

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ΕΝΣΩΜΑΤΩΣΗ MOSFET ΕΚV3 ΣΤΟΝ
ΠΡΟΣΟΜΟΙΩΤΗ NGSPICE

**-ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ-
ΑΝΘΟΥΛΗΣ ΣΤΥΛΙΑΝΟΣ**

Επιτροπή : Ματτίας Μπούχερ
 Κωνσταντίνος Μπάλας
 Ευτύχιος Κουτρούλης

Χανιά, Οκτώβριος 2013

*Θέλω να ευχαριστήσω ιδιαιτέρως τον κ.Μπούχερ
και τον κ.Μακρή για την συμβολή και την
καθοδήγηση που προσέφεραν σε όλα τα στάδια
διεξαγωγής της εργασίας.*

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|----------------------------------|---|
| ΕΙΣΑΓΩΓΗ – ABSTRACT | 7 |
|----------------------------------|---|

ΚΕΦΑΛΑΙΟ 1: ΙΣΤΟΡΙΚΗ ΠΟΡΕΙΑ ΤΩΝ MOSFET

| | |
|---|----|
| 1.1 Ιστορική πορεία των MOSFET | 10 |
|---|----|

ΚΕΦΑΛΑΙΟ 2: ΤΟ ΜΟΝΤΕΛΟ MOSFET EKV

| | |
|---|----|
| 2.1 Εισαγωγή | 12 |
| 2.2 Δομικά χαρακτηριστικά του MOSFET EKV | 13 |
| 2.2.1 Κυκλωματική διάταξη | 13 |
| 2.2.2 Περιγραφή ακροδεκτών | 14 |
| 2.3 Περιγραφή λειτουργίας του μοντέλου MOSFET EKV | 16 |
| 2.3.1 Βασικές έννοιες | 16 |
| 2.3.2 Φαινόμενα ανώτερης τάξης | 19 |
| 2.4 Παράθεση βασικών εξισώσεων μοντέλου | 20 |
| 2.5 Λειτουργικά χαρακτηριστικά του μοντέλου MOSFET EKV | 27 |

ΚΕΦΑΛΑΙΟ 3: ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΛΕΙΩΝ

| | |
|---|----|
| 3.1 Εισαγωγή | 28 |
| 3.2 Η γλώσσα Verilog-A | 29 |
| 3.2.1 Βασική δομή | 30 |
| 3.2.2 Εντολές ελέγχου ροής και επανάληψης | 33 |
| 3.2.3 Συναρτήσεις | 35 |
| 3.2.4 Λεξικοί κανόνες | 35 |
| 3.3 Υλοποιήσεις του μοντέλου | 36 |
| 3.3.1 Το μοντέλο σε Verilog-A | 36 |
| 3.3.2 Το μοντέλο σε C | 37 |

| | |
|---|----|
| 3.3.3 Πλεονεκτήματα και μειονεκτήματα μοντελοποίησης σε κάθε γλώσσα | 38 |
| 3.4 Ο προσομοιωτής Ngspice | 39 |
| 3.5 Το εργαλείο ADMS | 40 |

ΚΕΦΑΛΑΙΟ 4: ΕΝΣΩΜΑΤΩΣΗ ΜΟΝΤΕΛΟΥ ΕKV3 ΣΤΟΝ NGSPICE

| | |
|---|----|
| 4.1 Εισαγωγή | 42 |
| 4.2 Διαδικασία ενσωμάτωσης μοντέλου ΕKV3 στον προσομοιωτή Ngspice | 42 |
| 4.3 Διαδικασία μετάφρασης κώδικα μοντέλου ΕKV3 από Verilog-A σε C | 47 |

ΚΕΦΑΛΑΙΟ 5: ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΕΩΝ

| | |
|----------------------------|----|
| 5.1 Εισαγωγή | 48 |
| 5.2 Προσομοιώσεις τύπου DC | 48 |
| 5.3 Προσομοιώσεις τύπου AC | 67 |
| 5.4 Προσομοίωση κυκλώματος | 72 |

ΚΕΦΑΛΑΙΟ 6: ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

| | |
|---|----|
| 6.1 Εισαγωγή | 74 |
| 6.2 Ανασκόπηση της διαδικασίας υλοποίησης | 74 |
| 6.3 Συμπεράσματα | 75 |
| 6.4 Προοπτική μελλοντικής εξέλιξης | 76 |

ΠΑΡΑΡΤΗΜΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

| | |
|------|---|
| 1.1 | Διατομή MOSFET. |
| 2.1 | Κυκλωματική διάταξη MOSFET |
| 2.2 | Κυκλωματικές διατάξεις nMOS και pMOS |
| 2.3 | Διατομή nMOS.. |
| 2.4 | Διάγραμμα τάσης πύλης-ρεύματος. |
| 2.5 | Διάγραμμα τάσης drain /source-ρεύματος |
| 3.1 | Διασύνδεση των διαφορετικών ειδών Verilog |
| 3.2 | Βασική μορφή δήλωσης οντότητας |
| 3.3 | Παράδειγμα δήλωσης οντότητας |
| 3.4 | Παράδειγμα δήλωσης nature. |
| 3.5 | Βασική μορφή δήλωσης if |
| 3.6 | Βασική μορφή δήλωσης case |
| 3.7 | Βασική μορφή δήλωσης βρόγχων επανάληψης. |
| 3.8 | Βασική μορφή δήλωσης εντολών \$stop και \$finish. |
| 3.9 | Βασική μορφή δήλωσης εντολής cross |
| 3.10 | Οι πέντε εκδοχές του μοντέλου σε Verilog-A |
| 3.11 | Τα ενσωματωμένα χαρακτηριστικά του ADMS. |
| 4.1 | Διαδικασία μετάφρασης κώδικα μοντέλου EKV3 από Verilog-A σε C. |
| 5.1 | Διάγραμμα linear id-vg/ L=0.1.. |
| 5.2 | Διάγραμμα logarithmic id-vg/ L=0.1 |

| | |
|------|--|
| 5.3 | Διάγραμμα linear gm-vg/ L=0.1 |
| 5.4 | Διάγραμμα logarithmic gm-vg/ L=0.1 |
| 5.5 | Διάγραμμα logarithmic min.error-vg/ L=0.1..... |
| 5.6 | Διάγραμμα linear id-vg/ L=10.. |
| 5.7 | Διάγραμμα logarithmic id-vg/ L=10 |
| 5.8 | Διάγραμμα linear gm-vg/ L=10..... |
| 5.9 | Διάγραμμα logarithmic gm-vg/ L=10 |
| 5.10 | Διάγραμμα logarithmic min.error-vg/ L=10..... |
| 5.11 | Διάγραμμα linear id-vd/ L=0.1..... |
| 5.12 | Διάγραμμα logarithmic id-vd/ L=0.1 |
| 5.13 | Διάγραμμα linear gds-vd/ L=0.1..... |
| 5.14 | Διάγραμμα logarithmic gds-vd/ L=0.1 |
| 5.15 | Διάγραμμα logarithmic min.error-vd/ L=0.1..... |
| 5.16 | Διάγραμμα linear id-vd/ L=10..... |
| 5.17 | Διάγραμμα logarithmic id-vd/ L=10..... |
| 5.18 | Διάγραμμα linear gds-vd/ L=10 |
| 5.19 | Διάγραμμα logarithmic gds-vd/ L=10..... |
| 5.20 | Διάγραμμα logarithmic min.error-vd/ L=10 |
| 5.21 | Διάγραμμα linear id-vs/ L=0.1 |
| 5.22 | Διάγραμμα logarithmic id-vs/ L=0.1..... |
| 5.23 | Διάγραμμα linear gms-vs/ L=0.1..... |
| 5.24 | Διάγραμμα logarithmic gms-vs/ L=0.1 |

| | |
|------|---|
| 5.25 | Διάγραμμα logarithmic min.error-vs/ $L=0.1$ |
| 5.26 | Διάγραμμα linear id-vs/ $L=10$ |
| 5.27 | Διάγραμμα logarithmic id-vs/ $L=10$ |
| 5.28 | Διάγραμμα linear gms-vs/ $L=10$ |
| 5.29 | Διάγραμμα logarithmic gms-vs/ $L=10$ |
| 5.30 | Διάγραμμα logarithmic min.error-vs/ $L=10$ |
| 5.31 | CGB simulation-circuit |
| 5.32 | CGC simulation-circuit. |
| 5.33 | CGG simulation-circuit. |
| 5.34 | CGB simulation. |
| 5.35 | CGC simulation. |
| 5.36 | CGB και CGC simulation |
| 5.37 | CGG simulation. |
| 5.38 | Κύκλωμα - Διαιρέτης ρεύματος |
| 5.39 | Διάγραμμα I-IREF. |

ΕΙΣΑΓΩΓΗ

Στον χώρο της αυτοματοποίησης ηλεκτρονικής σχεδίασης (Electronic Design Automation – EDA) τα τελευταία χρόνια, παρατηρείται έντονη εξάπλωση των εργαλείων ελεύθερου λογισμικού, ένα από τα οποία είναι και ο προσομοιωτής ηλεκτρονικών κυκλωμάτων Ngspice.

Οι προοπτικές ανάπτυξης του Ngspice παρουσιάζουν έντονο ενδιαφέρον καθώς παρ' ότι αναπτύσσεται από μια ολιγομελή ομάδα, χρησιμοποιείται ήδη ευρέως σε πανεπιστήμια και εταιρείες ανά τον κόσμο. Ωστόσο, προς το παρόν περιλαμβάνει περιορισμένο αριθμό μοντέλων ενεργών στοιχείων, όπως είναι τα μοντέλα MOSFET τελευταίας γενιάς. Στα πλαίσια της παρούσας διπλωματικής εργασίας ενσωματώθηκε στον προσομοιωτή Ngspice το μοντέλο MOSFET EKV3 το οποίο αναπτύσσεται από το Εργαστήριο Ηλεκτρονικής.

Συγκρίνοντας τον προσομοιωτή Ngspice με παρεμφερές, μη ελεύθερο εργαλείο, τον προσομοιωτή Spectre, καταλήξαμε σε ένα διαφωτιστικό σύνολο αποτελεσμάτων που επιβεβαιώνουν την εγκυρότητα της λειτουργίας του μοντέλου EKV3 στο Ngspice. Η διαδικασία ενσωμάτωσης του μοντέλου ολοκληρώθηκε σε δύο στάδια: α) την ενσωμάτωση του μοντέλου MOSFET EKV3 στον προσομοιωτή Ngspice με χρήση του επίσης ελεύθερου λογισμικού-εργαλείου ADMS (εργαλείου μετατροπής γλώσσας Verilog-A σε γλώσσα προγραμματισμού C) και β) επαλήθευση σωστής λειτουργίας της ενσωμάτωσης του μοντέλου EKV3 στον προσομοιωτή Ngspice, μέσω σύγκρισης του με τον προσομοιωτή Spectre.

Η επαλήθευση σωστής λειτουργίας του μοντέλου MOSFET EKV3 διεξήχθη σε επίπεδο προσομοιώσεων τύπου στατικής συμπεριφοράς (DC) και μικρού σήματος (AC), με αντιπροσωπευτικό σύνολο προσομοιώσεων. Επίσης έγινε δοκιμή της σωστής λειτουργίας του μοντέλου EKV3 με βάση κυκλώματος - διαιρέτη ρεύματος (R-2R ή M-2M ladder circuit).

Το πρώτο κεφάλαιο περιλαμβάνει μία σύντομη ιστορική αναδρομή που αφορά τον τομέα της ηλεκτρονικής και την εξέλιξη των τρανζίστορς.

Στο δεύτερο κεφάλαιο περιγράφεται η δομή, το θεωρητικό υπόβαθρο, οι μαθηματικές εξισώσεις και τα φαινόμενα που διέπουν την λειτουργία του μοντέλου MOSFET EKV3.

Το τρίτο κεφάλαιο παρουσιάζει τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας και συγκεκριμένα τις γλώσσες προγραμματισμού Verilog-A και C, τους προσομοιωτές Ngspice και Spectre και τα μεταφραστικά εργαλεία ADMS και XML Scripts.

Το τέταρτο κεφάλαιο περιγράφει αναλυτικά την διαδικασία μετατροπής του κώδικα του μοντέλου EKV3 από Verilog-A σε γλώσσα C και την ενσωμάτωση του στον προσομοιωτή Ngspice.

Στο πέμπτο κεφάλαιο παρατίθενται τα αποτελέσματα των προσομοιώσεων σε διαγραμματική μορφή συνοδευόμενα από σχολιασμό και ερμηνεία, ενώ στο έκτο και τελευταίο κεφάλαιο ακολουθεί η σύνοψη της διαδικασίας υλοποίησης και η καταγραφή των συμπερασμάτων.

ABSTRACT

During recent years, a number of new open source tools have been made available in the field of Electronic Design Automation (EDA). The circuit simulator Ngspice is an open source tool, developed by a small group of engineers, which already has a large community of users in academia and industry. However, it yet includes a limited number of active device models and particularly MOSFET models of the latest generation.

Within the present diploma thesis, the EKV3 MOSFET model for analog and RF design and simulation, developed by the Electronics Laboratory, was implemented

in the circuit simulator Ngspice. By comparing Ngspice to another, licensed (non-open source) tool, circuit simulator Spectre, the correct implementation of the EKV3 model was verified. This work was carried out on two levels: a) the implementation of the EKV3 MOSFET model in Ngspice with the use of another open source tool, named ADMS (a tool that automates the translation of Verilog-A code into C-code) and b) quantitative comparison of the Ngspice's results, to circuit simulator's Spectre (non-open source) results, that lead to verification of Ngspice's correct functioning.

The latter verification was made on the level of static (DC) and small-signal dynamic (AC) simulations, with a representative set of simulations. Also, the correct behavior of the EKV3 model was tested on circuit level, using an R-2R (or M-2M) current divider circuit.

The first chapter is a historical reference that traces back to the birth of electronics and includes the most important steps of transistor's evolution.

The next chapter includes the structure, the theoretical background and mathematical description of the MOSFET EKV3 model.

The tools used in this project are explained in the third chapter and sum up to: the programming languages Verilog-A and C, Ngspice simulator, Spectre simulator and the translating tools ADMS and XML Scripts.

The fourth chapter elaborately describes the translation of the MOSFET EKV3 model's code from Verilog-A to C, as well as the procedure of its integration to Ngspice simulator.

The fifth chapter consists of the presentation, analysis and evaluation of the simulation results, followed by the sixth and final chapter where all the steps of the project are summarized and conclusions are presented.

ΚΕΦΑΛΑΙΟ 1

-ΙΣΤΟΡΙΚΗ ΠΟΡΕΙΑ ΤΩΝ MOSFET-

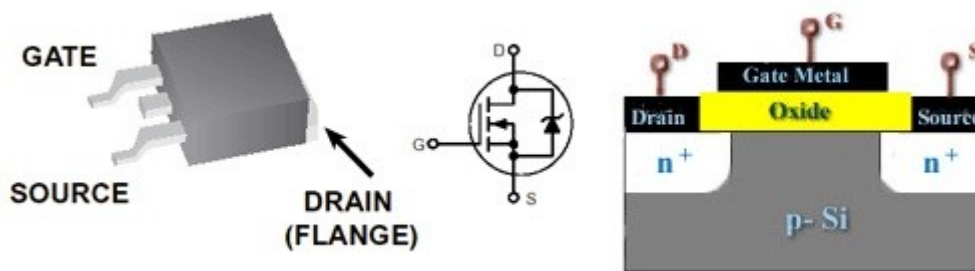
1.1 Ιστορική πορεία των MOSFET

Η γέννηση του τομέα της ηλεκτρονικής ανάγεται στις αρχές του 20ου αιώνα με την ανακάλυψη της λυχνίας, τεχνογνωσία που λίγο αργότερα οδήγησε στην δημιουργία της διόδου το 1905 και της τριόδου το 1906.

Το *τρανζίστορ πεδίου* σε επίπεδο σύλληψης, εισήγαγε το 1926 ο *Julius Edgar Lilienfeld*, ο οποίος κατοχύρωσε την εφεύρεσή του με δίπλωμα ευρεσιτεχνίας. Την ίδια διαδικασία ακολούθησε το 1934 και ο Oskar Heil για την ανακάλυψη ενός παρεμφερούς με το σημερινό MOSFET, στοιχείου. Οι παραπάνω εφευρέσεις ήταν κατά κύριο λόγο θεωρητικής φύσεως και όχι λειτουργικής μορφής.

Τα πρώτα λειτουργικά τρανζίστορς κάνουν την εμφάνισή τους το 1947 (διπολικά τρανζίστορς ένωσης-bipolar junction transistor ή BJT) από τους John Brattain, William Shockley, Walter Bardeen (Bell Labs) , για να ακολουθήσει το 1951 το πρώτο *τρανζίστορ επίδρασης πεδίου* (junction field-effect transistor ή J-FET), σε επίπεδο σύλληψης από τον W. Shockley. Το 1953 εμφανίστηκαν J-FET λειτουργικής μορφής από τους Dacey και Ross, καθιερώνοντας την βασική αρχή λειτουργίας αυτού του είδους, που υπαγορεύει ότι η συμπεριφορά τους σαν πηγή ρεύματος ελέγχεται από την **τάση**.

Το 1959 πρωτοεμφανίζεται το MOSFET, η ανακάλυψη του οποίου είναι απόρροια της τεχνογνωσίας ολοκληρωμένων κυκλωμάτων πυριτίου, γνώση που αποκτήθηκε σταδιακά καθ' όλη τη δεκαετία του '50. Το όνομα του στοιχείου προέκυψε από την διαδοχική κάθετη τοποθέτηση υλικών που το απαρτίζουν: μέταλλο-Metal, οξείδιο του πυριτίου-Oxide και πυρίτιο-Silicon, εξού και *Mos-fet*.



Εικόνα 1.1: Διατομή MOSFET.

Η κατασκευή των πρώτων ολοκληρωμένων κυκλωμάτων λαμβάνει χώρα τις αρχές της δεκαετίας '60, με την εμφάνιση της ολοκλήρωσης μικρής κλίμακας (SSI -small scale integration), στην οποία βασίζεται και η ανάπτυξη της TTL τεχνολογίας.

