Technical Note PR-TN 2006/00368

Issued: 05/2006

Modeling of transient programming and erasing of SONOS non-volatile memories

I. Remond

(Student at Illinois Institute of Technology, Chicago)

N. Akil

Philips Research Europe

TABLE OF CONTENTS

	Page
Abstract	3
Acknowledgments	4
List of symbols	5
List of constants	7
List of tables and figures	8
Introduction	12
1. Theory	13
1.1. Outline	13
1.2. Threshold voltage	13
1.3. Electric field calculation	16
1.4. Calculation of the injected currents in the nitride	18
1.5. Calculation/Update of the trap density in the silicone nitride	21
1.6. Charge density calculation	25
1.7. Transition from time *t* to time *t+dt*	25
2. Modeling results for the reference wafer (2.2/6/8nm ONO N-type gate)	26
2.1. Fitting parameters setting	26
2.1.1. Coefficients of the Fowler-Nordheim current	27
2.1.2. Mobility and current criterion	27
2.1.3. Carrier effective mass	28
2.1.4. Trap capture cross section in the nitride	29
2.1.5. Oxide and nitride tunneling barriers	29
2.2 Other physical quantities which are important for the model	30
2.2.1. Bottom oxide thickness	30
2.2.2. Current criterion	30
2.2.3. Total trap density	30
2.3. Programming	31
2.3.1. Threshold voltage	31
2.3.2. Electric field through the ONO	34
2.3.3. Injected current density in the nitride	35
2.3.4. Trap density in the nitride	37
2.3.5. Charge distribution in the nitride	39
2.4. Erasing	40
2.4.1. Threshold voltage	40
2.4.2. Electric field through the ONO	43
2.4.3. Injected current density in the nitride	43
2.4.4. Trap density in the nitride	45
2.4.5. Charge distribution in the nitride	47

3. Modeling results for other stacks 49
3.1. Influence of the gate doping 49
3.2. Impact of changes in the oxide thickness on the model 54
3.2.1. Top oxide thickness 54
3.2.2. Bottom oxide thickness 57
3.3. Impact of a change in the k-value of the top oxide on the model results 57

Conclusion 61

References 62

Appendix 63
A.1. General outline 63
A.2. Matlab program for the model 65
A.2.1. Program case 65
A.2.1.1. main_program.m 65
A.2.1.2. Vt_function_program.m 75
A.2.1.3. Efield_program.m 75
A.2.1.4. Jox_program.m 76
A.2.1.5. fun.m 78
A.2.2. Erase case 78
A.2.2.1. main_erase.m 78
A.2.2.2. Vt_function_erase.m 88
A.2.2.3. Efield_ erase.m 89
A.2.2.4. Jox_ erase.m 90
A.2.2.5. fun.m 91

ABSTRACT

SONOS (silicon - oxide - nitride - oxide - silicon) non volatile memories are attractive candidates for embedded flash memories in the 45nm CMOS generation and beyond. Most of the commercial device simulation software's, like Medici, are not able yet to simulate transient programming and erasing of a SONOS memory. A model for program and erase operations of a SONOS device was developed in this work, which yields good agreement with measurements. This model is implemented in a Matlab program and can be easily used to predict the program and erase curves (threshold voltage shift versus program/erase time for a defined gate voltage) for different ONO stack variations, such as oxide thickness variations.
The results presented in this work were obtained in the framework of an internship of 8 months of Indira Remond student at the Illinois Institute of Technology, Chicago, US.

ACKNOWLEDGMENTS

The authors would like to thank all the members of the non-volatile memory project for their support and for many helpful discussions.

LIST OF SYMBOLS

Symbol	Description	Unit
A_{FN}	Pre-exponential coefficient for Fowler-Nordheim current equation	
B_{FN}	Exponential coefficient for Fowler-Nordheim current equation	
C_{eff}	Equivalent capacitance per unit area of the ONO stack	[F/cm²]
C_{botox}	Equivalent capacitance per unit area of the bottom oxide	[F/cm²]
C_N	Equivalent capacitance per unit area of the nitride	[F/cm²]
C_{topox}	Equivalent capacitance per unit area of the top oxide	[F/cm²]
d_{botox}	Bottom oxide thickness	[cm]
d_N	Nitride thickness	[cm]
d_{topox}	Top oxide thickness	[cm]
E_g	Silicon band gap at T=300K	[eV]
E_{botox}	Electric field in the bottom oxide	[V/cm]
E_N	Electric field in the nitride	[V/cm]
E_{topox}	Electric field in the top oxide	[V/cm]
$\hbar$	Reduced Planck constant	[J.s]
I_D	Drain current	[A]
J_{DT}	Direct tunneling current density	[A/cm²]
J_{FN}	Fowler-Nordheim tunneling current density	[A/cm²]
J_{MFN}	Modified Fowler-Nordheim tunneling current density	[A/cm²]
J_{Ne}	Electron current density in the nitride	[A/cm²]
J_{Nh}	Hole current density in the nitride	[A/cm²]
k	Boltzmann constant	[J/K]
L	Gate length	[cm]
m_0	Electron rest mass	[kg]
m_N	Electron effective mass in the nitride	[kg]
m_{ox}	Electron effective mass in the oxide	[kg]
N_A	Bulk doping	[1/cm³]
N_t	Total trap density in the nitride	[1/cm³]
n_{te}	Density of traps filled with electrons in the nitride	[1/cm³]
n_{tf}	Density of free traps in the nitride	[1/cm³]
n_{th}	Density of traps filled with holes in the nitride	[1/cm³]
q	Elementary charge	[Coul]
Q_N	Charge per unit area in the nitride	[Coul/cm²]
Q_{ox}	Fixed oxide charge in the nitride	[Coul/cm²]
V_{cg}	Control gate voltage	[V]
V_D	Drain voltage	[V]

Symbol	Description	Unit
V_{fb}	Flat band voltage	[V]
V_t	Threshold voltage	[V]
W	Gate width	[cm]
γ	Ratio of the oxide and nitride dielectric constants	
ε_0	Permittivity in vacuum	[F/cm]
ε_{botox}	Dielectric constant in the bottom oxide	
ε_N	Dielectric constant in the nitride	
ε_{topox}	Dielectric constant in the top oxide	
ε_S	Dielectric constant in the substrate	
μ_e	Electron semiconductor mobility	[cm²/(V.s)]
μ_h	Hole semiconductor mobility	[cm²/(V.s)]
ν_0	Frequency factor	[1/s]
ρ_N	Charge density in the nitride	[Coul/cm³]
σ_t	Trap capture cross section in the nitride	[1/ cm²]
ϕ_1	Bottom oxide tunneling barrier	[V]
ϕ_2	Nitride tunneling barrier	[V]
ϕ_3	Top oxide tunneling barrier	[V]
ϕ_M	Gate work function	[V]
χ	Electron affinity in the substrate	[V]
ψ_B	Potential barrier between the Fermi level and the intrinsic Fermi level	[V]
ψ_S	Surface potential	[V]

LIST OF CONSTANTS

Symbol	Value	
A'_{FN}	6e-2	
B'_{FN}	0.95	
E_g	1.12 eV	
$\hbar$	1.055458e-34 J.s	
I_D	10 µA	
k	1.38066e-23 J/K	
L	0.2e-4 cm	
m_0	0.91095e-30 kg	
m_N	Program	$0.1.m_0$
	Erase	$0.4.m_0$
N_A	1e17 /cm³	
N_t	5e19 /cm³	
q	1.60218e-19 Coul	
V_D	0.5 V	
W	0.18e-4 cm	
ε_0	8.85418e-14 F/cm	
ε_{botox}	3.9	
ε_N	7.5	
ε_{topox}	3.9	
ε_S	11.8	
μ_e	130 cm²/(V.s)	
ν_0	10e9 1/s	
σ_t	1e-13 / cm²	
ϕ_1	Program	3.1 V
	Erase	4.8 V
ϕ_2	Program	1.05 V
	Erase	3.35 V
ϕ_3	Program	4.8 V
	Erase	3.1 V
ϕ_M	N-type gate	4.17 V
	P-type gate	5.2 V
χ	4.17 V	
ψ_B	0.4070818 V	

LIST OF TABLES AND FIGURES

Page

Table 1: Fitting parameters values for the reference SONOS stack (2.2/6/8nm N-type gate) 26

Table 2: Values of some important quantities for the modeling of the reference stack 30

Figure 1: I_D V_G curves for the reference wafer PLI9517 W#2 2.2/6/8nm ONO N-type gate 15

Figure 2: Energy band diagram for the programming case V_{cg} >0 19

Figure 3: Energy band diagram for the erasing case V_{cg} <0 20

Figure 4: Dependence on the applied control gate voltage of the currents injected into the nitride 20

Figure 5: Current density vs. electric field in the oxide for A'_{FN} = B'_{FN} =1 (default values) and for A'_{FN} =6e-2 and B'_{FN} =0.95. 27

Figure 6: m_{ox} influence on J_{MFN} and J_{DT} vs. E_{botox} 28

Figure 7: Influence of ϕ_2 on J_{MFN} and J_{DT} vs. E_{botox} 29

Figure 8: Programming curves for the reference stack with a 10V control gate voltage. The measurements were done on 1T mini array (256bits).. n_{th}(t=0) is taken $0.1e^{19}/cm^3$. 31

Figure 9: Programming curves for the reference stack with an 11V control gate voltage. The measurements were done on 1T mini array (256bits). n_{th}(t=0) is taken $0.1e^{19}/cm^3$. 32

Figure 10: Programming curves for the reference stack with a 12V control gate voltage. The measurements were done on 1T mini array (256bits). n_{th}(t=0) is taken $0.1e^{19}/cm^3$. 32

Figure 11: Programming curves for the reference stack with a 13V control gate voltage. The measurements were done on 1T mini array (256bits). n_{th}(t=0) is taken $0.1e^{19}/cm^3$. 33

Figure 12: Programming curves comparison between the modeling and measurements for different control gate voltages and different starting V_t. The starting Vt of the programming curves Vcg=11, 12 and 13V was the same. 34

Figure 13: Electric field in the top and bottom oxide vs. time extracted from the program, for a 12V programming control gate voltage 35

Figure 14: Distribution of the current injected from the bottom oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 36

Figure 15: Distribution of the current injected from the top oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 36

Figure 16: Density of traps filled with electrons in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 37

Figure 17: Density of traps filled with holes in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 38

Figure 18: Density of empty traps in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 38

Figure 19: Charge distribution in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 39

Figure 20: Erase curve for the reference stack with a -10V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$. 40

Figure 21: Erase curve for the reference stack with a -11V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$. 41

Figure 22: Erase curve for the reference stack with a -12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$. 41

Figure 23: Erase curve for the reference stack with a -13V control gate voltage. The measurements were done on 1T mini array (256bits).$n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$. 42

Figure 24: Erase curves comparison between model and measurement for different control gate voltages. 42

Figure 25: Electric field in the top and bottom oxide vs. time during erase, for a -12V control gate voltage 43

Figure 26: Distribution of the current injected from the top oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 44

Figure 27: Distribution of the current injected from the bottom oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 44

Figure 28: Density of traps filled with electrons in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 45

Figure 29: Density of traps filled with holes in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 46

Figure 30: Density of empty traps in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 46

Figure 31: Charge distribution in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 47

Figure 32: Comparison of the charge distribution in the nitride for -12V and -13V control gate voltages (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface) 48

Figure 33: Comparison between the program curves from measurements for 2.2/6/8nm SONOS stack with N-type and P-type gate, with V_{cg}=10V (starting from a virgin device). The measurements were done on 1T mini array (256bits). 49

Figure 34: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -11V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$. 50

Figure 35: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$. 51

Figure 36: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -13V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$. 51

Figure 37: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -14V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$. 52

Figure 38: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for an 11V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$. 52

Figure 39: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for a 12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$. 53

Figure 40: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for a 13V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$ 53

Figure 41: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for a 14V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$ 54

Figure 42: Program curves for a 2.2/6/5nm ONO stack with a 9Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$is taken $0.1e^{19}/cm^3$. 55

Figure 43: Program curves for a 2.2/6/5nm ONO stack with a 10Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$is taken $0.1e^{19}/cm^3$. 55

Figure 44: Erase curves for a 2.2/6/5nm ONO stack with a -9Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $2.1e^{19}/cm^3$. 56

Figure 45: Erase curves for a 2.2/6/5nm ONO stack with a -10Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $2.1e^{19}/cm^3$. 56

Figure 46: Erase curves for a 3/6/8nm ONO stack, N-type gate with a -14Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $0.75e^{19}/cm^3$ 57

Figure 47: Comparison between program/erase curves from measurements at Vcg=+/-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) with N- and P-type gates. 58

Figure 48: Comparison between two program/erase curves from measurements at Vcg=+/-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) with P-type gate. 58

Figure 49: Comparison between two program/erase curves from the model at Vcg=+/-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) with P-type gate. 59

Figure 50: Program curves for a SHINOS (2.2/6/10nmhfsiO2) stack with a 12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$is taken $0.1e^{19}/cm^3$. 59

Figure 51: Erase curves for a SHINOS (2.2/6/10nmhfsiO2) stack with a -12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$is taken $1.5e^{19}/cm^3$. 60

Figure 52: Structure of the Matlab program for the model 63

INTRODUCTION

The main specification of non-volatile memories is that they can retain stored data for a long time, at least ten years, without any power supply. In this work, we focus on one concept of embedded non-volatile memories, called SONOS (silicon - oxide - nitride - oxide - silicon). SONOS non-volatile memories were invented in the 1960's. This memory concept was abandoned in the 1980's because of technological obstacles. Today, these barriers have been overcome to a certain extend. Therefore, SONOS non-volatile memories are now the focus of many experimental and theoretical efforts, due to their ease of integration in CMOS baseline process and lower program/erase voltages compared to floating-gate devices. SONOS non-volatile memories represent attractive candidates for embedded flash memories in the 45nm CMOS generation and beyond.

However, most of the commercial device simulation software's, like Medici, are not able yet to simulate transient programming and erasing of a SONOS non-volatile memory. A model for program and erase operations is needed to enable fast SONOS development. Such a model was developed in this work and implemented in a Matlab program.

In the first chapter, we review the theory which is used for performing the program/erase of an N-type SONOS device. In the second chapter, we present the modeling results for the reference ONO stack (2.2/6/8nm N-type gate). Finally, we present in the third chapter the modeling results for different types of gate doping and ONO stack variations (such as oxide thickness variations or the substitution of the top oxide by a high-k material). The matlab program is given in the appendix.

CHAPTER I – THEORY

1.1. Outline

This chapter presents the basic theory which is used for performing the program/erase modeling of an N-type SONOS device.

In this work we will repeat the following five step procedure in order to simulate transient programming and erasing of a SONOS memory:

1) Threshold voltage calculation
2) Electric field calculation through the ONO
3) Calculation of the injected currents in the nitride
4) Calculation/Update of the carrier density in the silicone nitride
5) Charge density calculation in the silicone nitride

1.2. Threshold voltage

Definition

The ***threshold voltage*** corresponds to the bias that must be applied to the gate in order to invert the channel region.

If the gate voltage is below the threshold voltage, the transistor is turned off and ideally there is no current from the drain to the source of the transistor, due to the absence of a conductive channel.
If the gate voltage is larger than the threshold voltage, a conductive channel is created just below the interface and the transistor is tuned on. A current can flow from drain to source when the drain voltage is different from the source voltage $V_{DS} \neq 0$. This situation is called ***strong inversion*.**

1. Threshold voltage calculation

For fast measurements, often the threshold voltage V_t is determined by current criterion. This current criterion is chosen when the transistor is in strong inversion regime (e.g. conducting a current at a certain V_{DS} and $V_{GS} > V_t$)
The threshold voltage is given by [Sze1981]:

$$V_t = \frac{\sqrt{2\varepsilon_S q N_A \psi_S}}{C_{eff}} + \psi_S + V_{fb} + \frac{I_D}{\beta V_D} - \mathrm{X}, \quad (1.1)$$

where

- **1st term:**
  - C_{eff} denotes the equivalent capacitance per unit area of the ONO stack [F/cm²], given by:

$$\frac{1}{C_{eff}} = \frac{1}{C_{topox}} + \frac{1}{C_N} + \frac{1}{C_{botox}}, \quad (1.2)$$

  Where $C_{topox} = \frac{\varepsilon_{topox}}{d_{topox}}$, $C_N = \frac{\varepsilon_N}{d_N}$, and $C_{botox} = \frac{\varepsilon_{botox}}{d_{botox}}$ are the equivalent capacitance in the top oxide, the nitride, and the bottom oxide, respectively. ε_{topox}, ε_N, ε_{botox} are the dielectric constants in the top oxide, the nitride, and the bottom oxide, respectively, and d_{topox}, d_N, d_{botox} are the top oxide, nitride, and bottom oxide thickness, respectively [cm].
  - ε_S denotes the silicon dielectric constant
  - N_A denotes the doping concentration in the semiconductor substrate [1/cm³] (typically 1e17 / cm³).

- **2nd term: surface potential**

  ψ_S and ψ_B respectively denote the surface potential and the energy difference between the Fermi level and the intrinsic Fermi level in the substrate, both in [V].
  In case of strong inversion regime, the surface potential is given by:

$$\psi_S = 2 \cdot \psi_B, \quad (1.3)$$

  Where $\psi_B = \frac{kT}{q} \ln(\frac{N_A}{n_i})$ and n_i the intrinsic carrier density.

- **3rd term: flat band voltage.**

  Definition: The flat band voltage is the gate voltage required to counterbalance the work function difference between the gate and the semiconductor of a SONOS structure, to have $\psi_S = 0$.

$$V_{fb} = \phi_{MS} - \left(Q_N + Q_{ox}\right)\left[\frac{d_{topox}}{\varepsilon_{topox}\varepsilon_0} + \frac{\left(d_N - \overline{d_N}\right)}{\varepsilon_N \varepsilon_0}\right], \quad (1.4)$$

- $\overline{d_n}$ denotes the charge centroid in the nitride, i.e. the gravity center of the charge distribution in the nitride layer, given by:

$$\overline{d_n} = \frac{\int_0^{d_N} x\rho_N dx}{Q_N}$$

- $Q_N = \int_0^{d_N} \rho_N(x)dx$ is the charge per unit area in the silicon nitride [Coul/cm²].
- Q_{ox} is the fixed charge in the oxide [Coul/cm²].
- $\rho_N = q \cdot \left(n_{th} - n_{te}\right)$ is the charge density in the silicon nitride [Coul/cm³].
- n_{th} and n_{te} denote respectively the number of traps in the silicon nitride filled with holes and electrons [1/ cm³].
- ϕ_{MS} is work function difference between the gate and the bulk $\phi_{MS}=\phi_M - \phi_{S=}\phi_M - (\chi + Eg/2 + \psi_B)$, where χ is the silicon electron affinity=4.17V. The work function of n+ - and p+ -type gate are different (respectively $\phi_M=4.17V$ and $\phi_M=5.2V$).

➢ **4th term: current criterion.**

As we mentioned earlier, there is a practical need to calculate the V_t based on current criterion to allow the final program to cope directly with measurements, because most often the measurements are done in this way.

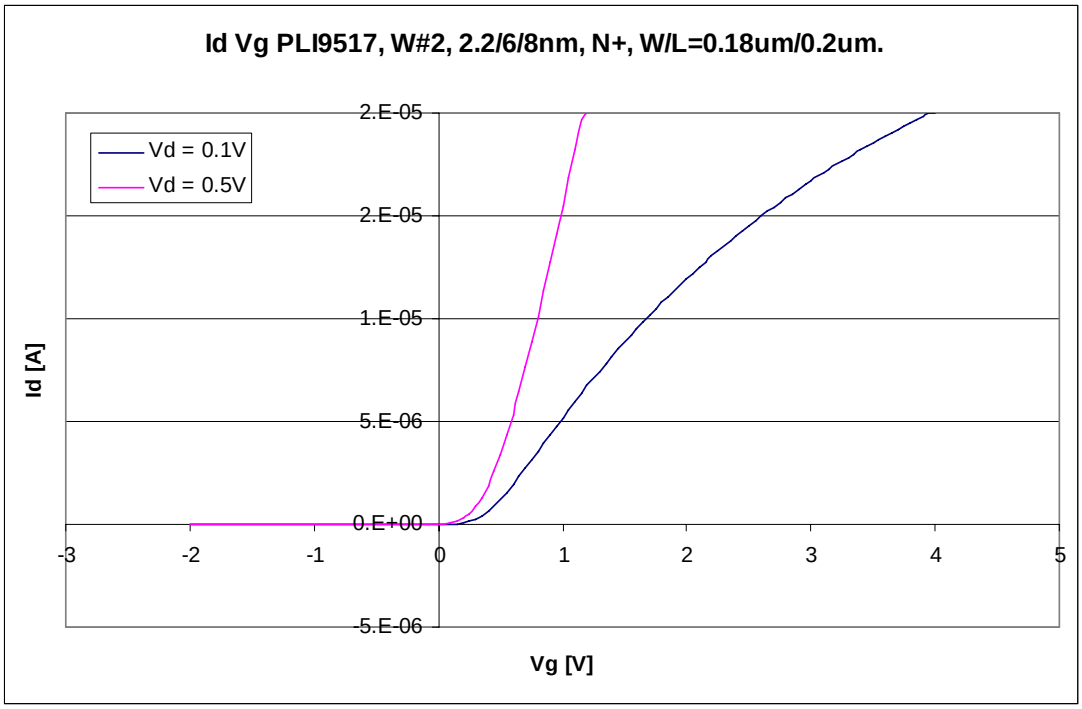

Figure 1: I_D V_G ***curves for the reference wafer PLI9517 W#2 2.2/6/8nm ONO N-type gate***

Therefore, we calculate the control gate voltage at a certain drain current, considering that the transistor is operating in the linear region i.e. $|V_G - V_{th}| > |V_{DS}|$, which implies:

$$I_D = \beta \cdot (V_G - V_{th}) V_D \quad , (1.5)$$

- I_D is the drain current [A], which we take equal to 10 µA for the current criterion.
- V_D is the drain voltage [V], which we take equal to 0.5 V.
- $\beta = \frac{W}{L} \mu_n C_{eff}$, where μ_n is the electron mobility [cm²/V/s], W the gate width [cm] and L the gate length [cm]. The mobility value is often much lower than the theoretical value and can be extracted from the linear slope of the IdVg. We found a value of 130 cm²/V/s for the characteristics in figure 1 at Vd=0.5V. This value is not fixed and should be extracted from the IdVg of a given ONO stack.

➢ **5th term: narrow width effect.**

The term X in equation 1.1 accounts for the well known narrow width and short channel effects. In fact when dealing with narrow and short transistors the V_t is few hundred mV lower than the theoretical value (depending on technology parameters, like STI shape, boron segregation etc..). For the IdVg of figure 1, we have to choose X=0.4V to account for the narrow width effects.

1.3. Electric field calculation

In this part, we calculate the electric field in each layer of the ONO stack:

- E_{topox} is the electric field in the top oxide.
- E_{botox} is the electric field in the bottom oxide.
- E_N is the electric field in the nitride.

The control gate voltage over the ONO stack is given by:

$$V_{cg} = E_{topox} d_{topox} + E_{botox} d_{botox} + \int_0^{d_N} E_N(x)dx + \Delta V_{bulk} + \phi_{MS} \text{ , (1.6)}$$

Where ΔV_{bulk} is the voltage drop in the silicon. E_N is accessible through the Poisson equation. The electric potential ψ_N in the nitride is given by:

$$\frac{\partial^2 \psi_N}{\partial x^2} = -\frac{\rho_N}{\varepsilon_N \varepsilon_0} \text{ , (1.7)}$$

By integrating the Poisson equation (1.7) for the potential in the nitride layer, we yield the electric field in the nitride layer E_N:

$$E_N(x) = -\frac{\partial \psi_N}{\partial x} \text{ , (1.8)}$$

Hence:

$$E_N(x) = \frac{1}{\varepsilon_N \varepsilon_0} \int_0^x \rho_N(x)dx + E_N(x=0) \text{ , (1.9)}$$

Now, we can use the continuity equations at the nitride interface:

- at the interface between the bottom oxide and the nitride:

$$\varepsilon_{botox} E_{botox} = \varepsilon_N E_N(x=0) \text{ , (1.10)}$$

- at the interface between the top oxide and the nitride:

$$\varepsilon_{topox} E_{topox} = \varepsilon_N E_N(x=d_{topox}) \text{ , (1.11)}$$

Hence, by combining equations (1.9) and (1.10), we yield:

$$E_N(x=0) = \frac{\varepsilon_{botox}}{\varepsilon_N} E_{botox} \text{ , (1.12)}$$

Thus, we plug equation (1.9) in equation (1.6), and we yield:

$$V_{cg} = E_{topox} d_{topox} + E_{botox} d_{botox} + \Delta V_{bulk} + \phi_{MS} + \int_0^{d_N} \left[\frac{1}{\varepsilon_N \varepsilon_0} \int_0^x \rho_N(x) dx + \frac{\varepsilon_{botox}}{\varepsilon_N} E_{botox} \right] \cdot dx \,, \tag{1.13}$$

i.e.

$$V_{cg} = E_{topox} d_{topox} + E_{botox} d_{botox} + \Delta V_{bulk} + \phi_{MS} + \frac{1}{\varepsilon_N \varepsilon_0} \int_0^{d_N} \left(\int_0^x \rho_N(x) dx \right) dx + \frac{\varepsilon_{botox}}{\varepsilon_N} E_{botox} d_N \,, \tag{1.14}$$

Where ΔV_{bulk} is taken $(2 \cdot \psi_B$+0.2V-X) for programming case and -(kT/q)Ln($P_{interface}/P_{bulk}$)~-0.21V for erasing case. In the programming case the value of 0.2V was added to the $2 \cdot \psi_B$ to account for strong inversion (normally the onset of inversion is described by $2 \cdot \psi_B$ but in case of programming we are in strong inversion which adds ~0.2V to the voltage drop in the silicon according to device simulation performed with Medici) and X was subtracted to account for narrow width and short channel effects as described earlier. In case of erasing (Vcg<0) ΔV_{bulk} is negative and it accounts for the voltage drop in the silicon due to the difference in hole concentration at the interface $P_{interface}$ and deep in the bulk P_{bulk} which is ~-0.21V according to Medici.

From the continuity equation (1.11) at the nitride and top oxide interface, we get:

$$\varepsilon_{topox} E_{topox} = \varepsilon_N \left[\frac{1}{\varepsilon_N \varepsilon_0} \int_0^{x=d_N} \rho_N(x) dx + \frac{\varepsilon_{botox}}{\varepsilon_N} E_{botox} \right], \tag{1.15}$$

i.e.

$$\varepsilon_{topox} E_{topox} - \varepsilon_{botox} E_{botox} = \frac{1}{\varepsilon_0} \int_0^{x=d_N} \rho_N(x) dx \,, \tag{1.16}$$

Therefore, in order to solve for E_{topox} and E_{botox}, we must solve the following system:

$$\begin{cases} E_{topox}d_{topox} + E_{botox}\left(d_{botox} + \dfrac{\varepsilon_{botox}}{\varepsilon_N}d_N\right) = V_{cg} - \psi_S - \dfrac{1}{\varepsilon_N\varepsilon_0}\displaystyle\int_0^{d_N}\left(\int_0^x \rho_N(x)dx\right)dx \\ \varepsilon_{topox}E_{topox} - \varepsilon_{botox}E_{botox} = \dfrac{1}{\varepsilon_0}\displaystyle\int_0^{x=d_N}\rho_N(x)dx \end{cases}, \quad (1.17)$$

1.4. Calculation of the injected currents in the nitride

Depending on the control gate voltage applied (program or erase), we must take into account different injection mechanisms.

We will consider the three following injection mechanisms through the oxide [Bach2001]:

- **Direct tunneling** (DT) occurs in the bottom oxide for $\dfrac{\phi_1 - \phi_2}{d_{botox}} \le E_{botox} \le \dfrac{\phi_1}{d_{botox}}$, where ϕ_1 and ϕ_2 are the bottom oxide and the nitride tunneling barriers respectively [V], as shown in the energy band diagrams in Fig. 2 and 3.
  The Direct tunneling current density through the bottom oxide is given by:

$$J_{DT} = \frac{m_0}{m_{ox}}\frac{q^3}{16\pi^2\hbar}E_{botox}^2\frac{1}{\left[\sqrt{q\phi_1} - \sqrt{q\phi_1 - qE_{botox}d_{botox}}\right]^2} \times \exp\left[-\frac{4\sqrt{2m_{ox}}\left(\sqrt[3]{q\phi_1} - \sqrt[3]{q\phi_1 - qE_{botox}d_{botox}}\right)}{3q\hbar E_{botox}}\right], \quad (1.22)$$

- **Modified Fowler – Nordheim tunneling** (MFN) occurs in the bottom oxide for $\dfrac{\phi_1 - \phi_2}{d_{botox} - \gamma d_N} \le E_{botox} \le \dfrac{\phi_1 - \phi_2}{d_{botox}}$, where $\gamma = \dfrac{\varepsilon_N}{\varepsilon_{botox}}$,

  The Modified Fowler – Nordheim tunneling current density through the bottom oxide is given by:

$$J_{MFN} = \frac{m_0}{m_{ox}}\frac{q^3}{16\pi^2\hbar}E_{botox}^2\frac{1}{\left[\sqrt{q\phi_1} - \sqrt{q\phi_1 - qE_{botox}d_{botox}} + \gamma\sqrt{\frac{m_N}{m_{ox}}}\sqrt{q\phi_1 - q\phi_2 - qE_{botox}d_{botox}}\right]^2} \times \exp\left[-\frac{4\sqrt{2m_{ox}}\left(\sqrt[3]{q\phi_1} - \sqrt[3]{q\phi_1 - qE_{botox}d_{botox}}\right) + 4\gamma\sqrt{2m_N}\sqrt[3]{q\phi_1 - q\phi_2 - qE_{botox}d_{botox}}}{3q\hbar E_{botox}}\right], \quad (1.21)$$

- **Fowler – Nordheim tunneling** (FN) occurs in the top oxide for $E_{topox} \geq \frac{\phi_3}{d_{topox}}$.

  Then the tunneling current density through the top oxide is given by:

$$J_{FN} = A_{FN} E_{topox}^2 \exp\left(-\frac{B_{FN}}{E_{topox}}\right), \quad (1.18)$$

  Where A_{FN} and B_{FN}, are the pre-exponential and exponential constants given by:

$$A_{FN} = 10 \cdot 6.32e-7 \cdot \frac{3.1}{\phi_3}, \quad (1.19)$$

$$B_{FN} = 2.4e8 \cdot \sqrt[3]{\frac{\phi_3}{3.1}}, \quad (1.20)$$

  Where ϕ_3 is the top oxide tunneling barrier [V].

