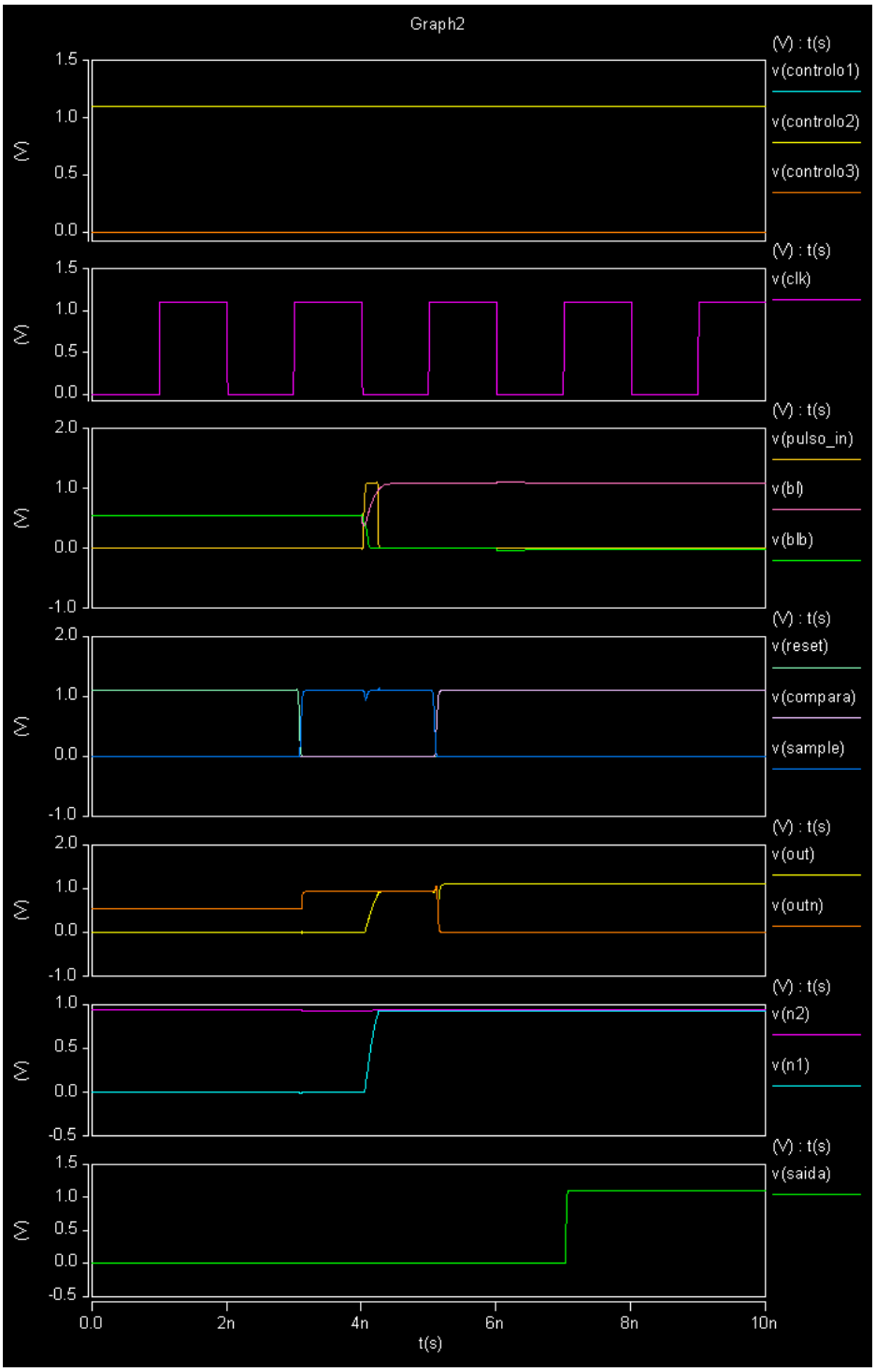


Figure 4.6 - Performance Sensor Simulation for SRAM initialization to VDD/2 – Error detected.

For a better understanding of the difference in the sensibility controls, Figure 4.7 illustrates a bigger difference in the voltage in the *n1* and *n2* nodes (respectively Vsensed and Vref signals), when the 3 Sensibility controls are active.