Χρονιά σταθμό στην εξελικτική πορεία των MOS αποτελεί το 1963, με την ανακάλυψη των CMOS (complementary MOS-συμπληρωματικό MOS), στα οποία και θα βασιστεί η ανάπτυξη των ηλεκτρονικών υπολογιστών. Περί τα τέλη της ίδιας δεκαετίας εμφανίζονται τα πρώτα κυκλώματα ψηφιακής μνήμης και μία νέα κλίμακα σχεδίασης, η MSI (medium scale integration), με δεκαπλάσια χωρητικότητα πυλών σε κάθε ολοκληρωμένο κύκλωμα από αυτήν της SSI. Παράλληλα αναπτύσσεται η BiCMOS, μία υβριδική τεχνική σχεδίασης που περιλαμβάνει CMOS και BJT.

Την δεκαετία του '70 γίνεται η μετάβαση στην LSI (large scale integration) κλίμακα σχεδίασης, την οποία θα διαδεχθεί η VLSI (very large scale integration) κλίμακα την επόμενη δεκαετία, με τον αριθμό των πυλών να ξεπερνά και τις 5.000 ανά ολοκληρωμένο [1,2,7].

ΚΕΦΑΛΑΙΟ 2

- ΤΟ ΜΟΝΤΕΛΟ MOSFET EKV -

2.1 Εισαγωγή

Το μοντέλο EKV δημιούργησαν οι: **Christian Enz, Francois Krummenacher** και **Eric A. Vittoz**, το 1995 [3] στο Swiss Federal Institute of Technology of Lausanne, ενώ το όνομα “EKV” προέρχεται από τα αρχικά των επιθέτων τους. Το μοντέλο βασισμένο στα φορτία δημιουργήθηκε από τους **Bucher** (PhD 1999, SSE'2003), **Sallese, e. a.**

Το μοντέλο EKV δανείζεται στοιχεία της δουλειάς μίας μακράς σειράς ερευνητών, η οποία ανάγεται στα πρώτα στάδια διερεύνησης της φυσικής ημιαγωγών. Επίσης, καταβολές του τοποθετούνται στις πρώτες ερευνητικές προσπάθειες για ενσωματωμένα κυκλώματα πολύ χαμηλής ισχύος και τάσης που θα επέτρεπαν την περιγραφή της συμπεριφοράς των τρανζίστορς μέσα σε πλαίσια συγκεκριμένων αναγκών.

Το μοντέλο EKV αφορά την εξομοίωση του τρόπου λειτουργίας ενός MOS τρανζίστορ και περιγράφει με συνεχή τρόπο τον θόρυβο, τα στατικά και δυναμικά χαρακτηριστικά στο τρανζίστορ έως και πολύ χαμηλά επίπεδα ρεύματος. Παράλληλα απαιτεί ένα πολύ περιορισμένο σύνολο παραμέτρων, οι οποίες είναι άμεσα συνδεδεμένες με βασικές, ανεξάρτητες φυσικές παραμέτρους.

Οι εξισώσεις που περιγράφουν την λειτουργία του μοντέλου, χαρακτηρίζονται από απλότητα και γι' αυτό παρέχουν την ξεχωριστή δυνατότητα για διαισθητική κατανόηση του μοντέλου και των παραμέτρων του. Το καθιστούν έτσι πιο “εύπλαστο” και διαχειρίσιμο στα χέρια του προγραμματιστή, γεγονός που ευνοεί την συστηματοποίηση του σχεδιασμού.

Το μοντέλο που θα μας απασχολήσει στην παρούσα εργασία είναι η τρίτη έκδοση του MOSFET EKV (Bazigos, Bucher e.a. 2008), που δημιουργήθηκε και διανέμεται από την ομάδα *Μικροηλεκτρονικής του Πολυτεχνείου Κρήτης* (κ.Βucher) [1,3,4,5,6].

2.2 Δομικά χαρακτηριστικά του μοντέλου MOSFET EKV

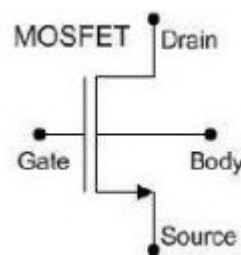
Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, η ονομασία MOSFET είναι άμεσα συνδεδεμένη με την δομή του στοιχείου. Το συνθετικό Mos- προκύπτει από την παρατήρηση της κάθετης διατομής του στοιχείου, ξεκινώντας από το επάνω στρώμα που αποτελείται από μέταλλο (Metal), συνεχίζοντας στο επόμενο στρώμα που αποτελείται από οξείδιο του πυριτίου (Oxide) και καταλήγοντας στο τρίτο και τελευταίο στρώμα από πυρίτιο (Silicon).

Το δεύτερο συνθετικό της ονομασίας -fet, είναι τα αρχικά περιγραφής του τρόπου λειτουργίας του, “field-effect transistor”, που μεταφράζεται σε *τρανζίστορ επίδρασης πεδίου*. Η επίδραση πεδίου, περιγράφει ένα πολύ σημαντικό χαρακτηριστικό της λειτουργίας του στοιχείου, όσον αφορά τον ρόλο του ως πηγή ρεύματος. Αυτού του τύπου τα τρανζίστορς είναι πηγές ρεύματος ελεγχόμενες από τάση και όχι από ρεύμα [7].

2.2.1 Κυκλωματική διάταξη

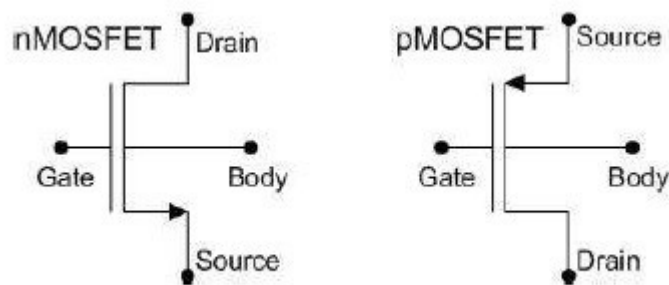
- Το στοιχείο MOSFET περιλαμβάνει 4 ακροδέκτες:

1. drain – υποδοχέας
2. gate - πύλη
3. source - πηγή
4. bulk ή body - σώμα



Εικόνα 2.1: Κυκλωματική διάταξη MOSFET.

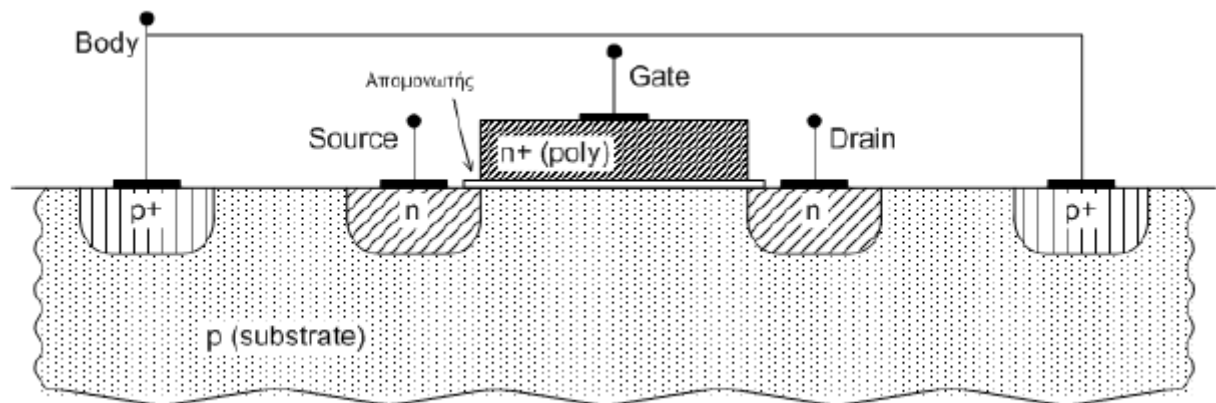
- Αναλόγως με τους φορείς πλειονότητας του κάθε ακροδέκτη (ηλεκτρόνια ή οπές), τα MOSFET κατηγοριοποιούνται σε *nMOS* και *pMOS*.



Εικόνα 2.2: Κυκλωματικές διατάξεις *nMOS* και *pMOS*.

Οι δύο αυτοί τύποι είναι απόλυτα συμμετρικοί, οπότε η λειτουργία του ενός αντικατοπτρίζει τη λειτουργία του άλλου και επαρκεί η επεξήγηση του ενός, για την κατανόηση και των δύο [8].

2.2.2 Περιγραφή ακροδεκτών



Εικόνα 2.3: Διατομή *nMOS*.

Το άνω στρώμα του στοιχείου αποτελεί τον μεταλλικό ακροδέκτη gate - πύλη. Η πύλη είναι σχήματος ορθογωνίου με πλάτος (W-width) και μήκος (L-length) που αντιστοιχεί στο διάστημα μεταξύ των ακροδεκτών drain και source.

Κάτωθεν της πύλης βρίσκεται ένα λεπτό στρώμα από οξείδιο του πυριτίου (SiO_2) που εφάπτεται αυτής και παρέχει μόνωση. Το στρώμα αυτό καλείται απομονωτής και οι διαστάσεις του πρέπει να είναι ίσες ή μεγαλύτερες από αυτές της πύλης, για να διασφαλίζεται η επίτευξη μόνωσης.

Στην βάση του MOSFET συναντούμε τον ημιαγωγό, που είναι το κατώτατο στρώμα του στοιχείου. Στο σημείο αυτό ξεκινάει η διαφοροποίηση μεταξύ nMOS-pMOS. Στα πρώτα, ο ημιαγωγός περιλαμβάνει υπόστρωμα με φορείς πλειονότητας-οπές. Στα δεύτερα, υπόστρωμα με φορείς πλειονότητας-ηλεκτρόνια.

Στο υπόστρωμα σχεδιάζονται οι δύο εναπομείναντες ακροδέκτες του στοιχείου, drain και source. Αυτοί τοποθετούνται πριν και μετά το συσσωμάτωμα οξειδίου-μετάλλου, σχηματίζονται εξ' ολοκλήρου στο χώρο που καλύπτει το υπόστρωμα και δεν εισέρχονται σχεδόν καθόλου στην περιοχή κάτω από το μέταλλο της πύλης. Η ονοματολογία τους προκύπτει απ' την πραγματική ροή του ρεύματος που δημιουργείται στα nMOS, με φορά από το source στο drain.

Οι ακροδέκτες αυτοί είναι εμπλουτισμένοι με φορείς πλειονότητας αντίθετους απ' αυτούς του υποστρώματος και συνεπώς δημιουργούν δύο θύλακες αντίθετου φορτίου μέσα σε αυτό. Οι θύλακες δεν συνδέονται ηλεκτρικά μεταξύ τους, ενώ σε όλο το μήκος της ενδιάμεσης απόστασης παρεμβάλλεται ημιαγωγός αντίθετου φορτίου. Έτσι, δεν δύναται υπό τις δεδομένες συνθήκες να υπάρξει ροή ηλεκτρικού ρεύματος ανάμεσά τους.

Ο τέταρτος ακροδέκτης εμφανίζεται στο κάτω μέρος του υποστρώματος και πρακτικά είναι ολόκληρο το υπόστρωμα. Γι' αυτό εκτός από bulk ονομάζεται και substrate. Επίσης ονομάζεται *πίσω πύλη* διότι η διαφορά τάσης του με την πύλη ελέγχει την λειτουργία του στοιχείου.

Συχνά οι ακροδέκτες πηγή και υπόστρωμα ενώνονται, σχηματίζοντας μια δίοδο, οπότε και θεωρούμε το στοιχείο σαν τριών ακροδεκτών [7,8].

2.3 Περιγραφή λειτουργίας του μοντέλου MOSFET EKV

Η προαναφερθείσα δομή οδηγεί στην δημιουργία παρασιτικών διόδων μεταξύ drain-bulk και source-bulk. Η ορθή λειτουργία του στοιχείου απαιτεί οι διόδοι αυτές, να είναι μονίμως ανάστροφα πολωμένες. Έτσι επιτυγχάνεται ηλεκτρική απομόνωση μεταξύ των ακροδεκτών drain, source και του υποστρώματος.

Ενώ μεταξύ των ακροδεκτών drain και source δεν υπάρχει ηλεκτρική σύνδεση, με την εφαρμογή τάσης στην πύλη αναπτύσσεται ένα κάθετο ηλεκτροστατικό πεδίο που διέρχεται από το στρώμα του οξειδίου και συνεχίζει στο σώμα του υποστρώματος. Αναλόγως με την έντασή του, το πεδίο αυτό έλκει φορείς πλειονότητας ίδιου τύπου με τους φορείς των περιοχών drain και source. Ξεκινά δηλαδή αντίστροφη συγκέντρωση φορέων πλειονότητας κατά μήκος της επιφάνειας του υποστρώματος, που εξελίσσεται σε στρώμα και επιτρέπει την ηλεκτρική σύνδεση των δύο ακροδεκτών drain και source. Στο σημείο αυτό, με άσκηση τάσης στους ακροδέκτες - άκρα του στρώματος, δημιουργείται ροή ηλεκτρικού φορτίου, δηλαδή ρεύμα καναλιού.

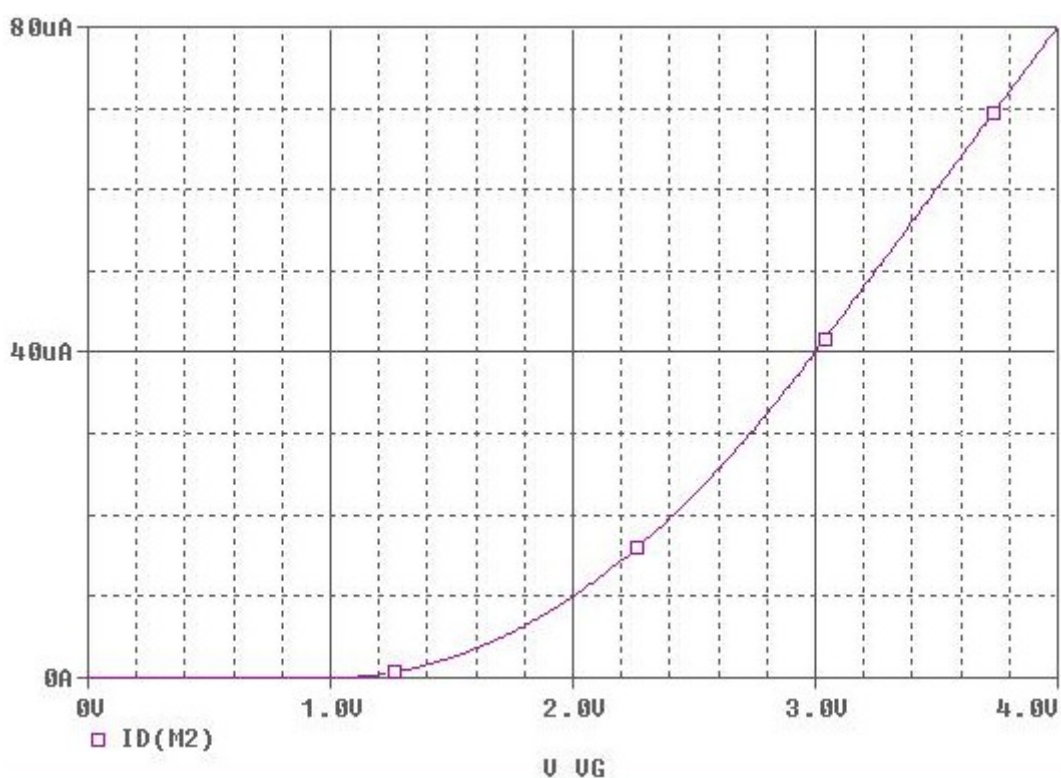
Το αγώγιμο στρώμα φορέων ονομάζεται *κανάλι* - *channel*. Ο χαρακτηρισμός αυτός επεκτείνεται και πέραν του φαινομένου της συγκέντρωσης φορέων, προσδιορίζοντας την περιοχή του ημιαγωγού όπου παρατηρείται το φαινόμενο [7,8].

2.3.1 Βασικές έννοιες

Η περιγραφή της συμπεριφοράς ενός MOSFET, περιλαμβάνει κάποιες έννοιες με ιδιαίτερη σημασία:

1. Τάση κατωφλίου, είναι η ελάχιστη διαφορά δυναμικού μεταξύ πύλης - υποστρώματος, υπό την επίδραση της οποίας δημιουργείται στο υπόστρωμα συγκέντρωση φορέων ικανή ώστε να συνδέσει ηλεκτρικά τους ακροδέκτες drain και source.

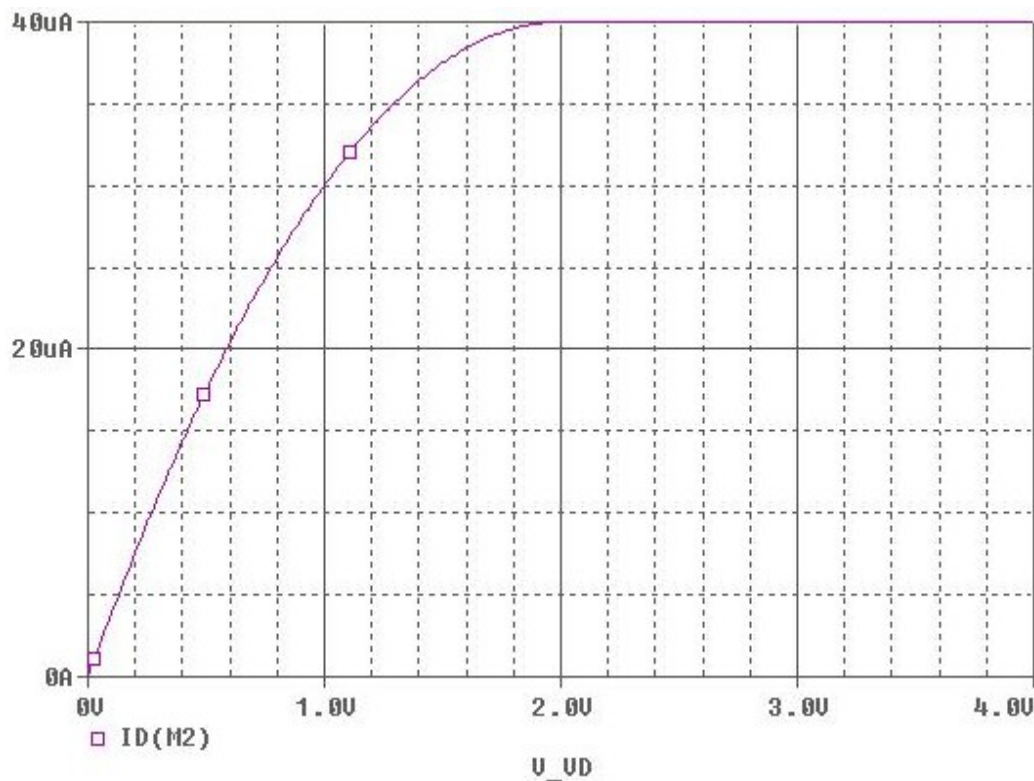
2. Αναστροφή, καλείται η ύπαρξη καναλιού στο οποίο η συγκέντρωση αντίθετων φορέων είναι μεγαλύτερη απ' την συγκέντρωση φορέων υποστρώματος. Υπάρχουν τρία είδη αναστροφής ανάλογα με την ένταση του φαινομένου : ασθενής, μέτρια και ισχυρή. Για την ασθενή αναστροφή η εξάρτηση τάσης - ρεύματος έχει εκθετικό χαρακτήρα, για την ισχυρή γραμμικό ενώ για την μέτρια αναστροφή η εξάρτηση τάσης - ρεύματος έχει περίπλοκο χαρακτήρα.