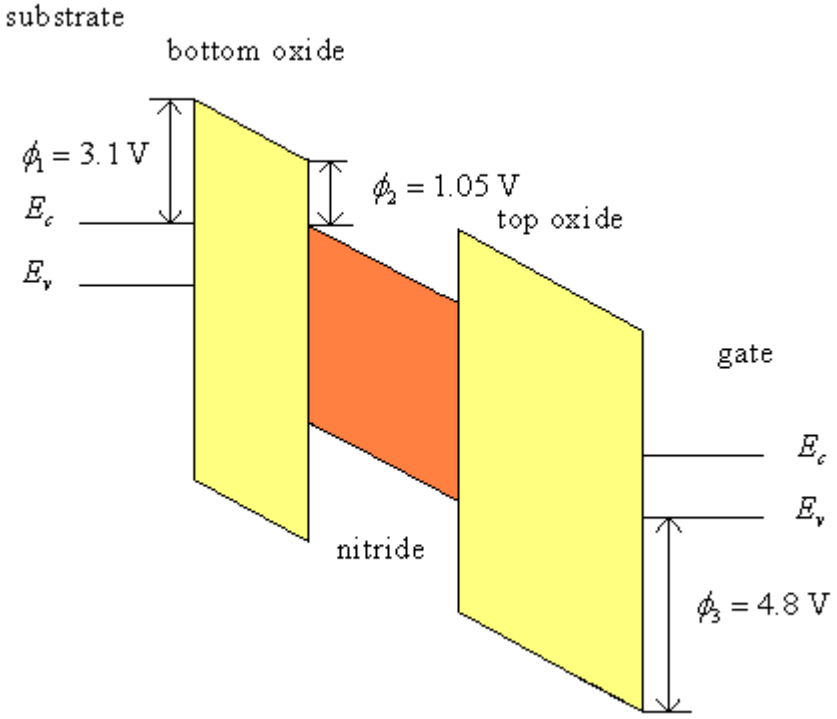

Figure 2: Energy band diagram for the programming case V_{cg} >0

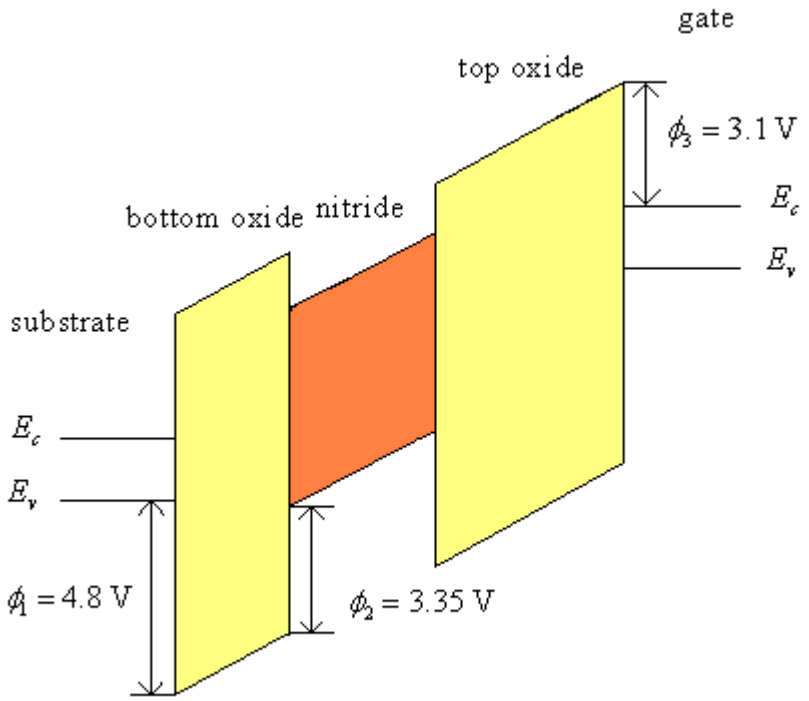

Figure 3: Energy band diagram for the erasing case V_{cg} <0

The importance of these conduction mechanisms in the oxide depends on the applied electric field or voltage, on the top and bottom oxide thickness, and on the barriers height.

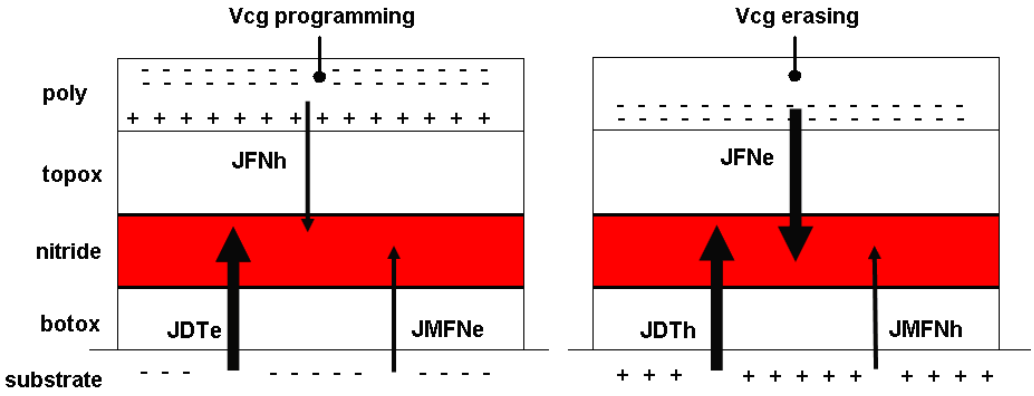

Figure 4: Dependence on the applied control gate voltage of the currents injected into the nitride

- **Under a positive control gate voltage (programming)**, Direct (respectively Modified Fowler – Nordheim) tunneling electron injection from the semiconductor to the nitride will dominate at high (respectively low) electric field in the bottom oxide.

  Hole injection (described by the Fowler – Nordheim tunneling mechanism) from

the gate is negligible because of the high potential barrier for holes in the oxide ($\phi_1 = 4.8$ V).

- **Under a negative control gate voltage (erasing),** two mechanisms compete with each other in the bottom and top oxide.

  Holes are injected from the substrate to the nitride through the bottom oxide. This injection is described by the Direct tunneling mechanism at high electric field, and by Modified Fowler – Nordheim tunneling mechanism at low electric field. In this work, we neglect the electrons back tunneling from the nitride to the substrate.

  On the other side of the stack, electrons are injected from the gate to the nitride through the top oxide. This Fowler – Nordheim electron injection is easily performed, due to the low potential barrier ($\phi_3 = 3.1$ V).

In both cases, we will neglect the Poole-Frenkel conduction of electrons through the nitride.

Knowing these carrier injection mechanisms, we yield the current density values in the top and the bottom oxide at the nitride interface.

We can achieve the current density distribution within the nitride, by solving the following first order differential equation for electrons and holes, respectively:

$$\begin{cases} \dfrac{\partial J_{Ne}(x,t)}{\partial x} = -J_{Ne}(x,t)\cdot\sigma_t\cdot\left[N_t - n_{te}(x,t)\right] \\ \dfrac{\partial J_{Nh}(x,t)}{\partial x} = -J_{Nh}(x,t)\cdot\sigma_t\cdot\left[N_t - n_{th}(x,t)\right] \end{cases}, \quad (1.23)$$

Where J_{Ne} and J_{Nh} respectively denote the electron and hole current distribution in the nitride.

The total number of traps in the nitride N_t is given by:

$$N_t = n_{tf} + n_{te} + n_{th}, \quad (1.24)$$

where n_{tf} denotes the number of empty traps within the nitride, and σ_t, the trap capture cross section which is taken $\sigma_t = 1e^{-13}$ /cm².

Once the current density distribution for holes and electrons is known within the nitride, we can proceed to the carrier density calculation in the nitride.

1.5. Calculation/Update of the trap density in the silicone nitride

The trap density distribution at time $t + dt$ can be reached from the distribution at time t, thanks to the following continuity equations:

$$\begin{cases} n_{te}(x,t+dt) = n_{te}(x,t) + \dfrac{\partial n_{te}(x,t)}{\partial t}\Delta t \\ n_{th}(x,t+dt) = n_{th}(x,t) + \dfrac{\partial n_{th}(x,t)}{\partial t}\Delta t \\ n_{tf}(x,t+dt) = N_t - n_{te}(x,t+dt) - n_{th}(x,t+dt) \end{cases} \quad (1.25)$$

Where n_{te}, n_{th}, and n_{tf} respectively denote the density of traps filled with electrons, holes, and the density of empty traps [1/ cm³].

In equation (1.25), $\dfrac{\partial n_{te}(x,t)}{\partial t}\Delta t$ (respectively $\dfrac{\partial n_{th}(x,t)}{\partial t}\Delta t$) denotes the amount of traps which got filled with electrons (respectively holes) within the elementary time Δt. This amount is also equal to $\dfrac{1}{q}\cdot\dfrac{J_{Ne}(x+dx,t)-J_{Ne}(x,t)}{dx}\Delta t$, ie. to $\dfrac{1}{q}\dfrac{\partial J_{Ne}(x,t)}{\partial x}\Delta t$.

Therefore equation (1.25) is equivalent to:

$$\begin{cases} n_{te}(x,t+dt) = n_{te}(x,t) + \dfrac{1}{q}\dfrac{\partial J_{Ne}(x,t)}{\partial x}\Delta t \\ n_{th}(x,t+dt) = n_{th}(x,t) + \dfrac{1}{q}\dfrac{\partial J_{Nh}(x,t)}{\partial x}\Delta t \\ n_{tf}(x,t+dt) = N_t - n_{te}(x,t+dt) - n_{th}(x,t+dt) \end{cases} , (1.26)$$

By plugging the first order differential equation giving the current distribution for electrons and holes into equation (1.26), we yield:

$$\begin{cases} n_{te}(x,t+dt) = n_{te}(x,t) + \dfrac{1}{q}J_{Ne}(x,t)\cdot\sigma_t\cdot\left[N_t - n_{te}(x,t)\right]\Delta t \\ n_{th}(x,t+dt) = n_{th}(x,t) + \dfrac{1}{q}J_{Nh}(x,t)\cdot\sigma_t\cdot\left[N_t - n_{th}(x,t)\right]\Delta t \\ n_{tf}(x,t+dt) = N_t - n_{te}(x,t+dt) - n_{th}(x,t+dt) \end{cases} , (1.27)$$

In order to perform a correct program/erase simulation of a SONOS device, we need to divide the trap density calculation into two steps. Inside these two steps, we use intermediate times t_1, t_{11}, t_2, and t_{22}, which have no physical meaning, and denote the order of the update within time Δt.

Programming case

- *1st step: t_{11} and t_1*

  In equation (1.27), we update n_{te}, n_{th}, and n_{tf} with the new values for J_{Ne} and J_{Nh}.

  For a given position x in the nitride and a given time t, we first update the

amount of new traps filled with electrons due to electron injection in the nitride. This can be calculated by multiplying the amount of electrons at position x in the nitride by the amount of available traps at this position, and by the trap capture cross section, as shown in equation (1.28).

$$n_{te}(x,t_1) = n_{te}(x,t) + \frac{1}{q} J_{Ne}(x,t) \cdot \sigma_t \cdot \left[n_{th}(x,t) + n_{tf}(x,t) \right] \cdot \frac{n_{tf}(x,t)}{n_{th}(x,t) + n_{tf}(x,t)} \Delta t \text{ , (1.28)}$$

Then, we update the free traps density in the nitride:

$$n_{tf}(x,t_{11}) = N_t - n_{te}(x,t_1) - n_{th}(x,t) \text{ , (1.29)}$$

Now, we must subtract to the density of traps filled with holes the amount of traps filled with holes at time t, which have been filled with electrons injected to the nitride at time t_1. We then yield the hole density at time t_1:

$$n_{th}(x,t_1) = n_{th}(x,t) - \frac{1}{q} J_{Ne}(x,t) \cdot \sigma_t \cdot \left[n_{th}(x,t) + n_{tf}(x,t_{11}) \right] \cdot \frac{n_{th}(x,t)}{n_{th}(x,t) + n_{tf}(x,t_{11})} \Delta t \text{ , (1.30)}$$

Finally, we update a second time the free traps density in the nitride:

$$n_{tf}(x,t_1) = N_t - n_{te}(x,t_1) - n_{th}(x,t_1) \text{ , (1.31)}$$

- *2nd step:* t_{22} *and* t_2

  We first update the density of traps filled with holes by adding to the amount of traps filled with holes in the nitride at time t_1 the amount of traps filled with electrons injected from to the nitride at time t_2:

$$n_{th}(x,t_2) = n_{th}(x,t_1) + \frac{1}{q} J_{Nh}(x,t) \cdot \sigma_t \cdot \left[n_{te}(x,t_1) + n_{tf}(x,t_1) \right] \cdot \frac{n_{tf}(x,t_1)}{n_{te}(x,t_1) + n_{tf}(x,t_1)} \Delta t \text{ , (1.32)}$$

  Then, we update the free traps density in the nitride:

$$n_{tf}(x,t_{22}) = N_t - n_{te}(x,t_2) - n_{th}(x,t_1) \text{ , (1.33)}$$

  Now, we must subtract to the density of traps filled with electrons the amount of traps that was originally filled with electrons at time t, which have been filled with holes injected from the gate at time t_2. We then yield the density of traps filled with holes at time t_2:

$$n_{te}(x,t_2) = n_{te}(x,t_1) - \frac{1}{q} J_{Nh}(x,t) \cdot \sigma_t \cdot \left[n_{te}(x,t_2) + n_{tf}(x,t_{22}) \right] \cdot \frac{n_{te}(x,t_{22})}{n_{te}(x,t_1) + n_{tf}(x,t_{22})} \Delta t \text{ , (1.34)}$$

  Finally, we update a second time the free traps density in the nitride:

$$n_{tf}(x,t_2) = N_t - n_{te}(x,t_2) - n_{th}(x,t_2) \text{ , (1.35)}$$

Erasing case

- *1st step:* t_{11} *and* t_1

  We first update the density of traps filled with holes by adding the amount of traps

filled with holes injected to the nitride at time t , to the initial amount of traps filled with holes in the nitride:

$$n_{th}(x,t_1) = n_{th}(x,t) + \frac{1}{q} J_{Nh}(x,t) \cdot \sigma_t \cdot \left[n_{te}(x,t) + n_{tf}(x,t) \right] \cdot \frac{n_{tf}(x,t)}{n_{te}(x,t) + n_{tf}(x,t_1)} \Delta t \text{ , (1.36)}$$

Then, we update the free traps density in the nitride:

$$n_{tf}(x,t_{11}) = N_t - n_{te}(x,t) - n_{th}(x,t_1) \text{ , (1.37)}$$

Now, we must subtract to the density of traps filled with electrons the amount of traps filled with electrons at time t which have been filled with holes injected to the nitride at time t_1. We then yield the density of traps filled with electron at time t_1:

$$n_{te}(x,t_1) = n_{te}(x,t) - \frac{1}{q} J_{Nh}(x,t) \cdot \sigma_t \cdot \left[n_{te}(x,t) + n_{tf}(x,t_{11}) \right] \cdot \frac{n_{te}(x,t_{11})}{n_{te}(x,t_1) + n_{tf}(x,t_{11})} \Delta t \text{ , (1.38)}$$

Finally, we update a second time the free traps density in the nitride:

$$n_{tf}(x,t_1) = N_t - n_{te}(x,t_1) - n_{th}(x,t_1) \text{ , (1.39)}$$

- *2nd step:* t_{22} *and* t_2

  We first update the density of traps filled with electrons by adding the amount of traps filled with electrons injected to the nitride at time t_2, to the amount of traps filled with electrons in the nitride at time t_1 :

$$n_{te}(x,t_2) = n_{te}(x,t_1) + \frac{1}{q} J_{Ne}(x,t) \cdot \sigma_t \cdot \left[n_{th}(x,t_1) + n_{tf}(x,t_1) \right] \cdot \frac{n_{tf}(x,t_1)}{n_{th}(x,t_1) + n_{tf}(x,t_1)} \Delta t \text{ , (1.40)}$$

  Then, we update the free traps density in the nitride:

$$n_{tf}(x,t_{22}) = N_t - n_{te}(x,t_2) - n_{th}(x,t_1) \text{ , (1.41)}$$

  Now, we must subtract to the hole density the amount of traps that was originally filled with holes at time t, which have been filled with electrons injected from the gate at time t_2. We then yield the hole density at time t_2 :

$$n_{th}(x,t_2) = n_{th}(x,t_1) - \frac{1}{q} J_{Ne}(x,t) \cdot \sigma_t \cdot \left[n_{th}(x,t_2) + n_{tf}(x,t_{22}) \right] \cdot \frac{n_{th}(x,t_{22})}{n_{th}(x,t_2) + n_{tf}(x,t_{22})} \Delta t \text{ , (1.42)}$$

  Finally, we update a second time the free traps density in the nitride:

$$n_{tf}(x,t_2) = N_t - n_{te}(x,t_2) - n_{th}(x,t_2) \text{ , (1.43)}$$

The values of $n_{te}(x,t_2)$, $n_{th}(x,t_2)$, and $n_{tf}(x,t_2)$ correspond to the values at the next time step (time $t + dt$).

Overall, we started from the knowledge of the trap distribution in the nitride at time t, and after solving for the current density and updating the trap distribution, we finally end up with the trap distribution in the nitride at time $t + dt$.

1.6. Charge density calculation

Knowing the trap density distribution in the nitride at the next time step (i.e. at time $t + dt$), we calculate the charge distribution in the silicon nitride ρ_N, by using:

$$\rho_N = q \cdot \left(n_{th} - n_{te} \right), \quad (1.44)$$

We then yield Q_N, the charge per unit area in the silicon nitride, at the next time step. The charge per unit area directly depends on ρ_N, through the following formula:

$$Q_N = \int_0^{d_N} \rho_N(x) dx, \quad (1.45)$$

1.7. Transition from time *t* to time *t+dt*

Once we know Q_N at time $t + dt$, we can calculate the threshold voltage at time $t + dt$ and perform the same five step procedure we have just described for time $t + dt$.

By repeating this procedure over and over again, we finally get the value of the threshold voltage for any time t. Thus, we can draw the erasing or programming curves.

CHAPTER 2 –MODELING RESULTS FOR THE REFERENCE WAFER (2.2/6/8nm ONO N-type gate)

In this chapter, we first discuss the fitting parameters we had to tune to reach good agreement between the model and measurement data. Then, we discuss other physical quantities of importance for the model. Finally, we present the results of the model for programming and erasing and compare them with measurements on the reference wafer, which has 2.2nm thermal oxide, 6nm nitride, and 8nm HTO ONO stack with N-type gate.

2.1. Fitting parameters setting

In order to perform a good fitting of the $V_t(t)$ measurement curves of the reference SONOS stack (2.2/6/8nm N-type gate), we choose the following parameters setting in the simulation program.

Parameters	Definition	Values	
A'_{FN}	Correction coefficient of the pre-exponential factor of the Fowler-Nordheim current equation	6e-2	
B'_{FN}	Correction coefficient of the exponential factor of the Fowler-Nordheim current equation	0.95	
m_N	Carrier effective mass in the nitride	Program (electrons)	$0.1 \cdot m_0$
		Erase (holes)	$0.4 \cdot m_0$
m_{ox}	Carrier effective mass in the oxide	Program (electrons)	$0.32 \cdot m_0 \cdot \left(\frac{1e7}{E_{botox}} \right)^{1.25}$
		Erase (holes)	$0.325 \cdot m_0 \sqrt{\frac{1e7}{E_{botox}}}$
μ_e	Electron semiconductor mobility	130 cm²/V/s	
σ_t	Trap capture cross section in the nitride	1e-13/ cm²	
ϕ_1	Bottom oxide tunneling barrier	For electrons	3.1 V
		For holes	4.8 V
ϕ_2	Nitride tunneling barrier	For electrons	1.05 V
		For holes	3.35 V
ϕ_3	Top oxide tunneling barrier	For holes	4.8 V
		For electrons	3.1 V

Table 1: Fitting parameters values for the reference SONOS stack (2.2/6/8nm N-type gate)

2.1.1. Coefficients of the Fowler-Nordheim current

$A_{FN}^{'}$ and $B_{FN}^{'}$ are the correction coefficients of the pre-exponential (A_{FN}) and exponential (B_{FN}) of the Fowler-Nordheim current equation. They matter for the erasing case, when Fowler-Nordheim injection is relevant. Equations (1.18) and (1.19) are now replaced by [Bach2001]:

$$A_{FN} = A_{FN}^{'} \cdot 10 \cdot 6.32 \cdot (1e-7) \cdot \frac{3.1}{\phi_3}, \quad (2.1)$$

$$B_{FN} = B_{FN}^{'} \cdot 2.4(1e8) \cdot \sqrt[3]{\frac{\phi_3}{3.1}}, \quad (2.2)$$

These correction coefficients affect the injected current density from the gate into the top oxide to the nitride, as shown in Figure 5. An increase in A_{FN} shifts the J_{FN} vs. E_{topox} curve up, while a decrease in B_{FN} decreases the slope of the curve for high E_{topox}.

Therefore, by playing on $A_{FN}^{'}$ and $B_{FN}^{'}$, we can change the location of the Fowler-Nordheim current curve with respect to the Direct tunneling curve's. Thus, we can adjust the erase saturation, which happens when the currents in the bottom and the top oxide become comparable.

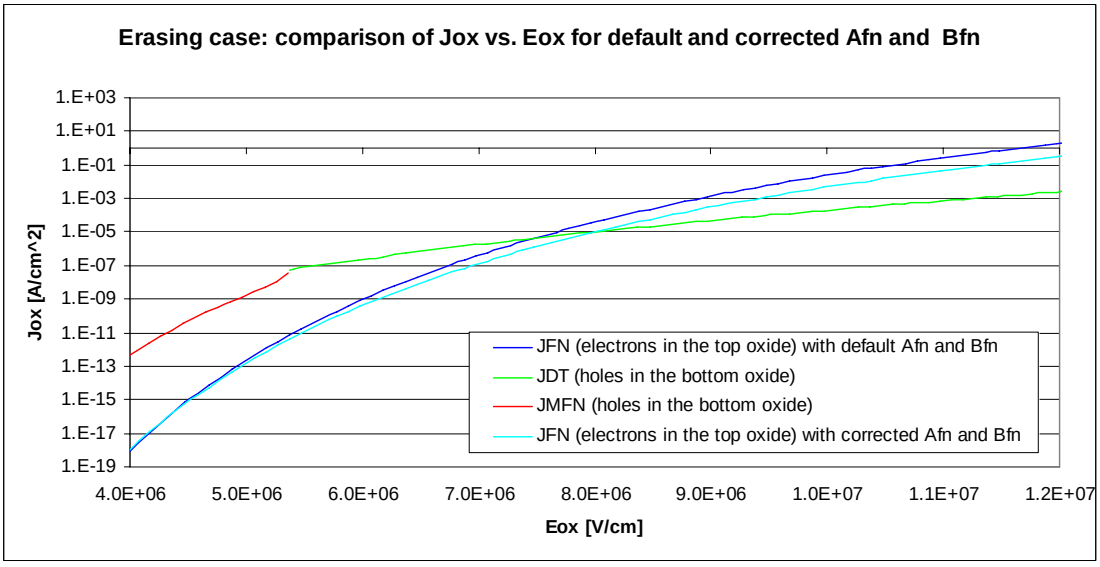

Figure 5: Current density vs. electric field in the oxide for $A_{FN}^{'}$ = $B_{FN}^{'}$ =1 (default values) and for $A_{FN}^{'}$ =6e-2 and $B_{FN}^{'}$ =0.95.

2.1.2. Mobility and current criterion

The values for I_D and V_D in the model should be chosen according to the values chosen during measurements. However, β should be extracted from measured I_D vs.

V_g curve for a given ONO stack, using the following equation:

$$\beta = \frac{W}{L}\mu_e C_{eff}\,, \quad (2.3)$$

Where μ_e is used as a fitting parameter and set to the appropriate value (see Table 1).

2.1.3.Carrier effective mass

The carrier (electron in case of programming and hole in case of erasing) effective mass in the nitride and in the oxide (respectively m_N and m_{ox}) are also used as fitting parameters.

The carrier effective mass in the nitride m_N affects the Modified Fowler-Nordheim current density. An increase in m_N decreases the slope of the J_{MFN} vs. E_{botox} curve. An increase in m_{ox} will shift the entire J_{MFN} and J_{DT} vs. E_{botox} curve down, as shown in Figure 6. The carrier effective mass in the oxide m_{ox} is taken equal to $0.32 \cdot m_0 \cdot \left(\frac{1e7}{E_{botox}}\right)^{1.25}$ for program (electron), and to $0.325 \cdot m_0 \sqrt{\frac{1e7}{E_{botox}}}$ for erase (holes). These formulas are used to soften the direct tunneling current at low electric field in the bottom oxide for better fitting of the program/erase curves. This suggests that for our devices the pre-exponential factor in the direct tunneling equation (1.22) depends on $(E_{botox})^{0.75}$ or $(E_{botox})^{1.5}$ in case of programming or erasing respectively and not on $(E_{botox})^2$, like in reference [Bach2001].

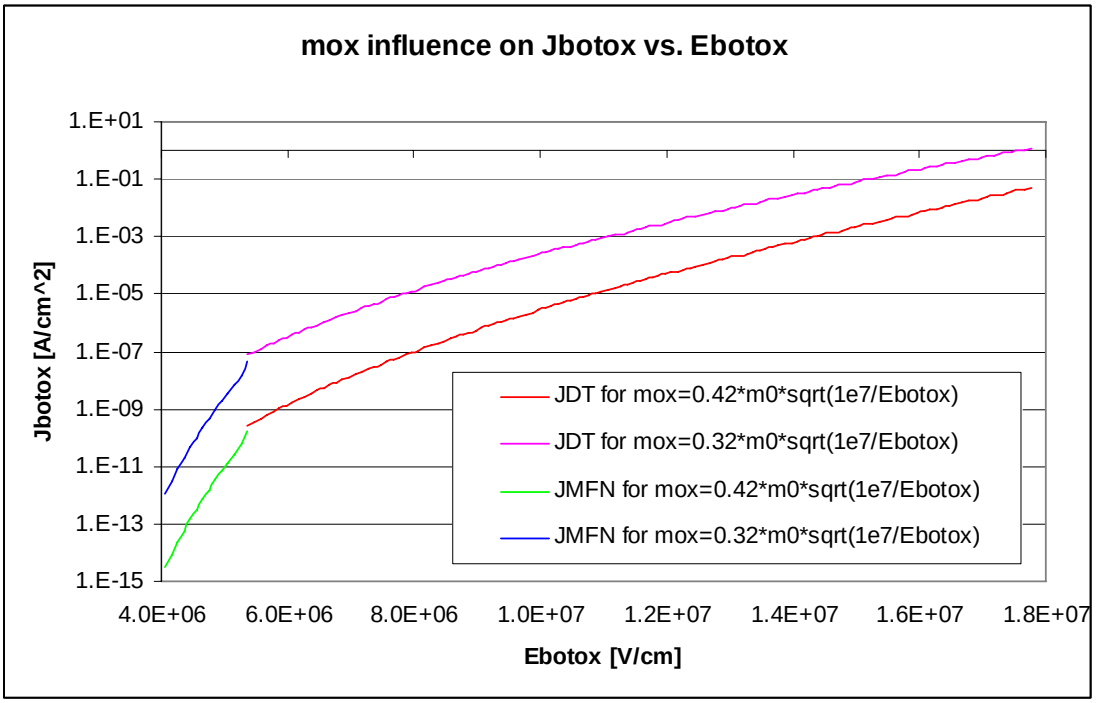

Figure 6: m_{ox} ***influence on*** J_{MFN} ***and*** J_{DT} ***vs.*** E_{botox}

Thanks to these fitting parameters, we can adjust the position of the J_{botox} vs. E_{botox} curve with respect to the J_{FN} vs. E_{topox}. Thus, we can also adjust the erase saturation.

2.1.4. Trap capture cross section in the nitride

The trap capture cross section in the nitride σ_t is taken to a smaller value than in publications [Bach2001] (1e-13 /cm² instead of 1e-14 /cm²), for better fitting of the program/erase curves. This parameter influences the slope of the injected currents in the nitride. For high values of σ_t, the injected current in the bottom oxide was not reaching the top oxide (similar for injected currents in the top oxide) and this results in erasing the bottom part of the nitride while programming the top part.

2.1.5. Oxide and nitride tunneling barriers

Finally, the oxide tunneling barriers are taken 3.1 V and 4.8 V for electron and holes respectively.
We set the nitride tunneling barrier for holes to 3.35V (instead of 2.85V), so that the Direct tunneling current is preponderant for a larger window of electric field values, as shown in Figure 7.

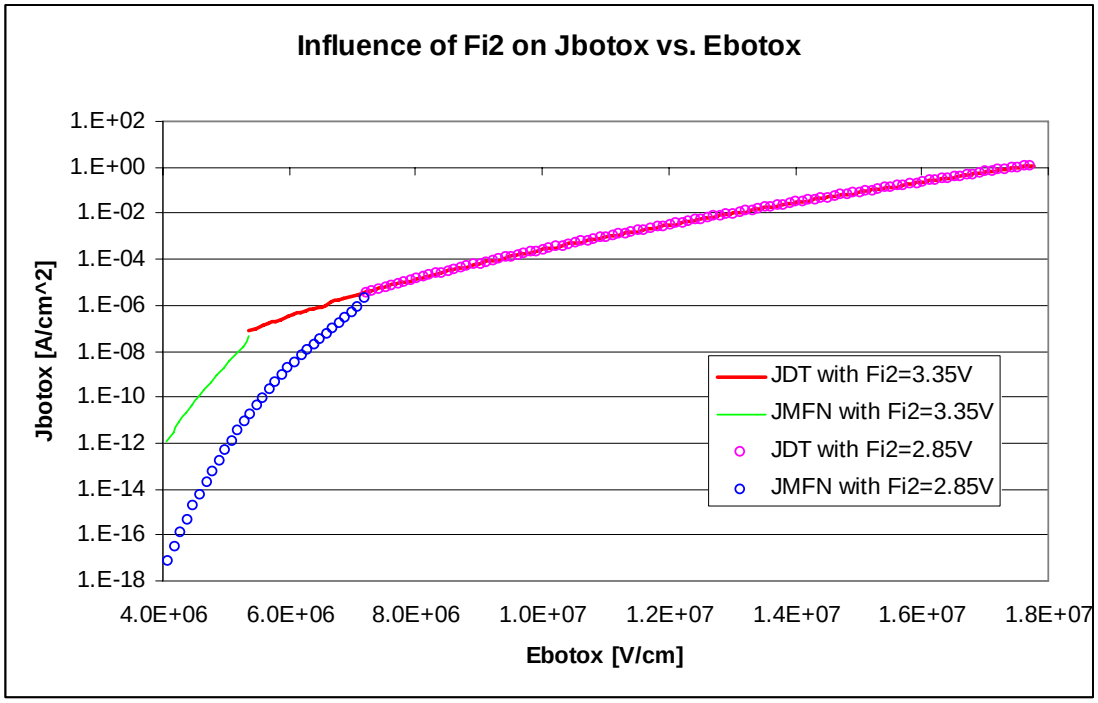

Figure 7: Influence of ϕ_2 on J_{MFN} and J_{DT} vs. E_{botox}

The nitride tunneling barrier for electrons is 1.05 V.