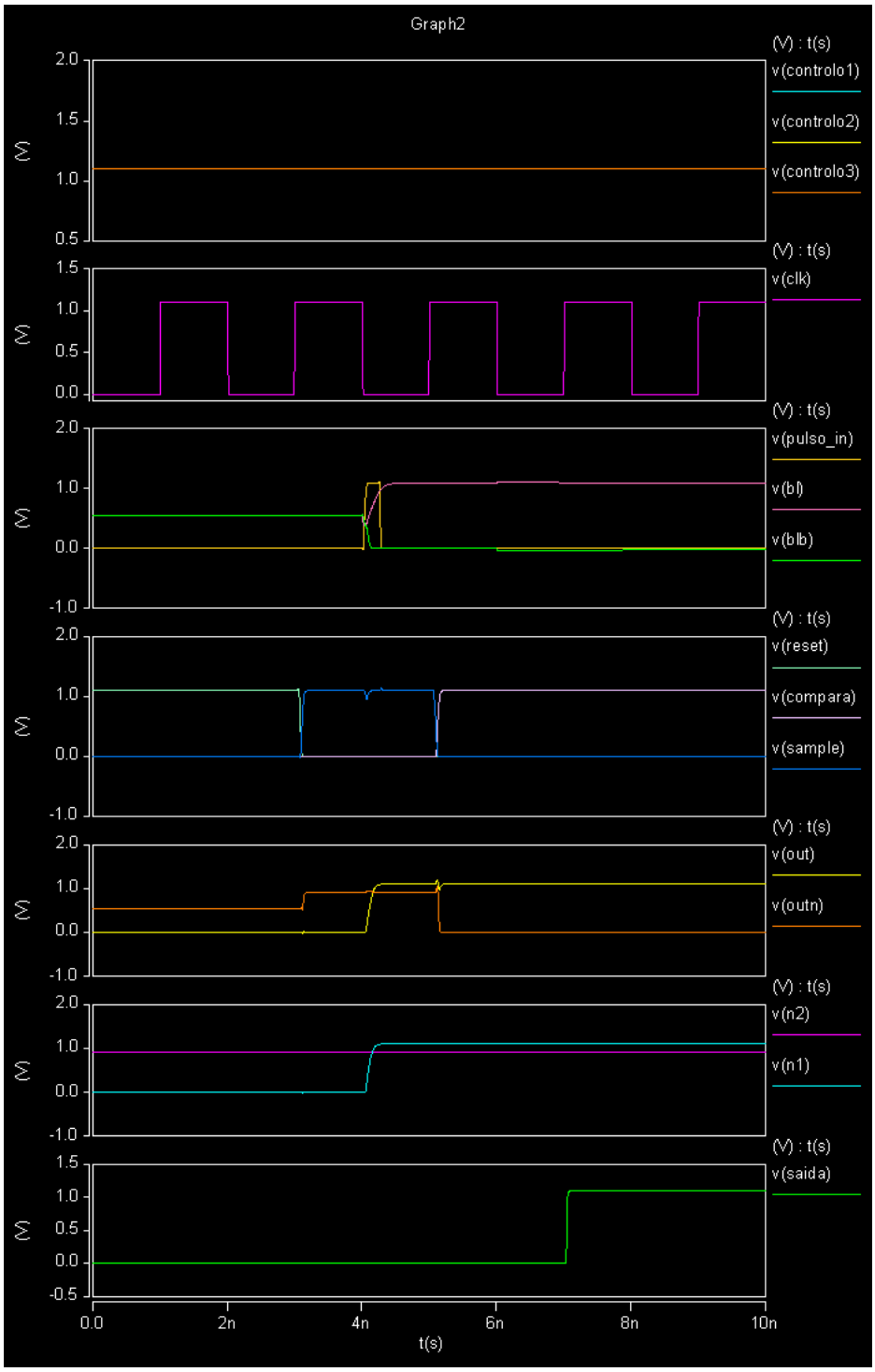


Figure 4.7 - Performance Sensor Simulation for SRAM initialization to VDD/2 –Error detected with the 3 controls

In Figure 4.8, and repeating the same simulating conditions reported in section 4.1.1, the circuit was simulated with aging (10% increase in |Vth|) and using a VDD = 0.8V, and also using a low sensibility (only controlo1 was activated). However, this was not enough to detect an error. If with these values one wishes to detect an error, the sensibility controls should be readjusted. Anyway, this result shows us that in this application we have a lower sensibility

when comparing with the SRAM sensor initialized to VDD. This is because in this case, due to the bit line initialization to VDD/2, the transition time of the bit line is smaller, making the sensor less sensitive, when comparing the SRAM initialized to VDD, for the same implementation and simulating conditions.

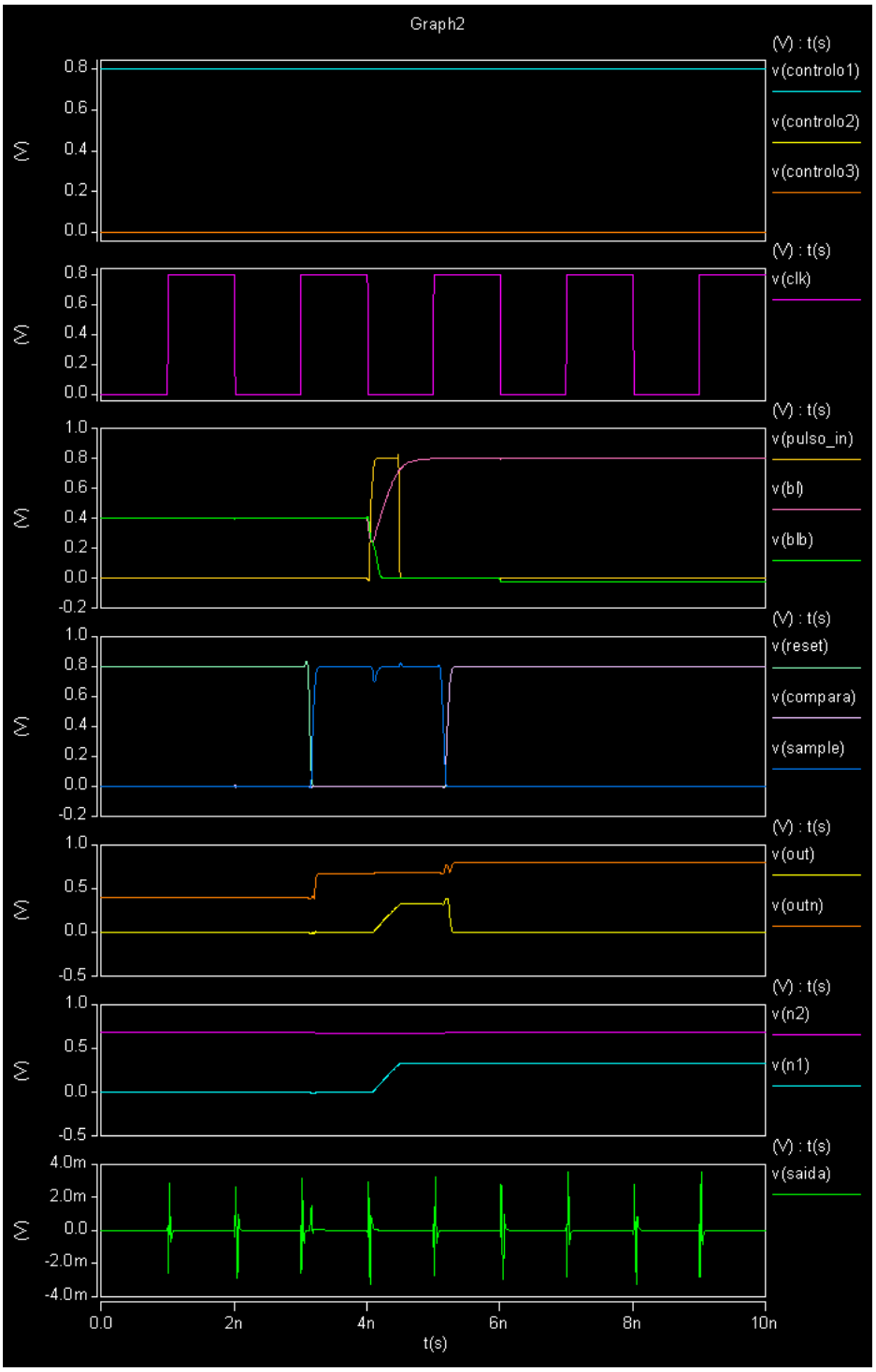


Figure 4.8 - Performance Sensor Simulation for SRAM initialization to VDD/2 – No error detected with aging.