Εικόνα 2.4: Διάγραμμα τάσης πύλης-ρεύματος.

Στο παραπάνω διάγραμμα αρχικά δεν υπάρχει κανάλι και έτσι δεν παρατηρείται ρεύμα. Καθώς όμως πλησιάζουμε την τάση κατωφλίου αρχίζει η περιοχή της ασθενούς αναστροφής. Εκεί παρουσιάζεται εκθετική εξάρτηση ρεύματος - τάσης, που εξελίσσεται σε ισχυρή αναστροφή με την σχέση να γραμμικοποιείται.

3. Κορεσμός η λειτουργία στον κόρο, συμβαίνει όταν το ρεύμα που διαρρέει το κανάλι βρίσκεται σε μία συγκεκριμένη τιμή - ανώφλι, όπου και θα παραμείνει σε γενικές γραμμές, όσο και αν αυξηθεί η τάση V_{DS} .



Εικόνα 2.5: Διάγραμμα τάσης *drain /source*-ρεύματος.

Στο παραπάνω διάγραμμα παρατηρούμε ότι από ένα σημείο και μετά η τιμή του ρεύματος παραμένει σταθερή ανεξάρτητα από την αύξηση της τάσης στα άκρα του καναλιού, υπάρχει δηλαδή κορεσμός [7].

2.3.2 Φαινόμενα ανώτερης τάξης

Το μοντέλο EKV3 [4,5,8,9] είναι ένα πλήρες μοντέλο προσομοίωσης κυκλωμάτων. Εμπεριέχει όλα τα φαινόμενα ανώτερης τάξης, όπως:

- Φαινόμενο σώματος, καλείται η εξάρτηση της τάσης κατωφλίου V_T από την τάση V_{BS} .
- Φαινόμενο Early, παρατηρείται κατά τον κορεσμό, με αύξηση της αγωγιμότητας εξόδου $G_{DS} \approx I_{DS} / V_E$.
- Εξάρτηση της ευκινησίας από το κάθετο πεδίο
- Κβαντικά φαινόμενα
- Polydepletion
- Φαινόμενο κορεσμού ταχύτητας φορέων (velocity saturation)
- Διαμόρφωση μήκους καναλιού (channel length modulation, CLM)
- Φαινόμενο DIBL

καθώς και πλήθος παρασητικών στοιχείων, όπως:

- σειριακές αντιστάσεις
- παρασητικές χωρητικότητες.

Ο κώδικας Verilog-A που ενσωματώνει όλα τα προαναφερθέντα δημιουργήθηκε και διανέμεται από την ομάδα Μικροηλεκτρονικής του Εργαστηρίου Ηλεκτρονικής του Πολυτεχνείου Κρήτης [7, 8, 9].

2.4 Παράθεση βασικών εξισώσεων μοντέλου MOSFET EKV

-QG, είναι το φορτίο του ακροδέκτη gate.

-Qox, είναι το φορτίο του στρώματος οξειδίου.

-Qinv, είναι το φορτίο του καναλιού.

-QB, είναι το φορτίο του υποστρώματος.

-Η δομή του στοιχείου περιλαμβάνει έναν πυκνωτή που σχηματίζεται μεταξύ της πύλης και του ημιαγωγού. Ο τύπος που συνδέει το πάχος του στρώματος οξειδίου με την χωρητικότητα του πυκνωτή (ανά μονάδα εμβαδού) και την διηλεκτρική σταθερά του οξειδίου του πυριτίου είναι ο:

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}} \quad , \quad \mu\epsilon \quad \epsilon_{ox} = 3.45 \cdot 10^{-11} F/m$$

-Ο κανονικοποιημένος παράγοντας - θερμοδυναμική τάση UT που εξαρτάται από την θερμοκρασία, δίνεται από τον τύπο:

$$U_T = \frac{k \cdot T}{q}$$

Όπου k είναι η σταθερά του Boltzmann ($1.381 \cdot 10^{-23}$ J/K), T η θερμοκρασία σε βαθμούς Kelvin και q η απόλυτη τιμή του φορτίου του ηλεκτρονίου.

-Η πύλη βρίσκεται υπό μία τάση V_G . Κάποιο μέρος της αντιστοιχεί στην πτώση τάσης μεταξύ του υλικού της πύλης και του ημιαγωγού (ΦMS), ενώ κάποιο άλλο στην διαφορά τάσης λόγω του φορτίου στην πύλη. Το δεύτερο περιγράφεται απ'τον τύπο:

$$\Psi_{OX} = \frac{Q_G}{C_{OX}}$$

-Για την V_G το άθροισμα των φορτίων ισχύει:

$$V_G = \Phi_{MS} + \Psi_{OX} + \Psi_S$$

$$Q_G + Q_{OX} + Q_C = 0$$

- Ψ_S είναι το δυναμικό στην επιφάνεια του υποστρώματος σε σχέση με την τάση στην πύλη. Η σχέση αυτή είναι η ακόλουθη:

$$\Psi_S \cong V_G - V_{FB} - \gamma \left(\sqrt{V_G - V_{FB} + \frac{\gamma^2}{4}} - \frac{\gamma}{2} \right)$$

- Ψ_S στην ισχυρή αναστροφή:

$$\Psi_S = \Psi_0 + V_{ch}$$

Όπου Ψ_0 είναι μία σταθερά που ισούται με το διπλάσιο της τάσης Fermi επαυξημένο κατά λίγες φορές την τάση UT.

-VFB (flat-band voltage), είναι η τάση που θα πρέπει να τεθεί στην πύλη για να έχουμε ηλεκτρικά ουδέτερο το κανάλι.

$$V_{FB} \equiv \Phi_{MS} - \frac{Q_{OX}}{C_{OX}}$$

- Φ_F , είναι η τάση Fermi του υποστρώματος και V_{ch} η τάση στο κανάλι. Ισχύει ότι:

$$\Phi_F = U_T \ln\left(\frac{N_{sub}}{N_i}\right)$$

$$Q_C = -\gamma C_{OX} \sqrt{U_T} \sqrt{\frac{\Psi_S}{U_T} + \exp\left(\frac{\Psi_S - 2\Phi_F - V_{ch}}{U_T}\right)}$$

Με γ συμβολίζουμε τον συντελεστή φαινομένου σώματος και ισούται με:

$$\gamma = \frac{\sqrt{2q\epsilon_{si}N_{sub}}}{C_{OX}}$$

-Προσεγγιστικός υπολογισμός QB:

$$Q_B = -\gamma C_{OX} \sqrt{U_T} \sqrt{\frac{\Psi_S}{U_T}}$$

-Το Q_{inv} περιγράφεται απ' τον τύπο:

$$Q_{INV} = -\gamma C_{OX} \sqrt{U_T} \left[\sqrt{\frac{\Psi_S}{U_T} + \exp\left(\frac{\Psi_S - 2\Phi_F - V_{ch}}{U_T}\right)} - \sqrt{\frac{\Psi_S}{U_T}} \right]$$

-VP (pinch-off voltage), είναι η τάση μέσω της οποίας γίνεται ο υπολογισμός του ρεύματος στο στοιχείο. Η VP είναι μία συνάρτηση της τάσης στην πύλη:

$$V_P \equiv V_{ch} \Big|_{Q_{inv}=0} = V_G - V_{FB} - \gamma * \left\{ \sqrt{V_G - V_{FB} + \frac{\gamma^2}{4}} - \frac{\gamma}{2} \right\} - \Psi_0$$

-Η τάση κατωφλίου δίνεται από την εξής σχέση:

$$V_{TO} \equiv V_G \Big|_{Q_{inv}=0, V_{ch}=0} = V_{FB} + \Psi_0 + \gamma * \Psi_0$$

-Παράγοντας κλίσης (n):

$$n \equiv \frac{dV_G}{dV_P} = 1 + \frac{\gamma}{2 * \sqrt{\Psi_0 + V_P}}$$

-Βασική εξίσωση περιγραφής του ρεύματος στο μοντέλο:

$$I_D = \mu W (-Q_{inv}) \frac{dV_{ch}}{dx}$$

Όπου μ είναι η κινητικότητα των φορέων του ρεύματος και W το πλάτος όπου θεωρούμε το ρεύμα.

-Σε αυτήν την εξίσωση το ρεύμα υπολογίζεται σαν μία έκφραση ολοκληρώματος ως προς την τάση στο κανάλι για τιμές της τάσης από τον ένα ακροδέκτη (source) στον άλλον (drain):

$$I_D = \mu W \int_{V_S}^{V_D} (-Q_{inv}(V_{ch})) dV_{ch}$$

-Υπολογισμός αγωγιμότητας του καναλιού:

$$G_{ch}(x) \equiv \frac{dI}{dV_{ch}} = \mu \frac{W}{L} (-Q_i(x))$$

Με τον όρο εξάρτησης x εννοούμε την εξάρτηση της αγωγιμότητας από την θέση στο κανάλι.

-Αγωγιμότητες ξεχωριστά ως προς κάθε μία τάση στα άκρα και αγωγιμότητα του όλου καναλιού:

$$G_{ch} = \frac{dI}{dV_{ch}} = \mu \frac{W}{L} (-Qi)$$

$$G_{ms} = \left. \frac{dI_D}{dV_S} \right|_{V_G, V_D} = - \left. \frac{dI_F}{dV_S} \right|_{V_G} = \mu \frac{W}{L} (-Qi(S))$$

$$G_{md} = \left. \frac{dI_D}{dV_D} \right|_{V_G, V_S} = \left. \frac{dI_R}{dV_D} \right|_{V_G} = \mu \frac{W}{L} (-Qi(D))$$

-Κανονικοποιημένη αγωγιμότητα:

$$g_{ms} = \frac{G_{ms}}{G_0} = q_S \quad g_{md} = \frac{G_{md}}{G_0} = q_D \quad g_{ch} = \frac{G_{ch}}{G_0} = q_{inv} \quad , G_0 = \frac{I_0}{U_T}$$

-Συνάρτηση κανονικοποιημένου λόγου αγωγιμότητας προς κανονικοποιημένο ρεύμα:

$$g_i(i) \equiv \frac{g_{ch}}{i} = \frac{qi}{i} = \frac{\sqrt{\frac{1}{4} + i} - \frac{1}{2}}{i} = \frac{1}{\sqrt{\frac{1}{4} + i} + \frac{1}{2}}$$

-Υπολογισμός φορτίων στους ακροδέκτες:

$$q_S = F^{-1}\left(\frac{V_P - V_S}{U_T}\right)$$

$$q_D = F^{-1}\left(\frac{V_P - V_D}{U_T}\right)$$

-Συντελεστής κανονικοποίησης ρεύματος:

$$I_{Spec} = 2n\mu C_{OX}U_T^2 \frac{W}{L}$$

-Υπολογίζουμε τα δύο ρεύματα στο κανάλι και η διαφορά τους (πολλαπλασιασμένη επί το I0), είναι το πραγματικό ρεύμα του καναλιού:

$$i_F = q_S^2 + q_S$$

$$i_R = q_D^2 + q_D$$

$$I_D = I_{Spec}(i_F - i_R)$$

[7]

2.5 Λειτουργικά χαρακτηριστικά του μοντέλου MOSFET EKV

Για ένα μοντέλο τύπου MOS, τα ρεύματα που διέρχονται αυτού καθορίζονται από τις τάσεις στους ακροδέκτες του. Στην περίπτωση του EKV παρατηρείται ένα επιπλέον χαρακτηριστικό καθοριστικής σημασίας: ο υπολογισμός των φορτίων περιλαμβάνει εκτός απ τις τάσεις στους ακροδέκτες και το δημιουργούμενο κανάλι του τρανζίστορ μεταξύ πηγής(source) - υποδοχέα(drain). Με τον τρόπο αυτό, η λειτουργία του καθίσταται πολύ συγγενική με τους νόμους της φυσικής που διέπουν τους ημιαγωγούς και την διέλευση φορτίου μέσα απ' αυτούς.

Επίσης η συμπεριφορά του μοντέλου EKV, περιγράφεται σε όλα τα επίπεδα λειτουργίας από το ίδιο σύνολο μαθηματικών εξισώσεων, ανεξαρτήτως απ' το αν το μοντέλο βρίσκεται σε ασθενή, μέτρια, ισχυρή αναστροφή, σε κόρο ή σε τρίοδο. Επαρκεί έτσι το ίδιο σύνολο εξισώσεων για να περιγράψει την συνολική συμπεριφορά του, χαρακτηριστικό που προσδίδει στο μοντέλο συνέχεια στα περάσματα μεταξύ περιοχών.

Τέλος το EKV χαρακτηρίζεται από συμμετρία κατά την οποίαν υπολογίζουμε το φορτίο στον υποδοχέα, το συμμετρικό του στην πηγή και από την διαφορά τους προκύπτει το ρεύμα καναλιού [6,7].

ΚΕΦΑΛΑΙΟ 3

-ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΛΕΙΩΝ-

3.1 Εισαγωγή

Η υλοποίηση του εργαστηριακού τμήματος της εργασίας διεξήχθη με τα εξής εργαλεία: τους προσομοιωτές ηλεκτρικών κυκλωμάτων **Ngspice** και **Spectre**, τα εργαλεία μετάφρασης **ADMS** και **XML** αρχεία. Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την μοντελοποίηση του στοιχείου MOSFET EKV3 είναι οι: γλώσσα περιγραφής μοντέλων **Verilog-A** και η γλώσσα προγραμματισμού **C**. Τα στάδια της εργαστηριακής υλοποίησης είναι τα εξής:

- α) Μετάφραση του κώδικα του MOSFET EKV3 από Verilog-A σε C και παράλληλη εγκατάστασή της παραγόμενης έκδοσης στον προσομοιωτή Ngspice. Η διαδικασία αυτή πραγματοποιήθηκε με την χρήση των εργαλείων ADMS και των αρχείων XML.
- β) Εγκατάσταση της Verilog-A έκδοσης του MOSFET EKV3 στον προσομοιωτή Spectre.
- γ) Εκτέλεση του ίδιου κύκλου προσομοιώσεων του στοιχείου EKV3 στους δύο προσομοιωτές και σύγκριση των αποτελεσμάτων.
- δ) Δημιουργία και προσομοίωση κυκλώματος (διαίρετη ρεύματος-r 2r ladder circuit) που περιλαμβάνει το MOSFET EKV3 σαν ηλεκτρικό στοιχείο.

Επίσης χρησιμοποιήθηκαν τα εξής “βοηθητικά” προγράμματα: οι μεταγλωττιστές *bison* και *flex*, ο compiler *g++* και οι εξειδικευμένες για την εγκατάσταση των εργαλείων συναρτήσεις, *gawk* και *libreadline6-dev*. Στο παρόν κεφάλαιο ακολουθεί μία συνοπτική παρουσίαση και περιγραφή των παραπάνω εργαλείων και γλωσσών.

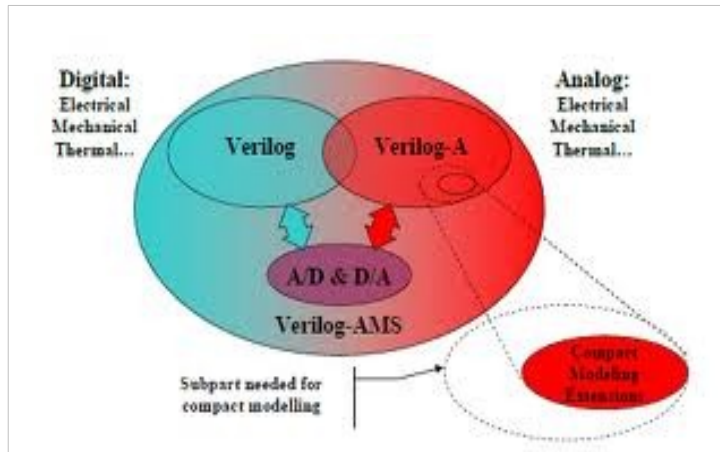
3.2 Η γλώσσα Verilog-A

Η γλώσσα Verilog-A είναι μία γλώσσα ανώτερου επιπέδου περιγραφής αναλογικών δομών και δημιουργήθηκε στα πλαίσια ανάπτυξης της οικογένειας περιγραφικών γλωσσών HDL (Hardware Description Language).

Στόχος αυτού του συνόλου γλωσσών, ήταν η βελτίωση της διαδικασίας περιγραφής μοντέλων και ο περιορισμός των διαφορετικών υλοποιήσεων του καθενός για το πλήθος των διαθέσιμων προσομοιωτών.

Το A στην ονομασία αντιστοιχεί στην αγγλική λέξη *Analogue*, που σημαίνει αναλογικό και υποδηλώνει ότι η Verilog-A είναι γλώσσα περιγραφής αναλογικών δομών. Παράλληλα αποτελεί παρακλάδι της ευρύτερης γλώσσας Verilog-AMS (Analogue and Mixed Signal) η οποία καλύπτει συστήματα μεικτών σημάτων (τόσο αναλογικά όσο και ψηφιακά).

Η Verilog-AMS αναπτύσσεται από την εταιρία με το όνομα Accellera Consortium (www.accellera.org) η οποία είναι μη κερδοσκοπικού χαρακτήρα και σχηματίστηκε από την ενοποίηση των Open Verilog International και VHDL International.



Εικόνα 3.1: Διασύνδεση των διαφορετικών ειδών Verilog.