2.2 Other physical quantities which are important for the model

In this sub-section, we explain the meaning of the values taken for some important physical quantities listed in Table 2.

Parameters	Definition	Values	
d_{botox}	Bottom oxide thickness	2.7 nm	
I_D	Drain current	10 µA	
N_t	Total trap density in the nitride	5e19 /cm³	
$n_{te}(t=0)$	Density of traps filled with electrons in the nitride at time zero	Program	0
		Erase	1.15e19 /cm³
$n_{th}(t=0)$	Density of traps filled with holes in the nitride at time zero	Program	0.17e19 /cm³
		Erase	0
V_D	Drain voltage	0.5 V	

Table 2: Values of some important quantities for the modeling of the reference stack

2.2.1. Bottom oxide thickness

In order to account for the quantum mechanical effects, we assumed that the inversion or accumulation layers are -5Å below the interface, so that the electric thickness of the bottom oxide we use is equal to the physical thickness +5 Å, e.g. for the reference device we use d_{botox}=2.2nm + 5 Å, i.e. d_{botox}=2.7nm.

2.2.2. Current criterion

Refer to sub-section 2.1.3. Mobility and current criterion.
It should be noted that I_D was taken relatively high in the program as well as in measurements (most often in measurements we use 5 µA instead of 10 µA for a device with W/L=0.18/0.2 µm) in order to be sure that we are not sensitive to any sub-threshold hump in IdVg characteristics that might be present for some devices.

2.2.3. Total trap density

We set the total trap density in the nitride N_t to an arbitrary value, which has no impact on the simulation results. The value chosen for N_t guaranties a programming window of about 7 V when all traps are filled.
However, the density of traps filled with electrons (n_{te}) or holes (n_{th}) in the nitride do matter for the simulation results. Indeed, they affect the threshold voltage value at time zero, when the simulation starts.
We took n_{th}=0.1e19 /cm³ for programming, and n_{te}=1.15e19 /cm³ for erasing in order to adjust the value of V_t at t = 0 to the measurement for the reference stack.

2.3. Programming

In this section we will show the results of the model when programming the reference SONOS device and we will compare the results to measurements.

In order to make the calculations performed by the program more visible for the user, several physical parameters or quantities are plotted next to the V_t at each point in time. In this way, the user can better follow the calculation in order to have deep analysis of the results. First we will show the results of the modeling of V_t, then we will address the physical quantities plotted with the V_t.

Let us first consider the programming case, where a positive voltage is applied to the control gate.
In this case, electron tunneling from the substrate to the nitride is the preponderant injection mechanism, as discussed in Chapter I.

2.3.1. Threshold voltage

We find a good agreement between simulation and measurement data for the threshold voltage curves vs. time measured on 1T mini array (256bits), as shown in Figures 8 to 11. In Figure 12, we see that the threshold voltage shift between two curves with consecutive values of V_{cg} is larger for simulation than for measurement data. However in both cases, this threshold voltage shift is constant, whatever the V_{cg} value. Moreover, we see that the program gives reasonable results for different starting V_t (from virgin where nitride charge is assumed 0, or when the device was erased to lower V_t which is the case of the the programming curves 11, 12 and 13V).

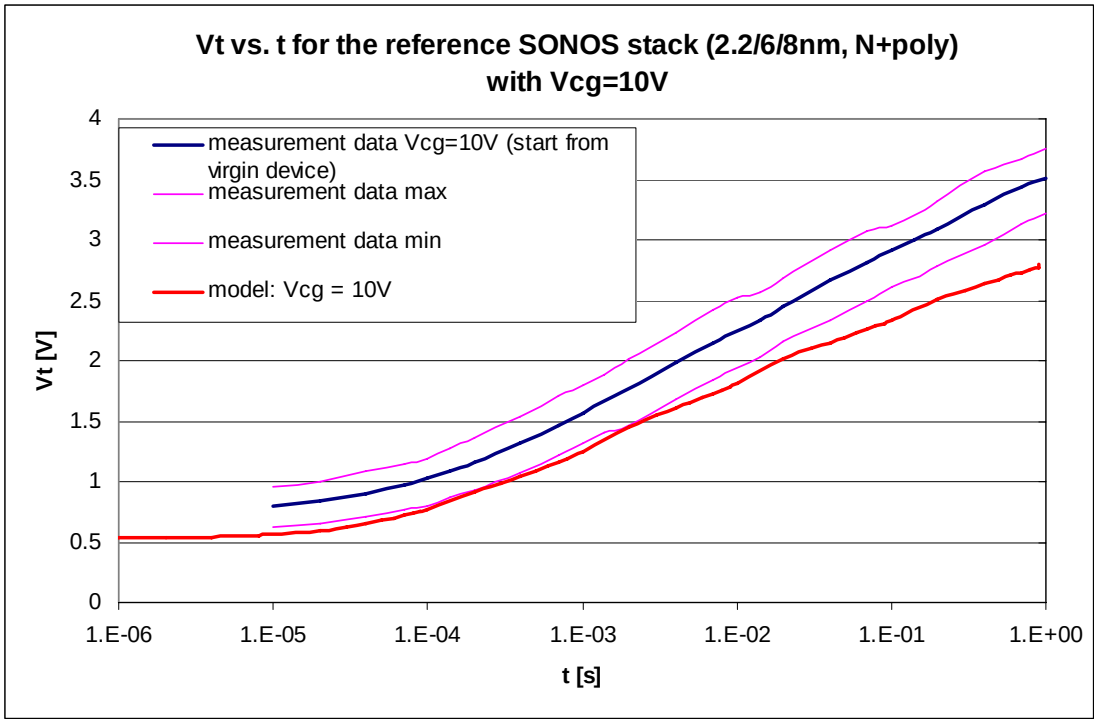

Figure 8: Programming curves for the reference stack with a 10V control gate

voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken zero.

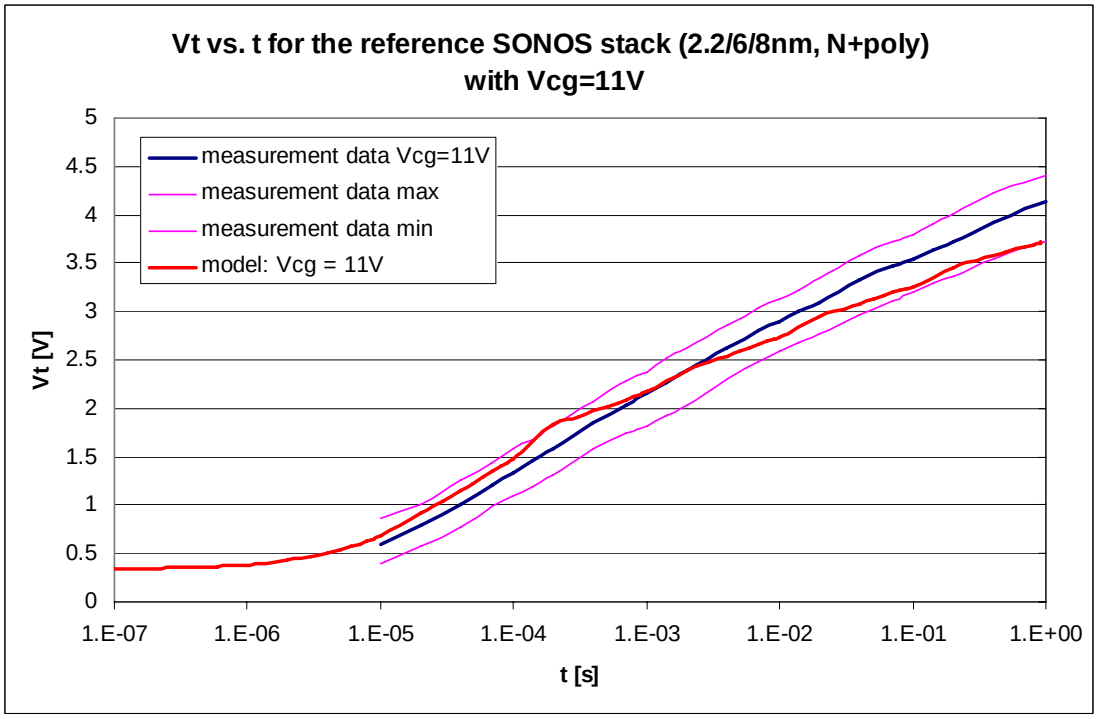

Figure 9: Programming curves for the reference stack with an 11V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.17e^{19}/cm^3$.

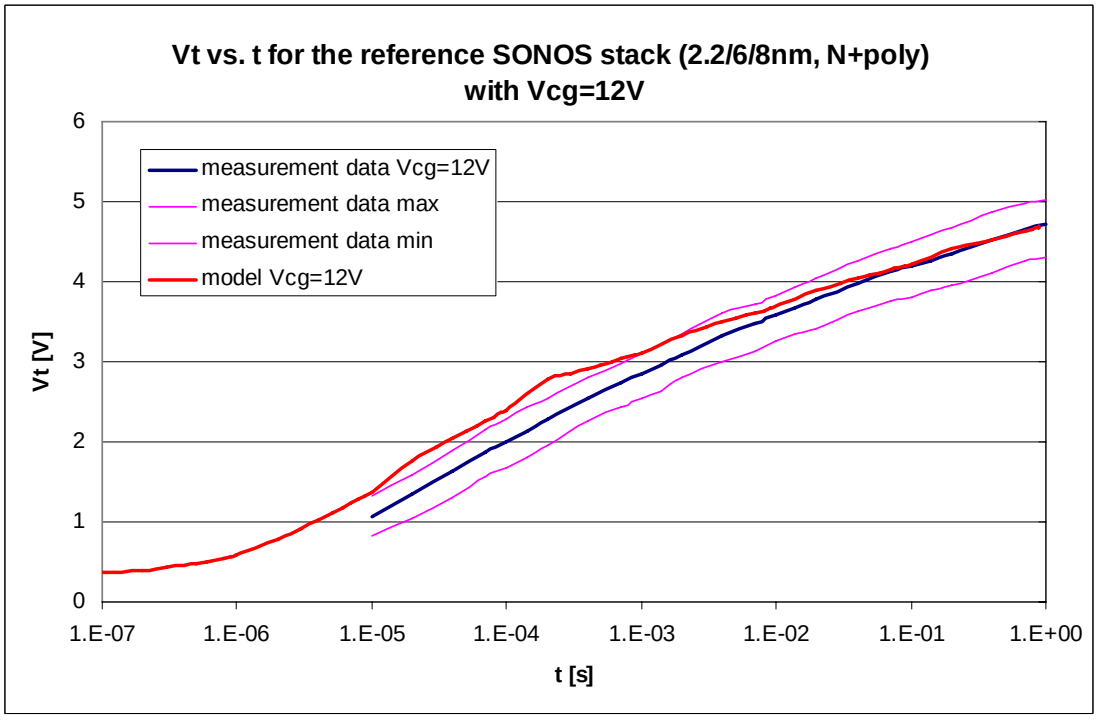

Figure 10: Programming curves for the reference stack with a 12V control gate

voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$.

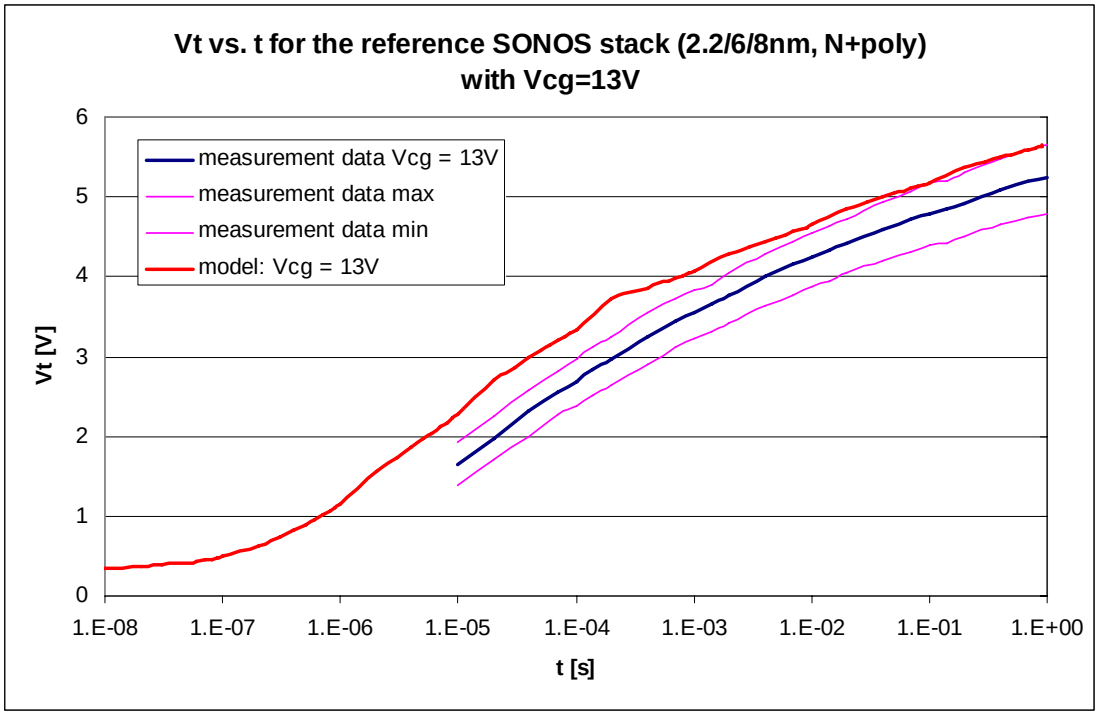

Figure 11: Programming curves for the reference stack with a 13V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.17e^{19}/cm^3$.

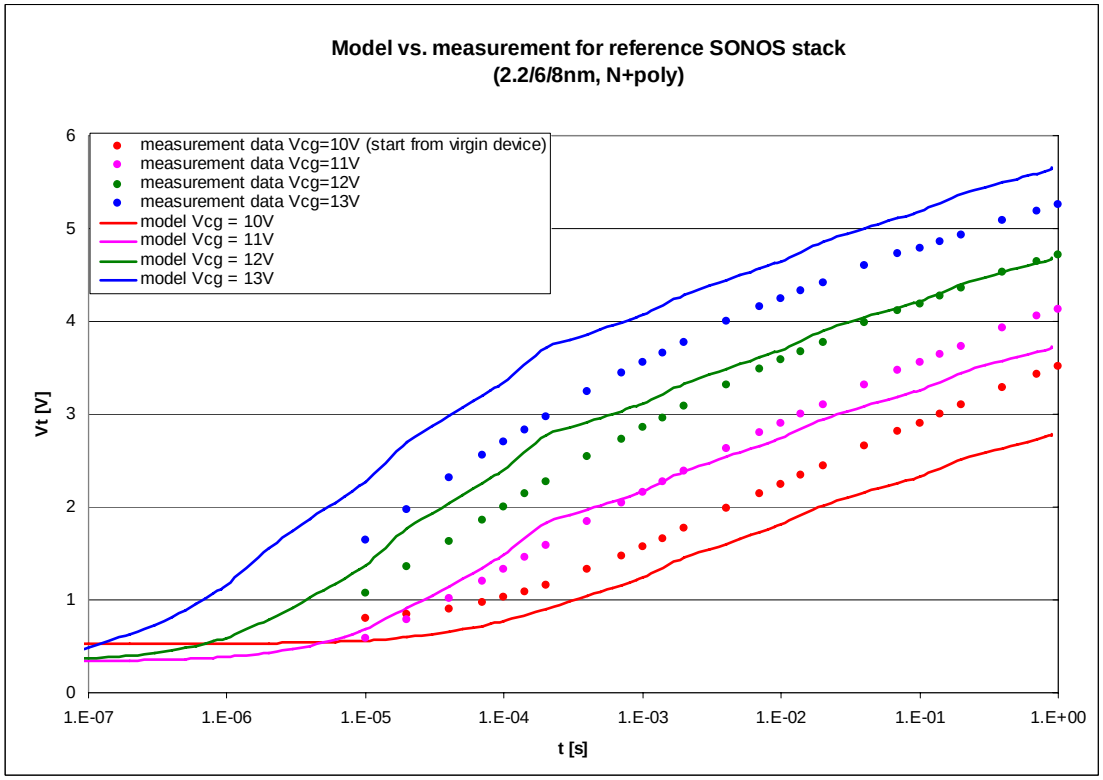

Figure 12: Programming curves comparison between the modeling and measurements for different control gate voltages and different starting V_t. The starting Vt of the programming curves Vcg=11, 12 and 13V was the same.

2.3.2. Electric field through the ONO

The electric fields in the top and bottom oxide are plotted at each point in time as shown in Figure 13. Since we start from a partly erased nitride, E_{botox} is larger than E_{topox} at time zero. Then, E_{botox} decreases and E_{topox} increases, since electrons are injected from the substrate to the nitride.

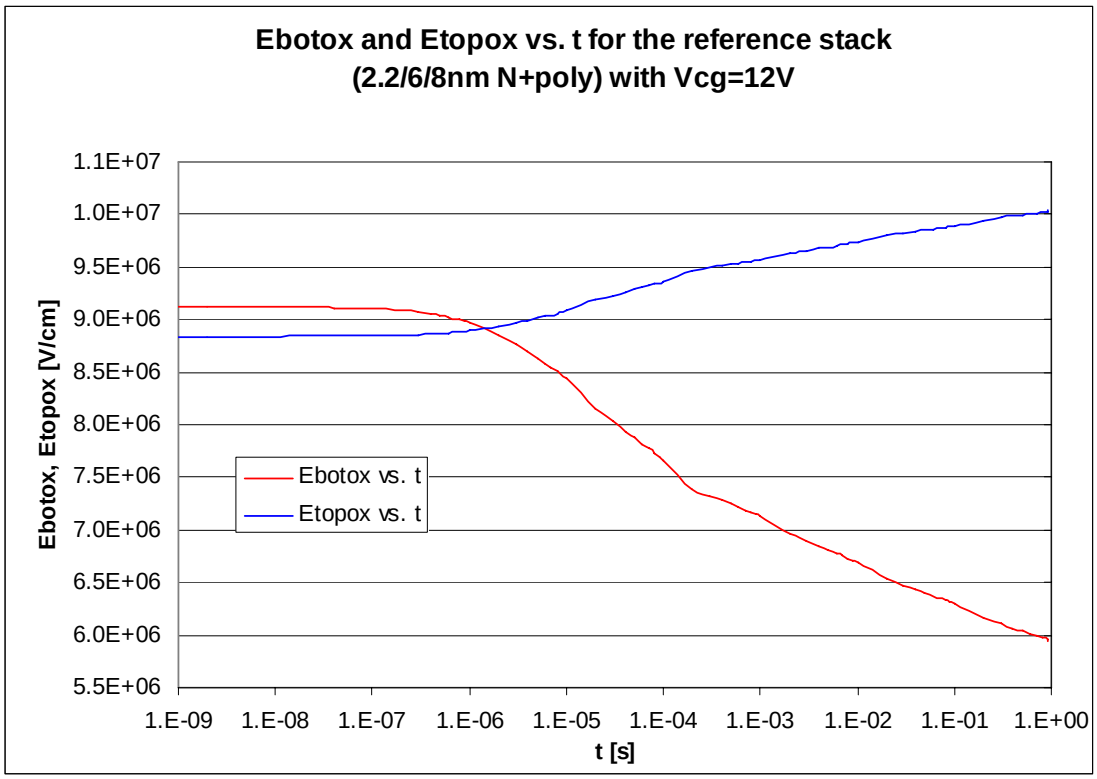

Figure 13: Electric field in the top and bottom oxide vs. time extracted from the program, for a 12V programming control gate voltage

2.2.3. Injected current density in the nitride

Figure 14 and 15 show the profile of the electron and hole currents injected in the nitride (respectively J_{Ne} from the substrate and J_{Nh} from the gate). It can be observed that J_{Nh} is negligible compared to J_{Ne}, since the Fowler-Nordheim tunneling current of holes from the gate is negligible compared to the Direct tunneling current of electrons, due to a large difference in barrier heights. We can see that J_{Ne} is de-

creasing, while J_{Nh} is increasing with time, due to the decrease (increase) of E_{botox} (E_{topox}) respectively with time.

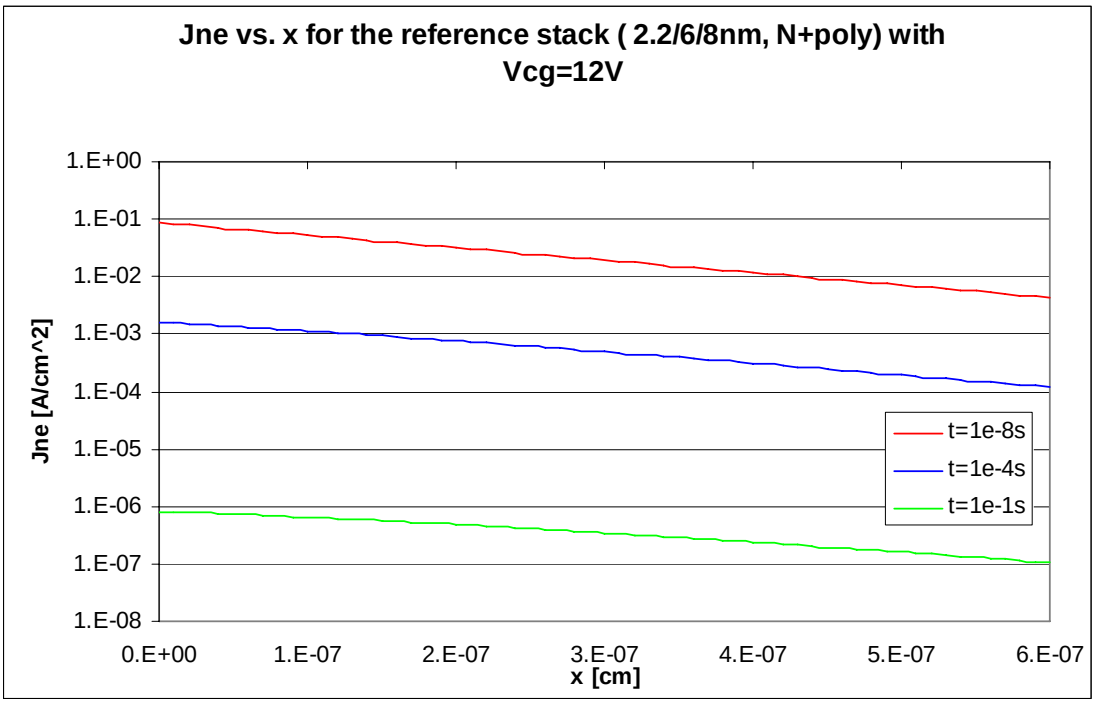

Figure 14: Distribution of the current injected from the bottom oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

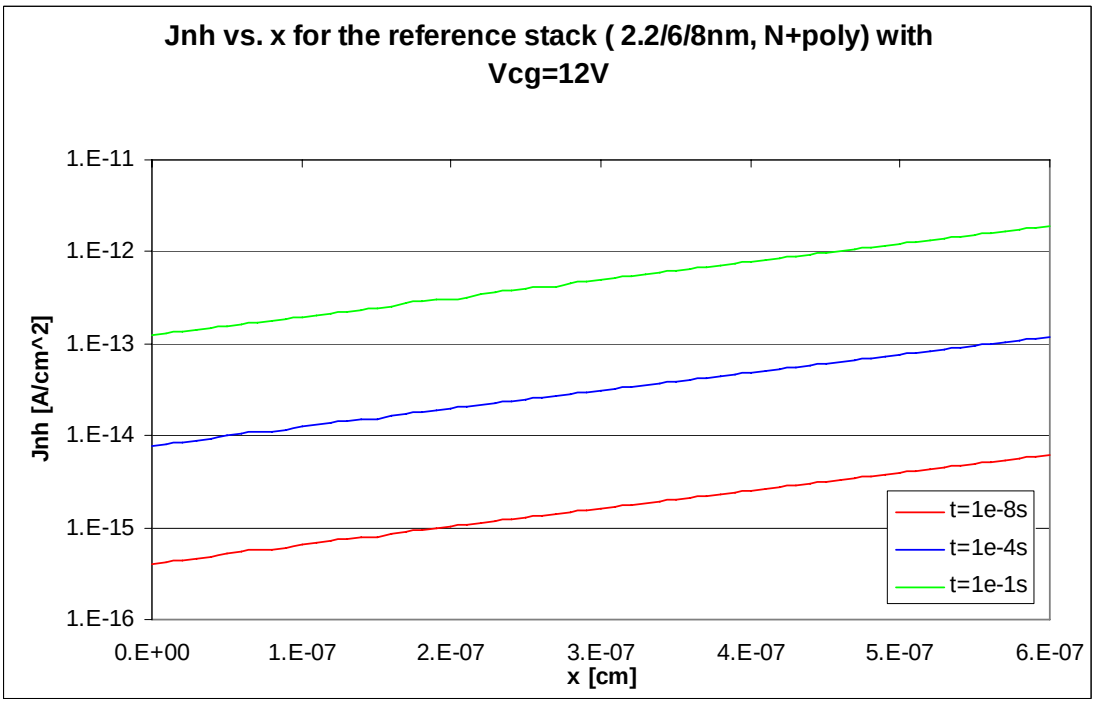

Figure 15: Distribution of the current injected from the top oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

2.2.4. Trap density in the nitride

Figures 16 to 18 show the density of traps filled with electrons n_{te}, holes n_{th}, and the density of empty traps n_{tf}, during programming.

At the beginning of programming, most of the traps in the nitride are either filled with holes or empty. Few of them are filled with electrons, which are injected from the substrate to the nitride.

Progressively n_{te} increases –especially near the bottom oxide interface–, as electrons are still injected from the substrate. Consequently, n_{tf} and n_{th} decrease.

When we reach the end of programming, traps are being filled with electrons even near the top oxide interface. Then, n_{th} has dramatically decreased in the nitride and most of the remaining empty traps are concentrated near the top oxide interface.