4.2. PERFORMANCE SENSOR FOR DRAM

DRM memories work with an initialization at VDD/2, and the memory cell is connected to only one bit line. Therefore, each sensor should work connected to an individual bit line.

As shown in Figure 3.8, the Transition Detector for a DRAM with initialization to VDD/2 consists of one set of double-path inverters, connected to an XOR gate. The inverters are chosen so that *p*-type inverters and *n*-type inverters are placed alternately along the paths, to create two paths with different (opposing) delay characteristics, one being faster for a low-to-high transition, and the other to be faster for a high-to-low transistor. Like in the SRAM with VDD/2 initialization, the XOR gate is connected to a NAND gate, along with the Sense Amplifier signal of the DRAM memory, so that the transitions for the pre-charge of the bit lines to VDD/2 does not generate additional pulses. The delays created in the inverters' paths generate a pulse in the XOR gate, which will be used in sensor to decide whether the bit line transition should be considered as an error or as a success.

The simulations will be performed based on a bit line parasitic capacity of 10ff, the C1 capacitor's capacity of 50ff, temperature of 27º and a period of 2n with VDD at 1.1V.

In Figure 4.9 it is possible to see the simulation of a successful transition in a DRAM memory initialized to VDD/2. In this case we see the bit line initialized to VDD/2, while the value to be stored in the memory is the logical value of 1. Therefore, when the Sense Amplifier signal (that activates the memory sense amplifier) is activated, the bit line (node *bl* in simulation) change from VDD/2 to VDD. Then, a pulse is created in the Pulse Detector (*pulso_in* node in simulation), and the voltages in *n1* (Vsensed) and *out* nodes are updated with a new value measured (a voltage proportional to the bit line transition time). In this case, Sensibility1 and Sensibility2 signals (nodes *controlo1* and *controlo2* in simulation) are active, while Sensibility3 signal (node *controlo3*) is deactivated, so that in the Pulse Detector the two smaller transistors are conducting and charging C1.

The n1 node (representing the Vsensed signal) has the value coming from the Pulse Detector, and the n2 node (representing the Vref signal) has the reference value, as mentioned before. Looking now at the FSM's signals, when entering the Sample state, the value that was in *n1* passes to the out node, and the value that was in *n2* passes to the complementary out node. When the Compare signal (node compara in simulation) is activated, it causes the

comparator to amplify the values in the out and complementary out (*outn* node in simulation) nodes to VSS and VDD limits. In this case, has the value in *n1* is lower than the value in *n2*, *out* is triggered to VSS and *outn* is triggered to VDD, making this a successful transition.

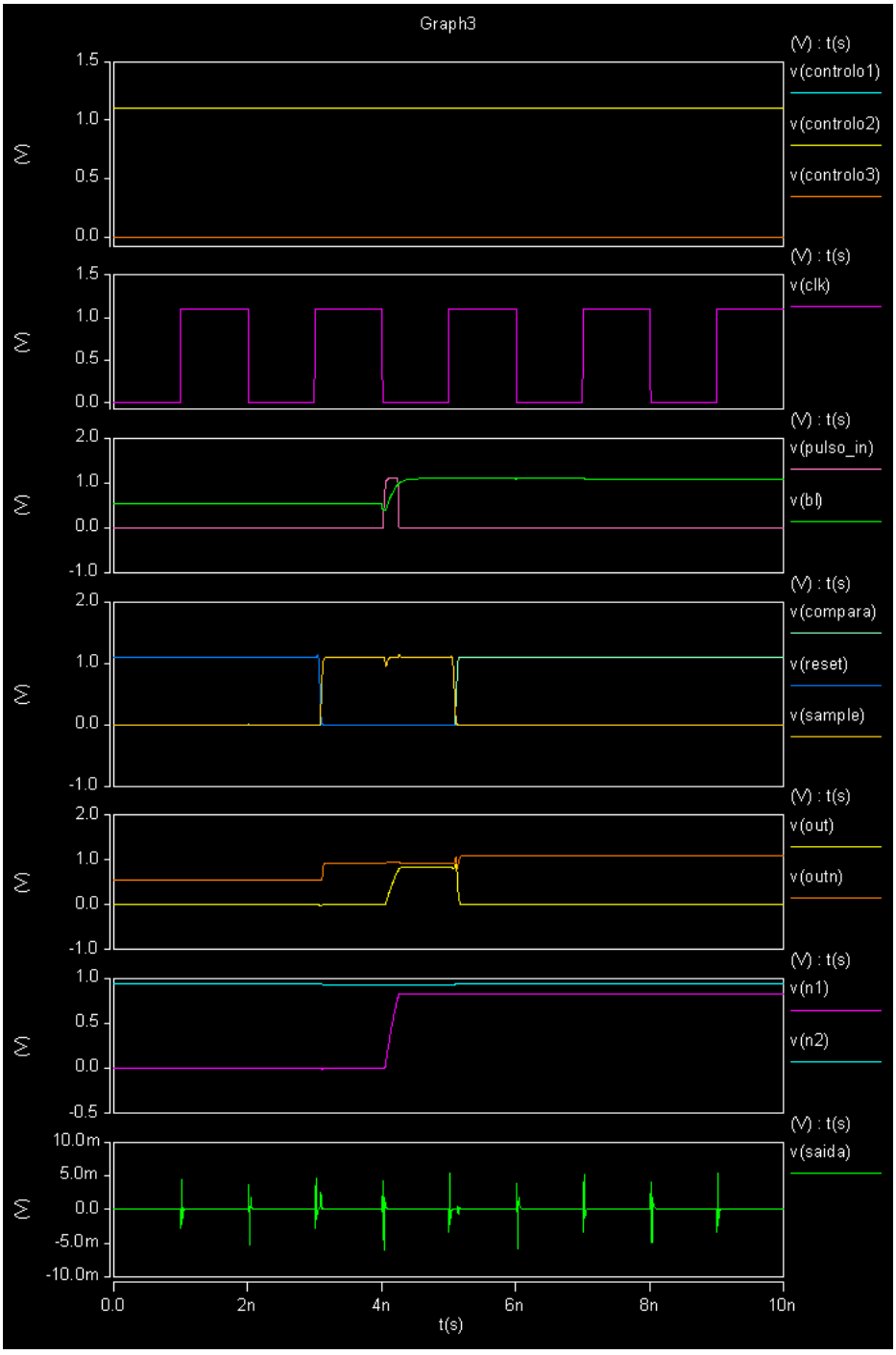


Figure 4.9 - Performance Sensor Simulation for DRAM initialization to VDD/2 – No error detected.