Η Verilog-A καλύπτει ένα ευρύ φάσμα εφαρμογών σε τομείς όπως η ηλεκτρονική, η μηχανική και η θερμοδυναμική, εφόσον όμως τα συστήματα που περιλαμβάνονται είναι *συνεχούς χρόνου* και *συντηρητικά*. Το δεύτερο σημαίνει ότι η συμπεριφορά τους υπακούει τους νόμους του Kirchhoff.

Η Verilog-A ακολουθεί τα πρότυπα των γλωσσών C, C++ και δύναται να εκτελέσει εντολές τους. Από την C++ επιπλέον, δανείζεται την βασική διάρθρωση τρόπου γραφής με την χρήση των κλάσεων και άλλα δομικά χαρακτηριστικά.

Η πρώτη έκδοση της Verilog-A κυκλοφόρησε το 1995 με τον αριθμό 1.0 και ήταν αποτέλεσμα σύμπραξης εταιριών από τον χώρο της ηλεκτρονικής ημιαγωγών [7,8].

3.2.1 Βασική δομή

Βασική δομική μονάδα της Verilog-A είναι οι *οντότητες* (modules). Κάθε στοιχείο του οποίου την συμπεριφορά θέλουμε να περιγράψουμε, μετουσιώνεται προγραμματιστικά σε module και τεστάρεται σαν μέρος κάποιου μεγαλύτερου συστήματος, με την βοήθεια ενός προσομοιωτή. Έτσι ελέγχεται ο τρόπος λειτουργίας του.

Η συμπεριφορά ενός στοιχείου αποτυπώνεται ως η μαθηματική εξάρτηση μεταξύ των εισόδων και των εξόδων του. Η αλληλεπίδραση των στοιχείων στην προσομοίωση καθορίζεται από την συμπεριφορά και την μεταξύ τους συνδεσμολογία. Ο προσομοιωτής εξάγει ένα σύστημα εξισώσεων στο οποίο καταλήγει βάσει της εφαρμογής των δύο κανόνων Kirchhoff στο κύκλωμα. Στην συνέχεια λύνει το σύστημα και παράγει τα αποτελέσματα της προσομοίωσης.

-Η βασική μορφή δήλωσης μιας οντότητας είναι η:

```
module <όνομα> (<κόμβοι - θύρες>)    // αναγνωριστικό οντότητας, είσοδοι-έξοδοι
    <declares>                        // δήλωση αντικειμένων δεδομένων
    <module items>                    // δομές αρχικοποίησης και υλοποίησης
end module
```

Εικόνα 3.2: Βασική μορφή δήλωσης οντότητας.

-Ως **κόμβος** ορίζεται ένα σημείο φυσικής σύνδεσης μεταξύ συντηρητικών οντοτήτων περιγραφής συνεχούς χρόνου. Ο κόμβος δηλώνεται είτε σαν **input**, οπότε η τιμή του ορίζεται από εξωτερικό σήμα και μπορεί να συμπεριληφθεί στις εξισώσεις του στοιχείου, είτε σαν **output**, οπότε η τιμή του καθορίζεται εσωτερικά και δεν μπορεί να εισαχθεί στις εξισώσεις του στοιχείου, είτε τέλος, ως **inout** συνδυάζοντας και τις δύο παραπάνω ιδιότητες.

-Υπάρχει όμως ακόμα μία διάσταση στην ύπαρξη του κόμβου που πρέπει να δηλωθεί και αφορά την φύση του. Καθώς η χρήση της Verilog-A εκτείνεται σε πολλά επιστημονικά πεδία (ηλεκτρονική, θερμοδυναμική κλπ), με την δήλωση **discipline** αποσαφηνίζεται στο σύστημα το είδος του κόμβου και κατ' επέκταση το είδος των μονάδων που του αντιστοιχούν, όπως επίσης και η κατάλληλη ονοματολογία. Στο σημείο αυτό δηλώνονται και οι εσωτερικοί κόμβοι, εάν υπάρχουν.

```

module voltage_amplifier (out, in) ;
input in ;
output out ;
voltage out ,      // Discipline voltage defined elsewhere
           in ;     // with potential access function V()
parameter real GAIN_V = 10.0 ;

analog
    V(out) <+ GAIN_V * V(in) ;

endmodule

```

Εικόνα 3.3: Παράδειγμα δήλωσης οντότητας.

-Η περιγραφή των discipline συσχετίζεται με την χρήση των **nature**. Τα τελευταία είναι μία συλλογή από μετρήσιμες οντότητες (τάση, φορτίο, ταχύτητα, μήκος κλπ.) και είναι απαραίτητο για την τιμή κάθε κόμβου, να δηλωθεί η αντίστοιχη nature πριν την δήλωση του discipline. Η δήλωση των nature γίνεται σε top level.

```

nature current
    units = "A" ;
    access = I ;
    idt_nature = charge ;
    abstol = 1u ;
endnature

nature voltage
    units = "V" ;
    access = V;
endnature

```

Εικόνα 3.4: Παράδειγμα δήλωσης nature.

-Το θεμελιώδες χαρακτηριστικό των κόμβων είναι ότι η τιμή τους προκύπτει από ταυτόχρονη επίλυση εξισώσεων. Οι εξισώσεις καθορίζονται από τα στιγμιότυπα των στοιχείων που συνδέονται με αυτούς και διαμορφώνονται βάσει των νόμων του Kirchhoff.

-Οι δηλώσεις των κόμβων, των discipline και των παραμέτρων του στοιχείου λαμβάνουν χώρα στο κομμάτι **declares**. Ακολουθεί το κομμάτι **module items**, που περιλαμβάνει τις απαιτούμενες αρχικοποιήσεις καθώς και τις εξισώσεις που περιγράφουν την λειτουργία του στοιχείου [7,10,11].

3.2.2 Εντολές ελέγχου ροής και επανάληψης

Η Verilog-A περιλαμβάνει όλες τις γνωστές δομές ελέγχου και επανάληψης που συναντώνται στις διάφορες γλώσσες προγραμματισμού:

-**if** statement:

```
if_else_if_statement ::=
    if (expression) statement
    { else if (expression) statement }
    else statement
```

Εικόνα 3.5: Βασική μορφή δήλωσης if.

-**case** statement:

```
case_statement ::=
    case ( expression ) case_item { case_item } endcase
case_item ::=
    expression { , expression } : statement
    | default [ : ] statement
```

Εικόνα 3.6: Βασική μορφή δήλωσης case.

-**for**, **while**, **repeat** statements:

```
looping_statement ::=
    forever statement
    | repeat ( expression ) statement
    | while ( expression ) statement
    | for ( procedural_assignment ; expression ;
          procedural_assignment ) statement
```

Εικόνα 3.7: Βασική μορφή δήλωσης βρόγχων επανάληψης.

Επιπλέον δίνει την δυνατότητα για ακόμα μεγαλύτερη αμεσότητα στην διαχείριση ροής του προγράμματος με εντολές όπως οι:

-**\$stop**, **\$finish**, που είναι εντολές εξόδου από το σύστημα:

```
Syntax:  
  
    $stop [(n)] ;  
  
    $finish [(n)]
```

Εικόνα 3.8: Βασική μορφή δήλωσης εντολών *\$stop* και *\$finish*.

-**initial_step**, **final_step**, **cross**, **timer**. που είναι εντολές αναγνώρισης αλλαγών σε σήματα και καταστάσεις. Οι δύο πρώτες αφορούν την έναρξη και λήξη της εκτέλεσης αντίστοιχα, η τρίτη την εκτέλεση συγκεκριμένης εντολής με την υλοποίηση μίας συνθήκης και η τέταρτη την εκτέλεση εντολής σε προκαθορισμένη χρονική στιγμή.

```
analog begin  
  
    @ cross (V(smpl)-2.5,+1)  
        V(out) <+ 0 ;  
  
end
```

Εικόνα 3.9: Βασική μορφή δήλωσης εντολής *cross*.

Στο παραπάνω παράδειγμα το V(out) γίνεται 0, όταν η παράσταση (V(smpl)-2.5) γίνει από αρνητική, θετική. Το πέρασμα στον θετικό ημιάξονα εκφράζει η χρήση του +1 [7,11].

3.2.3 Συναρτήσεις

-white_noise, **flicker_noise**, **noise_table**. Αυτές οι συναρτήσεις της Verilog-A χρησιμοποιούνται για την έκφραση θορύβου. Η πρώτη εκφράζει λευκό θόρυβο, η δεύτερη θόρυβο flicker ενώ η τρίτη περιλαμβάνει όλα τα είδη θορύβου.

-\$temperature, **\$vt**, **\$abstime**, **analysis**. Οι πρώτες τρεις είναι συναρτήσεις που επιτρέπουν την επικοινωνία του χρήστη με το περιβάλλον προσομοίωσης, παρέχοντας στοιχεία όπως η θερμοκρασία του κυκλώματος και το χρονικό σημείο της προσομοίωσης. Η τελευταία επιτρέπει στον κώδικα να αναγνωρίσει το είδος της προσομοίωσης που εκτελείται, δίνοντας στον χρήστη την δυνατότητα διαδραστικής παρέμβασης στην συμπεριφορά του στοιχείου [7].

3.2.4 Λεξικοί κανόνες

-Οι λεξικοί κανόνες της Verilog-A παρουσιάζουν πολλές ομοιότητες με αυτούς της C. Τα σχόλια εισάγονται στον κώδικα ακολουθώντας τα σύμβολα `//`, εάν πρόκειται για μία γραμμή ή με `/*σχόλια*/`, αν πρόκειται για ομάδα σχολίων.

-Η γλώσσα είναι ευαίσθητη στον διαχωρισμό μεταξύ κεφαλαίων και μικρών γραμμάτων (case sensitive).

-Οι αριθμοί αποτυπώνονται με τη συνηθισμένη μορφή ή ως εξής:

<πρόσημο><μέγεθος><βάση αριθμού><αριθμός>,

όπου το μέγεθος είναι ο αριθμός των ψηφίων που αποτελούν τον αριθμό και η βάση ο χαρακτήρας `<< ' >>`, ακολουθούμενος από το αρχικό γράμμα του συστήματος που χρησιμοποιείται, στα αγγλικά (**b**inary,**d**ecimal,**o**ctal,**h**ex) [10].

3.3 Υλοποιήσεις του μοντέλου

Το MOSFET EKV3 υλοποιείται με πέντε διαφορετικούς τρόπους, εκ των οποίων ο καθένας παρουσιάζει μοναδικά χαρακτηριστικά. Η διαφοροποίηση αυτή καθιστά την κάθε έκδοση του μοντέλου κατάλληλη για ενσωμάτωση σε συγκεκριμένα κυκλώματα, ανάλογα με την συχνότητα λειτουργίας κάθε κυκλώματος.

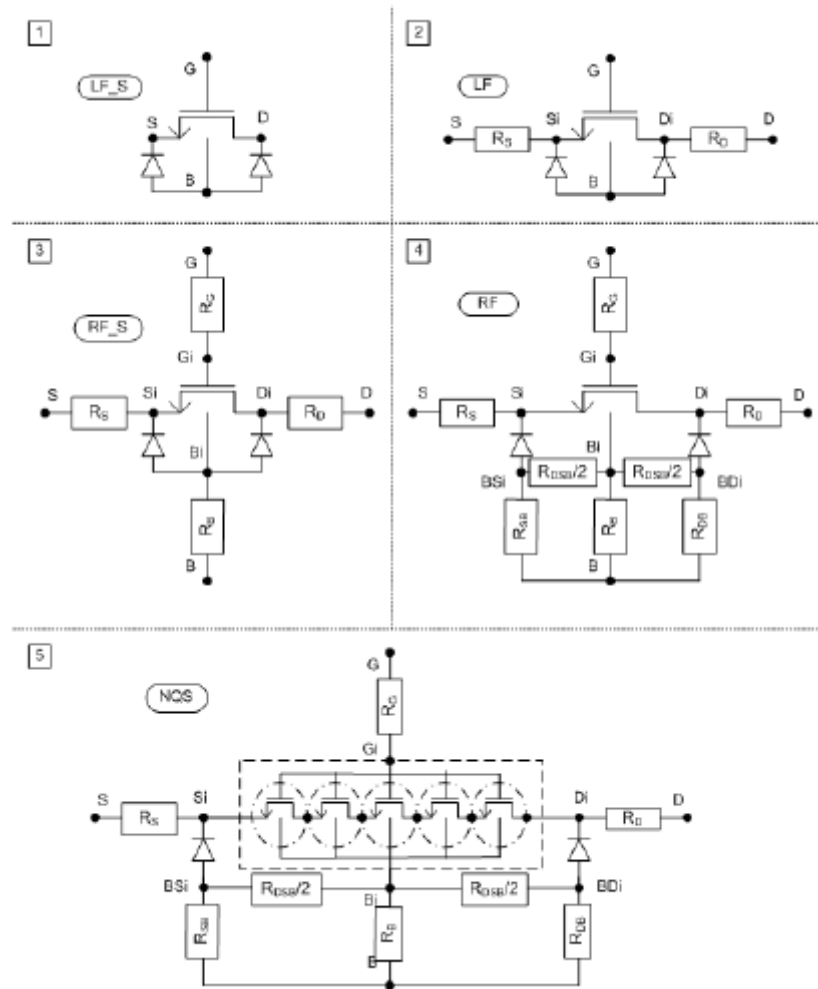
3.3.1 Το μοντέλο σε Verilog-A

Η υλοποίηση του μοντέλου σε Verilog-A απαρτίζεται από 19 αρχεία. Το βασικό αρχείο (ekv3.va) “καλεί” τα υπόλοιπα που περιλαμβάνουν δηλώσεις παραμέτρων, συναρτήσεις, μοντελοποίηση φαινομένων και ενημερώσεις.

Ο κώδικας περιγράφει πέντε διαφορετικές υλοποιήσεις του μοντέλου, που διαφέρουν κατά κύριο λόγο στον αριθμό των εξωτερικών αντιστατών. Στην απλούστερη έκδοση δεν γίνεται καμία εισαγωγή εξωτερικής αντίστασης, ενώ στην επόμενη εισάγονται δύο σειριακές αντιστάσεις στους ακροδέκτες drain και source, δημιουργώντας δύο εσωτερικούς κόμβους. Η έκδοση αυτή είναι κατάλληλη για κυκλώματα χαμηλών συχνοτήτων.

Στην τρίτη έκδοση προστίθενται δύο ακόμα αντιστάσεις στους ακροδέκτες gate και body, δημιουργώντας τέσσερις εσωτερικούς κόμβους και καθιστώντας την έκδοση κατάλληλη για κυκλώματα υψηλότερων συχνοτήτων. Η τέταρτη έκδοση του μοντέλου περιλαμβάνει ένα δικτύωμα πέντε αντιστατών στο υπόστρωμα και έξι εσωτερικών κόμβων. Προτιμάται για υψίσυχνα κυκλώματα.

Η πέμπτη και πολυπλοκότερη εκδοχή του μοντέλου χρησιμοποιεί το εξωτερικό υποδίκτυο της προηγούμενης έκδοσης ενώ παράλληλα εφαρμόζει την τεχνική της μινιμαλιστικής κατάτμησης καναλιού σε πέντε τμήματα, διαμορφώνοντας συνολικά δέκα εσωτερικούς κόμβους. Αυτή η εκδοχή χρησιμοποιείται σε κυκλώματα που εμφανίζουν φαινόμενα μη στατικής συμπεριφοράς [8].



Εικόνα 3.10: Οι πέντε εκδοχές του μοντέλου σε Verilog-A.

3.3.2 Το μοντέλο σε C

Οι προσωμοιωτές ηλεκτρικών κυκλωμάτων είναι στην πλειοψηφία τους γραμμένοι σε γλώσσα C. Η εισαγωγή ενός ηλεκτρικού στοιχείου σε έναν από αυτούς ακολουθεί την εξής διαδικασία:

- Γραφή των εξισώσεων του στοιχείου σε γλώσσα C, βάσει των ειδικών χαρακτηριστικών λειτουργίας και της ορολογίας του προσωμοιωτή.
- Μεταγλώττιση κώδικα σε μορφή δυναμικής βιβλιοθήκης συναρτήσεων.
- Φόρτωση βιβλιοθήκης απ'τον προσομοιωτή και διαχείριση του μοντέλου σαν στοιχείο κυκλώματος. Η διαδικασία μετάφρασης του μοντέλου EKV3 από Verilog-A σε C περιγράφεται στο κεφάλαιο 4 [8].

3.3.3 Πλεονεκτήματα και μειονεκτήματα μοντελοποίησης σε κάθε γλώσσα

Verilog-A:

-Το σημαντικότερο πλεονέκτημα της περιγραφής μοντέλου σε γλώσσα Verilog-A, είναι ότι παρέχει την δυνατότητα ενσωμάτωσής του μοντέλου στην πλειοψηφία των σύγχρονων προσομοιωτών, δίχως να απαιτούνται προσθήκες ή αλλαγές. Επαρκεί δηλαδή η δημιουργία ενός αρχείου για την περιγραφή και ενσωμάτωση του στοιχείου σε οποιονδήποτε σχεδόν προσομοιωτή.

-Αυτή η καθολική συμβατότητα ωστόσο, ενδεχομένως να μην εκμεταλλεύεται πλήρως τις δυνατότητες του κάθε προσομοιωτή, γεγονός που αποτελεί μειονέκτημα.

-Το σημαντικότερο μειονέκτημα της Verilog-A σχετίζεται με την αποδοτικότητα των περιγραφόμενων μοντέλων, καθώς απαιτεί την χρήση ενός μεταφραστή (interpreter), που παρέχει σημαντικά πιο αργά αποτελέσματα από τον μεταγλωττιστή (compiler) της γλώσσας C. Το χαρακτηριστικό αυτό καθιστά την Verilog-A ακατάλληλη για πραγματικά κυκλώματα με υψηλή πολυπλοκότητα [8].

C:

-Αντίστοιχα με τα παραπάνω, πλεονέκτημα της περιγραφής ενός μοντέλου σε γλώσσα C, είναι η απόλυτη συμβατότητα με τον εκάστοτε προσομοιωτή. Παρέχεται έτσι στον χρήστη η δυνατότητα για πλήρη εκμετάλλευση των ιδιαίτερων χαρακτηριστικών και δυνατοτήτων του εργαλείου.

-Ωστόσο η συγγραφή διαφορετικού κώδικα για κάθε προσομοιωτή είναι βαριά διαδικασία που συχνά οδηγεί σε λάθη και διαφοροποιήσεις μεταξύ των υλοποιήσεων του ίδιου μοντέλου.