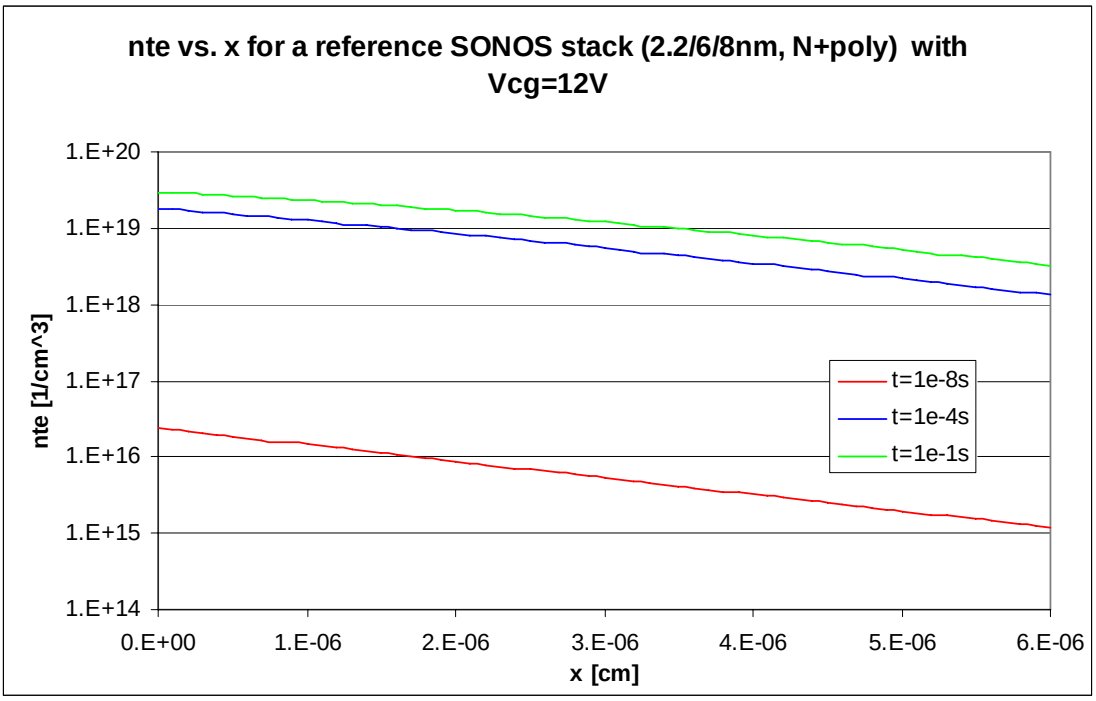

Figure 16: Density of traps filled with electrons in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

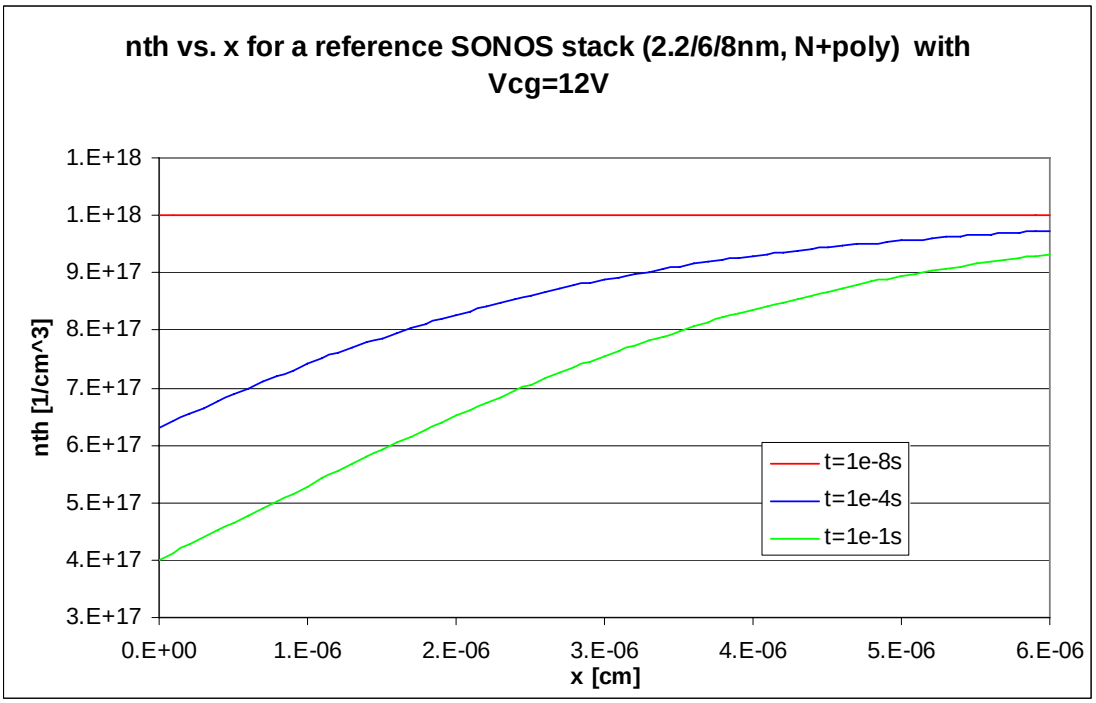

Figure 17: Density of traps filled with holes in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

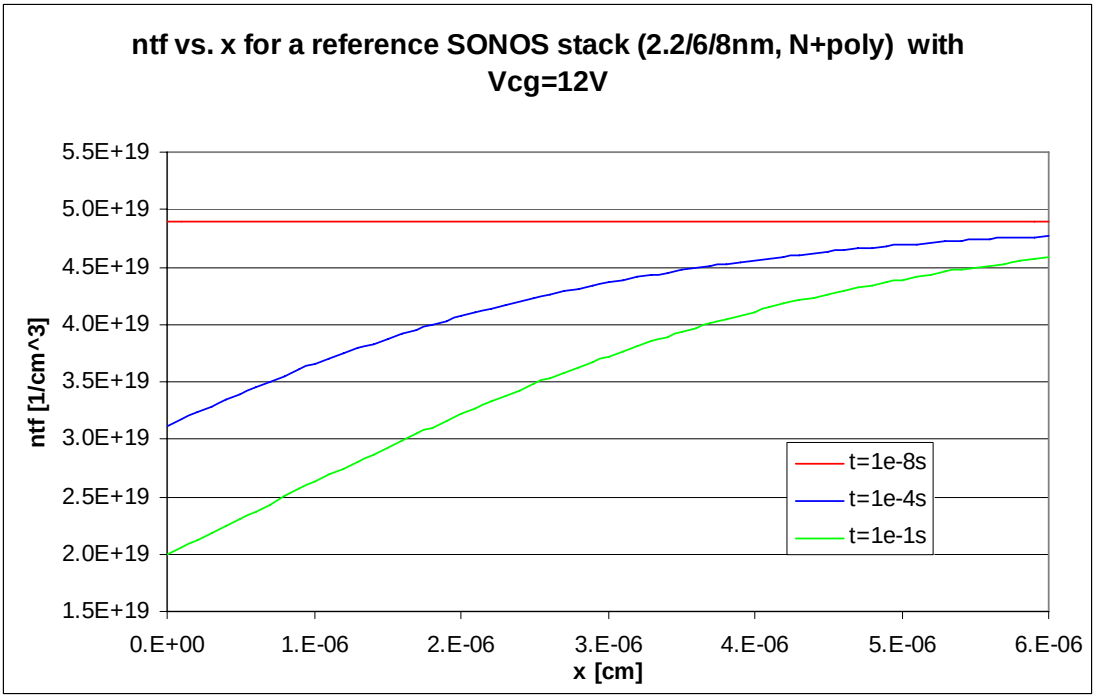

Figure 18: Density of empty traps in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

2.2.5. Charge distribution in the nitride

The charge distribution in the nitride ρ_N is shown in Figure 19. ρ_N becomes more and more negative as electrons are injected from the substrate to the nitride.

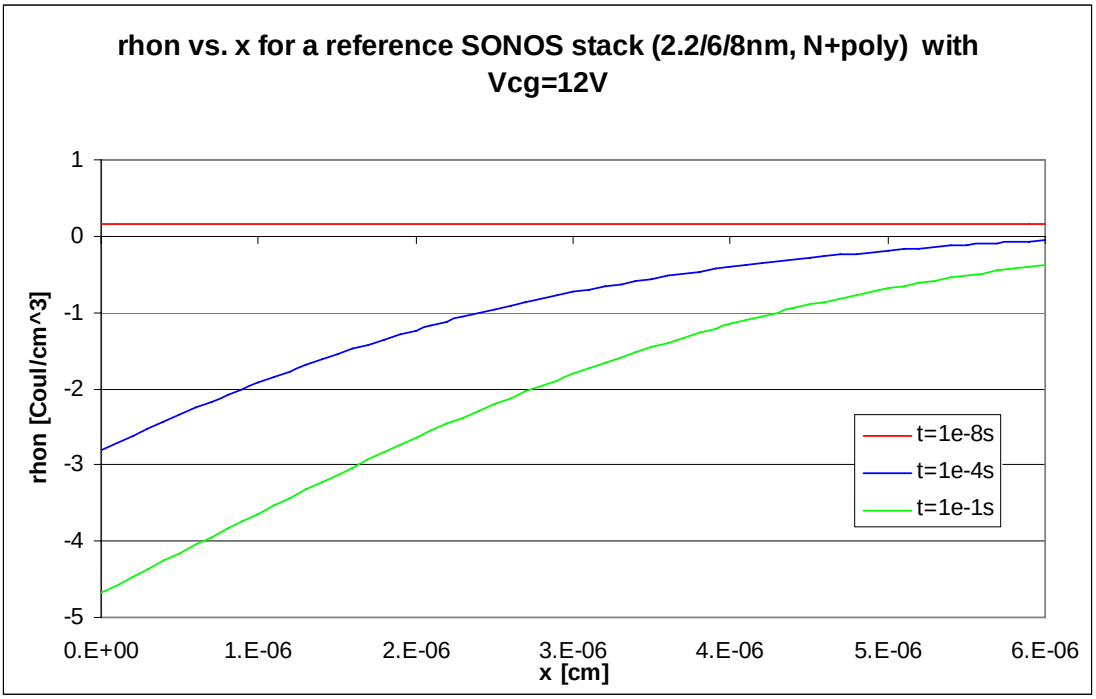

Figure 19: Charge distribution in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

2.3. Erasing

For negative control gate voltages (erasing), two injection mechanisms (Direct tunneling of holes in the bottom oxide and Fowler-Nordheim tunneling of electrons in the top oxide) compete with each other in the nitride.

2.3.1. Threshold voltage

While the nitride starts being erased, the charge in the nitride becomes increasingly positive. Therefore, E_{botox} decreases and E_{topox} increases, and electrons from the gate are injected in the nitride, as will be shown later on.
At a certain point, the currents injected from the bottom and top oxide in the nitride become comparable.
We find a good agreement between the model and measured erasing curves for a wide range of V_{cg}, as shown in Figures 20 to 23. We verify that erase saturation is achieved for high V_{cg}. In Figure 24, we see that the threshold voltage shift between two curves with consecutive values of V_{cg} is larger with the model than for measurement data. However in both cases, this threshold voltage shift is constant, whatever the V_{cg} value.
It should be noted that at the beginning of the erasing (t ~ 1e-4 s) our model gives almost no V_t shift, while the measured curves show about 0.2 to 0.5 V. This discrepancy between model and measurements is most likely caused by the fact that in this time frame the dominant mechanism is electron de-trapping (electrons tunnel back from the nitride to the bulk), which is not taken into account by our model.

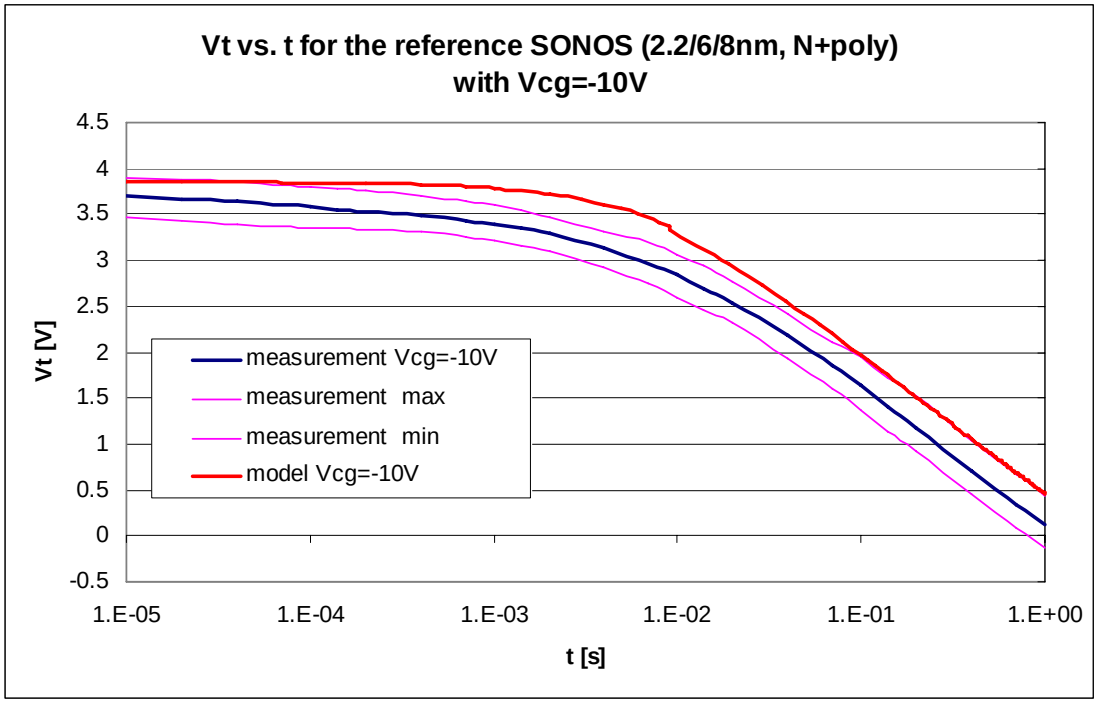

Figure 20: Erase curve for the reference stack with a -10V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$.

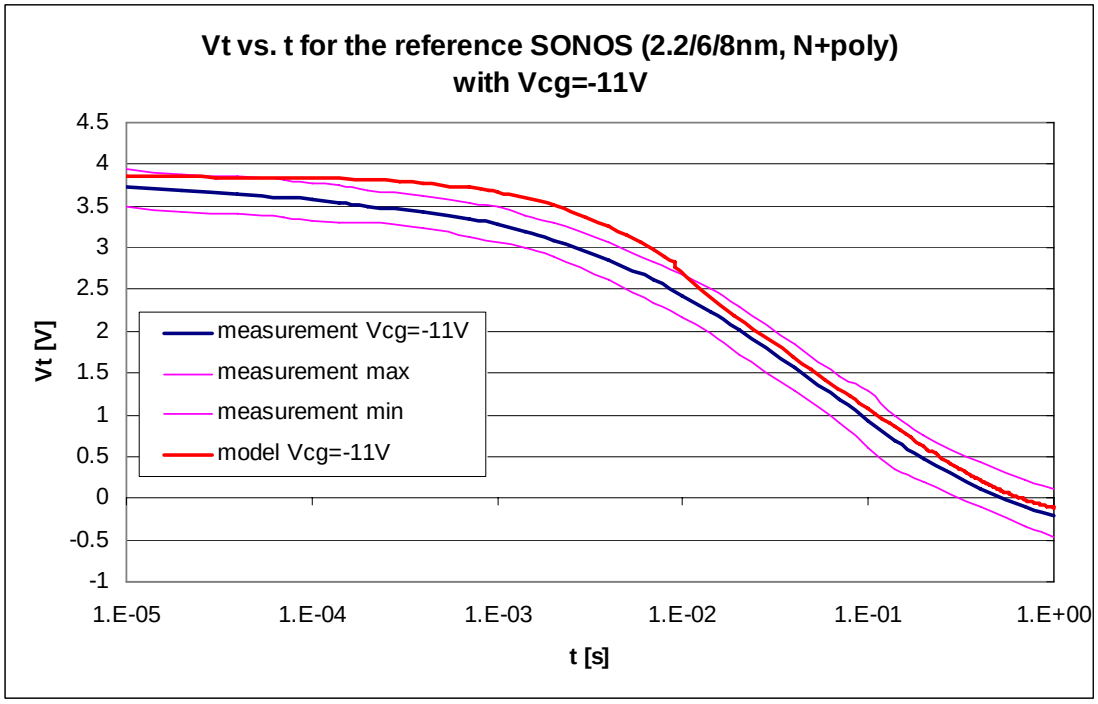

Figure 21: Erase curve for the reference stack with a -11V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$.

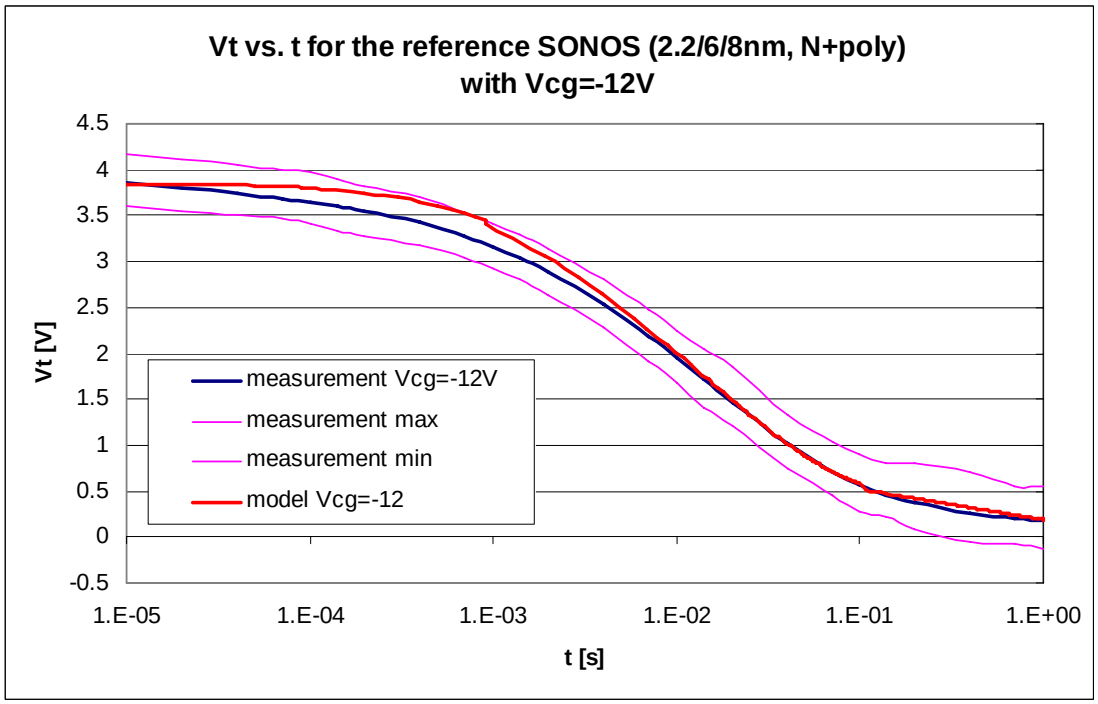

Figure 22: Erase curve for the reference stack with a -12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$.

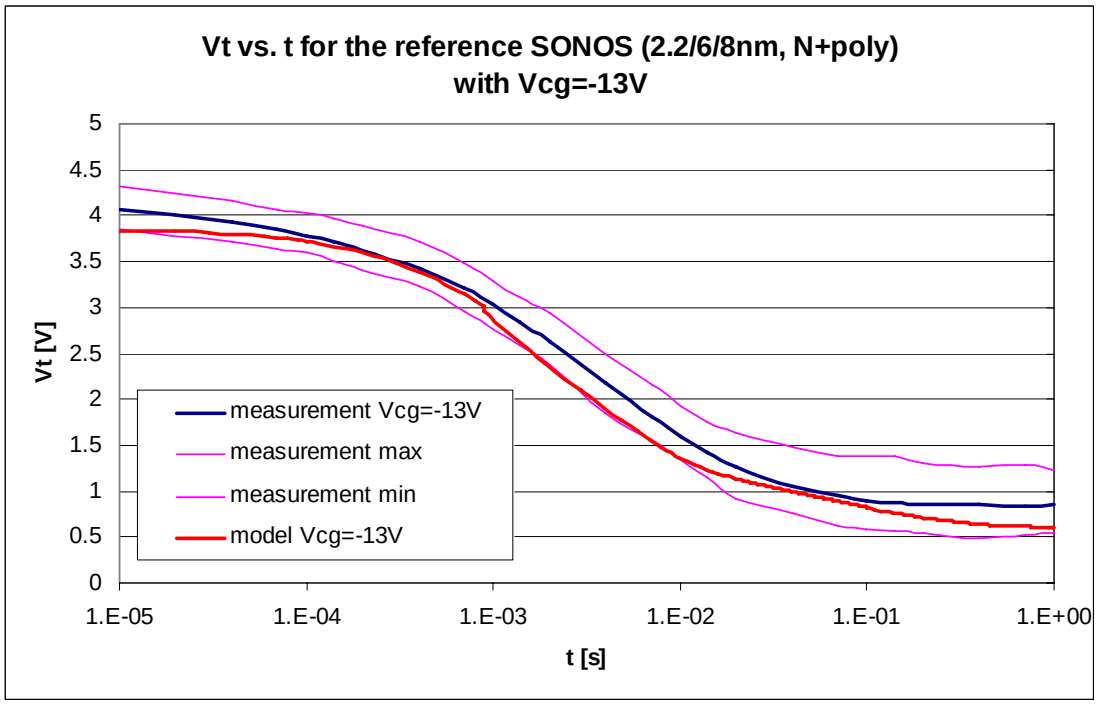

Figure 23: Erase curve for the reference stack with a -13V control gate voltage. The measurements were done on 1T mini array (256bits).$n_{te}(t=0)$ is taken $1.15e^{19}/cm^3$.

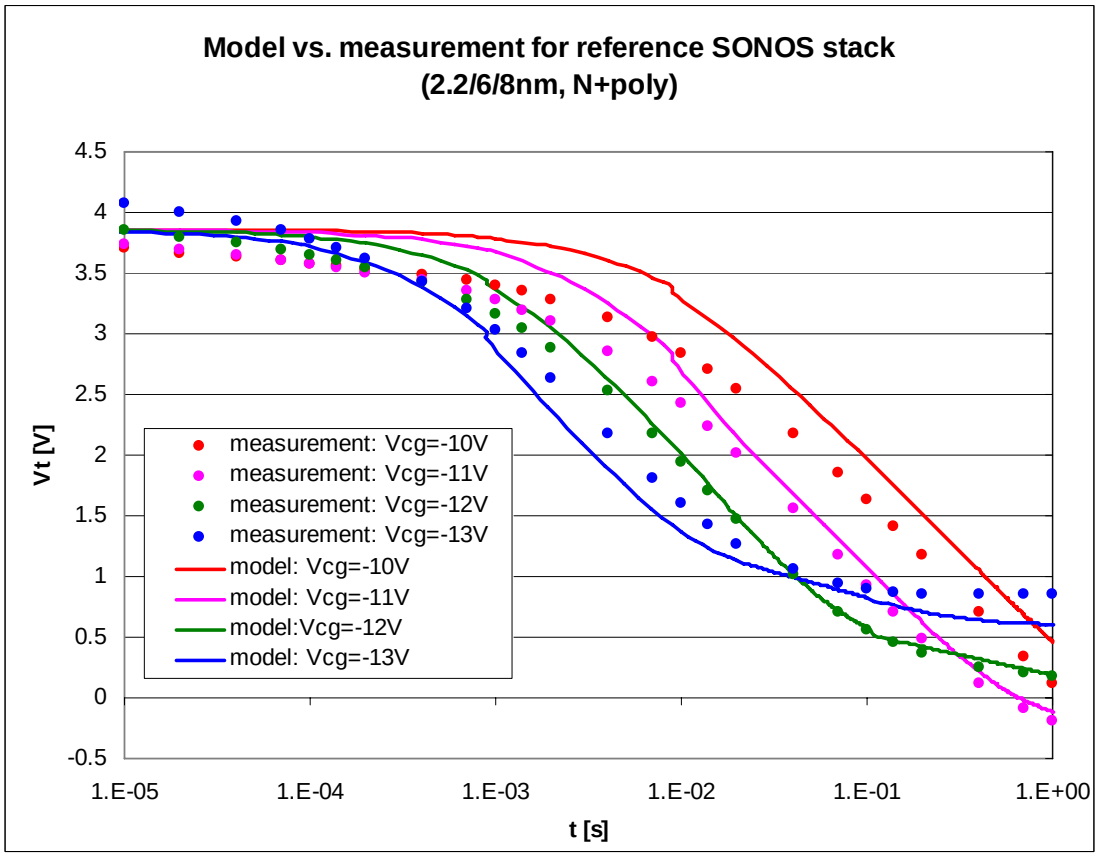

Figure 24: Erase curves comparison between model and measurement for different control gate voltages.

2.3.2. Electric field through the ONO

When we start to erase the SONOS, the charge in the nitride becomes increasingly positive. Therefore, E_{botox} decreases and E_{topox} increases. Figure 25 illustrates the E_{botox} and E_{topox} at each point in time.

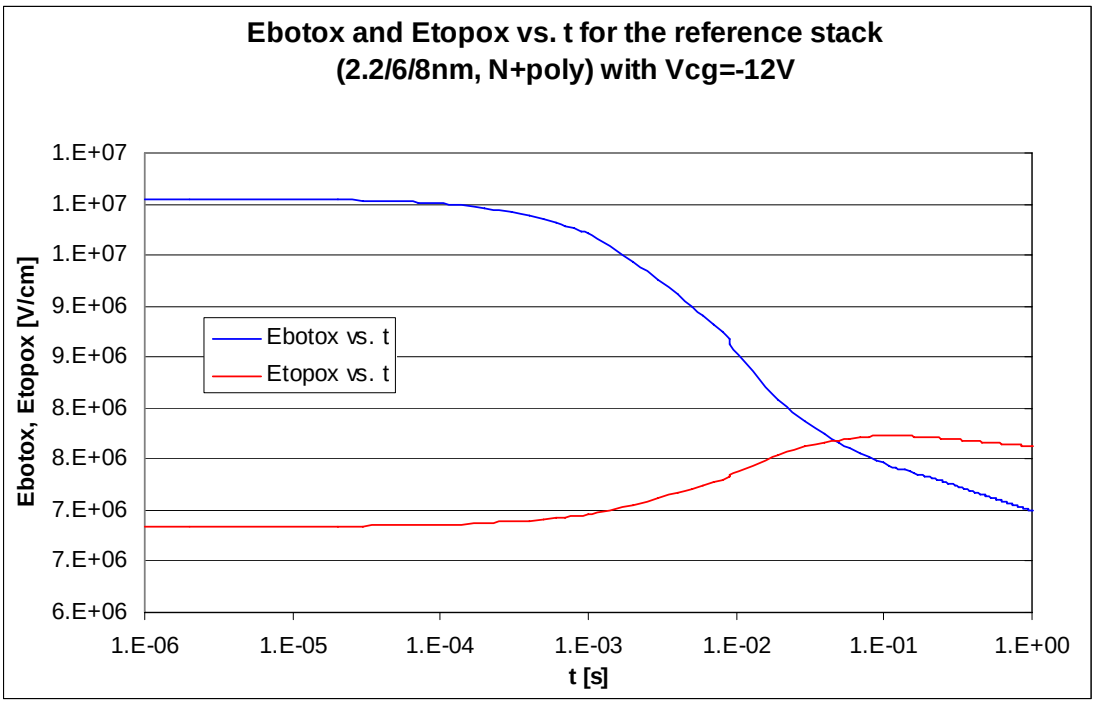

Figure 25: Electric field in the top and bottom oxide vs. time during erase, for a -12V control gate voltage

2.3.3. Injected current density in the nitride

Figure 26 and 27 show the profile of the injected current densities in the nitride (J_{Ne} from the gate and J_{Nh} from the substrate). It can be observed that J_{Ne} becomes comparable to J_{Nh}, when we reach the erase saturation regime. We can see that J_{Nh} is decreasing, while J_{Ne} is increasing with time.

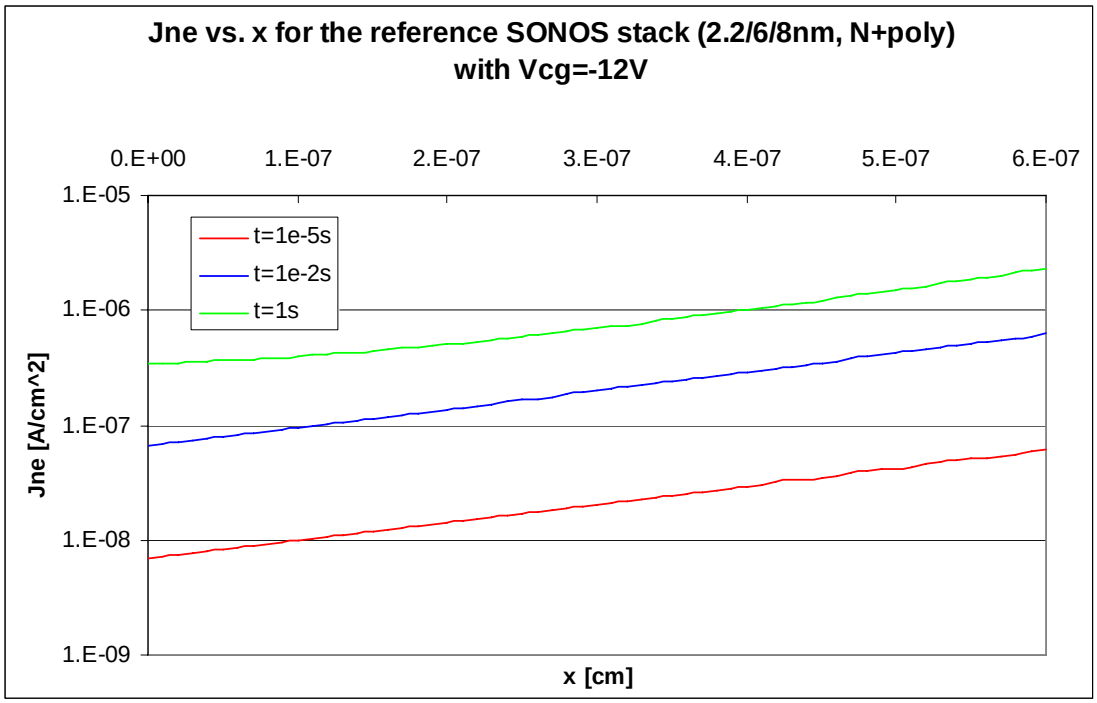

Figure 26: Distribution of the current injected from the top oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

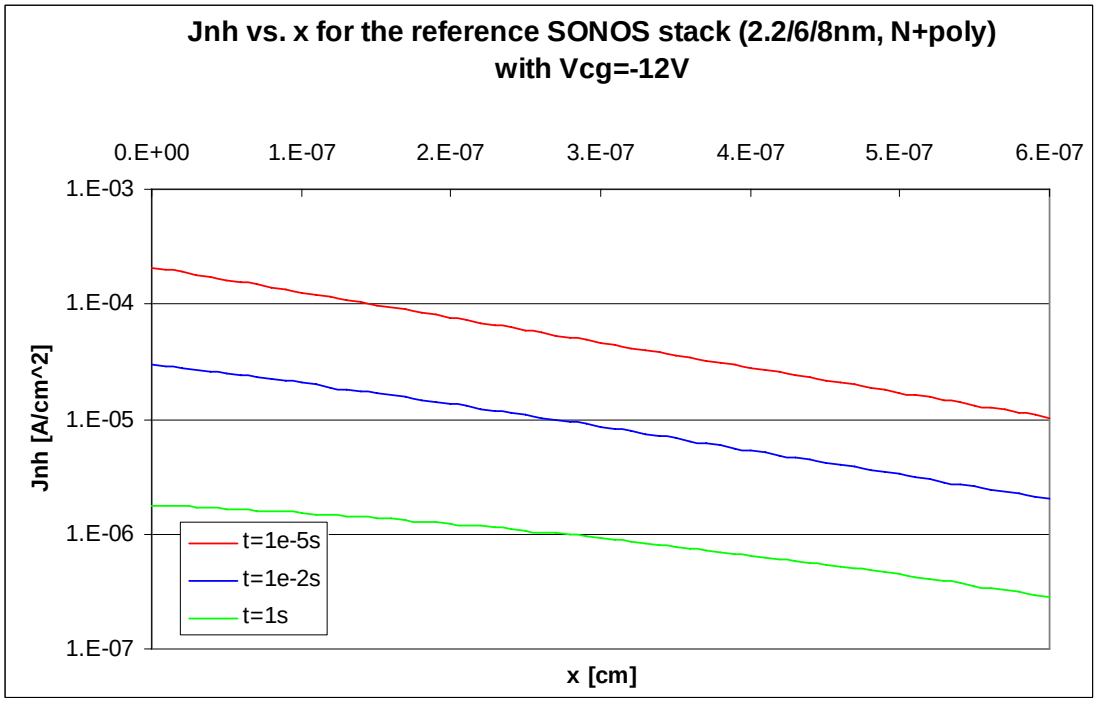

Figure 27: Distribution of the current injected from the bottom oxide to the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

2.3.4. Trap density in the nitride

Figures 28 to 30 show the density of traps filled with electrons n_{te}, holes n_{th}, and the density of empty traps n_{tf},during erasing.

At the beginning of erasing, most of the traps in the nitride are either filled with electrons or empty. Few of them are filled with holes, which are injected from the substrate to the nitride.