A similar simulation was again performed, but this time changing the sensibility of the sensor to activate only the Sensibility3 signal (and turning off Sensibility2 and Sensibility3).

The result obtained show that the same transition no longer is considered successful but, instead, is considered as an error, as it may be seen in Figure 4.10.

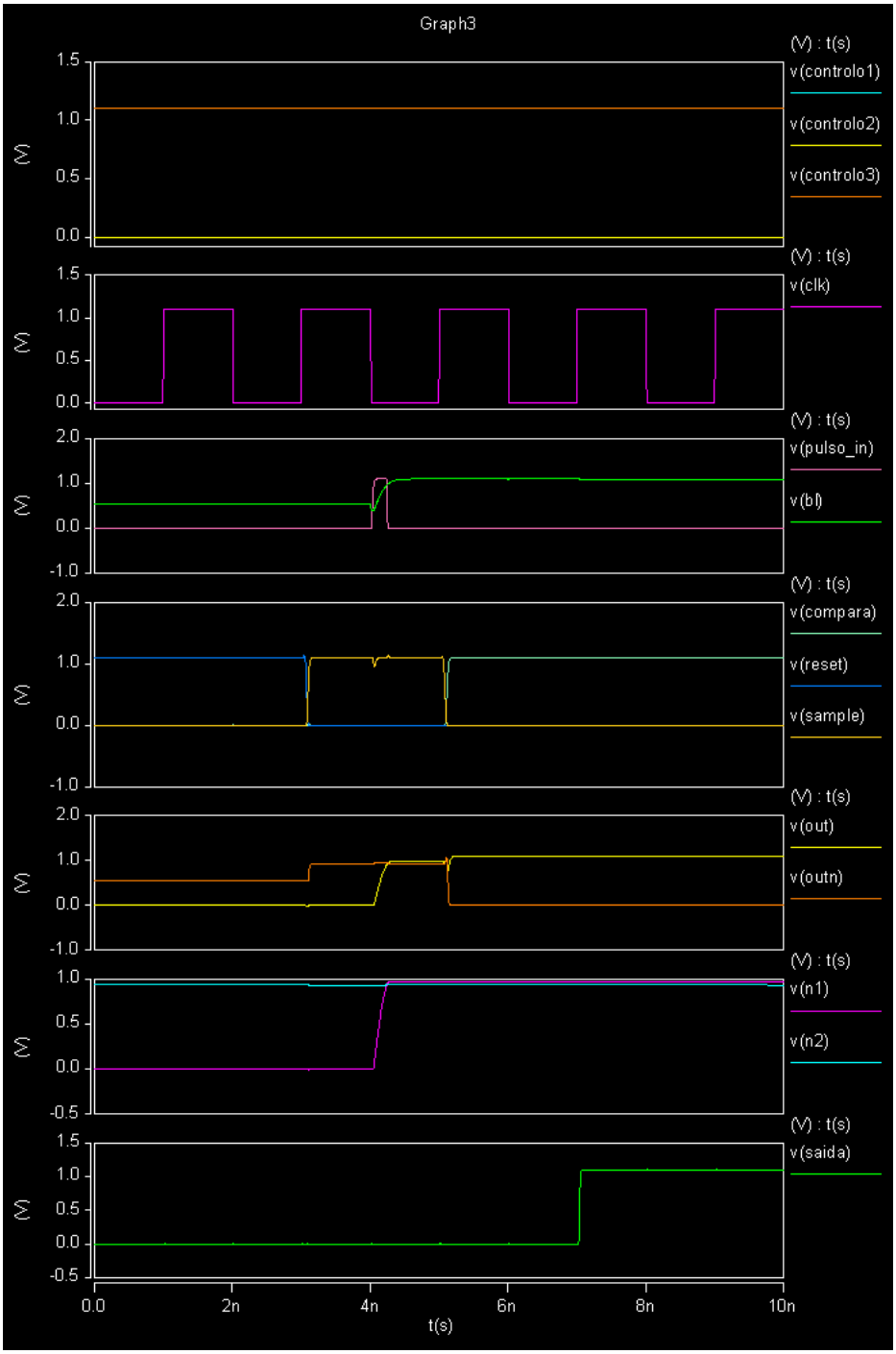


Figure 4.10 - Performance Sensor Simulation for DRAM initialization to VDD/2 – Error detected.

Now, activating the 3 sensibility controls, it is also possible to see that the value in *n1* (VSensed) increases its voltage, since the C1 capacitor will charge faster. Then, an error is

identified in the sensor, due to an unsafe transition for the sensibility used, as it can be seen in Figure 4.11.