-Επιπλέον περιπλέκεται η διαδικασία εξέλιξης του μοντέλου, καθώς η παραμικρή αλλαγή ή βελτίωση, μπορεί να συνεπάγεται επανεγγραφή μεγάλου μέρους του κώδικα [8].

3.4 Ο προσομοιωτής Ngspice

Το πρόγραμμα Ngspice είναι ένας γενικού τύπου και ελεύθερης πρόσβασης προσομοιωτής κυκλωμάτων, για μη γραμμικές dc, μη γραμμικές transient και γραμμικές ac αναλύσεις. Τα προσομοιούμενα κυκλώματα μπορεί να περιέχουν αντιστάτες, πυκνωτές, πηνία, αμοιβαία πηνία, ανεξάρτητες πηγές τάσεως και πηγές ρεύματος, τεσσάρων ειδών εξαρτημένες πηγές, γραμμές μεταφοράς με και χωρίς απώλειες (δυο διαφορετικές υλοποιήσεις), διακόπτες, ομοιόμορφα κατανεμημένες γραμμές RC, και τις πέντε πιο γνωστές ημιαγωγικές διατάξεις: διόδους, BJT, JFET, MESFET και MOSFET.

Ο προσομοιωτής Ngspice αποτελεί συνέχεια του προγράμματος Spice3f5, τελευταίας ανανέωσης του Berkeley για την οικογένεια προσομοιωτών Spice3. Η δημιουργία του είχε ως στόχο την ανάπτυξη νέων χαρακτηριστικών και την διόρθωση λαθών της τελευταίας έκδοσης.

Ο Ngspice περιλαμβάνει κάποια ενσωματωμένα μοντέλα ηλεκτρικών στοιχείων – ημιαγωγών, που χρειάζονται απλή ρύθμιση παραμέτρων απ' τον χρήστη πριν προσομοιωθούν. Το EKV3 δεν περιλαμβάνεται σε αυτά και ενσωματώνεται στον προσομοιωτή χειρωνακτικά.

Επιπλέον ο Ngspice περιλαμβάνει ένα κύκλωμα μεικτών επιπέδων - προσομοιωτή μοντέλων, το CIDER, που παρέχει άμεση σύνδεση μεταξύ των τεχνολογικών παραμέτρων και της απόδοσης ενός κυκλώματος. Το CIDER παρέχει μεγαλύτερη ακρίβεια στην προσομοίωση απ'οτι ένα μεμονωμένο κύκλωμα μεικτών επιπέδων ή ένας μεμονωμένος προσομοιωτής μοντέλων, καθώς μοντελοποιεί αριθμητικά τα κρίσιμα στοιχεία ενός κυκλώματος.

Κάθε στοιχείο του προσομοιούμενου κυκλώματος καθορίζεται από το όνομά του, τους κόμβους του κυκλώματος με τους οποίους συνδέεται καθώς και τις τιμές των παραμέτρων που καθορίζουν τα ηλεκτρικά χαρακτηριστικά του. Το κύκλωμα περιγράφεται στον Ngspice από ένα σεντ δηλώσεων που καθορίζουν την τοπολογία του κυκλώματος και τις τιμές των στοιχείων του καθώς επίσης και από μία σειρά

δηλώσεων ελέγχου που καθορίζουν τις παραμέτρους του μοντέλου και τις προδιαγραφές της προσομοίωσης.

Ο προσομοιωτής Ngspice παρέχει την δυνατότητα των παρακάτω προσομοιώσεων: *DC* ανάλυση, *AC Small-Signal* ανάλυση, *Transient* ανάλυση, *Pole-Zero* ανάλυση, *Small-Signal Distortion* ανάλυση, *Sensitivity* ανάλυση, *Noise* ανάλυση, ενώ επιτρέπει την διεξαγωγή όλων των παραπάνω προσομοιώσεων σε διάφορες θερμοκρασίες κυκλώματος [12].

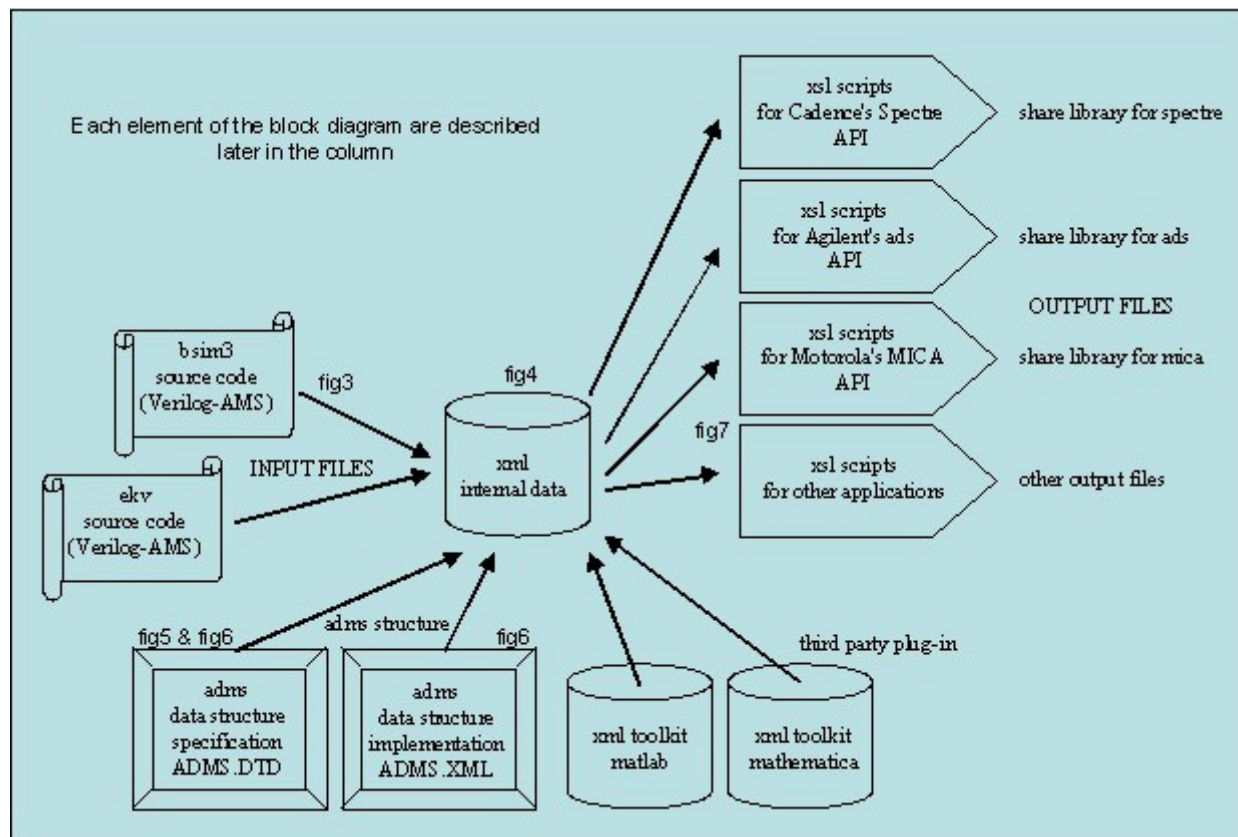
3.5 Το εργαλείο ADMS

Η μοντελοποίηση ενός ηλεκτρικού στοιχείου και η ενσωμάτωσή του σε κάποιον προσομοιωτή, αποτελούν χρονοβόρες και εκτενείς διαδικασίες που ενέχουν μεγάλο κίνδυνο λαθών και πολλά παρακλάδια προς διαχείριση.

Το ADMS (Automatic Device Model Synthesizer - Αυτόματος Συνθέτης Μοντέλων Στοιχείων) είναι ελεύθερο λογισμικό που υποστηρίζει και απλοποιεί την ανάπτυξη, διανομή και συντήρηση συμπαγών μοντέλων ηλεκτρικών στοιχείων. Το εργαλείο λαμβάνει σαν είσοδο τον κώδικα ενός μοντέλου σε γλώσσα Verilog-A και παράγει τον αντίστοιχο κώδικα σε γλώσσα C. Μερικές εκατοντάδες γραμμές του εισαγόμενου κώδικα, σε high-level γλώσσα Verilog-A, αντιστοιχούν σε δεκάδες χιλιάδες γραμμές παραγόμενου κώδικα, low-level γλώσσας C.

Το ADMS αυτοματοποιώντας την παραπάνω διαδικασία απαλλάσσει τον σχεδιαστή από την χειρωνακτική μετατροπή του κώδικα, με όλες τις δυσκολίες που αυτή συνεπάγεται, υπερπηδώντας έτσι ένα μεγάλο εμπόδιο στο τεστάρισμα και την ανάπτυξη των μοντέλων. Παράλληλα το ADMS παρέχει την δυνατότητα για εκμετάλλευση των πλεονεκτημάτων και των δύο γλωσσών καθώς διατηρεί την ύπαρξη ενός μόνον αρχείου σε γλώσσα Verilog-A, συμβατού με όλους σχεδόν τους σύγχρονους προσομοιωτές, ενώ επιτρέπει παράλληλα την μεταγλώττιση του εξαγόμενου C κώδικα, σε μορφή δυναμικής βιβλιοθήκης, πετυχαίνοντας πολύ ταχύτερη λειτουργία του μοντέλου.

Το ADMS έχει δομηθεί έτσι ώστε να εκμεταλλεύεται βέλτιστα τα υπάρχοντα εργαλεία λογισμικού. Η εικόνα 4.1 απεικονίζει την αρχιτεκτονική του ADMS και δείχνει πως τα διάφορα μέρη του αλληλεπιδρούν μεταξύ τους:



Εικόνα 3.11: Τα ενσωματωμένα χαρακτηριστικά του ADMS.

Το εργαλείο ADMS είναι ανοιχτού κώδικα και δεν σχετίζεται συγκεκριμένα με κάποιον προσομοιωτή. Ωστόσο η λειτουργία του βασίζεται στους ειδικούς κανόνες που διέπουν τον κάθε προσομοιωτή, οι οποίοι κατ' επέκταση καθορίζουν και την δομή του παραγόμενου απ' το εργαλείο, κώδικα [7,13].

ΚΕΦΑΛΑΙΟ 4

- ΕΝΣΩΜΑΤΩΣΗ ΜΟΝΤΕΛΟΥ EKV3 ΣΤΟΝ NGSPICE-

4.1 Εισαγωγή

Ο προσομοιωτής Ngspice περιλαμβάνει στην αρχική του εγκατάσταση διάφορα μοντέλα ηλεκτρικών στοιχείων, στα οποία όμως δεν συγκαταλέγεται το MOSFET EKV3. Το μοντέλο χρειάζεται να ενσωματωθεί χειρωνακτικά στον προσομοιωτή και να μεταφραστεί από Verilog-A σε C, πριν μπορέσουμε να το προσομοιώσουμε. Στο παρόν κεφάλαιο περιγράφονται οι διαδικασίες ενσωμάτωσης του μοντέλου στον προσομοιωτή Ngspice.

4.2 Διαδικασία ενσωμάτωσης μοντέλου EKV3 στον προσομοιωτή Ngspice

Ο κώδικας Verilog-A του `ekv3` περιγράφει πέντε διαφορετικές εκδόσεις του στοιχείου που πρέπει να ενσωματωθούν χειρωνακτικά στον προσομοιωτή καθώς δεν συγκαταλέγονται στα ήδη υπάρχοντα μοντέλα του. Συγκεκριμένα για την εισαγωγή κάθε μίας από τις εκδόσεις του μοντέλου, ακολουθήθηκε η εξής διαδικασία, που ενδεικτικά περιγράφεται για την έκδοση “`ekv3_s`” :

1.α) Αντιγραφή του φακέλου με την υλοποίηση του μοντέλου σε γλώσσα Verilog-A στο folder path: `~/ngspice/src/spicelib/devices/adms` και μετονομασία του φακέλου στο ακριβές όνομα της έκδοσης που θέλουμε να εισάγουμε (`ekv3_s`).

β)Μετονομασία του αρχείου **ekv3.va** που περιλαμβάνει ο παραπάνω φάκελος στο ακριβές όνομα της έκδοσης και παρέμβαση σε επίπεδο κώδικα στο πεδίο “define”, αποτυπώνοντας ξανά το ακριβές όνομα της προστιθέμενης έκδοσης:

```
`define DC_S  
    module ekv3_s (d,g,s,b);
```

2.Στο αρχείο **inpdomod.c**, με folder path: ~/ngspice/src/spicelib/parser/inpdomod.c, πρόσθεση στην λίστα των μοντέλων και συγκεκριμένα στο πεδίο των MOSFET, ενός “case” με το όνομα της έκδοσης που εισάγεται:

```
.case 84:  
  
    type = INPtypelook("ekv3_s");  
    if (type < 0) {  
        err =  
        INPmkTemp  
        ("Device type EKV3_S not available in this binary\n");  
    }  
    break;
```

3.Στο αρχείο **inp2m.c**, με folder path:~/ngspice/src/spicelib/parser/inp2m.c και συγκεκριμένα στο πεδίο “#ifndef ADMS” πρόσθεση της γραμμής:

```
&& thismodel->INPmodType != INPtypelook ("ekv3_s")
```

4.Στο αρχείο **dev.c**, με folder path:~/ngspice/src/spicelib/devices/dev.c, γίνονται οι εξής προσθήκες:

α) Στο πεδίο με τις δηλώσεις στοιχείων “#include” και συγκεκριμένα στο τμήμα “#ifndef ADMS”, προσθήκη της γραμμής:

```
#include "adms/ekv3_s/ekv3_sitf.h"
```

β) Στο πεδίο ορισμού τιμών για την λίστα “DEVices[]” και συγκεκριμένα στο τμήμα “#ifndef ADMS”, προσθήκη της γραμμής:

```
DEVices[62] = (SPICEdev*)get_ekv3_s_info();
```

γ) Στο πεδίο “#define DEVICES_USED{ }” και συγκεκριμένα στο τμήμα “#ifndef ADMS”, προσθήκη του ονόματος της έκδοσης που εισάγεται ως εξής:

```
,"ekv3_s"
```

5. Στο αρχείο **configure.ac**, με folder path: ~/ngspice/configure.ac, γίνονται οι εξής προσθήκες:

α) Στο πεδίο “VLADEVDIR” προσθήκη της γραμμής:

```
adms/ekv3_s \
```

β) Στο πεδίο “#VLAMKF” προσθήκη της γραμμής:

```
#VLAMKF src/spicelib/devices/adms/ekv3_s/Makefile
```

γ) Στο πεδίο “VLADEV=" " "” προσθήκη της γραμμής:

```
spicelib/devices/adms/ekv3_s/libekv3_s.la \
```

6. Στο αρχείο **ngspiceVersion.xml**, με folder path: ~/ngspice/src/spicelib/devices/adms
/admst/ngspiceVersion.xml, προστίθεται ο εξής κώδικας:

```
<admst:when test="[name='ekv3_s']">
  <admst:if test="[nilled(variable[name='nmos'])]">
    <admst:value-of select="analog"/>
    <admst:value-of select="."/>
    <admst:new datatype="variable" arguments="%p,nmos,%p">
      <admst:push into="module/variable" select="." onduplicate="ignore"/>
      <admst:value-to select="sizetype" value="scalar"/>
      <admst:value-to select="type" value="integer"/>
      <admst:value-to select="input" value="yes"/>
      <admst:value-to select="output" value="yes"/>
      <admst:value-to select="parametertype" value="model"/>
      <admst:value-to select="scope" value="global_model"/>
      <admst:new datatype="number" arguments="1">
        <admst:value-to select="scalingunit" value="1"/>
        <admst:value-of select="."/>
        <admst:value-of select="../module"/>
        <admst:new datatype="expression" arguments="%p,%p">
          <admst:value-of select="."/>
          <admst:value-to select="../default" value="%p"/>
        </admst:new>
      </admst:new>
    </admst:new>
  </admst:if>
  <admst:if test="[nilled(variable[name='pmos'])]">
    <admst:value-of select="analog"/>
    <admst:value-of select="."/>
    <admst:new datatype="variable" arguments="%p,pmos,%p">
      <admst:push into="module/variable" select="." onduplicate="ignore"/>
      <admst:value-to select="sizetype" value="scalar"/>
      <admst:value-to select="type" value="integer"/>
      <admst:value-to select="input" value="yes"/>
      <admst:value-to select="output" value="yes"/>
      <admst:value-to select="parametertype" value="model"/>
      <admst:value-to select="scope" value="global_model"/>
      <admst:new datatype="number" arguments="1">
        <admst:value-to select="scalingunit" value="1"/>
        <admst:value-of select="."/>
        <admst:value-of select="../module"/>
        <admst:new datatype="expression" arguments="%p,%p">
          <admst:value-of select="."/>
          <admst:value-to select="../default" value="%p"/>
        </admst:new>
      </admst:new>
    </admst:new>
  </admst:if>
</admst:when>
```


Ο παραπάνω κώδικας τοποθετείται πριν το τμήμα:

```
<admst:otherwise>
  <admst:value-of select="name"/>
  <admst:fatal format="%s: device not handled by the adms ngspice interface\n"/>
</admst:otherwise>
</admst:choose>
</admst:for-each>

</admst>
```

7.Τέλος στο αρχείο **ekv3_018.par**, με folder path: ~/ekv/ **ekv3_018.par**, καθορίζεται το νέο στοιχείο προς προσομοίωση, χρησιμοποιώντας τον αριθμό του “case” από το 2ο βήμα στο πεδίο “+ LEVEL”, ως εξής:

$$+ LEVEL = 84$$

Με την ολοκλήρωση αυτής της διαδικασίας έχουν γίνει οι απαραίτητες μετατροπές για την εισαγωγή της επιθυμητής έκδοσης του μοντέλου **ekv3** στον προσομοιωτή Ngspice.

Για την προσομοίωση του μοντέλου χρειάζεται απεγκατάσταση της τρέχουσας εφαρμογής του Ngspice και εγκατάσταση της νέας με τις αλλαγές. Η διαδικασία που ακολουθείται είναι η εξής:

- 1.Άνοιγμα του **terminal** των linux και μετάβαση στον φάκελο του Ngspice, με την εντολή **cd**.
- 2.Χρήση της εντολής “**sudo make clean**”, με την οποία διαγράφονται από το σύστημα όλα τα compiled αρχεία του προσομοιωτή.
- 3.Χρήση της εντολής “**sudo make uninstall**”, με την οποία απεγκαθίσταται το πρόγραμμα από το σύστημα.