Progressively n_{th} increases, especially near the bottom oxide interface. At a certain point, the current injected from the gate (Fowler-Nordheim tunneling mechanism) becomes important, and n_{te} starts increasing.

When we reach the end of erasing, the tunneling currents through the bottom and top oxide have become comparable, so that the net charge in the nitride is zero. Thus, we reach the erase saturation regime. Traps are being filled with electrons near the top oxide interface and with holes near the bottom oxide interface. Most of the remaining empty traps are then concentrated around the center of the nitride.

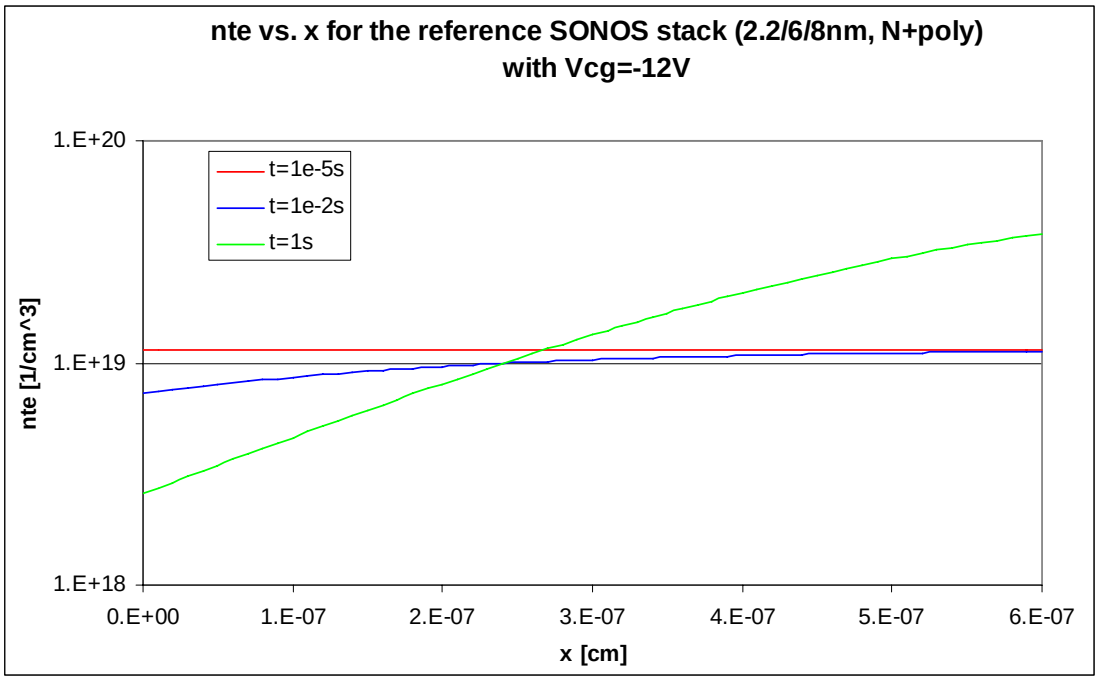

Figure 28: Density of traps filled with electrons in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

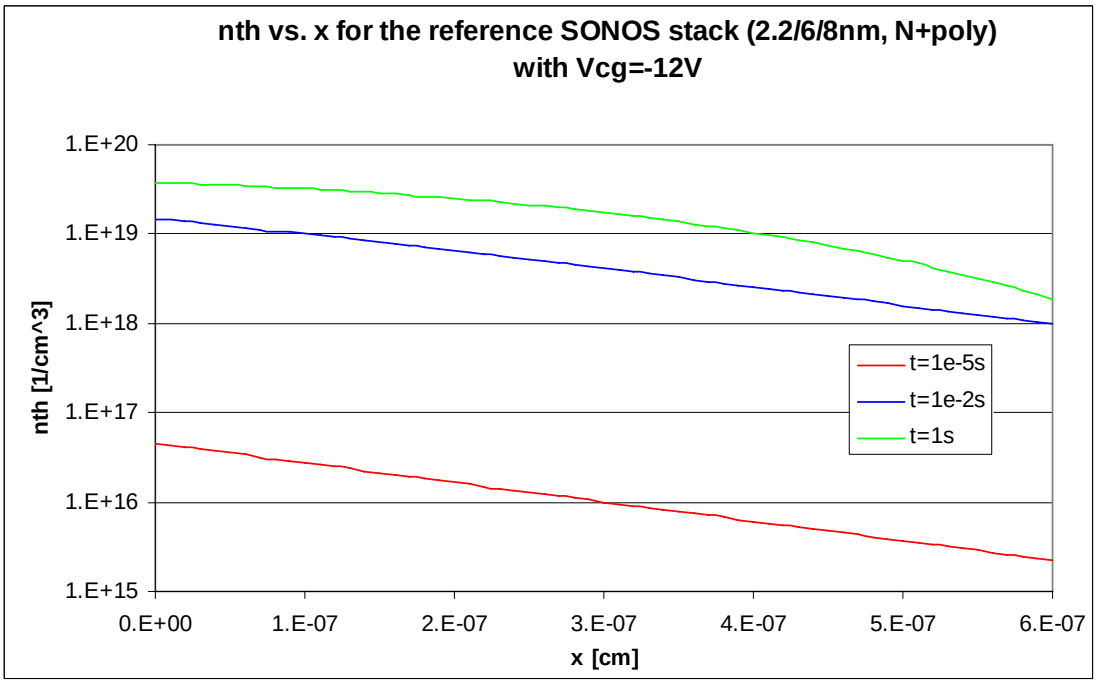

Figure 29: Density of traps filled with holes in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

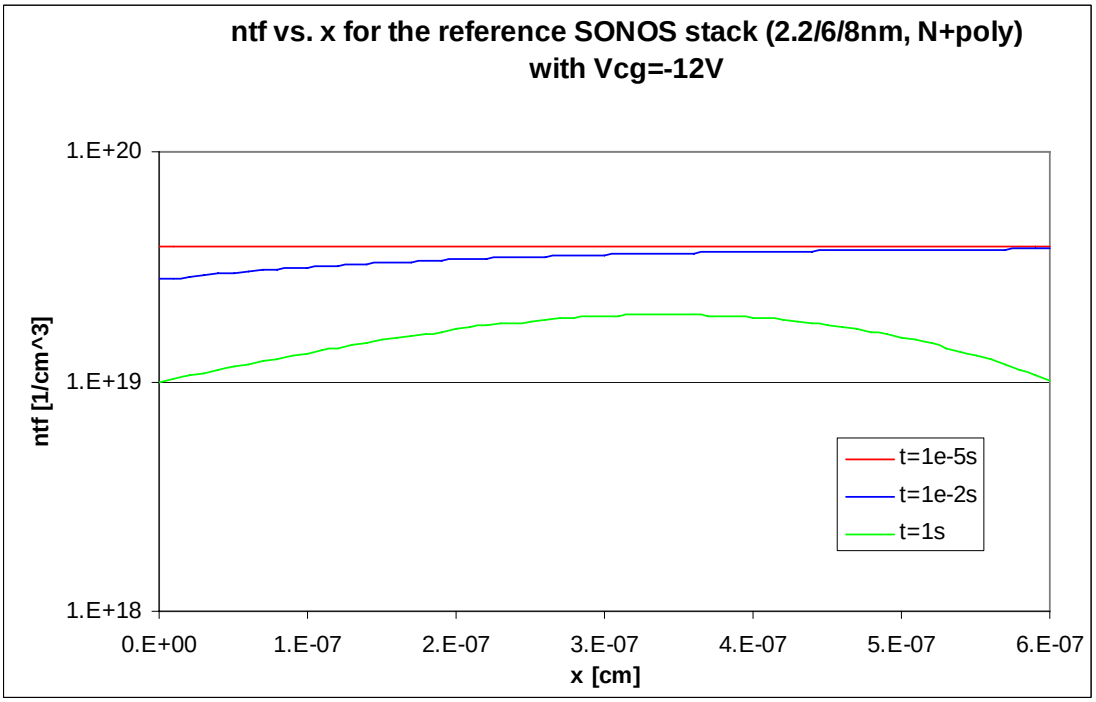

Figure 30: Density of empty traps in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

2.3.5. Charge distribution in the nitride

The charge distribution in the nitride, ρ_N, is shown in Figure 31. At the beginning of the erase, ρ_N is set to a constant negative value in the whole nitride layer. When holes start being injected from the substrate, ρ_N increases and becomes positive near the bottom oxide interface. As soon as the current injected from the top oxide becomes comparable to the one injected from the bottom oxide, ρ_N decreases near the top oxide interface, while it keeps on increasing near the bottom oxide interface.

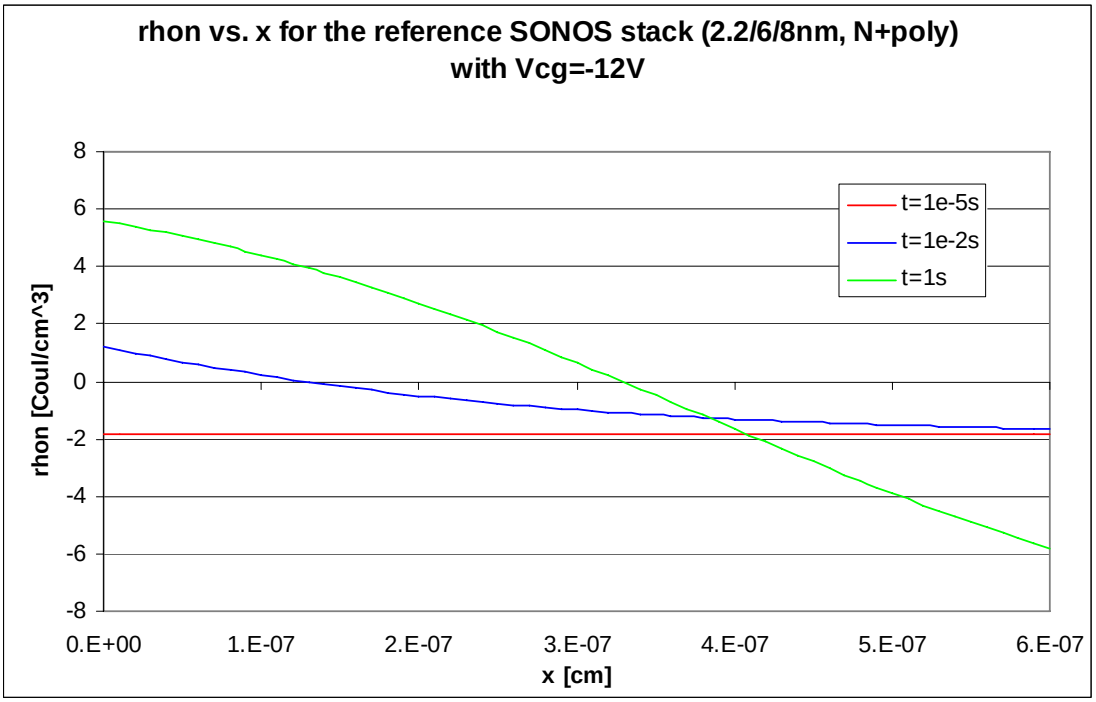

Figure 31: Charge distribution in the nitride (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

Figure 31 shows a comparison of the charge distribution in the nitride for -12V and -13V control gate voltages. We can see that at t = 1e^{-2} s more holes have been injected near the bottom oxide/nitride interface for the highest V_{cg} (-13V) which makes the erased V_t lower at this specific time with V_{cg}=-13V (see figure 24).
However, at t=1s more electrons have been injected near the top oxide/nitride interface for the highest V_{cg} (-13V) which makes the leveling off of the V_t at higher values (See figure 24). This result of the model is in line with measurements where we see clearly that going to higher V_{cg} erase voltage increases the leveling off level of the V_t.

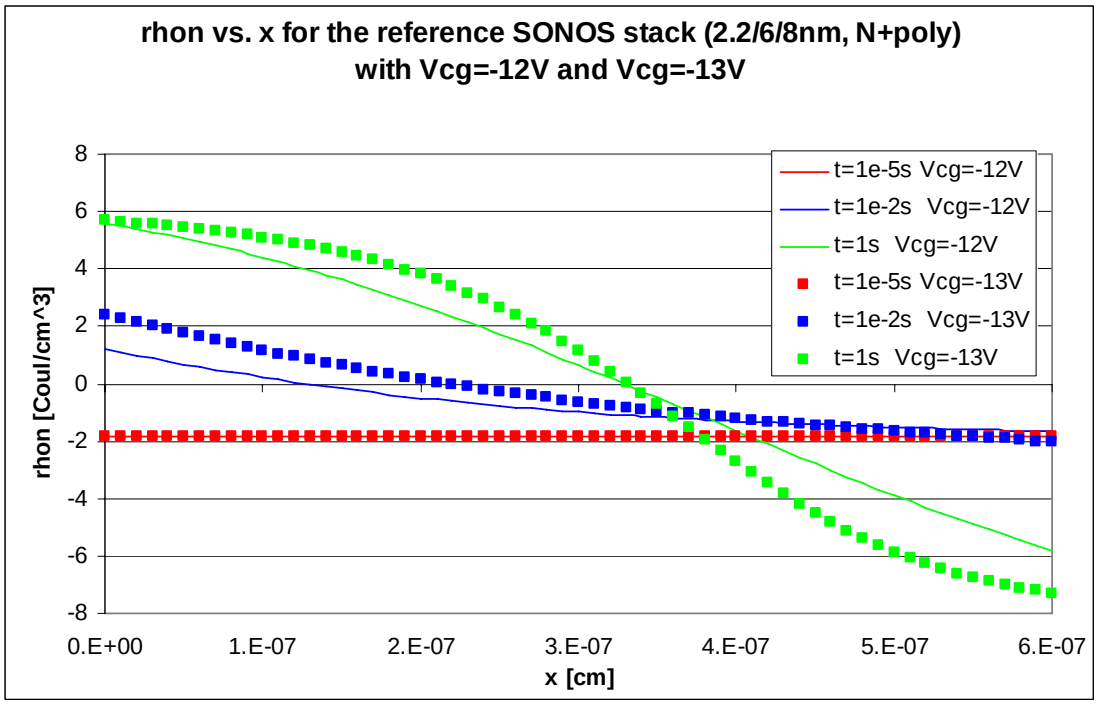

Figure 32: Comparison of the charge distribution in the nitride for -12V and -13V control gate voltages (x=0cm: bottom oxide/nitride interface, x=6e-7cm top oxide/nitride interface)

CHAPTER 3 – MODELING RESULTS FOR OTHER STACKS

In this chapter, we study how changes in some physical parameters can affect the program and erase curves described with the model, and in which respect we still find good correspondence with measurements.
We will first present the impact of the gate doping. Then we will present the influence of the top and bottom oxide thickness. Finally, we will show how a change in the k-value of the top oxide affects the results.

3.1. Influence of the gate doping

If we choose a P-type gate, instead of an N-type gate as in the previous chapter, we have to change the gate work function (for a P-type gate ϕ_M = 4.1 + 1.1 V, i.e. ϕ_M = 5.2 V) and the top oxide tunneling barrier for electrons (ϕ_3 = 3.1 + 1.1V, i.e. ϕ_3 = 4.2 V).
However, it can be observed from the comparison between program curves for N- and P-type gate that the threshold voltage does not shift by 1.1 V, as expected (see figure 33). We notice a V_t shift of 0.5 V. Therefore, we use in our model ϕ_M = 4.6 V and ϕ_3 = 3.6 V for electrons tunneling from the P-type gate to the nitride.

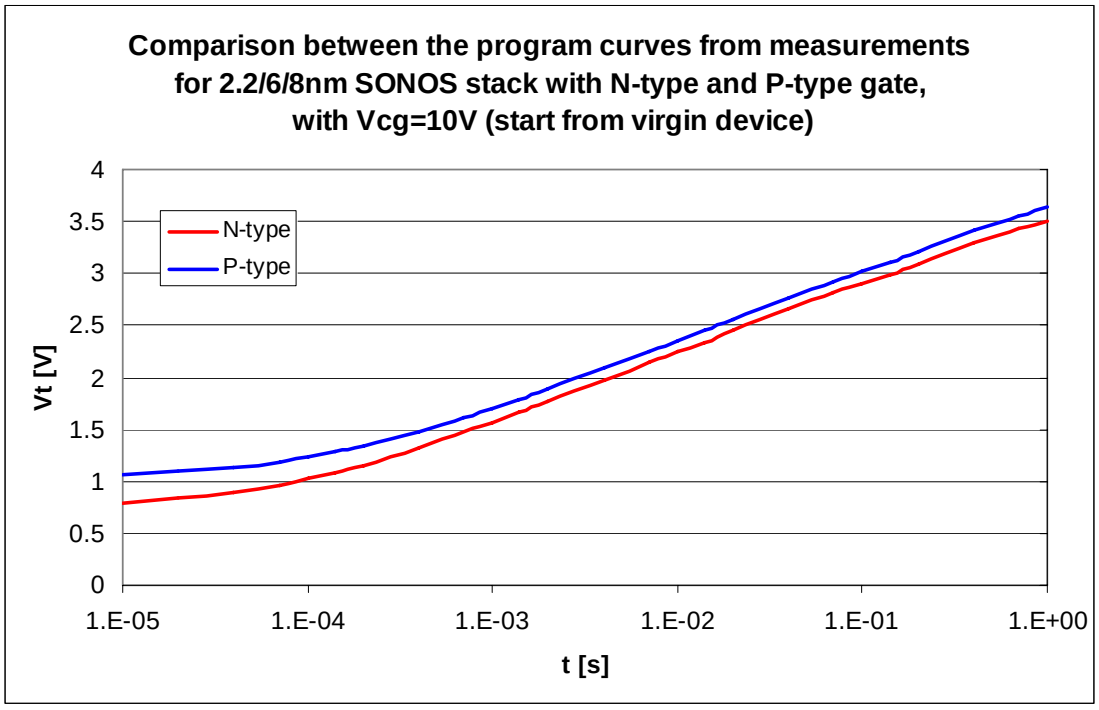

Figure 33: Comparison between the program curves from measurements for 2.2/6/8nm SONOS stack with N-type and P-type gate, with V_{cg} =10V (starting from a virgin device). The measurements were done on 1T mini array (256bits).

Increasing ϕ_3 results in delaying the erase saturation. Therefore the device can be erased to a deeper level than with an N-type gate, especially for high values of V_{cg}. Figures 34 to 37 show the comparison between erase curves from the model and measurements for a 2.2/6/8nm SONOS with P-type gate. It can be observed for high control gate voltages (V_{cg}=-13V and -14V) that erase saturation occurs for much lower V_t than with an N-type gate (see figures 36 and 37).

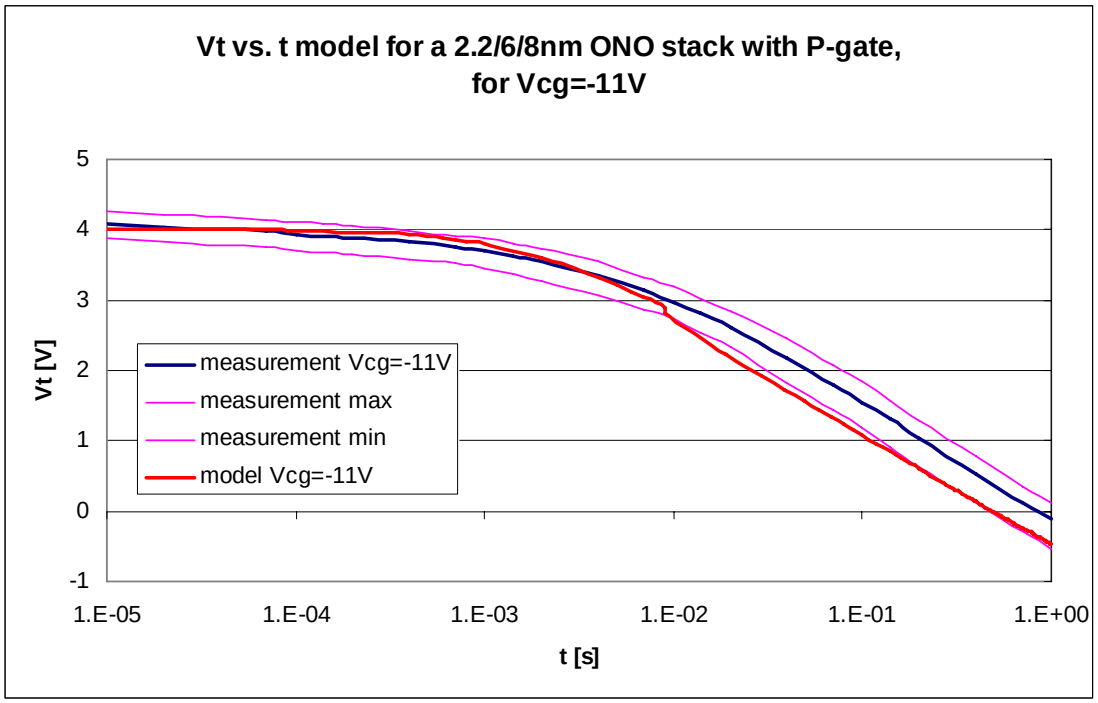

Figure 34: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -11V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$.

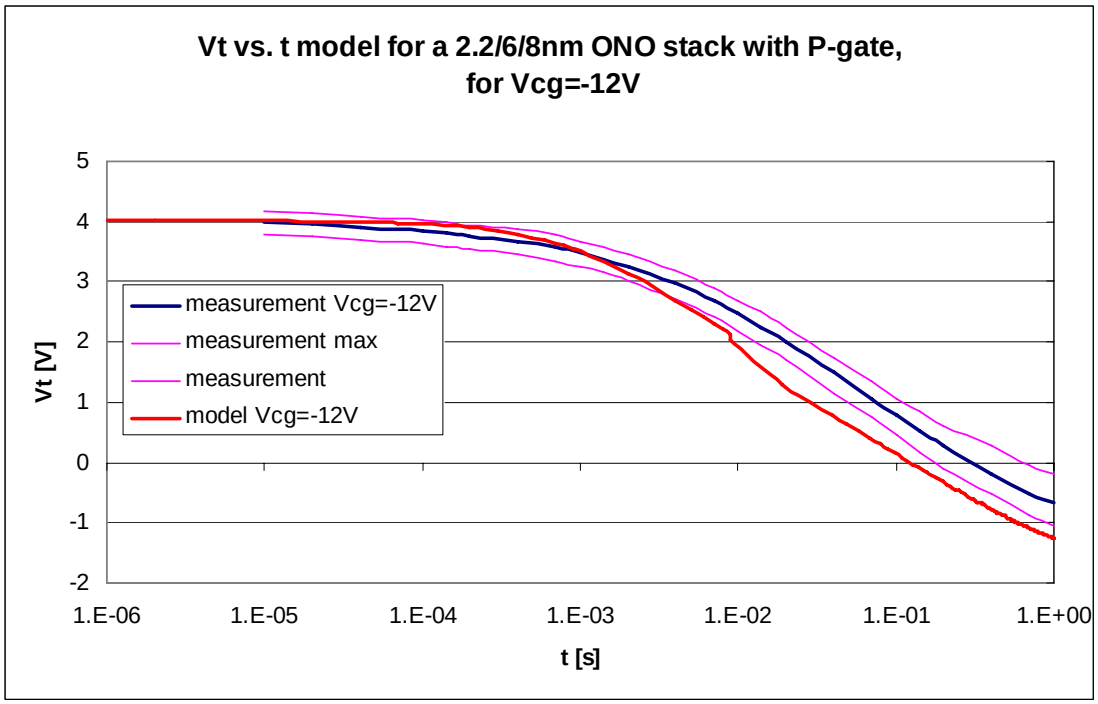

Figure 35: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$.

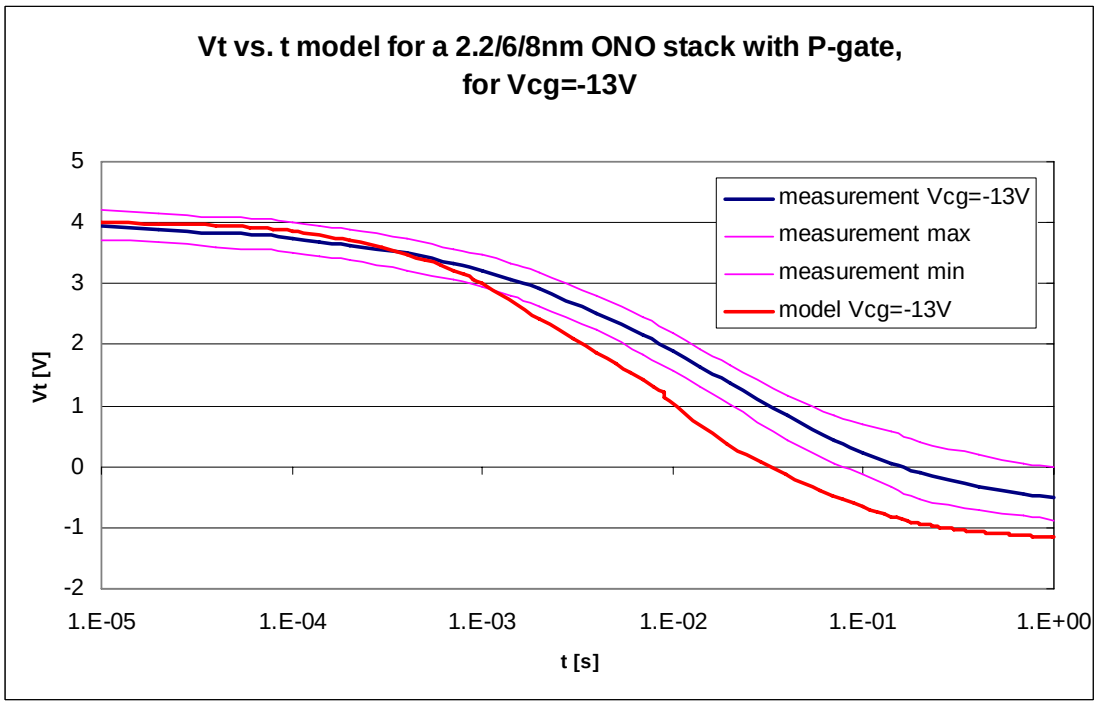

Figure 36: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -13V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$.

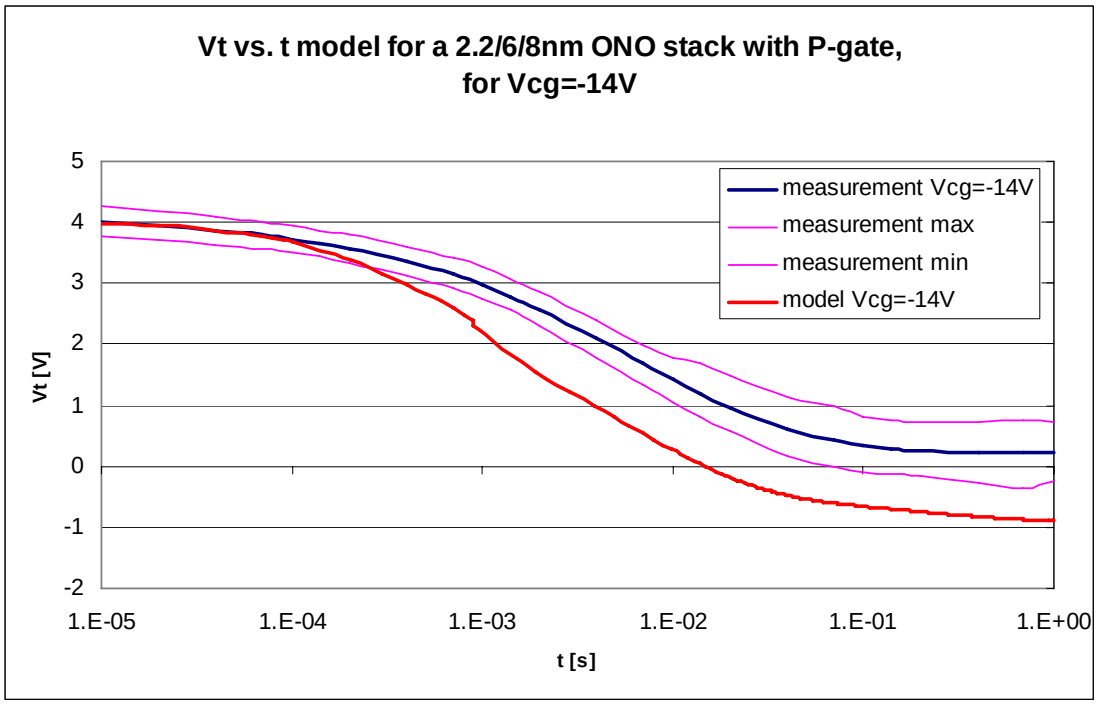

Figure 37: Erase curve from the model for a 2.2/6/8nm ONO with P-gate, for a -14V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $1.05e^{19}/cm^3$.

Figures show the comparison between program curves from the model and measurements for a 2.2/6/8nm SONOS with P-type gate. We took in our model $n_{th}(t=0) = 0.1e^{19}/cm^3$, $\phi_M = 4.6$ V and $\phi_3 = 3.6$ V for holes tunneling from the P-type gate to the nitride.