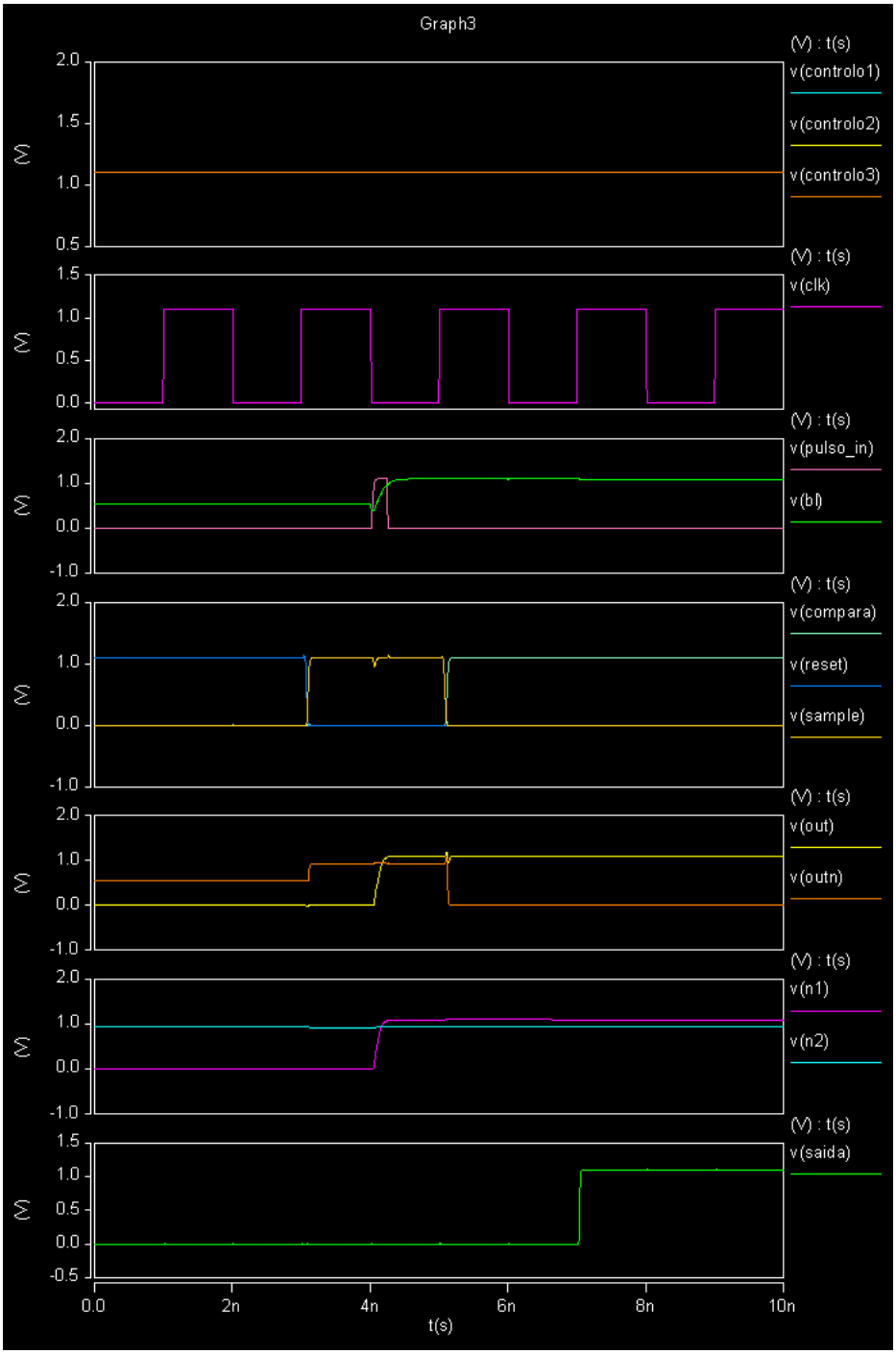


Figure 4.11 - Performance Sensor Simulation for DRAM initialization to VDD/2 –Error detected with the 3 controls.

In Figure 4.12, it is possible to see a simulation with similar conditions, when compared with the conditions from the simulation in Figure 4.9, with the particularity that the memory performance was changed by increasing aging degradations (transistors' |Vth| with

additional 10%) and VDD degradations (VDD was reduced to 0.8V) . In this case we are able to see that the same transition is now considered as an error (unsafe) transition.