4.Μετάβαση στο script `ng_sim`, με folder path: `~/ngspice/ng_sim`, που περιλαμβάνει τα πρώτα δύο βήματα της εγκατάστασης (εντολές: `./autogen.s` και `./configure`) και εκτέλεση με την εντολή `“./ng_sim”`.

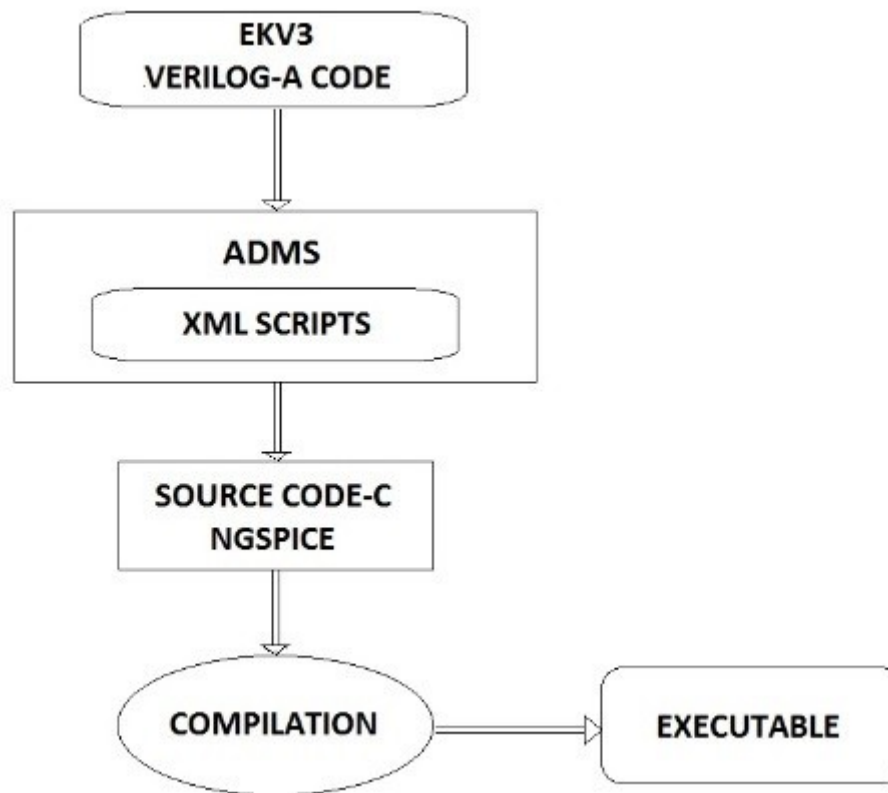
5.Χρήση της εντολής `“sudo make install”`, με την οποία εγκαθίσταται το νέο πρόγραμμα στο σύστημα.

Εαν η εγκατάσταση είναι επιτυχής, προχωρούμε σε προσομοίωση.

4.3 Διαδικασία μετάφρασης κώδικα μοντέλου EKV3 από Verilog-A σε C

Κατά την εγκατάσταση του προσομοιωτή Ngspice στο λειτουργικό σύστημα και συγκεκριμένα κατά την φάση του compilation (μεταγλώττιση), καλείται το εργαλείο ADMS. Ρόλος του είναι να “ενορχηστρώνει” την μετάφραση των μοντέλων του φακέλου ADMS, από Verilog-A σε γλώσσα C. Η διαδικασία αυτή γίνεται βάσει των XML Scripts που περιλαμβάνονται στον ίδιο φάκελο.

Ο προσομοιωτής Ngspice περιλαμβάνει τον φάκελο ADMS, ο οποίος φέρει το όνομα του αντίστοιχου εργαλείου. Ο φάκελος αυτός περιέχει διάφορα μοντέλα ηλεκτρικών στοιχείων, σε γλώσσα Verilog-A, που προορίζονται για μετάφραση (σε γλώσσα C) και ενσωμάτωση στον προσομοιωτή Ngspice. Ο φάκελος ADMS επίσης περιέχει τα XML Scripts που χρησιμοποιούνται ως οδηγός μετάφρασης και “υποδεικνύουν” στους μεταγλωττιστές (χρησιμοποιήθηκαν οι: `bison`, `flex`) την διαδικασία μετατροπής του κώδικα των μοντέλων από Verilog-A σε C.



Εικόνα 4.1: Διαδικασία μετάφρασης κώδικα μοντέλου *EKV3* από *Verilog-A* σε *C*.

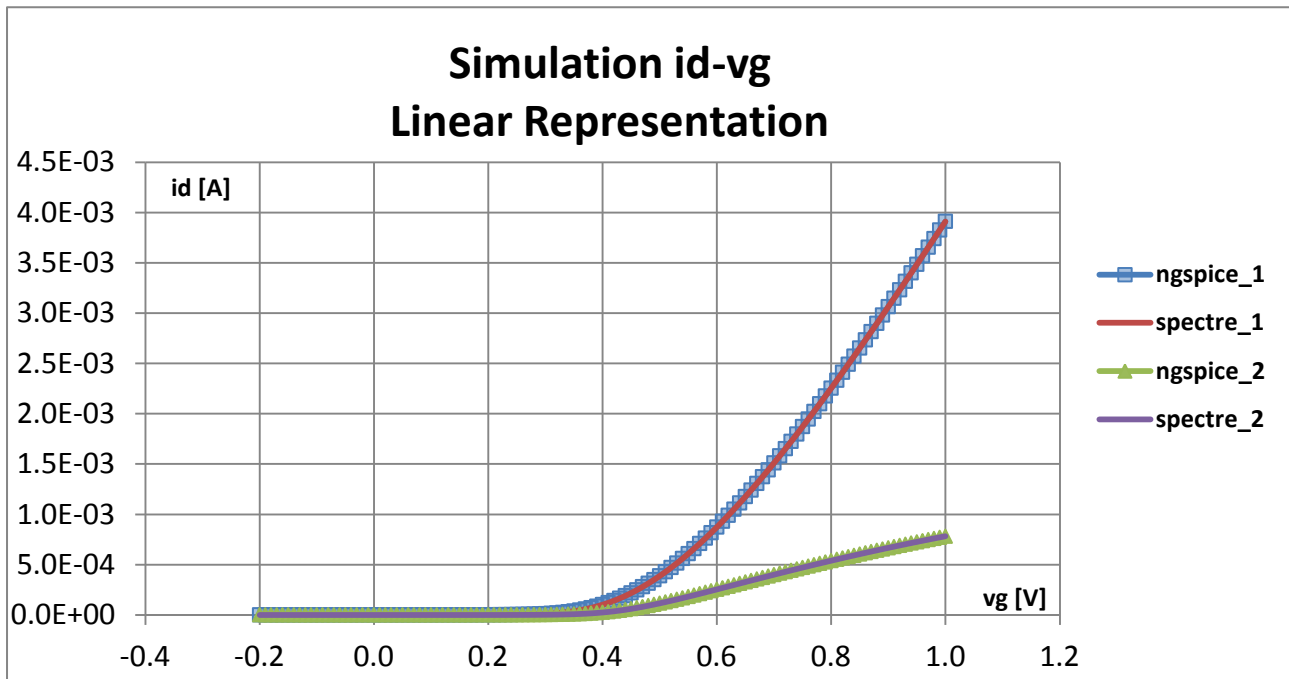
ΚΕΦΑΛΑΙΟ 5

-ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΕΩΝ-

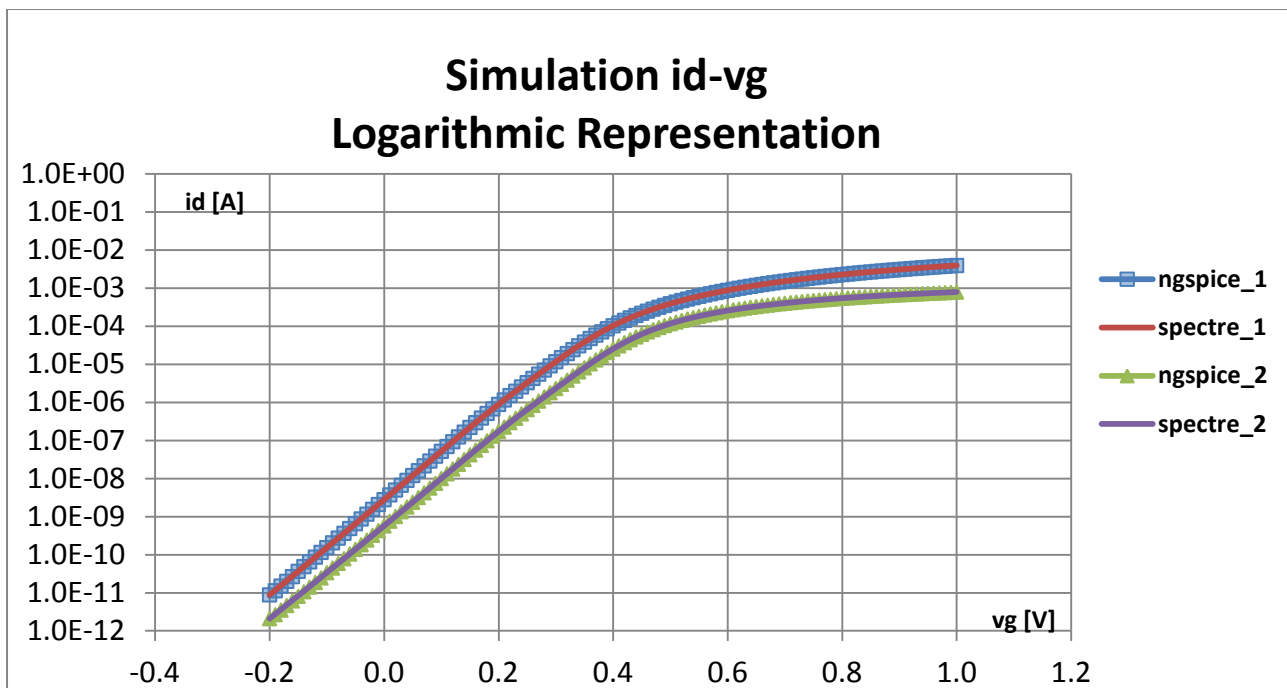
5.1 Εισαγωγή

Στο παρόν κεφάλαιο παρατίθενται τα διαγραμματικά αποτελέσματα και η ερμηνεία των προσομοιώσεων του μοντέλου MOSFET EKV3, που πραγματοποιήθηκαν στα πλαίσια της εργαστηριακής υλοποίησης. Συγκεκριμένα διεξήχθησαν προσομοιώσεις τύπου DC (id-vg, id-vd, id-vs), AC (Cgb, Cgc, Cgg) και CIRCUIT BASED (διαίρετης ρεύματος-r 2r ladder circuit) που περιλαμβάνει το μοντέλο MOSFET EKV3 ενσωματωμένο.

5.2 Προσομοιώσεις τύπου DC



Εικόνα 5.1: Διάγραμμα linear id-vg/ $L=0.1\mu$.



Εικόνα 5.2: Διάγραμμα logarithmic id-vg/ $L=0.1\mu$.

Προσομοιώσεις:

ngspice_1, spectre_1: id-vg => -200mv => 1v/10mv|w=10μ|L=0.1μ|vd=1v|vs=0v|vb=0v

ngspice_2, spectre_2: id-vg => -200mv => 1v/10mv|w=10μ|L=0.1μ|vd=0.05v|vs=0v|vb=0v

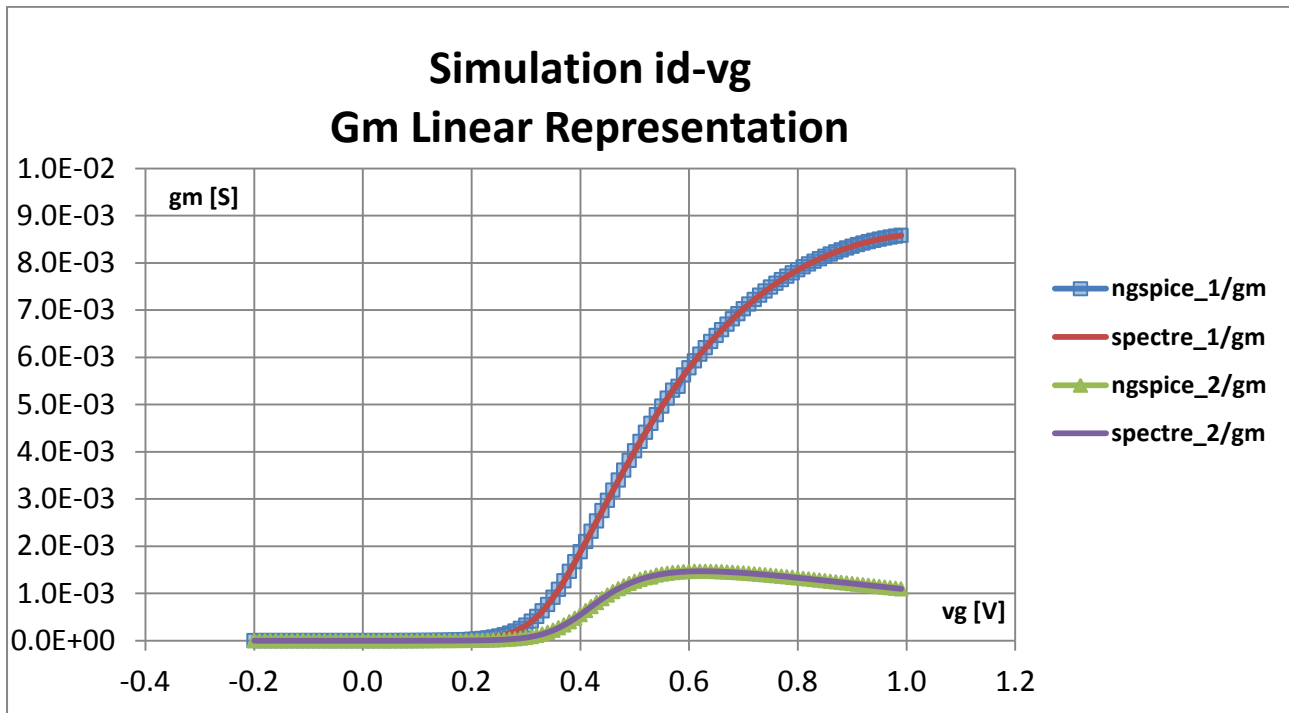
Ενδεικτική Παράθεση κώδικα προσομοίωσης **ngspice_1**:

```
.include ekv3_090.par
vd d 0 dc 1
vg g 0 dc 1
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

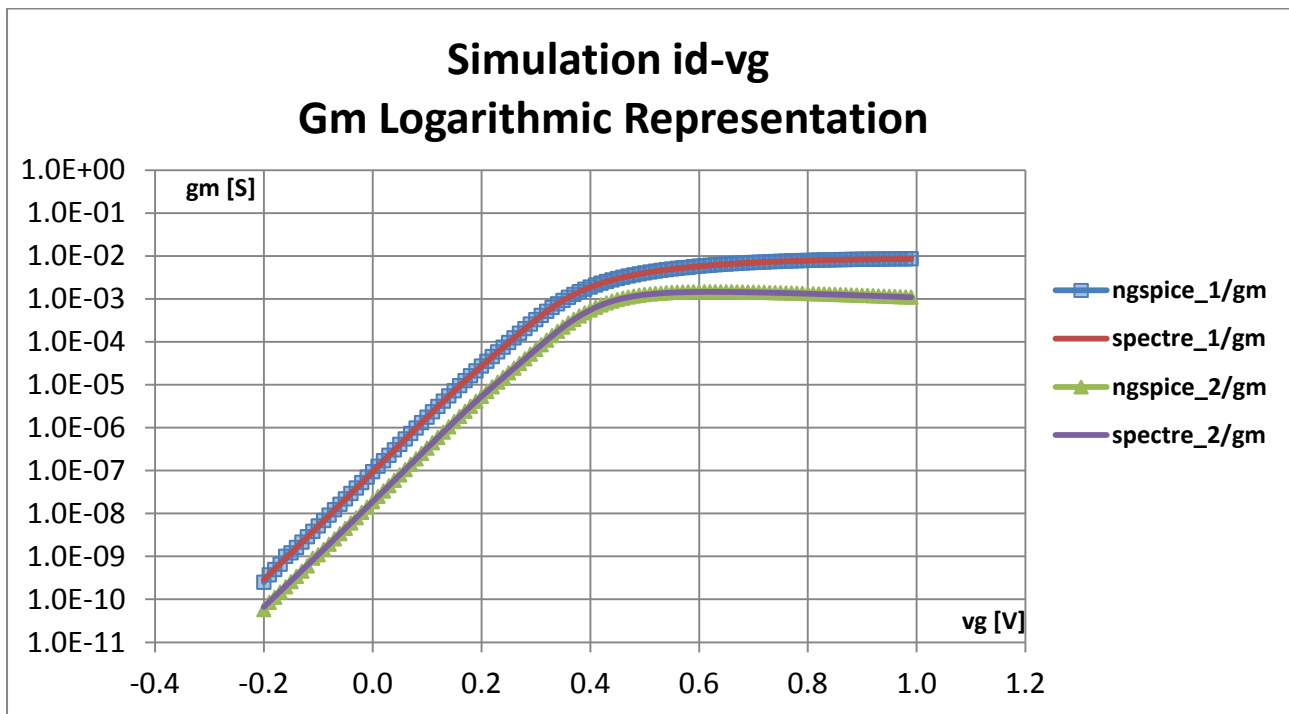
.control
dc vg -200E-3 1 0.01
set filetype=ascii
run
echo vg "          " id > a1.data
set appendwrite
wrdata a1 i(vs)
.endc

.end
```

Τα netlist των προσομοιώσεων του μοντέλου MOSFET EKV3 στον προσομοιωτή Ngspice, παρατίθενται στο Παράρτημα.



Εικόνα 5.3: Διάγραμμα linear g_m - v_g / $L=0.1\mu$.