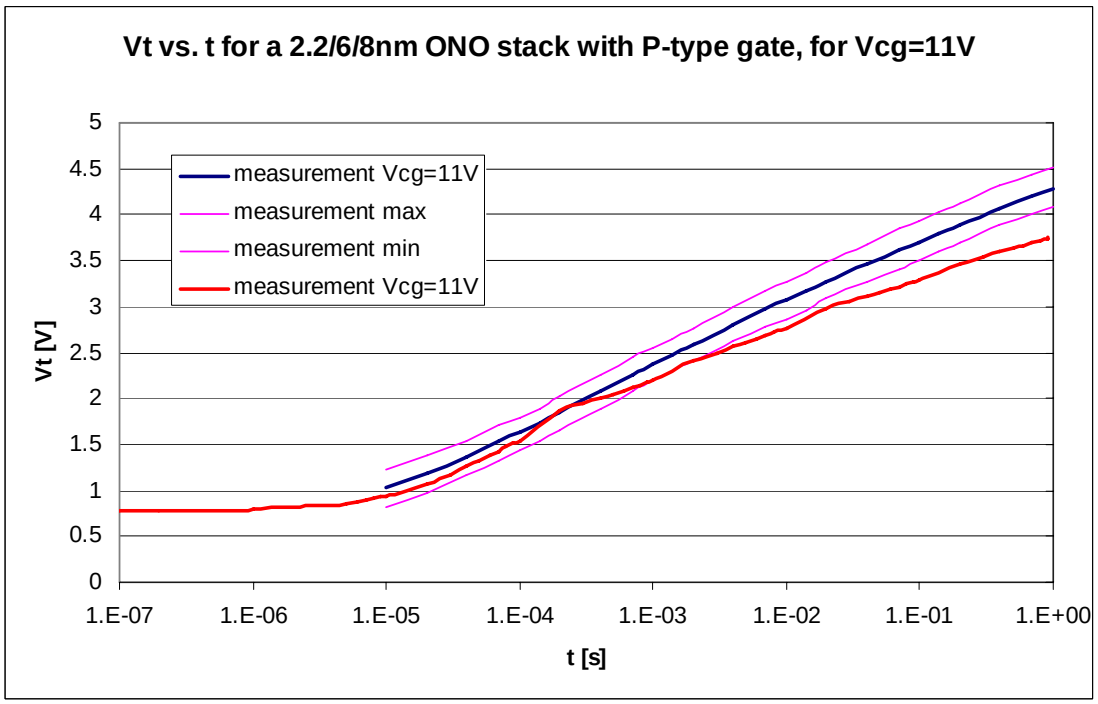

Figure 38: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for an 11V control gate voltage. The measurements were done on 1T mini array

(256bits). $n_{th}(t=0)$ is taken $0.18e^{19}/cm^3$.

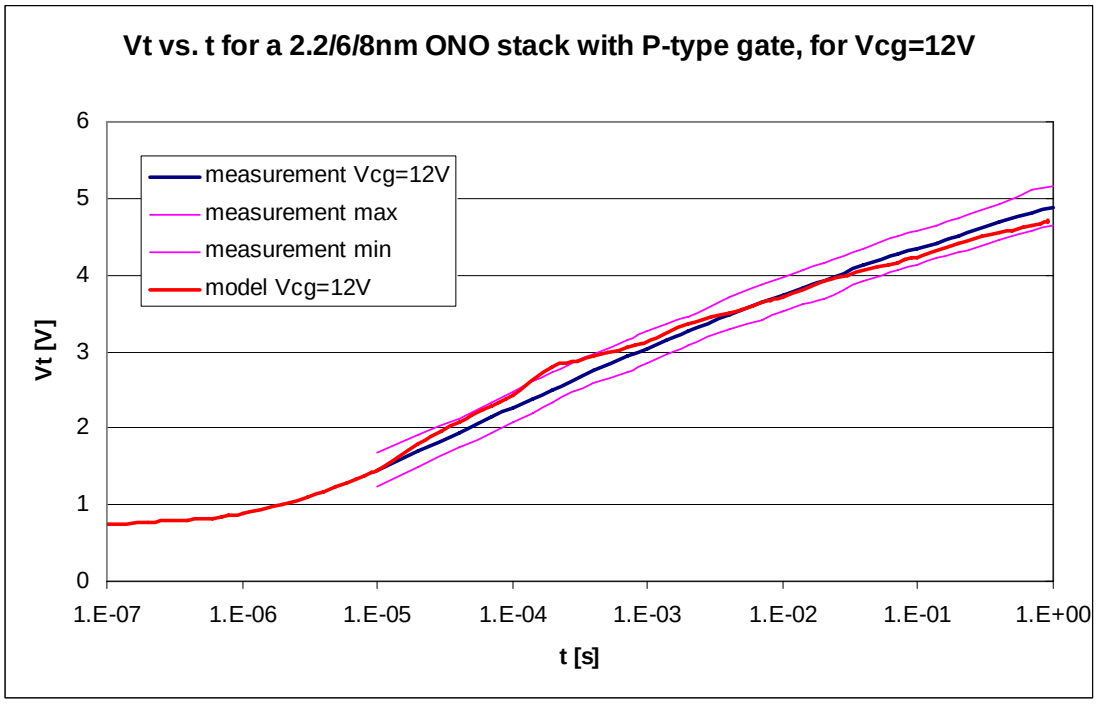

Figure 39: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for a 12V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.18e^{19}/cm^3$.

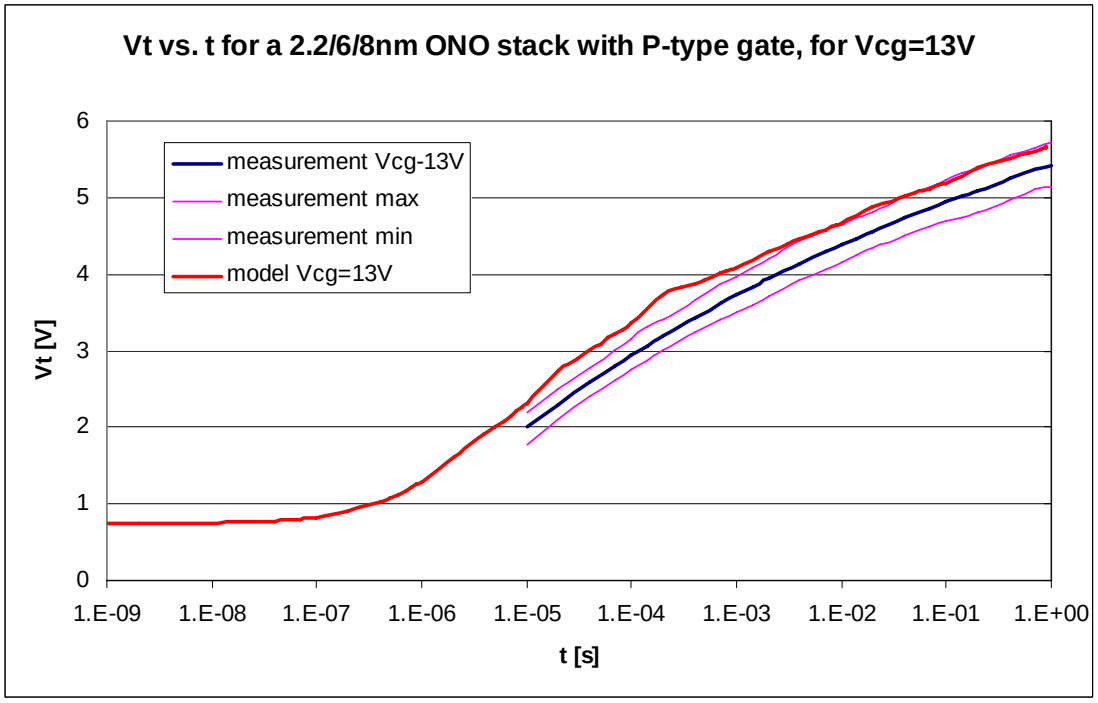

Figure 40: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for a 13V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.18e^{19}/cm^3$

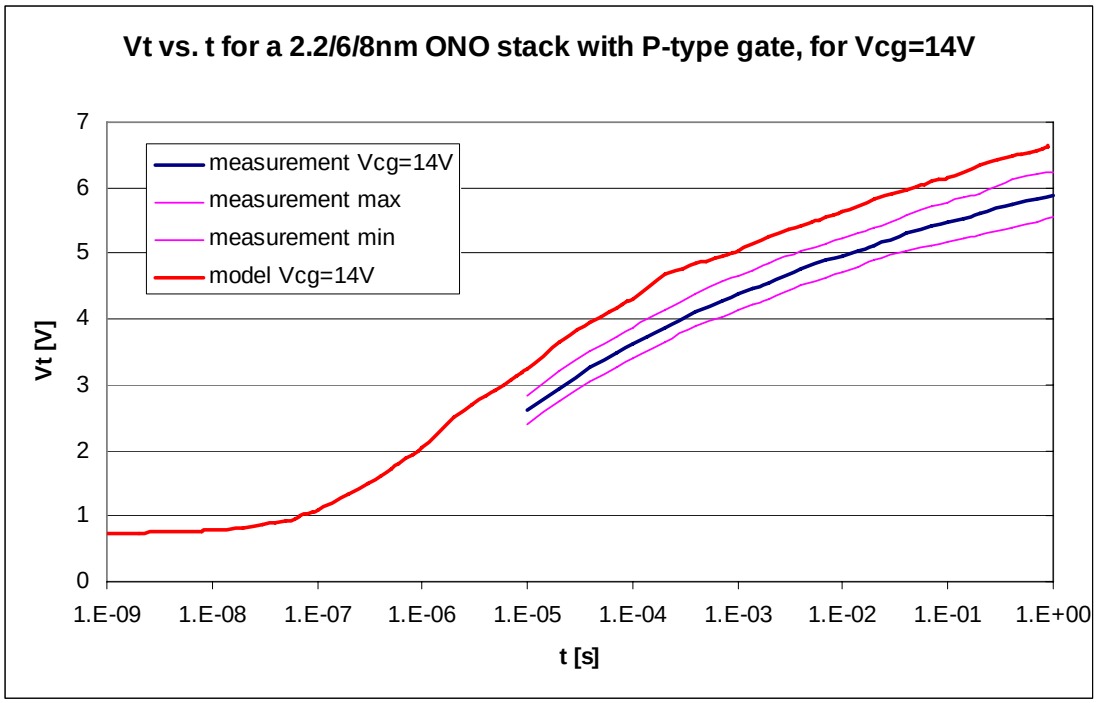

Figure 41: Program curves from the model for a 2.2/6/8nm ONO with P-gate, for a 14V control gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.18e^{19}/cm^3$

3.2. Impact of changes in the oxide thickness on the model

3.2.1. Top oxide

Decreasing the top oxide thickness enhances Fowler-Nordheim injection from the gate to the nitride. Therefore, we reach erase saturation earlier and for lower control gate voltages than for the reference stack (2.2/6/8nm ONO N-type gate). We find good agreement between the model and measurements for the program and erase curves for lower V_{cg} values, as shown in Fig. 42 to 45.

We had to reset the amount of traps filled with electrons to reach good correspondence Vt between the erase curves from model and measurement:

$n_{te}(t=0) = 2.1\text{e}19\ /\text{cm}^3$

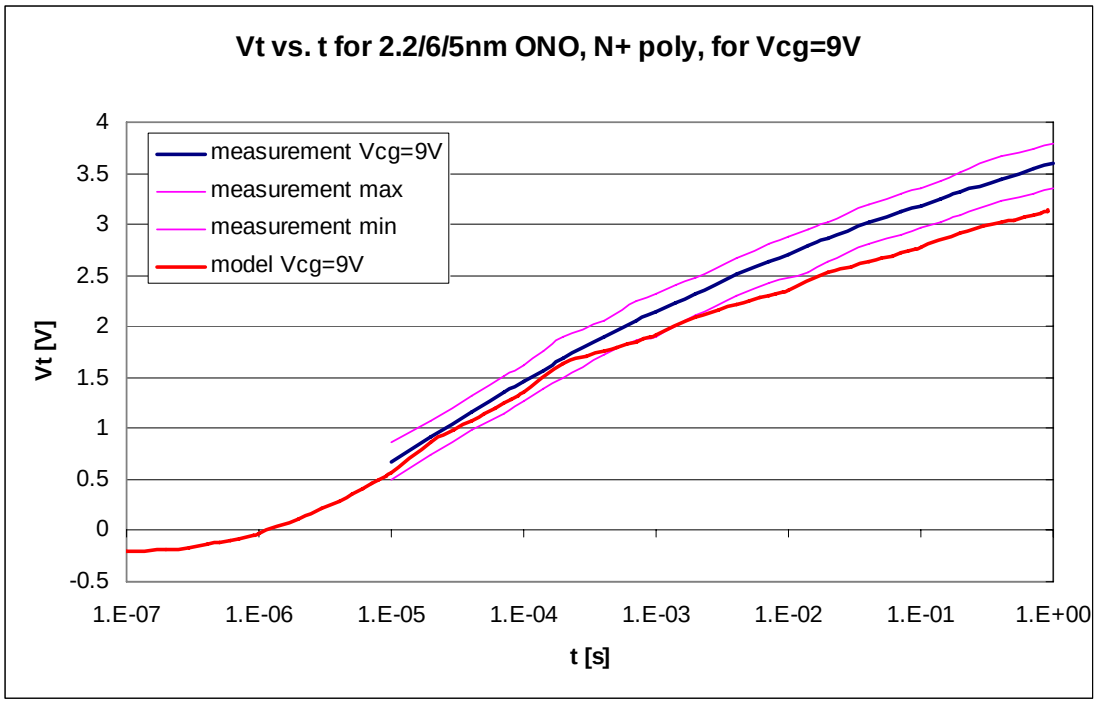

Figure 42: Program curves for a 2.2/6/5nm ONO stack with a 9Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$.

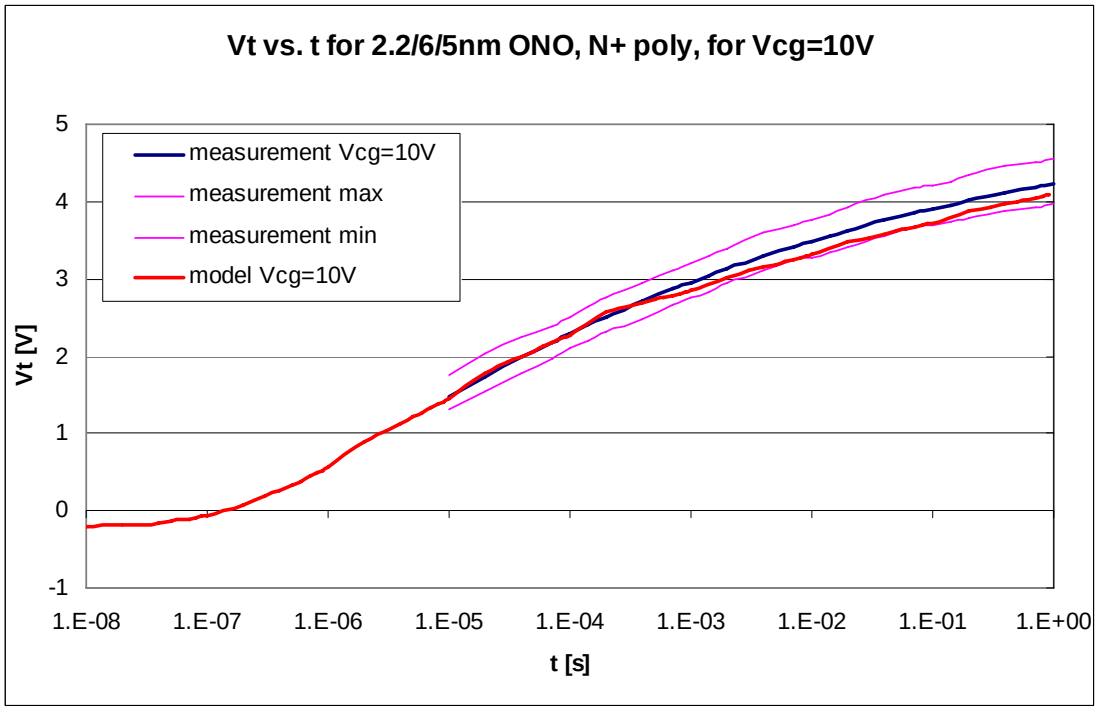

Figure 43: Program curves for a 2.2/6/5nm ONO stack with a 10Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{th}(t=0)$ is taken $0.1e^{19}/cm^3$.

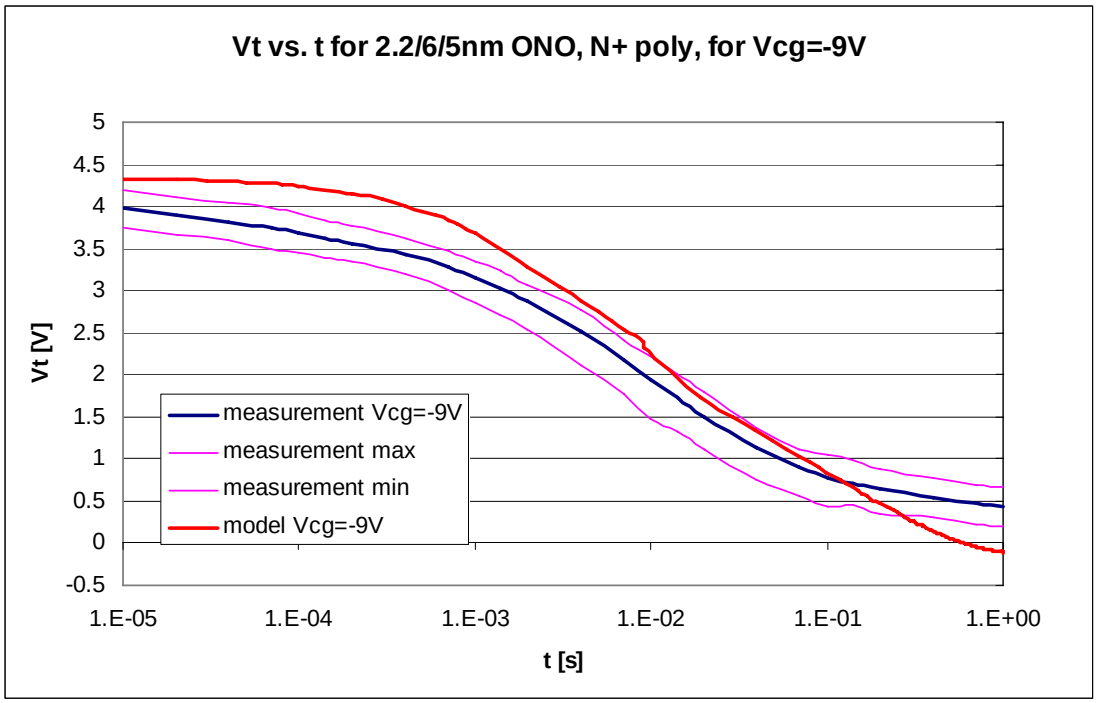

Figure 44: Erase curves for a 2.2/6/5nm ONO stack with a -9Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $2.1e^{19}/cm^3$.

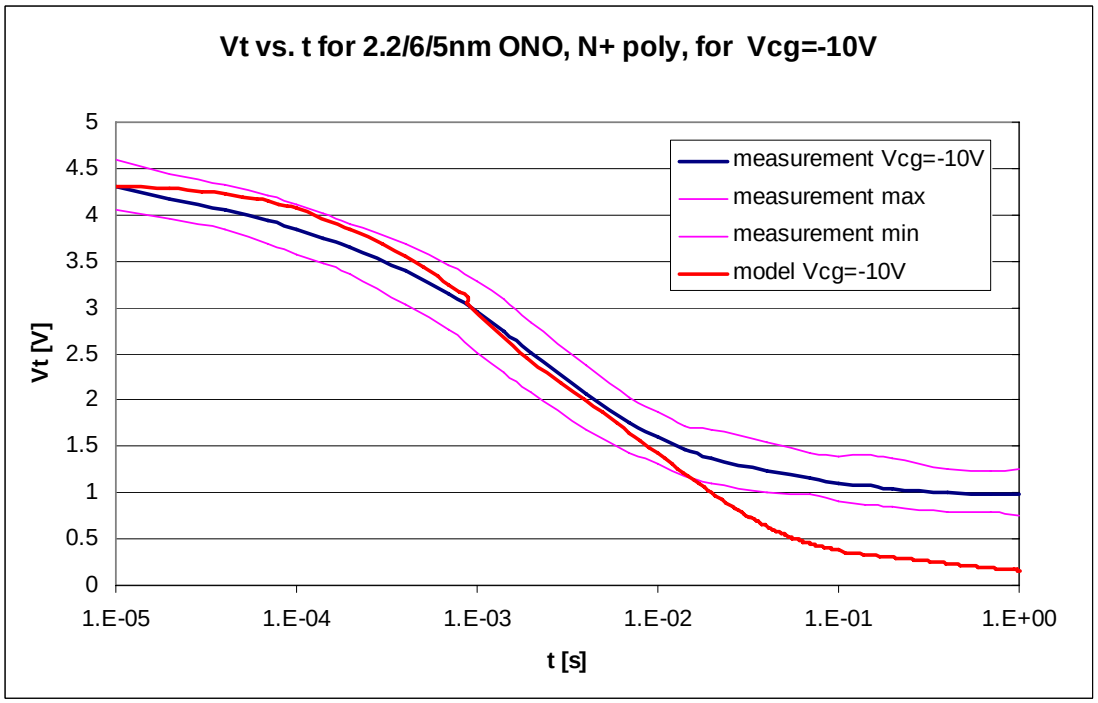

Figure 45: Erase curves for a 2.2/6/5nm ONO stack with a -10Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $2.1e^{19}/cm^3$.

3.2. Bottom oxide thickness

Increasing the bottom oxide thickness may enhance data retention. However, erasing the nitride becomes much more difficult, when the bottom oxide thickness increases. This can be explained by the fact that the direct tunneling current is exponentially dependent on the bottom oxide thickness. A slight increase in the bottom oxide thickness (e.g. 0.5 to 1nm) results in orders of magnitude slower programming or erasing.
Moreover, when a high V_{cg} is applied to erase a device with thick bottom oxide the results could be programming it rather than erasing. This can be explained by the extremely low direct tunneling of holes in the bottom oxide and the high injection of electrons from the gate to the nitride. We see in figure 46 that this phenomenon is observed in measurements and is well predicted by our program to a certain extend.

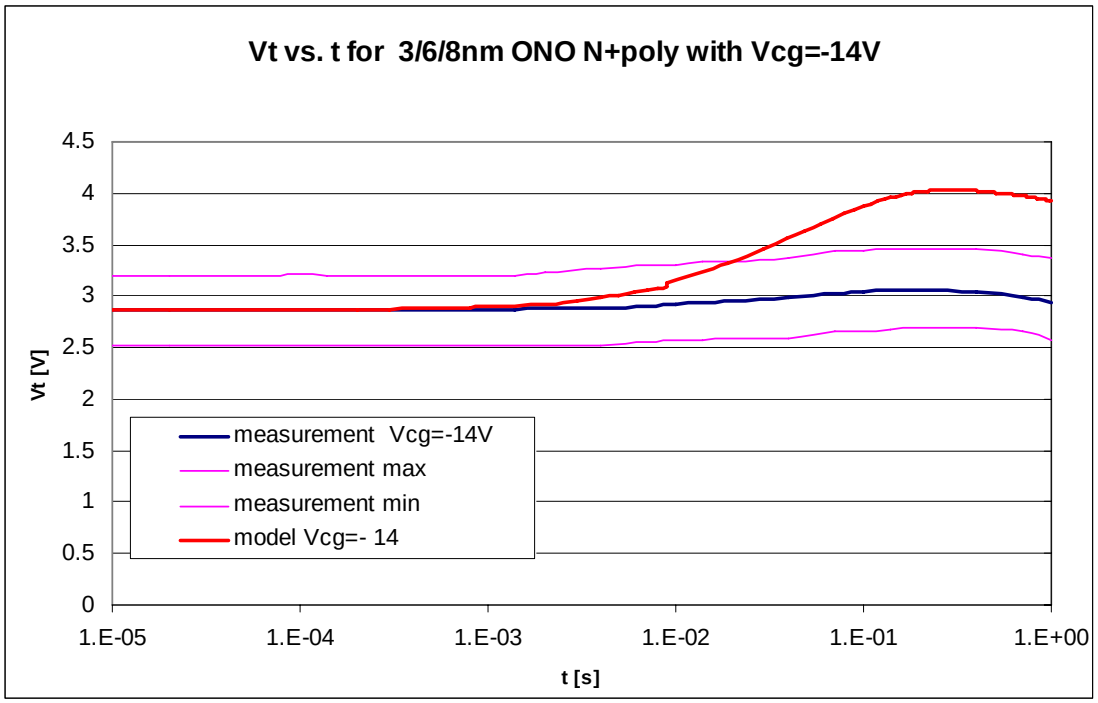

Figure 46: Erase curves for a 3/6/8nm ONO stack, N-type gate with a -14Vcontrol gate voltage. The measurements were done on 1T mini array (256bits). $n_{te}(t=0)$ is taken $0.75e^{19}/cm^3$

3.3. Impact of a change in the k-value of the top oxide on the model results

If we increase the k-value of the top oxide, we can erase the nitride to a lower V_t value without having erase saturation.
However, most often when using hi-k materials a fixed charge or interface charge can be present which can shift the V_t. The program is able to take the fixed charge into account by specifying a fixed Q_{ox} parameter.
A comparison between two erasing curves at Vcg=-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) is depicted in figure 47.

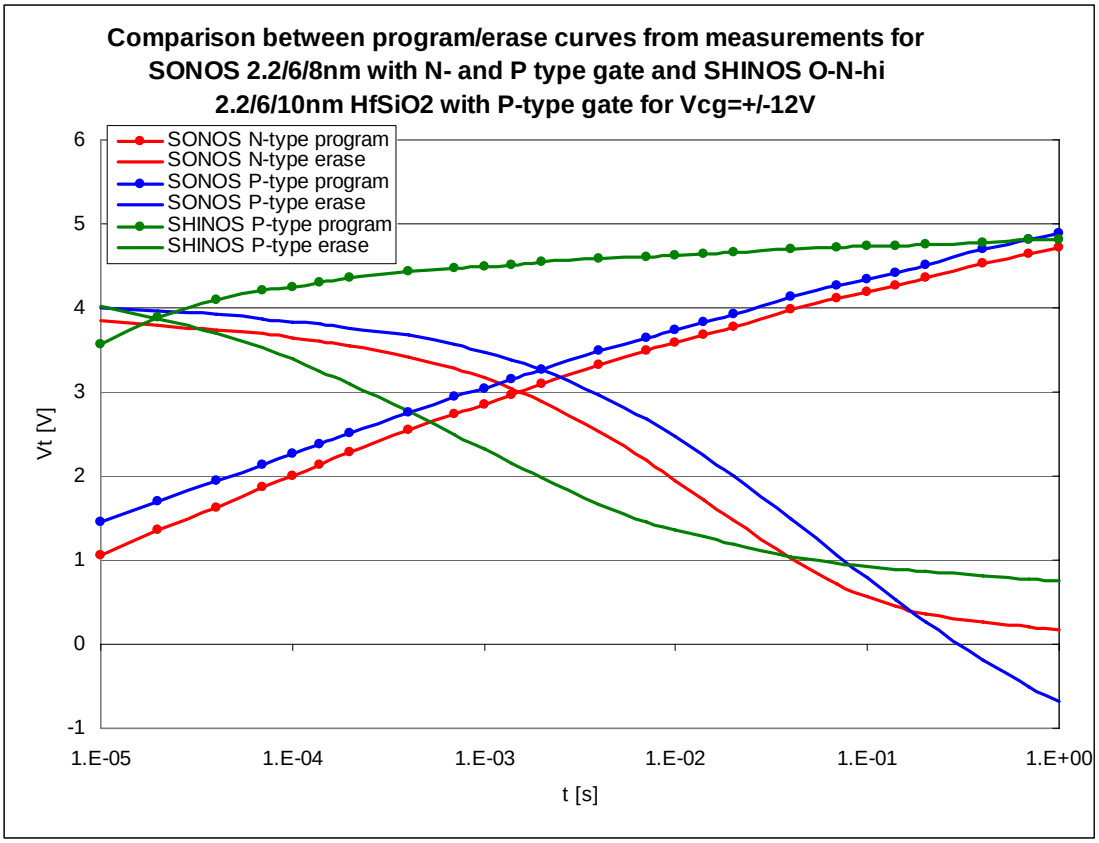

Figure 47: Comparison between program/erase curves from measurements at Vcg=+/-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) with N- and P-type gates.

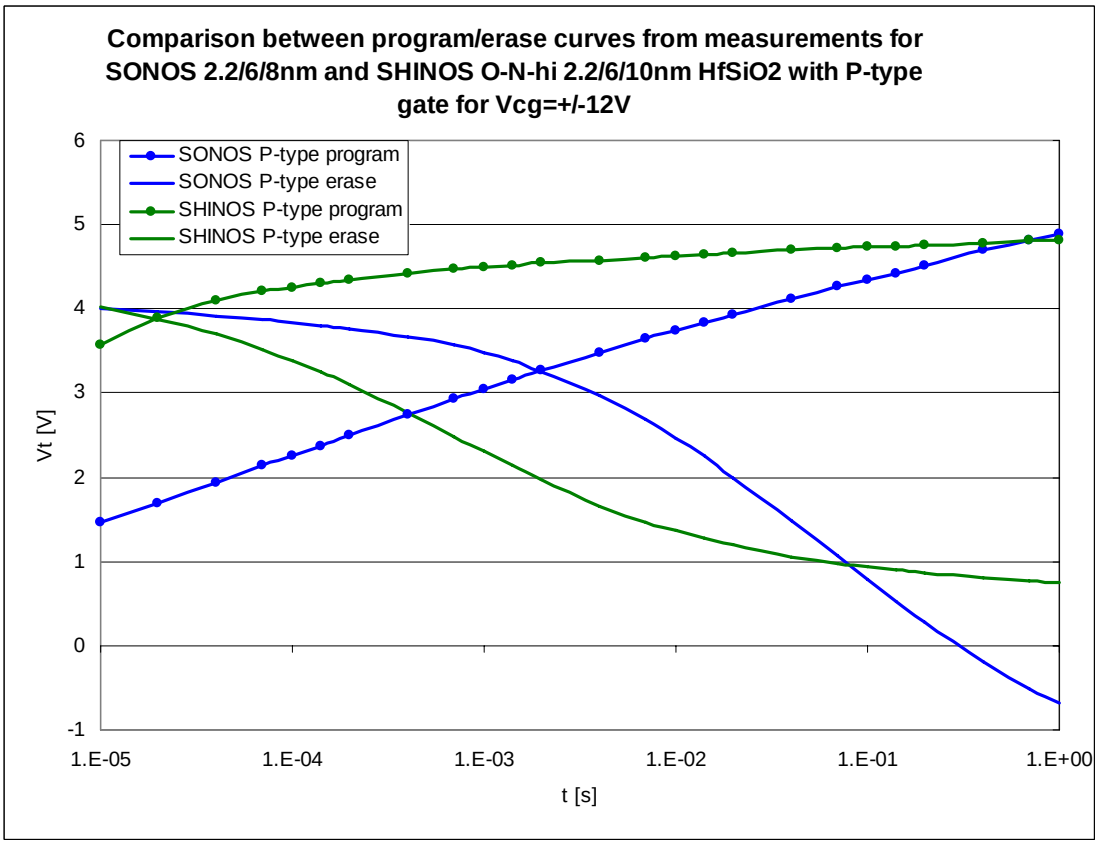

Figure 48: Comparison between two program/erase curves from measurements at Vcg=+/-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) with P-type gate.