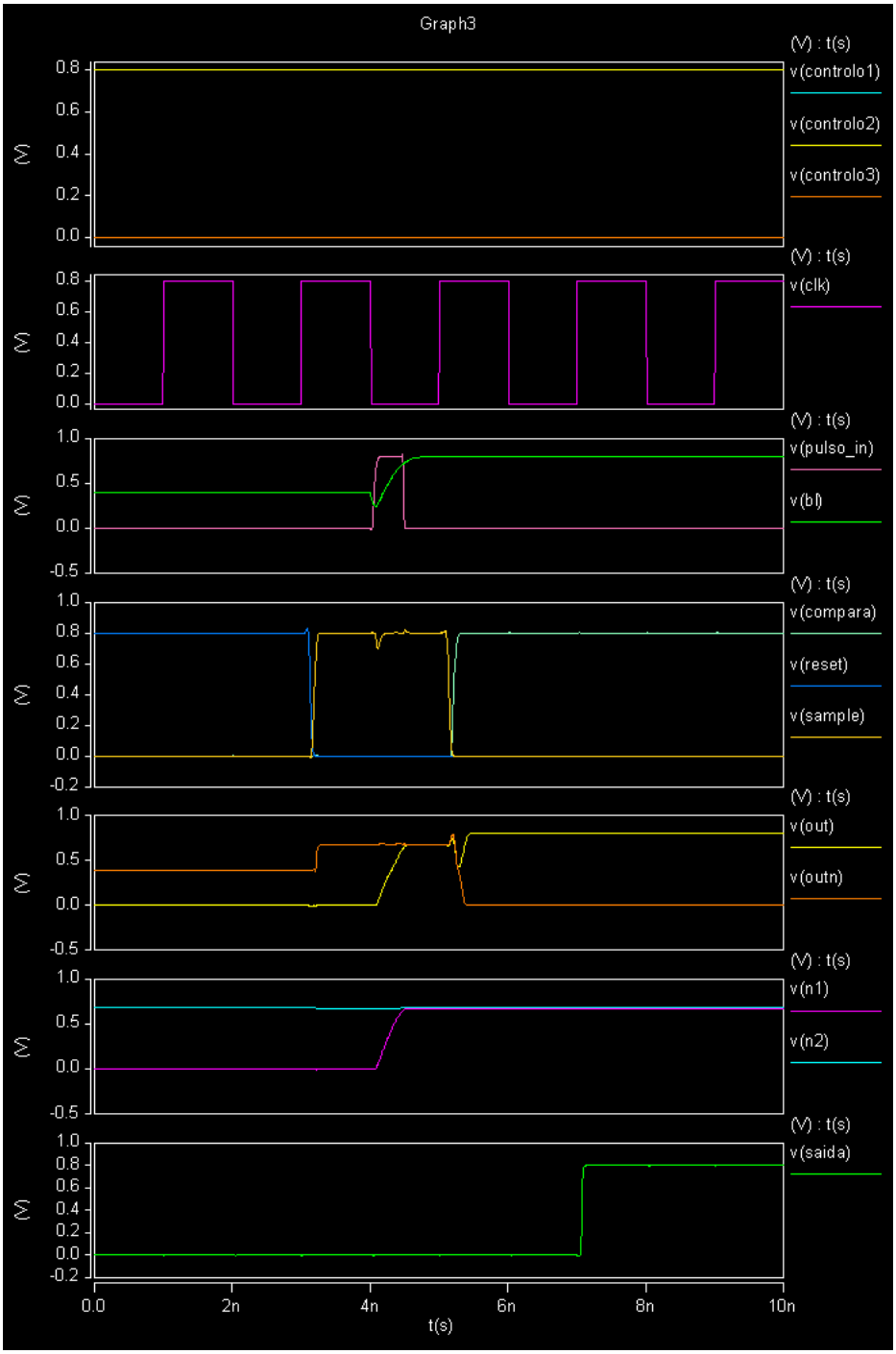


Figure 4.12 - Performance Sensor Simulation for DRAM initialization to VDD/2 –Error detected and aging.

5. CONCLUSIONS AND FUTURE WORK

5.1. CONCLUSIONS

This master's thesis is focused on the development of a performance sensor capable of detecting performance variations (or PVTA variations) in CMOS memories, signalizing when these variations impose unsafe memory operations, i.e., operations in the eminence of an error.

With the evolution of technology and the massive use of memories in virtually all the equipment we use today, which in most cases have quite small dimensions, new problems have begun to appear calling into question the performance of the equipment, thus creating the need to monitor its performance in order to predict the occurrence of an error. This error is usually observed in the increase of delays in signal transitions. If we consider memory circuits, bit lines signals are probably the most important signals, as they are the first main interface between the memory and the external circuitry. Therefore, performance that occur in the memory may be reflected in the bit lines, and its monitorization was the focus of this work.

From the several aging effects studied, such as PBTI and NBTI, it is the latter that has the greatest negative effect on the technologies we use today below 130nm, mainly on PMOS MOSFET transistors, whereas the effects of PBTI affect NMOS. The aging that is placed in the transistors consists of an increase of the Vth. However, there are other reasons that affect the performance of a memory and which have also been studied in this document, such as Process variations, Power-Supply Voltage variations, and Temperature variations, which together with the aging form the four parameters that may drastically change the performance of a memory, the PVTA variations.

In this thesis we identified two previous works on sensors that already address this problem. One was the On-Chip Aging Sensor (OCAS) [36], which appears as one of the first solutions to detect these problems. However, it displays some problems such as the need to have an offline memory in order to be able to carry out a test, among others. Another solution studied was the Performance Sensor [37], which shows a huge evolution concerning the OCAS. However, it still presents some limitations, such as the need to use the ascending and descending flanks of the clock signal to work, and it does not allow to change its sensibility,

or to be calibrated, throughout its operation and during its lifetime. Also, it only works for SRAM memories and it does not present solutions for the use in DRAM.

Based on these previous works and focusing on presenting a more sturdy solution, the purpose of this work was to detect the signals in the main tasks of a memory, i.e., Read and Write procedures. Moreover, it also allows the user to make calibrations throughout the use of the sensor, or change the sensibility of the sensor. Besides, it was also important to reduce the sensors dependency on synchronous clock signals, and create a solution that allows the use of the sensor in DRAM memories (being the first solution available for DRAM).

The performance sensor that was created, after its initialization, remains in the Sample state until a transition is detected in bit line, causing it to pass to the next state, the state for signal comparison. The sensor was developed with two types of Transition Detectors, which makes it possible to use in SRAM initialized to VDD or to VDD/2, as well as DRAM initialized to VDD/2. After the Transition Detector, it creates a pulse proportional to the transition delay time in the bit line. After the pulse is created, the signal goes to the Pulse Detector. In this block of the sensor, a new feature was introduced, the possibility of changing the sensibility of the sensor, by activating control signals. In this case, we change the sensibility of the sensor to detect a pulse by changing the current that charges a capacitor during the generated pulse width. This capacitor holds a DC voltage proportional to the pulse width, and therefore, also proportional to the bit line transition time. With this feature new feature, it is possible to create a sensor that can be more or less sensitive, according to what is to be monitored. The sensed voltage saved in the capacitor will then be compared (in the comparator block) with a reference value generated in the sensor. After this comparison, an output signal is generated, indicating whether the transition that occurred in the bit line should be considered as an unsafe transition (an error) or a safe transition (a successful transition).

From the developed work, it is concluded that there are several difficulties in creating a sturdy sensor, capable of detecting all the errors in all the existing types of memory. However, the solution presented is already an improved version to the previous works, especially regarding its application to different types of memories. In the simulations presented, it is possible to conclude that the performance sensor developed is a reliable, sturdy and versatile solution for the implementation in SRAM that are initialized both to VDD or VDD/2, as well as for DRAM initialized to VDD/2. In the simulations carried out it is possible to analyze the sensor's behavior to PVTA variations, which caused performance variations in the memory operation. The results demonstrate that the performance sensor has the ability to detect these problems, whether if a single parameter change, or in a combination of more than

one parameter with a degradation. The sensor allows the detection of changes in the bit line transitions in the memory when there is a degradation of its operating conditions, signaling errors according to the sensibility chosen in the Pulse Detector. It is also possible to see that changing only this calibration, it is possible to refine sensor operation and calibrate when the sensor should or should not consider as an error, a transition in the bit line. So, this solution is definitely more solid and versatile than the previous works. The results obtained in SRAM and DRAM are similar, proving its potential in both types of memory.

Other characteristic that may be observed is that, with the increase of the signal degradation, the sensor becomes more sensitive. This is an important feature, as the burden is not placed in the sensor design, as it happens in common sensors. Therefore, this is an advantage in its response capability, because sensor improves its sensibility when operating conditions are worse.