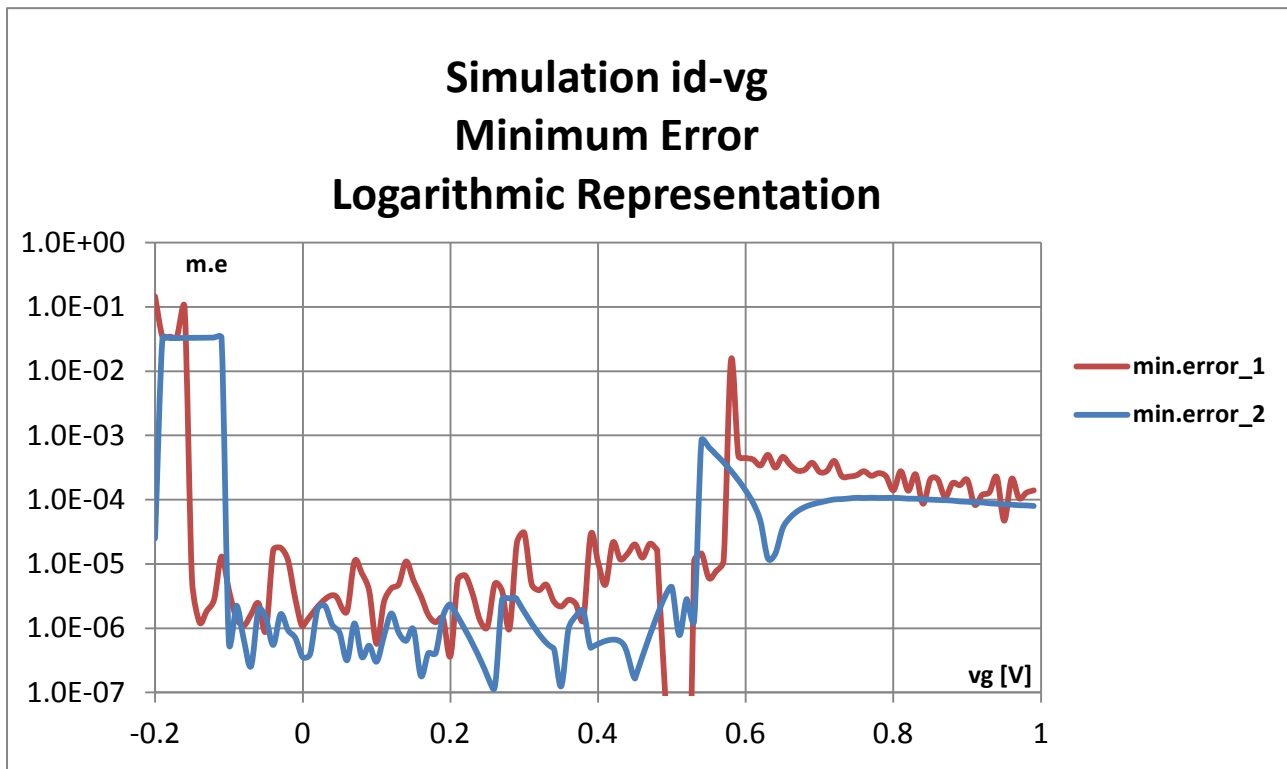


Εικόνα 5.4: Διάγραμμα logarithmic g_m - v_g / $L=0.1\mu$.

gm = Αναπαράσταση της διαγωγιμότητας για τις προσομοιώσεις:

ngspice_1, spectre_1: $\text{id-vg} \Rightarrow -200\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_d=1\text{v}|v_s=0\text{v}|v_b=0\text{v}$

ngspice_2, spectre_2: $\text{id-vg} \Rightarrow -200\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_d=0.05\text{v}|v_s=0\text{v}|v_b=0\text{v}$

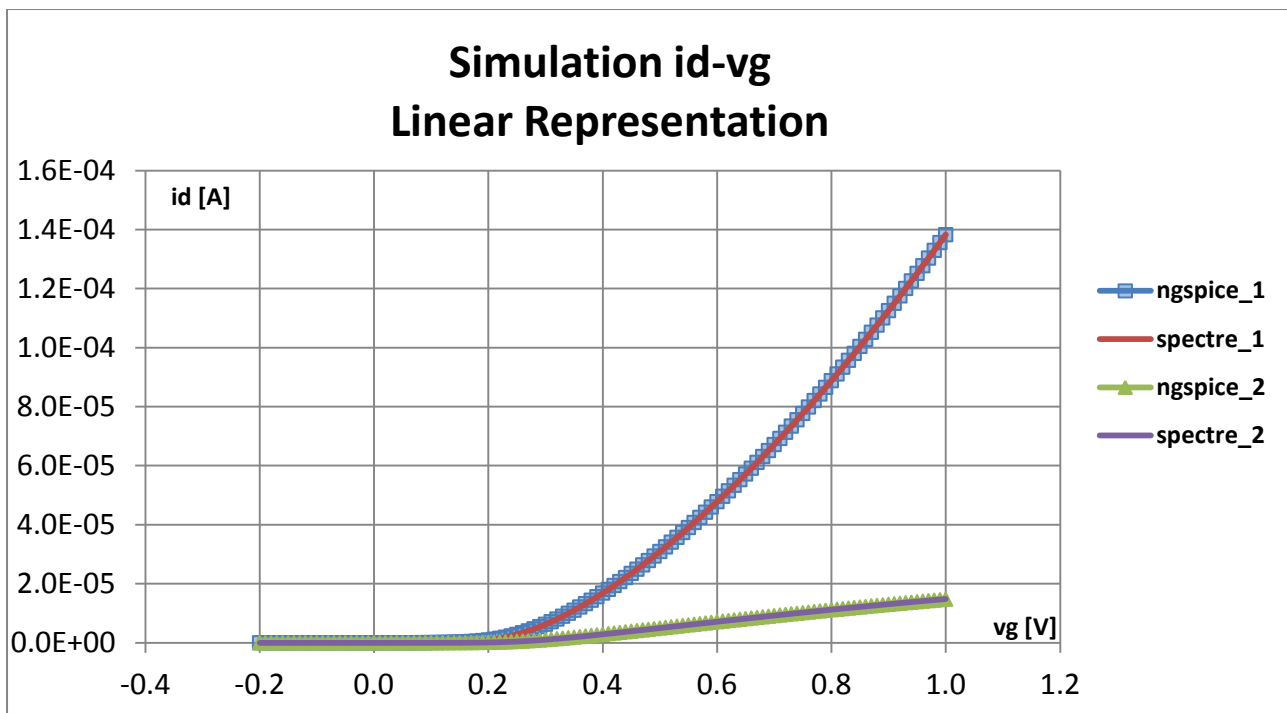


Εικόνα 5.5: Διάγραμμα logarithmic min.error-vg/ $L=0.1\mu$.

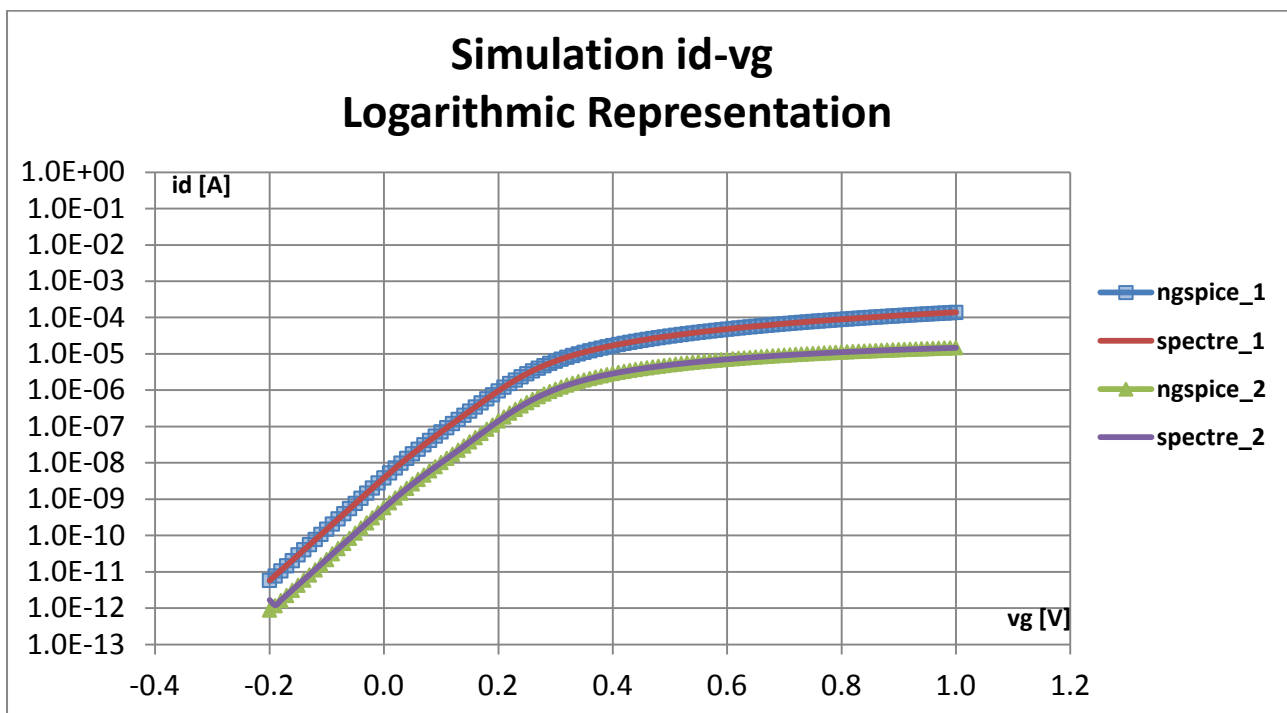
Min.error[1,2] = Απόκλιση μεταξύ προσομοιωτών ngspice, spectre για τις προσομοιώσεις:

ngspice_1, spectre_1: id-vg => -200mv => 1v/10mv|w=10μ|L=0.1μ|vd=1v|vs=0v|vb=0v

ngspice_2, spectre_2: id-vg => -200mv => 1v/10mv|w=10μ|L=0.1μ|vd=0.05v|vs=0v|vb=0v



Εικόνα 5.6: Διάγραμμα linear id - vg / $L=10\mu$.

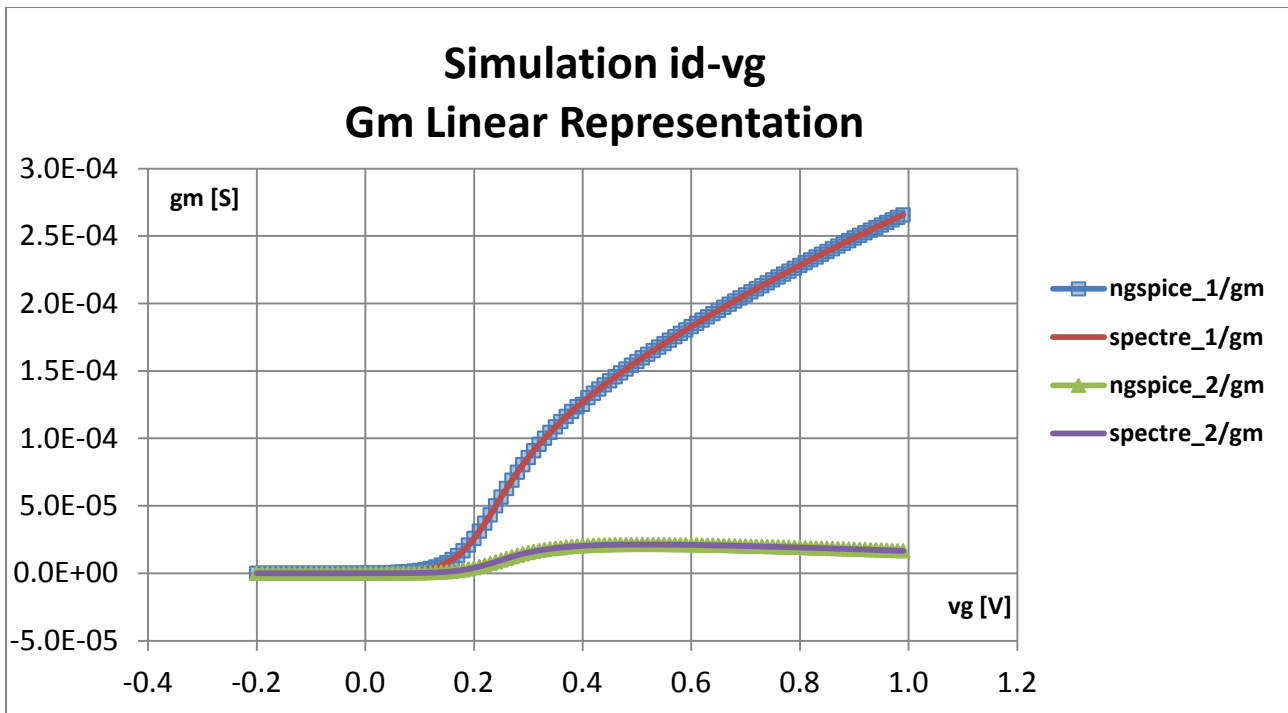


Εικόνα 5.7: Διάγραμμα logarithmic id - vg / $L=10\mu$.

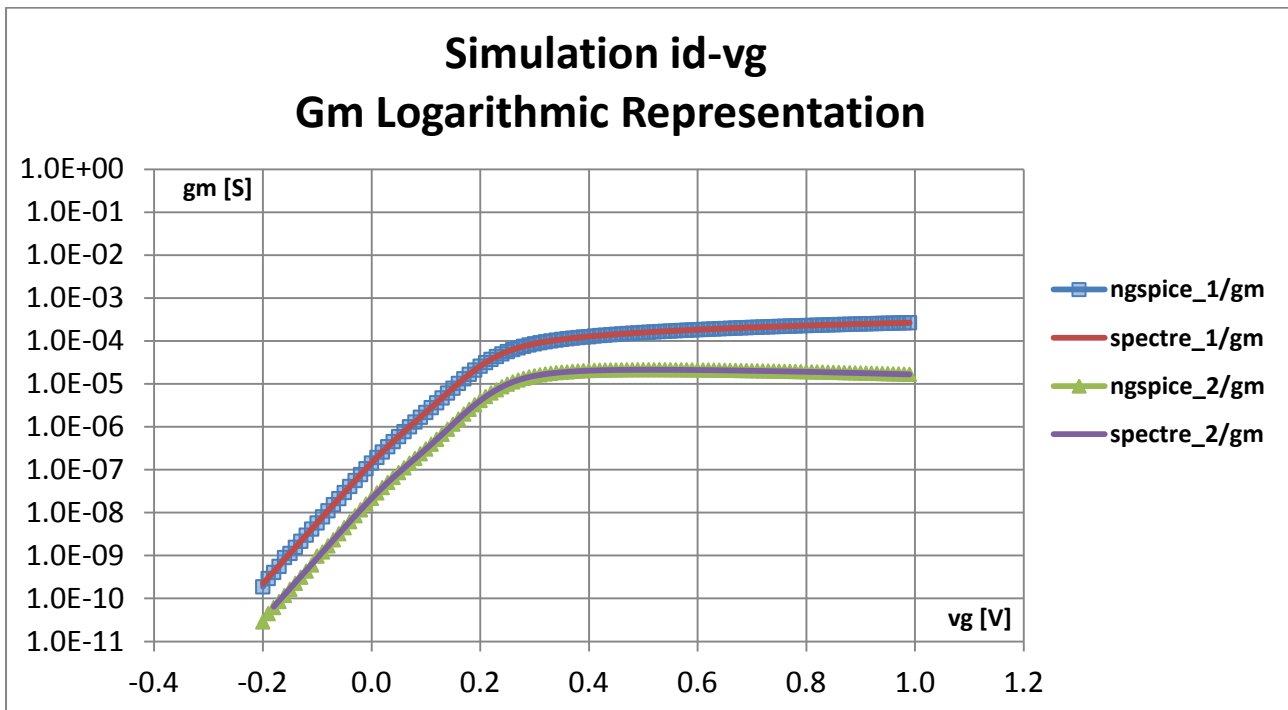
Προσομοιώσεις:

ngspice_1, spectre_1: id - $vg \Rightarrow -200mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|vd=1v|vs=0v|vb=0v$

ngspice_2, spectre_2: id - $vg \Rightarrow -200mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|vd=0.05v|vs=0v|vb=0v$



Εικόνα 5.8: Διάγραμμα linear gm - v_g / $L=10\mu$.