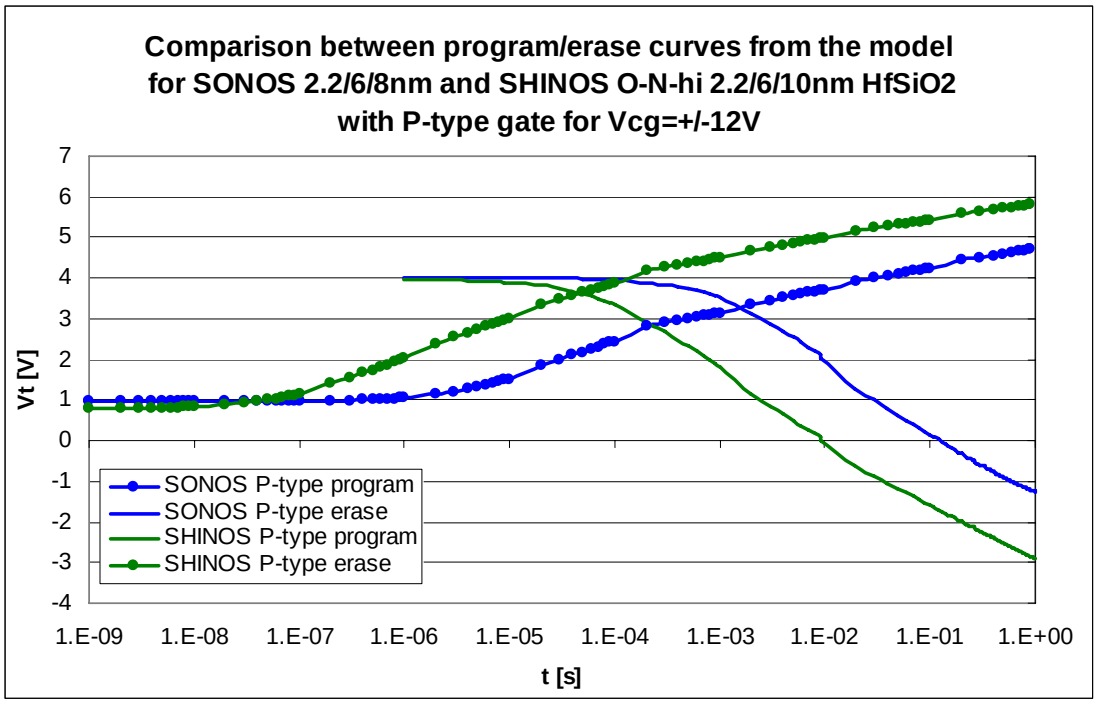

Figure 49: Comparison between two program/erase curves from the model at Vcg=+/-12V for a SONOS (2.2/6/8nm) and a SHINOS (2.2/6/10nmhfsiO2) with P-type gate.

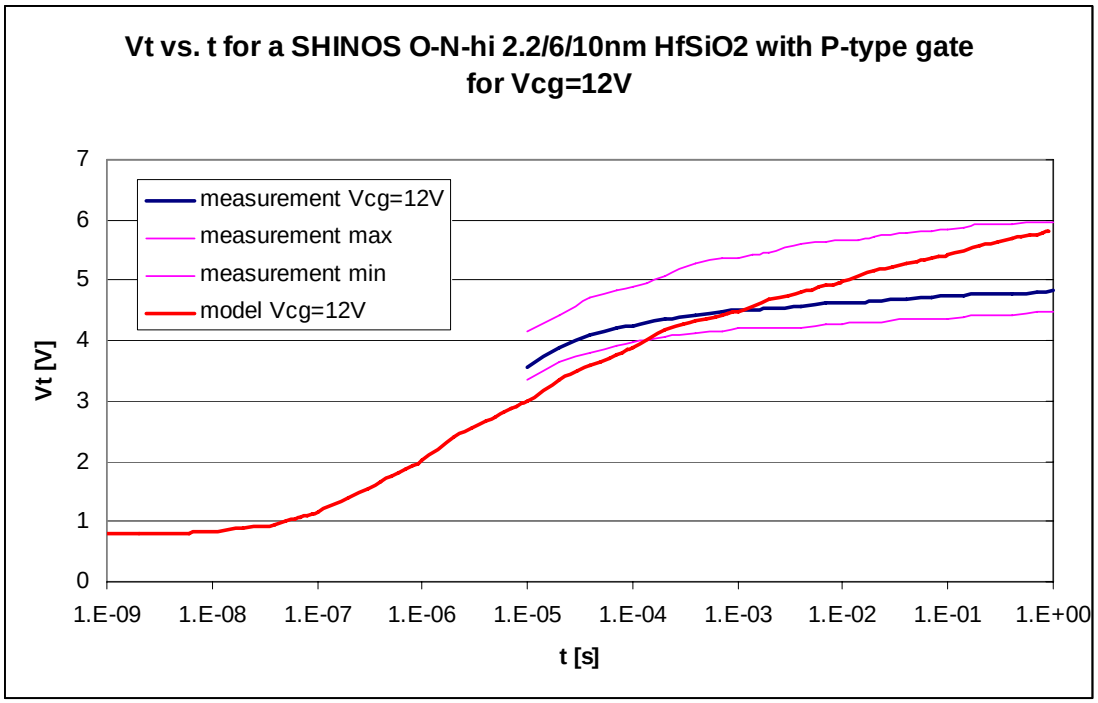

Figure 50: Program curves for a SHINOS (2.2/6/10nmhfsiO2) stack with a 12V

control gate voltage. The measurements were done on 1T mini array (256bits). n_{th}(t=0)is taken 0.1e^{19}/cm^3.

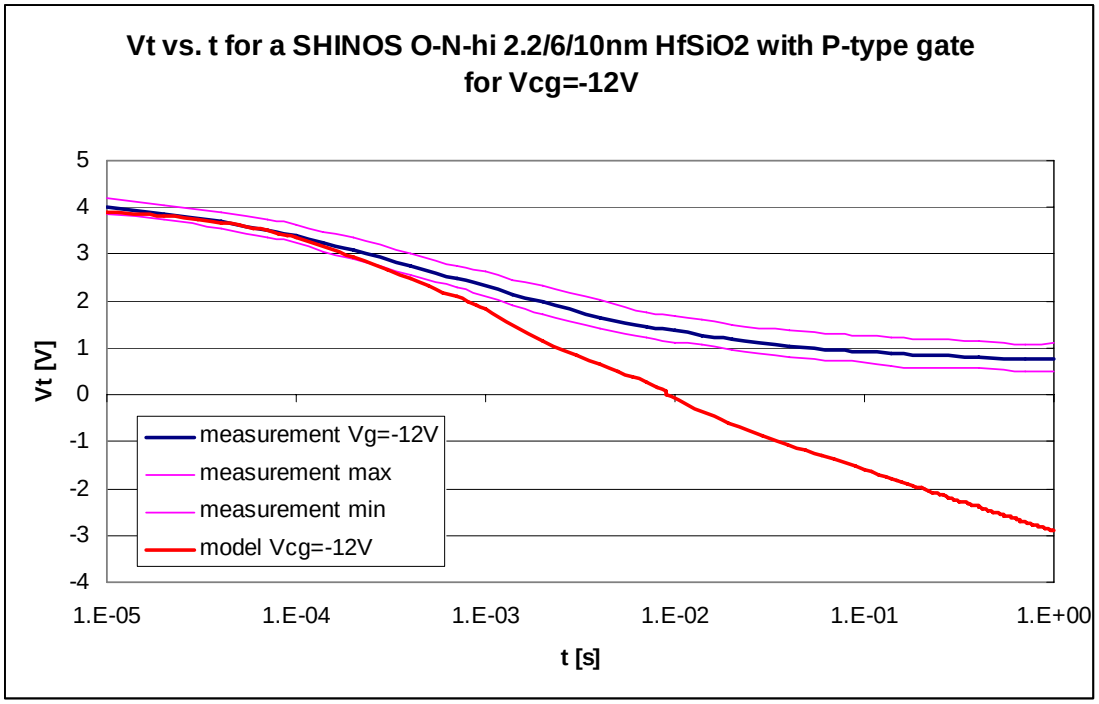

Figure 51: Erase curves for a SHINOS (2.2/6/10nmhfsiO2) stack with a -12V control gate voltage. The measurements were done on 1T mini array (256bits). n_{te}(t=0)is taken 1.5e^{19}/cm^3.

CONCLUSION

In this work, we developed a model implemented in a Matlab program, which can be easily used to predict the program/erase curves (threshold voltage shift of a SONOS device versus program/erase time for a defined control gate voltage). The user can easily modify several parameters within the model (e.g. bottom and top oxide thickness, k values, n or p-type gate, barrier heights) and see how these changes affect the behavior of physical quantities within the simulated stack (e.g. electric fields in the stack, injected currents, trapped charge location).

The model yields good agreement with the measurements for a reference SONOS stack, 2.2/6/8nm N-type gate, and also for P-type gate. Also good agreement with measurements is obtained for different ONO stack variations, such as bottom and top oxide thickness variations. In case the top oxide is substituted by a high-k material (SHINOS), the model yields qualitatively good results but the unkwon values of the barrier heights of the high-k makes it difficult to have quantitative agreement between the model and measurements, especially for the erase saturation level.

REFERENCES

[Bach2001] H. Bachhofer, H. Reisinger, E. Bertagnolli, H. von Philipsborn, "Transient conduction in multidielectric silicon – oxide – nitride – oxide semiconductor structures", J. Appl. Phys., Vol. 89, Num. 5, 1st March 2001

[Sze1981] S.M. Sze, "Physics of semiconductor devices", Wiley Inter-Science Publication, New York, 1981, Ch. 7.

APPENDIX

A.1. General outline

The model for program and erase was realized with Matlab and is structured as shown in Fig 52.

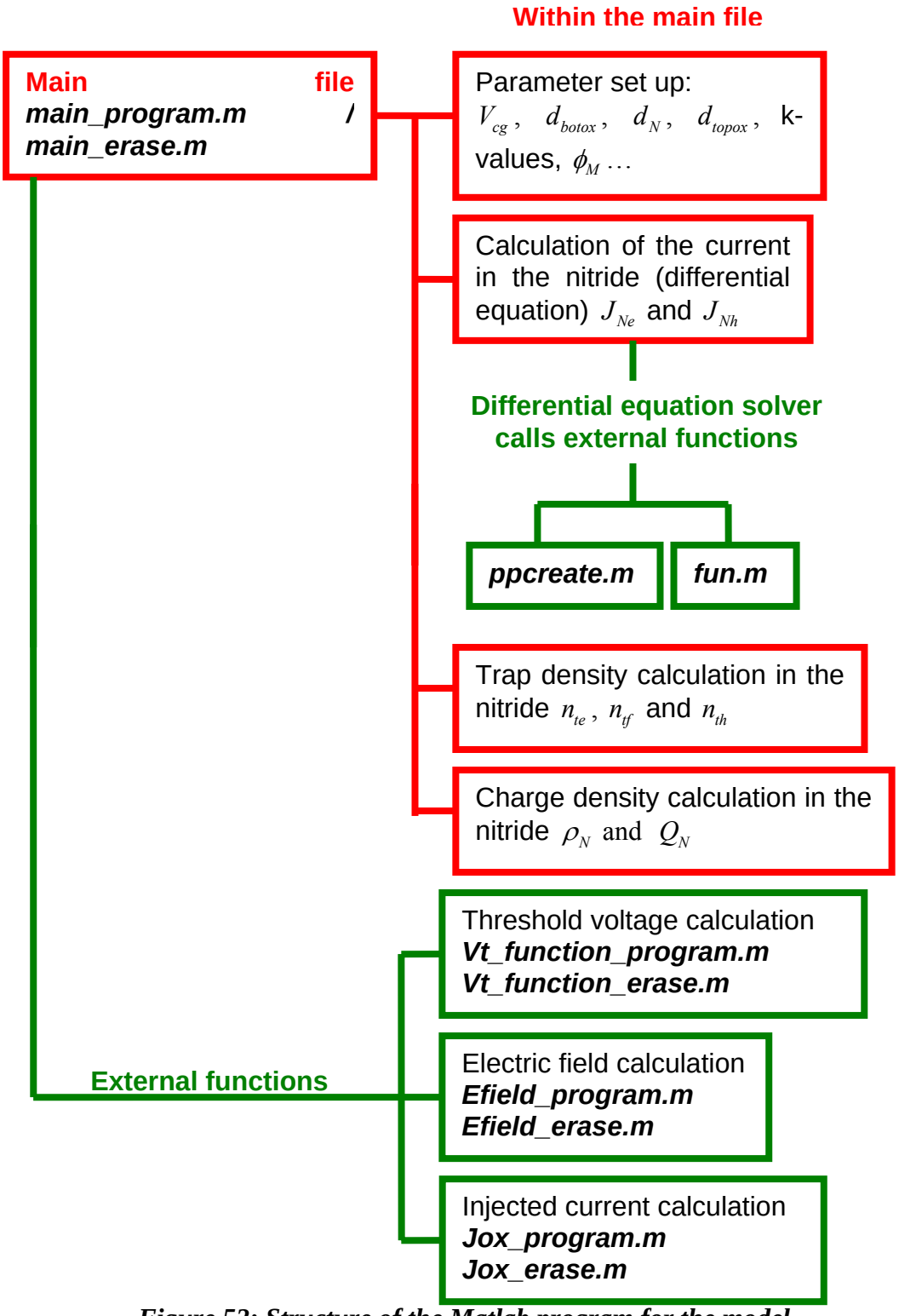

Figure 52: Structure of the Matlab program for the model

The program is easy to use. The user has to save the six Matlab files (six files for program and six files for erase) in the same folder and just run the main file for program or erase.

The main file (**main_program.m / main_erase.m**) is used to set up all the parameters and constants, which are shared with all the external functions.
The calculation of the electron and hole current densities in the nitride (Equation 1.23) is carried out in the main file. We call the external function **fun.m** to solve the differential equation (1.23). We also need to call the external function **ppcreate.m** (built in Matlab function), which provides a polynomial fit of n_{te} and n_{th}, in the right hand side of the differential equation (1.23).
The calculation of the trap density in the nitride is also carried out within the main file.
Finally, the charge density charge distribution in the silicon nitride (ρ_N) and the charge per unit area in the silicon nitride (Q_N) are calculated in the main file.

The main file calls three important external functions, apart from fun.m and ppcreate.m.

- **Vt_function_program.m** (resp. **Vt_function_erase.m**) calculates the threshold voltage at each point in time.
- **Efield_program.m** (resp. **Efield_erase.m)** calculates the electric field at each point in time.
- **Jox_program.m** (resp. **Jox_erase.m**) calculates the injected currents in the nitride at each point in time.

A.2. Matlab program for the model

A.2.1. Program case

A.2.1.1. main_program.m

```
global Ebotox Etopox Vg
global JDTe JMFNe psiB Qn nt rhon
global psiB
global Eg chi e0 etopox eni ebotox eS eox dtopox dn dbotox q
global Nt Na sigma v0 k Fit q pi e0 eopt
global X a x
global Ids Vd phiM
global mu_e W L
global Vt
global Fi1 Fi2 Fi3


%%%%%%%%%%%%%%%%% Definition of parameters %%%%%%%%%%%%%%%%%%%%%%%
disp('Programming case. Press Ctrl + C to abort this program.');
Vg   =   input('control   gate   voltage   for   PROGRAMMING   [V]:   ');
% Control gate voltage
phiM = input('gate work function phiM (Ntype=4.17, Ptype=5.2): ');
% Gate work function

dbotox         =         input('dbotox         [nm]:         ')*1e-7;
% thickness of the tunneling oxide (in cm)
dn                                =                                 6*1e-7;
% thickness of the nitride (in cm)
dtopox         =         input('dtopox         [nm]:         ')*1e-7;
% thickness of the blocking oxide (in cm)

Qox = input('fixed charge in the oxide: ');     % fixed oxide charge

dx = 1e-8;          % x-step = 0.1nm in the nitride
nstep = dn/dx;      % number of x-steps in the nitride


e0=8.85418*1e-14;   % permittivity in vaccuum (in F/cm)
ebotox    =    input('k-value    in    the    bottom    oxide:    ');
% k-value in the bottom oxide
eni                                =                                7.5;
%k-value in the nitride
etopox    =    input('k-value    in    the    top    oxide:    ');
% k-value in the top oxide
eox = 3.9;
eS = 11.8;

q = 1.60218*1e-19;  % elementary charge (in Coul)
psiB                              =                              0.4070818;
% potential barrier between the Fermi level and the intrinsic Fermi
level
chi = 4.17;         % electron affinity in Si
Eg = 1.12;          % Si band gap @ 300K (in eV)

W = 0.18e-4;        % gate width (in cm
L = 0.2e-4;         % gate length (in cm)
```



```
mu_e     =     input('electron     mobility,     default=130:     ');
% electron semiconductor mobility (in cm^2/(V*s))
Ids                              =                                10e-6;
% drain to source reading current (in A) used for the current crite-
rion in Vt
Vd = 0.5;               % drain voltage (in V)


Area = W*L;             % stack area (in cm^2)
Na = 1e17;              % bulk doping per cm-3

sigma=1.*1e-13;         % Trap capture cross section per cm-2.
v0=1.*10^9;             % Frequency factor per s-1.
k=(1.38066.*10^-23);    % Boltzmann constant J/K.
Fit=1.1;                % Trap depth in V.
q=1.60218*1e-19;        % Elementary charge (in Coul)
T=293;                  % Temperature in K.
pi=3.1416;              % pi
eopt= 4;                % optical dielectric constant


Ids = 10e-6;            % drain to source reading current (in A)
Vd = 0.5;               % drain voltage (in V)

Fi1= 3.1;               %Bottom oxide barrier height in V.
Fi2= 1.05;              %Nitride Tunneling barrier Height in V.
Fi3=input('Top oxide barrier height in V, reference value=4.8: ');
%Top oxide barrier height in V.
ygamma=eni/eox;


%%%%%%%%%%%%%%%%%%%%%%% End of definition of parameters %%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%% Step t=0: Initial conditions %%%%%%%%%%%%%%%%%
X                   =                   zeros(61,                   1);
% 61 = a+1 = number of steps from 0 to 6nm with a 0.1nm step
for i=1:61
    X(i,1) = i*dx;              % abscissa column vector
end

% Initialization
Nt=                                                                 5*1e19;
% total trap density per cm-3: accounts for traps filled with e/h AND
empty traps
Ntmatrix = Nt * ones(61, 1);

nth = input('enter the amount of traps filled with holes in the ni-
tride @t=0s, default=0.1e19/cm^3: ')*ones(61, 1); %density of traps
filled by holes
nte = zeros(61, 1);  % density of traps filled by electrons
ntf = Ntmatrix - nth - nte;       % density of free traps
```



```
rhon = nth - nte;   % charge distribution in the nitride
Qn = trapz(X,rhon); % integrates rhon using the trapezoidal rule
x = (0 : dx : dn)';

disp('enter the color of the Vt curve you want:');
R = input('red level: ');
G = input('green level: ');
B = input('blue level: ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t0 =1e-9;
t_end = 1;
t_switch = 1;
t = t0;
dt = t0;
Vt_data = zeros(2000,1);
t_data = zeros(2000,1);
j = 0;
Ebotox_data = zeros(2000,1);
Etopox_data = zeros(2000,1);

nte_data = zeros(61,1);
ntf_data = zeros(61,1);
nth_data = zeros(61,1);

Jne_data = zeros(61,1);
Jnh_data = zeros(61,1);

tic
while t <= t_end



    if dt >= t_switch
        disp('case dt=cte')
        disp('-----------------------------------------------------')
        t
        disp('-----------------------------------------------------')

        dt = t_switch

        Qn

        %%%%%%%%%%%%%%%%%%%%%%%% Preparation step for t=dt %%%%%%%%%

        % Call of the function which calculates Vt
        Vt = Vt_function_program(Qn, Qox)
        j =j+1;
        Vt_data(j) = Vt;
        t_data(j)=t;


        [Etopox, Ebotox] = Efield_program(nte, nth)
        Ebotox_data(j) = Ebotox;
        Etopox_data(j) = Etopox;

        [JDTe, JMFNe, JFNh] = Jox_program(Vg, nte, nth, phiM)    %
[A/cm^2]
```



```
        %%%%%%%% Step t=dt: solve for rhon(x,t=dt)%%%%%%%%%%%%%%

        % 1st boundary condition: @ botox interface Jn=JDTe
        if Ebotox >(Fi1-Fi2)/(dbotox) &  Ebotox < Fi1/(dbotox)
            disp('JDTe case')
            Jne                         =                         JDTe;
% boundary condition: @ botox interface Jn=JDTe
        end

        if  Ebotox  >(Fi1-Fi2)./(dbotox-ygamma*dn)  &  Ebotox  <=  (Fi1-
Fi2)/(dbotox)
            disp('JMFNe case')
            Jne = JMFNe;
        end
        y0e = Jne


        if Etopox >(Fi3)/dtopox & Etopox < 1.5*10^7
            disp('JFNh case')
            Jnh = JFNh;
        else
            Jnh = 0;
        end
        y0h = Jnh

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        u = (0 : dx : dn)';
        umax = dn;
        uspan = [0 umax]';
        du = 1e-8;
        ue = (0:du:umax)';
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Solve for Jnh  (differential equation)
        ye = nth;
        y0h;

        rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
        [u, y] = ode15s(rhp, (0:du:umax)', y0h);

        Jnh = y;

        for i = 1:61
            JJnh(i) = Jnh(62-i);
        end

        Jnh = JJnh';

        for i = 1:61
           Jnh_data(i)= Jnh(i);
        end

        for i = 1:61
           if Jnh(i) < 0
               Jnh(i) = 0;
           end
        end
```



```
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Solve for Jne  (differential equation)
        ye = nte;
        y0e;

        rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
        [u, y] = ode15s(rhp, (0:du:umax)', y0e);

        Jne = y;

        for i = 1:61
           Jne_data(i)= Jne(i);
        end
        for i = 1:61
           if Jne(i) < 0
               Jne(i) = 0;
           end
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        for i = 1:61
        %disp('1st update')
        nte(i) = nte(i) + 1/q.* Jne(i) .*sigma.* (nth(i) + ntf(i))
*ntf(i)/(nth(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
        nth(i) = nth(i) - 1/q.* Jne(i) .*sigma.* (nth(i) + ntf(i))
*nth(i)/(nth(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);

        %disp('2nd update')
        nth(i) = nth(i) + 1/q.* Jnh(i) .*sigma.* (nte(i) + ntf(i))
*ntf(i)/(nte(i) + ntf(i)) *dt;
        nte(i) = Ntmatrix(i) - nth(i) - nte(i);
        nte(i) = nte(i) - 1/q.* Jnh(i) .*sigma.* (nte(i) + ntf(i))
*nte(i)/(nte(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);

        nte_data(i) = nte(i);
        ntf_data(i) = ntf(i);
        nth_data(i) = nth(i);
        end


        % rhon(x,dt) update
        rhon = nth - nte;

        figure(1)
        semilogx(t, Vt,'.','color',[R,G,B])
        title('Vt vs. t')
        xlabel('t [s]')
        ylabel('Vt [V]')
        hold all

        figure(2)
        subplot(2,4,2)
        semilogy(Ebotox, y0e,'r.')
        title(' Jox vs Ebotox')
        xlabel('Ebotox [V/cm]')
```

```
        ylabel('Jox [A/cm2]')
        hold all

        subplot(2,4,3)
        semilogx(t, Ebotox, 'r.', t, Etopox, 'b.')
        title('Ebotox red Etopox blue')
        xlabel('t [s]')
        ylabel('Ebotox, Etopox [V/cm]')
        hold all

        %figure(4)
        subplot(2,4,4)
        plot(x, nth,'b.')
        title('hole traps')
        xlabel('x [cm]')
        ylabel('nt [1/cm3]')
        grid on

        subplot(2,4,5)
        plot(x, nte,'r.')
        title('electron traps')
        xlabel('x [cm]')
        ylabel('nt [1/cm3]')
        grid on

        subplot(2,4,6)
        plot(x, ntf,'g.')
        title('empty traps')
        xlabel('x [cm]')
        ylabel('nt [1/cm3]')
        grid on
        %hold all


        subplot(2,4,7)
        plot(x, Jne,'r.')
        title('e current')
        xlabel('x [cm]')
        ylabel('Jne, Jnh [A/cm2]')
        grid on

        subplot(2,4,8)
        plot( x, Jnh, 'b.')
        title('h current')
        xlabel('x [cm]')
        ylabel('Jne, Jnh [A/cm2]')
        grid on



        % Update of Qn
        Qn      =      trapz(x      ,      nth      -      nte);
%integrates rhon using the trapezoidal rule


        disp('-------------------------------------------------')

        %pause
```



```
        t = t + dt;


    else

        for i = 1: 9

        disp('case dynamic dt')
        disp('-----------------------------------------------------')
            t = i*dt
        disp('-----------------------------------------------------')




            %%%%%%%%%%%%%% Preparation step for t=dt %%%%%%%%%%%%

            % Call of the function which calculates Vt
            Vt = Vt_function_program(Qn, Qox)
            if t == t0
                Vt0 = Vt;
            end

            %%%%% t_switch  detection loop %%%%%%%%%%%%%%%%%%%%%%%%
            if abs(Vt0 - Vt) >7
                t_switch = t
                break
            end

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            j =j+1;
            Vt_data(j) = Vt;
            t_data(j)=t;

            Qn

            [Etopox, Ebotox] = Efield_program(nte, nth)
            Ebotox_data(j) = Ebotox;
            Etopox_data(j) = Etopox;

            [JDTe, JMFNe, JFNh] = Jox_program(Vg, nte, nth, phiM)


            %%%%%%%%%%% Step t=dt: solve for rhon(x,t=dt)%%%%%%%%%%
            % 1st boundary condition: @ botox interface Jn=JDTe
            if Ebotox >(Fi1-Fi2)/(dbotox) &  Ebotox < Fi1/(dbotox)
                disp('JDTe case')
                Jne                         =                       JDTe;
% boundary condition: @ botox interface Jn=JDTe
            end

            if  Ebotox  >(Fi1-Fi2)./(dbotox-ygamma*dn)  &  Ebotox  <=
(Fi1-Fi2)/(dbotox)
                disp('JMFNe case')
                Jne = JMFNe;
```

```
end
y0e = Jne


if Etopox >(Fi3)/dtopox & Etopox < 1.5*10^7
    disp('JFNh case')
    Jnh = JFNh;
else
    Jnh = 0;
end
y0h = Jnh

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u = (0 : dx : dn)';
umax = dn;
uspan = [0 umax]';
du = 1e-8;
ue = (0:du:umax)';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for Jnh  (differential equation)
ye = nth;
y0h;

rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
[u, y] = ode15s(rhp, (0:du:umax)', y0h);

Jnh = y;

for i = 1:61
    JJnh(i) = Jnh(62-i);
end

Jnh = JJnh';

for i = 1:61
   Jnh_data(i)= Jnh(i);
end
for i = 1:61
   if Jnh(i) < 0
       Jnh(i) = 0;
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for Jne  (differential equation)
ye = nte;
y0e;

rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
[u, y] = ode15s(rhp, (0:du:umax)', y0e);


Jne = y;

for i = 1:61
   Jne_data(i)= Jne(i);
end

for i = 1:61
```

```
            if Jne(i) < 0
                Jne(i) = 0;
            end
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        for i = 1:61
        %disp('1st update')
        nte(i)  =  nte(i)  +  1/q.*  Jne(i)  .*sigma.*  (nth(i)  +
ntf(i)) *ntf(i)/(nth(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
        nth(i)  =  nth(i)  -  1/q.*  Jne(i)  .*sigma.*  (nth(i)  +
ntf(i)) *nth(i)/(nth(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);

        %disp('2nd update')
        nth(i)  =  nth(i)  +  1/q.*  Jnh(i)  .*sigma.*  (nte(i)  +
ntf(i)) *ntf(i)/(nte(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
        nte(i)  =  nte(i)  -  1/q.*  Jnh(i)  .*sigma.*  (nte(i)  +
ntf(i)) *nte(i)/(nte(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);


        nte_data(i) = nte(i);
        ntf_data(i) = ntf(i);
        nth_data(i) = nth(i);
        end


        % rhon(x,t) update
        rhon = nth - nte;


        figure(1)
        semilogx(t, Vt,'.','color',[R,G,B])
        title('Vt vs. t')
        xlabel('t [s]')
        ylabel('Vt [V]')
        hold all

        figure(2)
        subplot(2,4,2)
        semilogy(Ebotox, y0e,'r.')
        title(' Jox vs Ebotox')
        xlabel('Ebotox [V/cm]')
        ylabel('Jox [A/cm2]')
        hold all

        subplot(2,4,3)
        semilogx(t, Ebotox, 'r.', t, Etopox, 'b.')
        title('Ebotox red Etopox blue')
        xlabel('t [s]')
        ylabel('Ebotox, Etopox [V/cm]')
        hold all

        subplot(2,4,4)
        plot(x, nth,'b.')
        title('hole traps')
```



```
            xlabel('x [cm]')
            ylabel('nt [1/cm3]')
            grid on

            subplot(2,4,5)
            plot(x, nte,'r.')
            title('electron traps')
            xlabel('x [cm]')
            ylabel('nt [1/cm3]')
            grid on

            subplot(2,4,6)
            plot(x, ntf,'g.')

            title('empty traps')
            xlabel('x [cm]')
            ylabel('nt [1/cm3]')
            grid on

            subplot(2,4,7)
            plot(x, Jne,'r.')
            title('e current')
            xlabel('x [cm]')
            ylabel('Jne, Jnh [A/cm2]')
            grid on

            subplot(2,4,8)
            plot( x, Jnh, 'b.')
            title('h current')
            xlabel('x [cm]')
            ylabel('Jne, Jnh [A/cm2]')
            grid on


            % Update of Qn
            Qn      =      trapz(x      ,      nth      -      nte);
% integrates rhon using the trapezoidal rule

            disp('-----------------------------------------------')
            %pause
        end
        dt = 10*dt;

    end


end

save my_Vt_data.out Vt_data -ASCII
save my_Ebotox_data.out Ebotox_data -ASCII
save my_Etopox_data.out Etopox_data -ASCII

save my_t_data.out t_data -ASCII

save my_nte_data.out nte_data -ASCII
save my_ntf_data.out ntf_data -ASCII
save my_nth_data.out nth_data -ASCII
```



```
save my_Jne_data.out Jne_data -ASCII
save my_Jnh_data.out Jnh_data -ASCII

disp('time elapsed in second')
time_elapsed = toc
disp('end of the calculation')
```