Finally, it is important to mention that sensor application in memories can be used as a global sensor, using one sensor externally to the circuit to be monitor, and connected to a dummy memory cell to mimic the operation of the real memory. Moreover, it can also be used as a local sensor, choosing specific locations to be monitored, and using some memory cells to statistically extrapolate the information, obtained from the key monitored memory cells, for the all memory. Also, sensor monitoring operation can be online, during the normal circuit operation, without the need to go off-line to execute the sensor test procedure.

5.2. FUTURE WORK

As usual, in all research works, it is not possible to reach a final version without the potential for it to grow and be optimized, since all research works are prone to be improved and worked on, so that new solutions appear.

One of the first future works that can be identified is to test and evaluate the sensor working in ultra-low power conditions, i.e., for subthreshold power-supply voltages, where the power-supply voltage is below the transistors' threshold voltages (Vth). Although important, these simulations were out of the scope of this work, because our main focus was to first define the sensor architectures that could work both on SRAM and DRAM memories. With these ultra-low power-supply voltage tests on the developed sensors, some corrections

could be necessary to do on the hardware, so that the sensor can work correctly at these ultra-small voltages.

A second future work, to be developed in the future, is to study the implementation of the sensor as a local sensor in SRAM and DRAM memories, namely knowing the amount of transistors to be used in the sensibility control, as well as the number of sensors to place in the memory and what would be the sampling monitoring methodology to implement.

Lastly, another future work is the implementation of the sensor in a chip, so that it would be possible to validate in silicon all sensor functionalities described.

BIBLIOGRAFIA

[1] Shashank Agrawal, Dario Vieira, “A survey on Internet of Things”, ABAKÓS – Instituto de Ciências exatas e Informáticas, Belo Horizonte,v. 1, n. 2, p. 78 – 95, maio 2013 – ISSN:2316–9451

[2] John Wiley & Sons, Ltd., “Internet of Things”, INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, Int. J. Commun. Syst. 2012; 25:1101–1102, Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2417

[3] J. Semião, H. Santos, R. Cabral, M. B. Santos and P. Teixeira, "PVTA-Aware Performance SRAM Sensor for IoT Applications", Proceedings of the 2nd INternational CongRess on Engineering and Sustainability in the XXI cEntury – INCREaSE 2019, Springer, Pages 337-353, October 09-11, 2019, DOI 10.1007/978-3-030-30938-1.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generat. Comput. Syst., vol. 29, no. 7, pp. 1645-1660, 2013.

[5] T. Park, N. Abuzainab, W. Saad, "Learning How to Communicate in the Internet of Things: Finite Resources and Heterogeneity", IEEE Access: Optimization for Emerging Wireless Networks: IoT, 5G and Smart Grid Communication Networks, Special Session, IEEE Access, vol. 4, pp. 7063 - 7073, Nov., 2016.

[6] H. Yu Shwe, T. King Jet, P. Han Joo Chong, "An IoT-oriented data storage framework in smart city applications", 2016 International Conference on Information and Communication Technology Convergence (ICTC), pp. 106-108, 2016.

[7] A. Mohon Ghosh, D. Halder, S K Alamgir Hossain, "Remote health monitoring system through IoT", 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp. 921-926, 2016.

[8] Tadaaki Yamauchi; Hiroyuki Kondo; Koji Nii, "Automotive low power technology for IoT society", 2015 Symposium on VLSI Circuits (VLSI Circuits), pp. T80-T81, 2015.

[9] H. He et al., "The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence," 2016

IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, 2016, pp. 1015-1021.

[10] T. Xu; J. B. Wendt; M. Potkonjak, "Security of IoT systems: Design challenges and opportunities", Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 417 - 423, 2014.

[11] K. Kaur, K. Kaur, "A study of power management techniques for Internet of Things (IoT)", Proc. Int. Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1781 - 1785, 2016-

[12] S. Carreon-Bautista, L. Huang, E. Sanchez-Sinencio, "An Autonomous Energy Harvesting Power Management Unit With Digital Regulation for IoT Applications", IEEE Journal of Solid-State Circuits, vol. 51, Issue 6, pp. 1457 - 1474, 2016.

[13] J. Semião, M. Irago, J. Rodríguez-Andina, L. Piccoli, F. Vargas, M. Santos, I. Teixeira, and J. Teixeira, "Signal Integrity Enhancement in Digital Circuits," in IEEE Design and Test of Computers, vol. 25, no. 5 pp. 452-461, September-October, 2008, DOI: http://dx.doi.org/10.1109/MDT.2008.146

[14] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, "The Impact of NBTI on the Performance of Combinational and Sequential Circuits," in Proc. of the ACM/IEEE Design Automation Conference, pp. 364-369, San Diego, CA, USA, 4-8 June, 2007, DOI:http://dx.doi.org/10.1109/DAC.2007.375188.

[15] T. Kim and Z. Kong, "Impact Analysis of NBTI/PBTI on SRAM VMIN and Design Techniques for Improved SRAM VMIN", in Journal of Semiconductor Technology and Science., vol. 13, no. 2, pp. 87-97, April, 2013, DOI:http://dx.doi.org/10.5573/JSTS.2013.13.2.87.

[16] A. Ceratti, T. Copetti, L. Bolzani, and F. Vargas, "On-Chip Aging Sensor to Monitor NBTI Effect in Nano-Scale SRAM," in Proc. of the 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits and Systems, DDECS 2012, pp. 354-359, Tallinn, Estonia, 18-20 April, 2012, DOI:http://dx.doi.org/10.1109/DDECS.2012.6219087.

[17] A. Gadhe, U. Shirode, "Read stability and Write ability analysis of different SRAM cell structures", International Journal of Engineering Research and Applications (IJERA), Vol. 3, Issue 1, January -February 2013, pp.1073-1078.

[18] M. Sharifkhani, M. Sachdev, "SRAM Cell Stability: A Dynamic Perspective", IEEE Journal of Solid-State Circuits, Vol. 44, No. 2, February 2009.

[19] J. Semiao, A. Romao, D. Saraiva, C. Leong, M. Santos, I. Teixeira, and P. Teixeira, "Performance Sensor for Tolerance and Predictive Detection of Delay-Faults", Proc. of the Int. Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT) 2014, Amsterdam, The Netherlands, October 1-3, 2014, DOI:http://dx.doi.org/10.1109/DFT.2014.6962092.