A.2.1.2. Vt_function_program.m

```
function Vt = Vt_function_program(Qn, Qox)

% Vt_function returns the value of the threshold voltage Vt
% Vt_function input arguments are: the charge in the nitride
% Qn, the fixed charge in the oxide Qox
% All the input arguments are set by the user in the main file

% List of global variables
global Vfb
global psiS psiB
global Eg chi e0 etopox eni ebotox eS dtopox dn dbotox q
global mu_e mu_h W L A Ceff Ids Vd
global Na phiM

% Vt calculation
phiMS      =      phiM      -(chi      +      Eg/2      +      psiB);
% metal semiconductor workfunction difference

dncentroid                              =                              dn/2;
% charge centroid (in cm)
Ceff = 1/(dtopox/(etopox*e0) + dn/(eni*e0) + dbotox/(ebotox*e0));

Vfb  =  phiMS  -  (Qn  +  Qox)*q*(dtopox/(etopox*e0)  +  (dn  -  dncen-
troid)/(eni*e0));  % flat-band voltage (in V)

psiS = 2*psiB ;

BBB=                                                    mu_e*Ceff*W/L;
% slope of the tangent of Ig(Vg) at the inflexion point

Vt  =  sqrt(2*eS*e0*q*Na*2*psiB)./Ceff  +  psiS  +  Vfb  +  Ids./(Vd*BBB)-
0.4;      % threshold voltage (in V) with current criterion

% we add 0.4V to account for the narrow width effect
return
```

A.2.1.3. Efield_program.m

```
function [Etopox, Ebotox] = Efield_program(nte, nth)

% Eox_function returns the values of Etopox and Ebotox,
% the electric field in the top and bottom oxide, respectively.
% Eox_function input arguments are: the amount of traps filled
% with electrons nte and holes nth
% All the input arguments are set by the user in the main file
```



```
% List of global variables
global Ebotox Etopox
global psiB
global Eg chi e0 etopox eni ebotox dtopox dn dbotox q
global a phiM x Vg t


% Calculation of the metal semiconductor workfunction difference
phiMS = phiM -(chi + Eg/2 + psiB);

% Calculation of the flat-band voltage (in V)
Vfb = phiM -(chi + Eg/2 + psiB);

% Programming case: if the control gate voltage Vg is positive
psiS = 1.03 - 0.4 + Vfb;
% potential drop in the bulk in case of depletion
% we add 1.03V to account for the inversion and quantum mechanical
effect
% and -0.4V to account for narrow width effects

% system of equations to solve
A = [ dtopox, dbotox + dn*ebotox/eni
        etopox, -ebotox];

pp0 = ppcreate(x,nth-nte,'notaknot');
pp1 = ppcreate(pp0,'int');
pp2 = ppcreate(pp1,'int');

firstelt =   Vg - psiS + (q)/(e0*eni)* (pp2(dn));

B = [firstelt
   -q/e0* (pp1(dn)) ];

X = A\B;

Etopox = X(1);   % Efield in the top oxide
Ebotox = X(2);   % Efield in the bottom oxide

% Plot of rhon and the fitting curve RHON
figure(2)
subplot(2,4,1)
pp0('plot');
hold on
plot(x, nth - nte, 'r.')
title('rhon(x)')
xlabel('x [cm]')
ylabel('rhon [1/cm3]')
hold off

return
```

A.2.1.4. Jox_program.m

```
function [JDTe, JMFNe, JFNh] = Jox_program(Vg, nte, nth, phiM)
```



```
% Jox_function returns the values of the tunneling current
%  through the oxide: Jtopox and Jbotox, the tunneling current
% in the top and bottom oxide, respectively.
% Jox_function input arguments are: the gate voltage Vg,
% the amount of traps filled with electrons nte and holes nth
% and the gate work function phiM.
% All the input arguments are set in the main file


% List of global variables
global Na Ebotox Etopox
global Vt JDTe JMFNe JFNh
global Vfb
global psiS psiB
global Eg chi e0 etopox eni ebotox eox dtopox dn dbotox
global eox ehik elowk hb k pi eopt Nt sigma v0 psis Fims qss T q
global Fi1 Fi2 Fi3


m0=0.91095.*10^-30;     % electron rest mass in Kg.
mn=0.1*m0;              % effective electron mass in nitride.
hb=(1.05458*10^-34);    % Reduced Planck constant in J-s.

% Call of the function Eox_function in order to get Etopox and Ebotox
values
[Etopox, Ebotox] = Efield_program(nte, nth);

% JDTe: Direct tunneling electron current density in A/cm2
if Ebotox >(Fi1 - Fi2)/(dbotox) &  Ebotox < Fi1/(dbotox)
    mox=0.32*m0*(1e7/Ebotox)^(1.25);

JDTe=(m0./mox).*((q.^3)./(16.*(pi^2).*hb)).*((Ebotox).^2).*(1./((((q.
*Fi1).^0.5)-((q.*Fi1-q.*Ebotox.*dbotox).^0.5)).^2)).*exp(-
((4.*((2.*mox).^0.5).*(0.01)).*(((q.*Fi1).^1.5)-((q.*Fi1-
q.*Ebotox.*dbotox).^1.5)))./(3.*q.*hb.*Ebotox));
end

% JMFNe: Modified Fowler Nordheim electron current density in A/cm2
ygamma=eni./eox;
if   Ebotox   >(Fi1-Fi2)./(dbotox-ygamma*dn)   &   Ebotox   <   (Fi1-
Fi2)/(dbotox)
    mox=0.32*m0*(1e7/Ebotox)^(1.25);

JMFNe=(m0/mox).*((q^3)/(16.*(pi^2).*hb)).*((Ebotox)^2).*(1/((((q.*Fi1
)^0.5)-((q.*Fi1-
q.*Ebotox.*dbotox)^0.5)+(ygamma.*(mn/mox)^0.5.*(q.*Fi1-q.*Fi2-
q.*Ebotox.*dbotox)^0.5))^2)).*exp(-
(((4.*((2.*mox)^0.5).*(0.01)).*(((q.*Fi1)^1.5)-((q.*Fi1-
q.*Ebotox.*dbotox)^1.5)))+ ygamma.*4.*(0.01).*((2.*mn)^0.5).*(q.*Fi1-
q.*Fi2-q.*Ebotox.*dbotox)^1.5)/(3.*q.*hb.*Ebotox));
end


% JFNh, Fowler Nordheim hole current density in A/cm^2
if Etopox >(Fi3)/dtopox & Etopox < 1.5*10^7
    AFN=1*10*6.32.*(10^-7).*(3.1/Fi3);                  %in A/V^2
    BFN=1*2.4.*(10^8).*((Fi3/3.1)^1.5);                 %in V/cm
    JFNh= AFN.*((Etopox)^2).*exp(-BFN/Etopox);
end
```



```
return
```

A.2.1.5. fun.m

```
function dy = fun(u, y, Nt, ue, ye, sigma)

% fun - get right-hand side using
% interpolation of experimental data

dy =  -sigma*y .*(Nt - interp1(ue, ye, u,'spline'));

return
```

A.2.2. Erase case

A.2.2.1. main_erase.m

```
global Ebotox Etopox Vg
global JDT JMFN psiB Qn nte nth rhon
global psiB
global Eg chi e0 etopox eni ebotox eS eox dtopox dn dbotox q
global Nt Na sigma v0 k Fit q pi e0 eopt
global X a x
global Ids Vd phiM
global mu_e W L
global Vt t
global Fi1 Fi2 Fi3


%%%%%%%%%%%%%%%%%% Definition of parameters %%%%%%%%%%%%%%%%%%%%%
disp('Erasing case. Press Ctrl + C to abort this program.');
Vg   =   input('control   gate   voltage   for   ERASING   [V]:   ');
% Control gate voltage
phiM = input('gate work function phiM (Ntype=4.17, Ptype=5.2): ');
% Gate work function

dbotox         =         input('dbotox         [nm]:         ')*1e-7;
% thickness of the tunneling oxide (in cm)
dn                              =                               6*1e-7;
% thickness of the nitride (in cm)
dtopox         =         input('dtopox         [nm]:         ')*1e-7;
% thickness of the blocking oxide (in cm)

Qox = input('fixed charge in the oxide: ');           % fixed oxide
charge

dx = 1e-8;       % x-step = 0.1nm in the nitride
nstep = dn/dx;   % number of x-steps in the nitride


e0=8.85418*1e-14;   % permittivity in vaccuum (in F/cm)
ebotox = input('k-value in the bottom oxide, reference value=3.9: ');
```

```
% k-value in the bottom oxide
eni                                 =                                 7.5;
% k-value in the nitride
etopox = input('k-value in the bottom oxide, reference value=3.9: ');
% k-value in the top oxide
eox = 3.9;
eS = 11.8;

q = 1.60218*1e-19;  % elementary charge (in Coul)
psiB                              =                             0.4070818;
% potential barrier between the Fermi level and the intrinsic Fermi
level
chi = 4.17;         % electron affinity in Si
Eg = 1.12;          % Si band gap @ 300K (in eV)

W = 0.18e-4;        % gate width (in cm)
L = 0.2e-4;         % gate length (in cm)



mu_e  =  input('mue  electron  mobility,  reference  value=130:  ');
% electron semiconductor mobility (in cm^2/(V*s))
Ids = 10e-6;            % drain to source reading current (in A)
Vd = 0.5;               % drain voltage (in V)

Area = W*L;             % stack area (in cm^2)
Na = 1e17;              % bulk doping per cm-3

sigma=1.*1e-13;         % Trap capture cross section per cm-2.
v0=1.*10^9;             % Frequency factor per s-1.
k=(1.38066.*10^-23);    % Boltzmann constant J/K.
Fit=1.1;                % Trap depth in V.
q=1.60218*1e-19;        % Elementary charge (in Coul)
T=293;                  % Temperature in K.
pi=3.1416;              % pi
eopt= 4;                % optical dielectric constant


Fi1=4.8;
%Bottom oxide barrier height in V.
Fi2=3.35;
%Nitride Tunneling barrier Height in V.
Fi3=input('Fi3,  Top  oxide  barrier  height  in  V,  default=3.1V:  ');
%Top oxide barrier height in V.
ygamma=eni/eox;


disp('enter the color of the Vt curve you want:')
R = input('red level: ');
G = input('green level: ');
B = input('blue level: ');
%%%%%%%%%%%%%%%% End of definition of parameters %%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%% Step t=0: Initial conditions %%%%%%%%%%%%%
X                   =                   zeros(61,                   1);
```



```
% 61 = a+1 = number of steps from 0 to 6nm with a 0.1nm step
for i=1:61
    X(i,1) = i*dx; % abscissa column vector
end

% Initialization
Nt=                                                          5*1e19;
% total trap density per cm-3: accounts for traps filled with e/h AND
empty traps
Ntmatrix = Nt * ones(61, 1);

nte = input('enter the amount of traps filled by electrons in the
nitride @t=0s, default value=1.15e19/cm^3: ')*ones(61, 1);         %
density of traps filled by holes
nth                  =                  zeros(61,                  1);
% density of traps filled by electrons
ntf        =        Ntmatrix        -        nth        -        nte;
% density of free traps

x = (0 : dx : dn)';
rhon              =              nth              -              nte;
% charge distribution in the nitride
Qn                           =                           trapz(x,rhon);
% integrates rhon using the trapezoidal rule


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t0 =1e-6;
t_end = 1;
t_switch = 1e-2; % temporary definition of t_switch
t = t0;
dt = t0;
dt_temp = 0;     % temporary definition of dt_temp
Vt_data = zeros(2000,1);
t_data = zeros(2000,1);
j = 0;
Ebotox_data = zeros(2000,1);
Etopox_data = zeros(2000,1);

nte_data = zeros(61,1);
ntf_data = zeros(61,1);
nth_data = zeros(61,1);

Jne_data = zeros(61,1);
Jnh_data = zeros(61,1);


tic
while t <= t_end



    if dt >= t_switch
        disp('case dt=cte')
        disp('-----------------------------------------------')
        t
        disp('-----------------------------------------------')
```



```
        dt = t_switch;

        Qn

        %%%%%%%%%%%% Preparation step for t=dt %%%%%%%%%%%%%%%%

        % Call of the function which calculates Vt
        Vt = Vt_function_erase(Qn, Qox)


        j =j+1;
        Vt_data(j) = Vt;
        t_data(j)=t;

        [Etopox, Ebotox] = Efield_erase(nte, nth)
        Ebotox_data(j) = Ebotox;
        Etopox_data(j) = Etopox;
        [JDT, JMFN, JFN] = Jox_erase(Vg, nte, nth, phiM) %[A/cm^2]



        %%%%%%%%% Step t=dt: solve for rhon(x,t=dt)%%%%%%%%%%%%%%

        % 1st boundary condition: @ botox interface Jn=JDTe
        if Ebotox >(Fi1-Fi2)/(dbotox) &  Ebotox < Fi1/(dbotox)
            disp('JDTh case')
            Jnh                          =                          JDT;
% boundary condition: @ botox interface Jn=JDTe
        end

        if  Ebotox  >(Fi1-Fi2)./(dbotox-ygamma*dn)  &  Ebotox  <=  (Fi1-
Fi2)/(dbotox)
            disp('JMFNh case')
            Jnh = JMFN;
        end
        y0h = Jnh


        if Etopox >(Fi3)/dtopox & Etopox < 1.5*10^7
            disp('JFNe case')
            Jne = JFN;
        else
            Jnh = 0;
        end
        y0e = Jne

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        u = (0 : dx : dn)';
        umax = dn;
        uspan = [0 umax]';
        du = 1e-8;
        ue = (0:du:umax)';
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Solve for Jnh  (differential equation)
        ye = nth;
        y0h;

        rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
```



```
        [u, y] = ode15s(rhp, (0:du:umax)', y0h);

        Jnh = y;

        for i = 1:61
           Jnh_data(i)= Jnh(i);
        end

        for i = 1:61
           if Jnh(i) < 0
               Jnh(i) = 0;
           end
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Solve for Jne  (differential equation)
        ye = nte;
        y0e;

        rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
        [u, y] = ode15s(rhp, (0:du:umax)', y0e);

        Jne = y;

        for i = 1:61
            JJne(i) = Jne(62-i);
        end

        Jne = JJne';

        for i = 1:61
           Jne_data(i)= Jne(i);
        end

        for i = 1:61
           if Jne(i) < 0
               Jne(i) = 0;
           end
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        for i = 1:61
        %disp('2nd update')
        nth(i) = nth(i) + 1/q.* Jnh(i) .*sigma.* (nte(i) + ntf(i))
*ntf(i)/(nte(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
        nte(i) = nte(i) - 1/q.* Jnh(i) .*sigma.* (nte(i) + ntf(i))
*nte(i)/(nte(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);

        %disp('1st update')
        nte(i) = nte(i) + 1/q.* Jne(i) .*sigma.* (nth(i) + ntf(i))
*ntf(i)/(nth(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
        nth(i) = nth(i) - 1/q.* Jne(i) .*sigma.* (nth(i) + ntf(i))
*nth(i)/(nth(i) + ntf(i)) *dt;
        ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
```



```
nte_data(i) = nte(i);
ntf_data(i) = ntf(i);
nth_data(i) = nth(i);
end



% rhon(x,dt) update
rhon = nth - nte;

figure(1)

semilogx(t, Vt,'.','color',[R,G,B])
title('Vt vs. t')
xlabel('t [s]')
ylabel('Vt [V]')
hold all

figure(2)
subplot(2,4,2)
semilogy(Ebotox, y0h,'c.')
hold on
semilogy(Etopox, y0e,'g.')
hold off
title(' Jox vs Eox')
xlabel('Ebotox blue,  Etopox red [V/cm]')
ylabel('Jox [A/cm2]')
hold all

subplot(2,4,3)
semilogx(t, Ebotox, 'b.', t, Etopox, 'r.')
title('Ebotox blue Etopox red')
xlabel('t [s]')
ylabel('Ebotox, Etopox [V/cm]')
hold all

subplot(2,4,4)
plot(x, nth,'b.')
title('hole traps')
xlabel('x [cm]')
ylabel('nt h [1/cm3]')
grid on

subplot(2,4,5)
plot(x, nte,'r.')
title('electron traps')
xlabel('x [cm]')
ylabel('nt e [1/cm3]')
grid on

subplot(2,4,6)
plot(x, ntf,'g.')
title('empty traps')
xlabel('x [cm]')
ylabel('nt f [1/cm3]')
grid on

subplot(2,4,7)
semilogy(x, Jne,'r.')
```

```
        title('e current')
        xlabel('x [cm]')
        ylabel('Jne, Jnh [A/cm2]')
        grid on

        subplot(2,4,8)
        semilogy( x, Jnh, 'b.')
        title('h current')
        xlabel('x [cm]')
        ylabel('Jne, Jnh [A/cm2]')
        grid on


        % Update of Qn
        Qn      =      trapz(x      ,      nth      -      nte);
% integrates rhon using the trapezoidal rule

        Vt_previous = Vt;


        disp('-----------------------------------------------------')

        %pause

        t = t + dt;


    else

        for i = 1: 9

        disp('case dynamic dt')
        disp('-----------------------------------------------------')
            t = i*dt
        disp('-----------------------------------------------------')




            %%%%%%%%%%%%%% Preparation step for t=dt %%%%%%%%%%%%%

            % Call of the function which calculates Vt
            Vt = Vt_function_erase(Qn, Qox)
            j =j+1;
            Vt_data(j) = Vt;
            t_data(j)=t;

            Qn

            [Etopox, Ebotox] = Efield_erase(nte, nth)
            Ebotox_data(j) = Ebotox;
            Etopox_data(j) = Etopox;

            [JDT, JMFN, JFN] = Jox_erase(Vg, nte, nth, phiM)
```



```
            %%%%%%%%%%% Step t=dt: solve for rhon(x,t=dt)%%%%%%%%%%
            % 1st boundary condition: @ botox interface Jn=JDTe
            if Ebotox >(Fi1-Fi2)/(dbotox) &  Ebotox < Fi1/(dbotox)
                disp('JDTh case')
                Jnh                         =                       JDT;
% boundary condition: @ botox interface Jnh=JDTh
            end

            if  Ebotox  >(Fi1-Fi2)./(dbotox-ygamma*dn)  &  Ebotox  <=
(Fi1-Fi2)/(dbotox)
                disp('JMFNh case')
                Jnh = JMFN;
            end
            y0h = Jnh


            if Etopox >(Fi3)/dtopox & Etopox < 1.5*10^7
                disp('JFNe case')
                Jne = JFN;
            else
                Jne = 0;
            end
            y0e = Jne

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            u = (0 : dx : dn)';
            umax = dn;
            uspan = [0 umax]';
            du = 1e-8;
            ue = (0:du:umax)';
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Solve for Jnh  (differential equation)
            ye = nth;
            y0h;

            rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
            [u, y] = ode15s(rhp, (0:du:umax)', y0h);

            Jnh = y;

            for i = 1:61
                JJnh(i) = Jnh(62-i);
            end

            for i = 1:61
               Jnh_data(i)= Jnh(i);
            end

            for i = 1:61
               if Jnh(i) < 0
                   Jnh(i) = 0;
               end
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Solve for Jne  (differential equation)
            ye = nte;
            y0e;

            rhp = @(u, y) fun(u, y, Nt, ue, ye, sigma);
```



```
            [u, y] = ode15s(rhp, (0:du:umax)', y0e);


            Jne = y;

            for i = 1:61
                JJne(i) = Jne(62-i);
            end

            Jne = JJne';

            for i = 1:61
               Jne_data(i)= Jne(i);
            end

            for i = 1:61
               if Jne(i) < 0
                   Jne(i) = 0;
               end
            end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for i = 1:61
            %disp('2nd update')
            nth(i)  =  nth(i)  +  1/q.*  Jnh(i)  .*sigma.*  (nte(i)  +
ntf(i)) *ntf(i)/(nte(i) + ntf(i)) *dt;
            ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
            nte(i)  =  nte(i)  -  1/q.*  Jnh(i)  .*sigma.*  (nte(i)  +
ntf(i)) *nte(i)/(nte(i) + ntf(i)) *dt;
            ntf(i) = Ntmatrix(i) - nth(i) - nte(i);

            %disp('1st update')
            nte(i)  =  nte(i)  +  1/q.*  Jne(i)  .*sigma.*  (nth(i)  +
ntf(i)) *ntf(i)/(nth(i) + ntf(i)) *dt;
            ntf(i) = Ntmatrix(i) - nth(i) - nte(i);
            nth(i)  =  nth(i)  -  1/q.*  Jne(i)  .*sigma.*  (nth(i)  +
ntf(i)) *nth(i)/(nth(i) + ntf(i)) *dt;
            ntf(i) = Ntmatrix(i) - nth(i) - nte(i);

            nte_data(i) = nte(i);
            ntf_data(i) = ntf(i);
            nth_data(i) = nth(i);
            end


            % rhon(x,t) update
            rhon = nth - nte;

            figure(1)
            semilogx(t, Vt,'.','color',[R,G,B])
            title('Vt vs. t')
            xlabel('t [s]')
            ylabel('Vt [V]')
            hold all

            figure(2)
            subplot(2,4,2)
            semilogy(Ebotox, y0h,'c.')
```



```
            hold on
            semilogy(Etopox, y0e,'g.')
            hold off
            title(' Jox vs Eox')
            xlabel('Ebotox blue,  Etopox red [V/cm]')
            ylabel('Jox [A/cm2]')
            hold all

            subplot(2,4,3)
            semilogx(t, Ebotox, 'b.', t, Etopox, 'r.')
            title('Ebotox blue Etopox red')
            xlabel('t [s]')
            ylabel('Ebotox, Etopox [V/cm]')
            hold all

            subplot(2,4,4)
            plot(x, nth,'b.')
            title('hole traps')
            xlabel('x [cm]')
            ylabel('nt h [1/cm3]')
            grid on

            subplot(2,4,5)
            plot(x, nte,'r.')
            title('electron traps')
            xlabel('x [cm]')
            ylabel('nt e [1/cm3]')
            grid on

            subplot(2,4,6)
            plot(x, ntf,'g.')
            title('empty traps')
            xlabel('x [cm]')
            ylabel('nt f [1/cm3]')
            grid on

            subplot(2,4,7)
            semilogy(x, Jne,'r.')
            title('e current')
            xlabel('x [cm]')
            ylabel('Jne, Jnh [A/cm2]')
            grid on

            subplot(2,4,8)
            semilogy( x, Jnh, 'b.')
            title('h current')
            xlabel('x [cm]')
            ylabel('Jne, Jnh [A/cm2]')
            grid on


            % Update of Qn
            Qn      =      trapz(x      ,      nth      -      nte);
% integrates rhon using the trapezoidal rule

            disp('-----------------------------------------------')
            %pause
        end
        if dt_temp ~= 0
```

```
            t_switch = 10*dt_temp
        end

        dt = 10*dt;

    end


end

save my_Vt_data.out Vt_data -ASCII
save my_t_data.out t_data -ASCII

save my_Ebotox_data.out Ebotox_data -ASCII
save my_Etopox_data.out Etopox_data -ASCII
save my_nte_data.out nte_data -ASCII
save my_ntf_data.out ntf_data -ASCII
save my_nth_data.out nth_data -ASCII

save my_Jne_data.out Jne_data -ASCII
save my_Jnh_data.out Jnh_data -ASCII

disp('time elapsed [s]')
time_elapsed = toc
disp('end of calculation')
```

A.2.2.2. Vt_function_erase.m

```
function Vt = Vt_function_erase(Qn, Qox)

% Vt_function returns the value of the threshold voltage Vt
% Vt_function input arguments are: the charge in the nitride Qn,
% the fixed charge in the oxide Qox
% All the input arguments are set by the user in the main file


% List of global variables
global Vfb
global psiS psiB
global Eg chi e0 etopox eni ebotox eS dtopox dn dbotox q
global mu_e mu_h W L A Ceff Ids Vd
global Na phiM


% Vt calculation
phiMS      =      phiM      -(chi      +      Eg/2      +      psiB);
% metal semiconductor workfunction difference

dncentroid = dn/2;           % charge centroid (in cm)
Ceff = 1/(dtopox/(etopox*e0) + dn/(eni*e0) + dbotox/(ebotox*e0));


Vfb  =  phiMS  -  (Qn+Qox)*q*(dtopox/(etopox*e0)  +  (dn  -  dncen-
troid)/(eni*e0)); % flat-band voltage (in V)
```



```
psiS = 2*psiB;


BBB=                                                    mu_e*Ceff*W/L;
% beta: slope of the tangent of Ig(Vg) at the inflexion point

Vt = sqrt(2*eS*e0*q*Na*2*psiB)./Ceff + psiS + Vfb + Ids./(Vd*BBB) -
0.4;
% threshold voltage (in V) with current criterion
%we add -0.4V to account for the narrow width effect

return
```

A.2.2.3. Efield_erase.m

```
function [Etopox, Ebotox] = Efield_program(nte, nth)

% Eox_function returns the values of Etopox and Ebotox,
% the electric field in the top and bottom oxide, respectively.
% Eox_function input arguments are: the amount of traps filled
% with electrons nte and holes nth
% All the input arguments are set by the user in the mainfile


% List of global variables
global Ebotox Etopox
global psiB
global Eg chi e0 etopox eni ebotox dtopox dn dbotox q
global a phiM x Vg t


% Calculation of the metal semiconductor workfunction difference
phiMS = phiM -(chi + Eg/2 + psiB);

% Calculation of the flat-band voltage (in V)
Vfb = phiM -(chi + Eg/2 + psiB);

% Programming case: if the control gate voltage Vg is positive
psiS = 1.03 - 0.4 + Vfb;
% potential drop in the bulk in case of depletion
% we add 1.03V to account for the inversion and quantum mechanical
effect
% and -0.4V to account for narrow width effects

% system of equations to solve
A = [ dtopox, dbotox + dn*ebotox/eni
        etopox, -ebotox];

pp0 = ppcreate(x,nth-nte,'notaknot');
pp1 = ppcreate(pp0,'int');
pp2 = ppcreate(pp1,'int');


firstelt =   Vg - psiS + (q)/(e0*eni)* (pp2(dn));
```



```
B = [firstelt
   -q/e0* (pp1(dn)) ];

X = A\B;

Etopox = X(1);   % Efield in the top oxide
Ebotox = X(2);   % Efield in the bottom oxide


% Plot of rhon and the fitting curve RHON
figure(2)
subplot(2,4,1)
pp0('plot');
hold on
plot(x, nth - nte, 'r.')
title('rhon(x)')
xlabel('x [cm]')
ylabel('rhon [1/cm3]')
hold off

return
```

A.2.2.4. Jox_ erase.m

```
function [JDTe, JMFNe, JFNh] = Jox_program(Vg, nte, nth, phiM)

% Jox_function returns the values of the tunneling current
%  through the oxide: Jtopox and Jbotox, the tunneling current
% in the top and bottom oxide, respectively.
% Jox_function input arguments are: the gate voltage Vg,
% the amount of traps filled with electrons nte and holes nth
% and the gate work function phiM.
% All the input arguments are set in the main file


% List of global variables
global Na Ebotox Etopox
global Vt JDTe JMFNe JFNh
global Vfb
global psiS psiB
global Eg chi e0 etopox eni ebotox eox dtopox dn dbotox
global eox ehik elowk hb k pi eopt Nt sigma v0 psis Fims qss T q
global Fi1 Fi2 Fi3


m0=0.91095.*10^-30;     % electron rest mass in Kg.
mn=0.1*m0;              % effective electron mass in nitride.
hb=(1.05458*10^-34);    % Reduced Planck constant in J-s.

% Call of the function Eox_function in order to get Etopox and Ebotox
values
[Etopox, Ebotox] = Efield_program(nte, nth);

% JDTe: Direct tunneling electron current density in A/cm2
```



```
if Ebotox >(Fi1 - Fi2)/(dbotox) &  Ebotox < Fi1/(dbotox)
    mox=0.32*m0*(1e7/Ebotox)^(1.25);

JDTe=(m0./mox).*((q.^3)./(16.*(pi^2).*hb)).*((Ebotox).^2).*(1./((((q.
*Fi1).^0.5)-((q.*Fi1-q.*Ebotox.*dbotox).^0.5)).^2)).*exp(-
((4.*((2.*mox).^0.5).*(0.01)).*(((q.*Fi1).^1.5)-((q.*Fi1-
q.*Ebotox.*dbotox).^1.5))./(3.*q.*hb.*Ebotox));
end

% JMFNe: Modified Fowler Nordheim electron current density in A/cm2
ygamma=eni./eox;
if   Ebotox   >(Fi1-Fi2)./(dbotox-ygamma*dn)   &   Ebotox   <   (Fi1-
Fi2)/(dbotox)
    mox=0.32*m0*(1e7/Ebotox)^(1.25);

JMFNe=(m0/mox).*((q^3)/(16.*(pi^2).*hb)).*((Ebotox)^2).*(1/((((q.*Fi1
)^0.5)-((q.*Fi1-
q.*Ebotox.*dbotox)^0.5)+(ygamma.*(mn/mox)^0.5.*(q.*Fi1-q.*Fi2-
q.*Ebotox.*dbotox)^0.5))^2)).*exp(-
(((4.*((2.*mox)^0.5).*(0.01)).*(((q.*Fi1)^1.5)-((q.*Fi1-
q.*Ebotox.*dbotox)^1.5)))+ ygamma.*4.*(0.01).*((2.*mn)^0.5).*(q.*Fi1-
q.*Fi2-q.*Ebotox.*dbotox)^1.5)/(3.*q.*hb.*Ebotox));
end


% JFNh, Fowler Nordheim hole current density in A/cm^2
if Etopox >(Fi3)/dtopox & Etopox < 1.5*10^7
    AFN=1*10*6.32.*(10^-7).*(3.1/Fi3);                    %in A/V^2
    BFN=1*2.4.*(10^8).*((Fi3/3.1)^1.5);                   %in V/cm
    JFNh= AFN.*((Etopox)^2).*exp(-BFN/Etopox);
end

return
```

A.2.2.5. fun.m

```
function dy = fun(u, y, Nt, ue, ye, sigma)

% fun - get right-hand side using
% interpolation of experimental data

dy =  -sigma*y .*(Nt - interp1(ue, ye, u,'spline'));

return
```