[20] C. Martins, J. Semião, J. Vazquez, V. Champac, M. Santos, I. Teixeira, and P. Teixeira, "Adaptive Error-Prediction Flip-flop for Performance Failure Prediction with Aging Sensors," in IEEE 29th- VLSI Test Symposium (VTS), pp. 203-208, Dana Point, CA, USA, 1-5 May, 2011, DOI:http://dx.doi.org/10.1109/VTS.2011.5783784.

[21] J. Semião, H. Santos, R. Cabral, A. Romão, M. Santos, I.C. Teixeira, J.P. Teixeira, "Aging and Performance Sensor for SRAM", Proc. of the 31th. Design of Circuits and Integrated Systems Conference (DCIS 2016), November, 23rd-25th, Granada (Spain), 2016.

[22] M. Valdés, J. Freijedo, M.J. Moure, J.J. Rodríguez-Andina, J. Semião, F. Vargas, I.C. Teixeira, J.P. Teixeira, "Design and Validation of Configurable On-Line Aging Sensors in Nanometer-Scale FPGAs", IEEE Transactions on Nanotechnology, Special Issue on "Defect & Fault Tolerance in VLSI and Nanotechnology Systems", vol. 12, nº 4, pp. 508-517, July 2013.

[23] J. Semião, R. Cabral, C. Leong, M. Santos, I. Teixeira, J. Teixeira, "Dynamic Voltage Scaling with Fault-Tolerance for Lifetime Operation", in the 4th. Workshop on Manufacturable and Dependable Multicore Architectures at Nanoscale (MEDIAN'15) / DATE'2015 Workshop W06, Grenoble, France, 13 March, 2015

[24] Bahar Farahani, Seyedamin Habibi, Saeed Safari, “A cross-layer SER analysis in the presence of PVTA variation” School of Electrical and Computer Engineering, University of Tehran, Tehran 14395-1515, Iran, http://dx.doi.org/10.1016/j.microrel.2015.04.008

[25] Mintarno E, Chandra V, Pietromonaco D, Aitken R, Dutton RW. Workload dependent NBTI and PBTI analysis for a sub-45nm commercial microprocessor. In: International reliability physics symposium; 2013. p. 3A–1.

[26] Jakson Pachito, “Metodologias para prever o envelhecimento de circuitos digitais”, Dissertação para a obtenção do grau de mestre em Engª. Eléctrica e Electrónica, Universidade do Algarve, Instituto Superior de Engenharia, Algarve, Portugal, Janeiro 2012

[27] Hugo Santos, “Aging Sensor for CMOS Memory Cells”, Universidade do Algarve, Instituto Superior de Engenharia, Algarve, Portugal, Setembro 2015

[28] J. G. Massey, “NBTI: What We Know and What We Need to Know - A Tutorial Addressing the Current Understanding and Challenges for the Future”, Integrated Reliability Workshop Final Report (IRW’04), IEEE, pp. 199-211, DOI: 10.1109/IRWS.2004.1422784, 18-21 Oct., 2004.

[29] A. Islam, H. Kufluoglu, D. Varghese, S. Mahapatra, and M. Alam, “Recent Issues in Negative-Bias Temperature Instability: Initial Degradation, Field Dependence of Interface Trap Generation, Hole Trapping Effects, and Relaxation,” in IEEE Transactions on Electron Devices, vol. 54, no. 9, pp. 2143–2154, September, 2007, DOI:http://dx.doi.org/10.1109/TED.2007.902883.

[30] S. Kumar, C. Kim, and S. Sapatnekar, “Impact of NBTI on SRAM Read Stability and Design for Reliability”, in ISQED 06 Proceeding of the 7th International Symposium on Quality Electronic Design, pp. 210–218, Washington, DC, USA, 2006, DOI:http://dx.doi.org/10.1109/ISQED.2006.73.

[31] Zhang, J. F., & Eccleston, W. (1998), “Positive bias temperature instability in MOSFETs”. IEEE Transactions on Electron Devices, 45(1), 116–124.doi:10.1109/16.658821

[32] Ioannou, D. P., Mittl, S., & La Rosa, G. (2009). “Positive Bias Temperature Instability Effects in nMOSFETs With HfO2/TiN Gate Stacks”. IEEE Transactions on Device and Materials Reliability, 9(2), 128–134.doi:10.1109/tdmr.2009.2020432

[33] A. Sedra and K. Smith, Microelectronic Circuits, 7th edition, Oxford University Press, Inc. New York, USA, 2014, cap. 16, pp. 1236–1283.

[34] Calhoun, B. H., & Chandrakasan, A. P. (2006). “Static Noise Margin Variation for Sub-threshold SRAM in 65-nm CMOS”. IEEE Journal of Solid-State Circuits, 41(7), 1673–1679. doi:10.1109/jssc.2006.873215

[35] Arandilla, C. D. C., Alvarez, A. B., & Roque, C. R. K. (2011). “Static Noise Margin of 6T SRAM Cell in 90-nm CMOS”. 2011 UkSim 13th International Conference on Computer Modelling and Simulation.doi:10.1109/uksim.2011.108

[36] Ceratti, A., Copetti, T., Bolzani, L., & Vargas, F. (2012). On-chip aging sensor to monitor NBTI effect in nano-scale SRAM. 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS).doi:10.1109/ddecs.2012.6219087

[37] Santos, H., Semiao, J., Cabral, R., Romao, A., Santos, M. B., Teixeira, I. C., & Teixeira, J. P. (2016). Aging and performance sensor for SRAM. 2016 Conference on Design of Circuits and Integrated Systems (DCIS).doi:10.1109/dcis.2016.7845354

[38] Jorge Semião, Ruben Cabral, Hugo Cavalaria, Marcelino Santos, Isabel C. Teixeira, J. Paulo Teixeira, "Ultra-Low-Power Strategy for Reliable IoE Nanoscale Integrated Circuits", book chopter in "Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities", IGI Global, February, 2019, DOI: https://doi.org/10.4018/978-1-5225-7332-6.ch011.

[39] Bruce Jacob, Spencer W. Ng, David T. Wang, Memory Systems Cache, Dram, Disk, Elsevier Inc, USA, 2008, ISBN: 978-0-12-379751-3