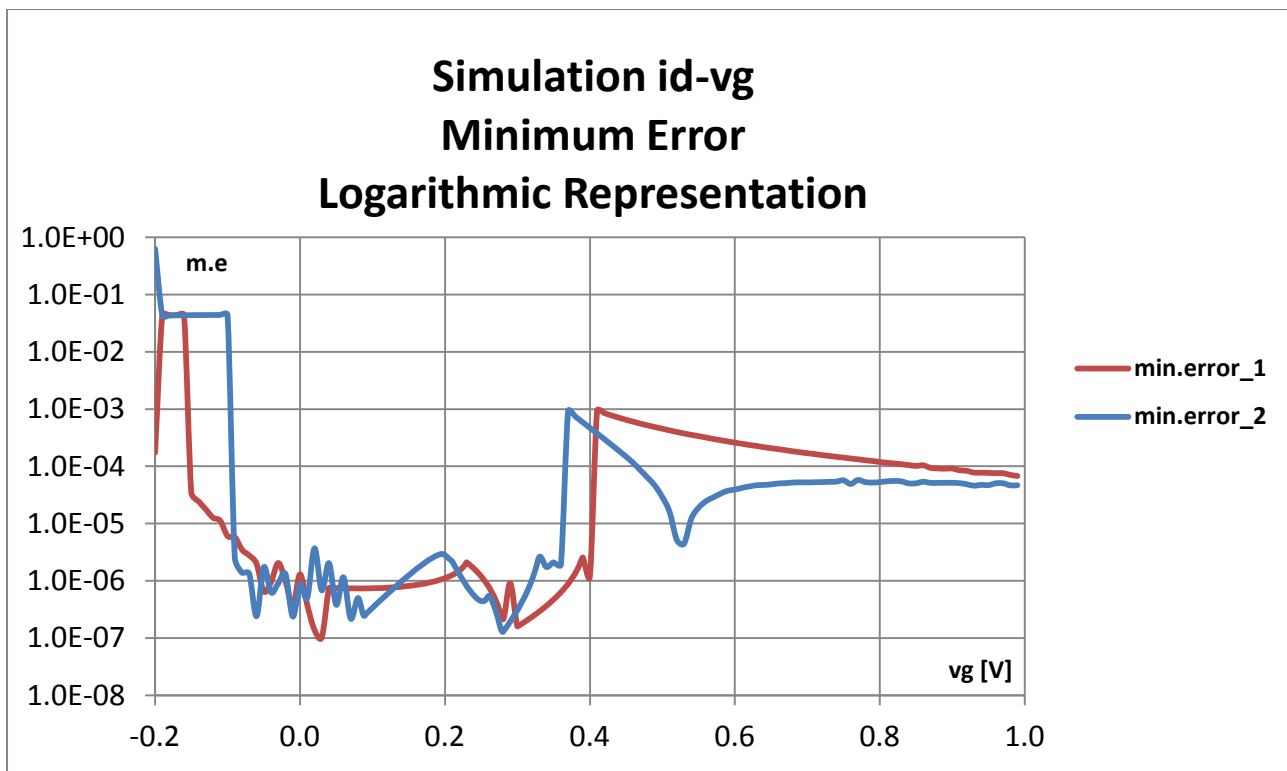


Εικόνα 5.9: Διάγραμμα logarithmic gm - v_g / $L=10\mu$.

gm = Αναπαράσταση της διαγωγιμότητας για τις προσομοιώσεις:

ngspice_1, spectre_1: $id-v_g \Rightarrow -200mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|v_d=1v|v_s=0v|v_b=0v$

ngspice_2, spectre_2: $id-v_g \Rightarrow -200mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|v_d=0.05v|v_s=0v|v_b=0v$



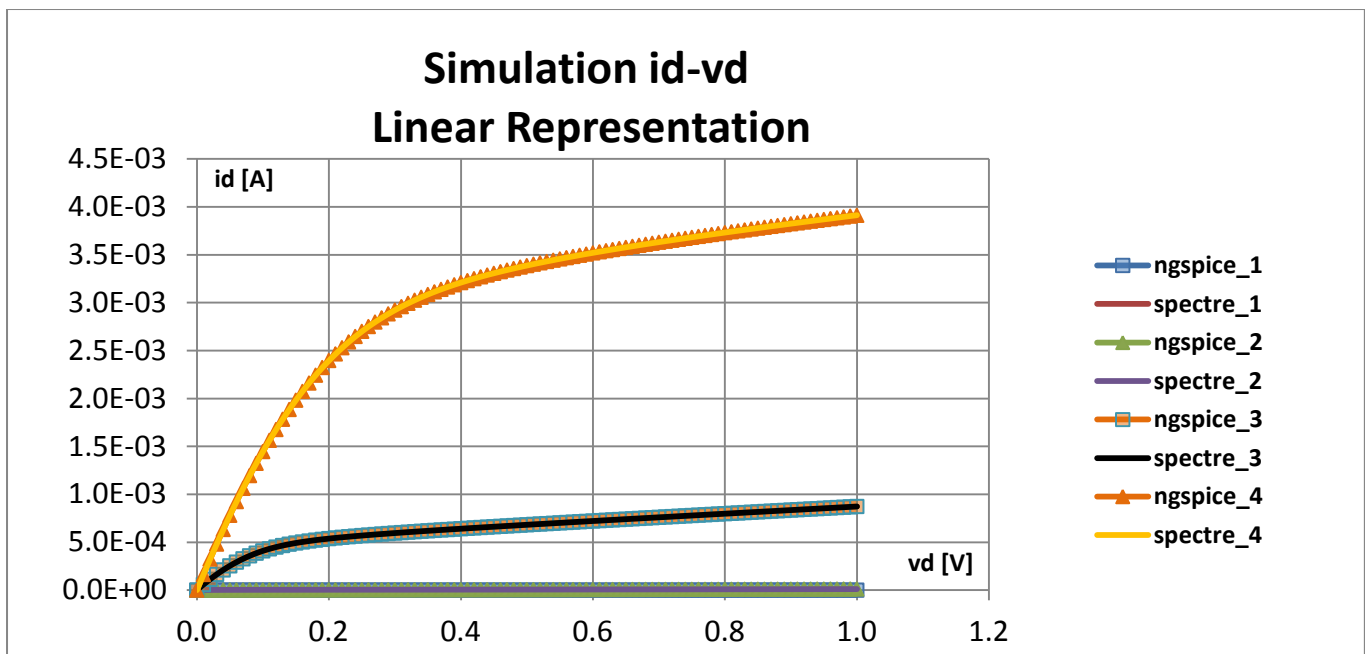
Εικόνα 5.10: Διάγραμμα logarithmic min.error-vg/ $L=10\mu$.

Min.error[1,2] = Απόκλιση μεταξύ προσομοιωτών ngspice, spectre για τις προσομοιώσεις:

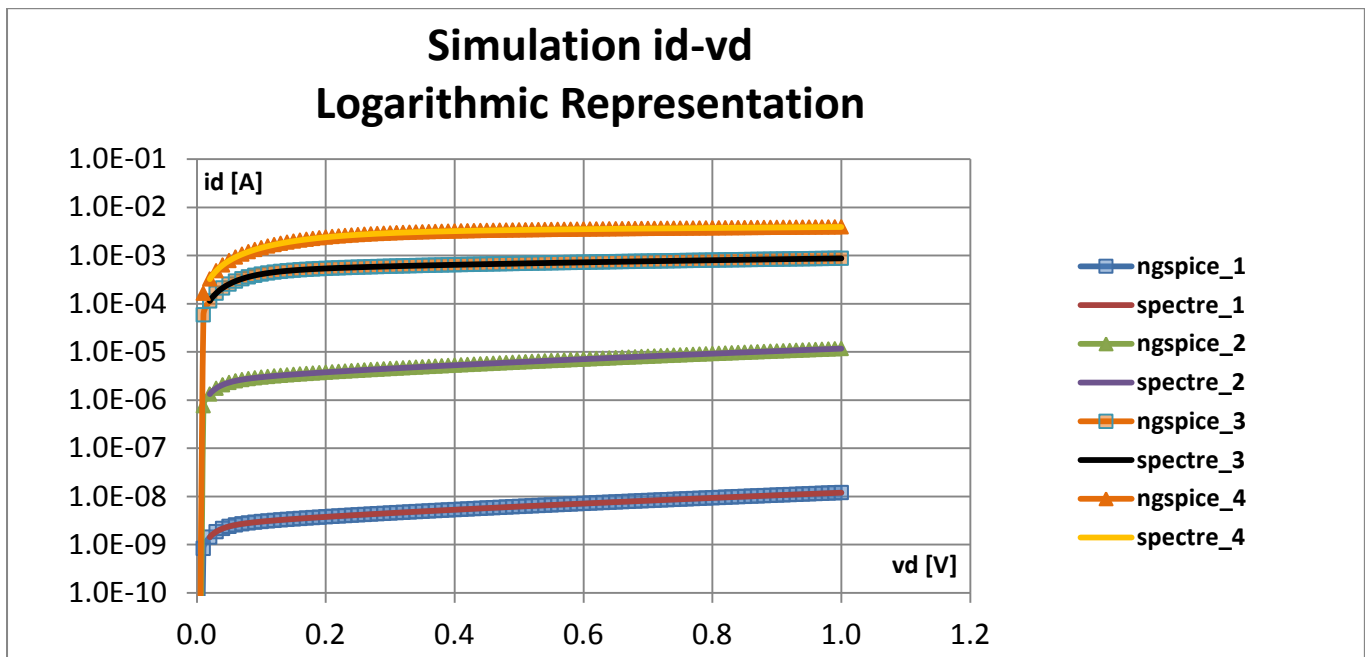
ngspice_1, spectre_1: id-vg => -200mv => 1v/10mv|w=10μ|L=10μ|vd=1v|vs=0v|vb=0v

ngspice_2, spectre_2: id-vg => -200mv => 1v/10mv|w=10μ|L=10μ|vd=0.05v|vs=0v|vb=0v

Παρατηρούμε ότι το σχετικό σφάλμα (Minimum Error) για τις id-vg προσομοιώσεις, αυξάνεται στις περιοχές χαμηλής και ισχυρής αναστροφής (Weak/Strong Inversion) που τοποθετούνται πριν και μετά την τάση κατωφλίου = 0.257V (παράμετρος +VTO=0.257), ενώ παραμένει χαμηλό στην περιοχή της μέτριας αναστροφής (medium inversion), όπως φανερώνουν οι εικόνες 5.5 και 5.10.



Εικόνα 5.11: Διάγραμμα linear id-vd/ $L=0.1\mu$.



Εικόνα 5.12: Διάγραμμα logarithmic id-vd/ $L=0.1\mu$.

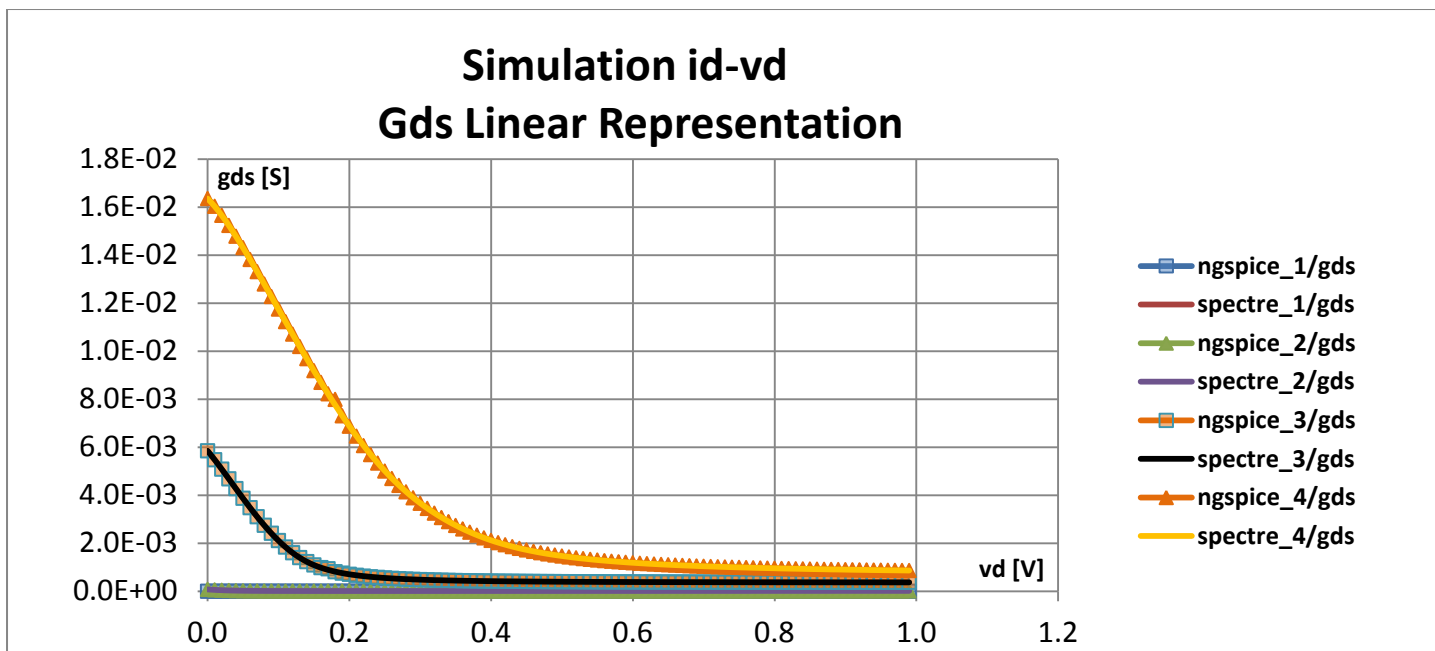
Προσομοιώσεις:

ngspice_1, spectre_1: $\text{id-vd} \Rightarrow 0\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_g=0.05\text{v}|v_s=0\text{v}|v_b=0\text{v}$

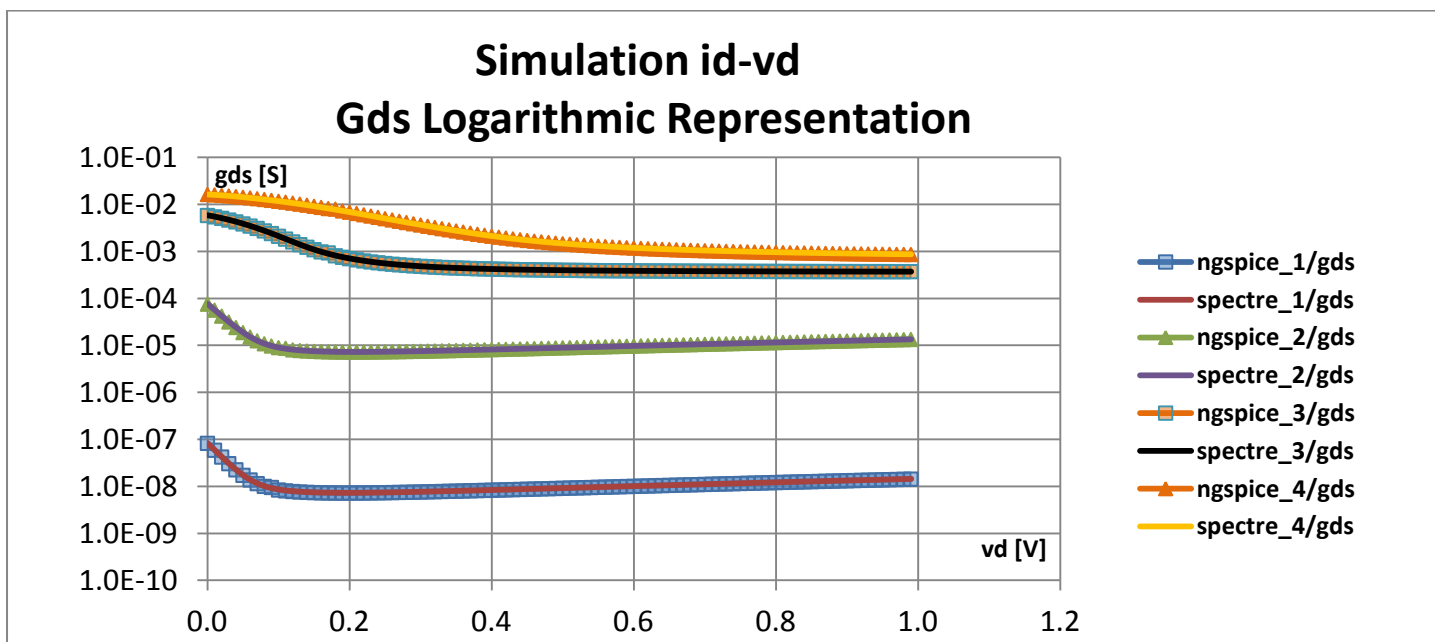
ngspice_2, spectre_2: $\text{id-vd} \Rightarrow 0\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_g=0.3\text{v}|v_s=0\text{v}|v_b=0\text{v}$

ngspice_3, spectre_3: $\text{id-vd} \Rightarrow 0\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_g=0.6\text{v}|v_s=0\text{v}|v_b=0\text{v}$

ngspice_4, spectre_4: $\text{id-vd} \Rightarrow 0\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_g=1\text{v}|v_s=0\text{v}|v_b=0\text{v}$



Εικόνα 5.13: Διάγραμμα linear g_{ds} - v_d / $L=0.1\mu$.



Εικόνα 5.14: Διάγραμμα logarithmic g_{ds} - v_d / $L=0.1\mu$.

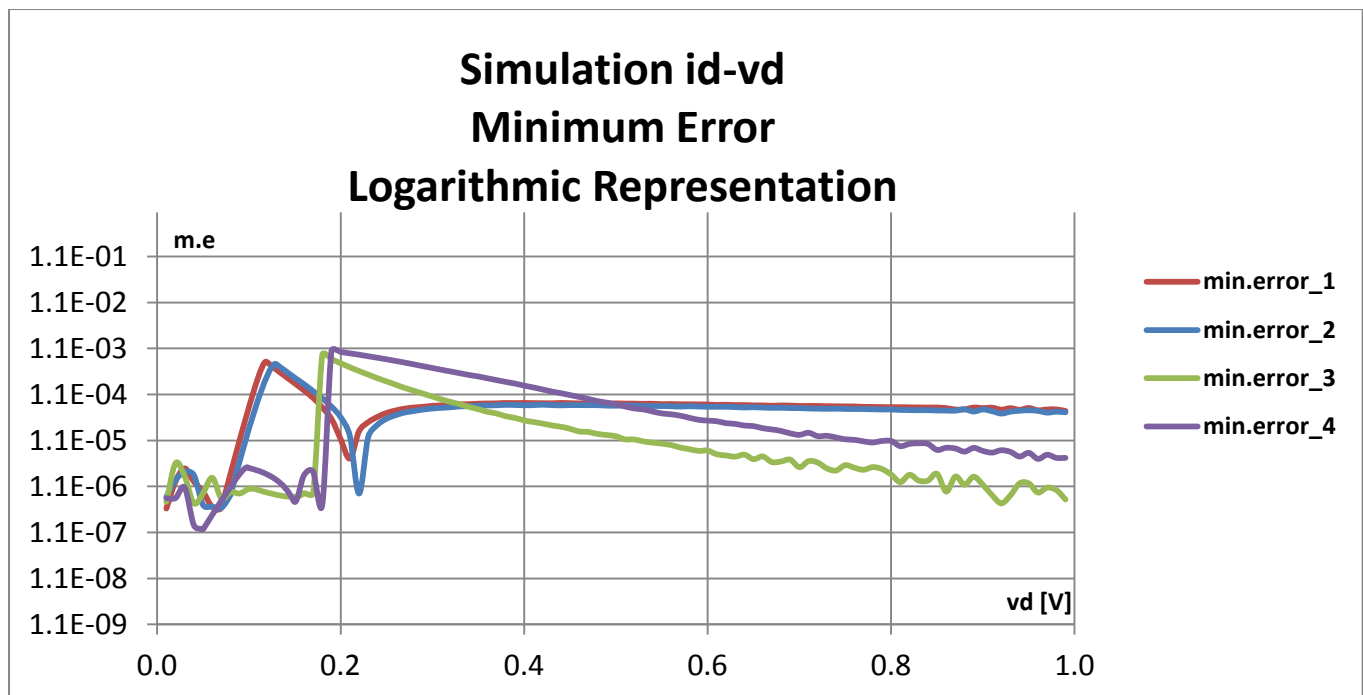
gds = Αναπαράσταση της διαγωγιμότητας για τις προσομοιώσεις:

ngspice_1, spectre_1: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=0.05v|vs=0v|vb=0v

ngspice_2, spectre_2: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=0.3v|vs=0v|vb=0v

ngspice_3, spectre_3: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=0.6v|vs=0v|vb=0v

ngspice_4, spectre_4: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=1v|vs=0v|vb=0v



Εικόνα 5.15: Διάγραμμα logarithmic min.error-vd/ $L=0.1\mu$.

Min.error[1,2,3,4]=

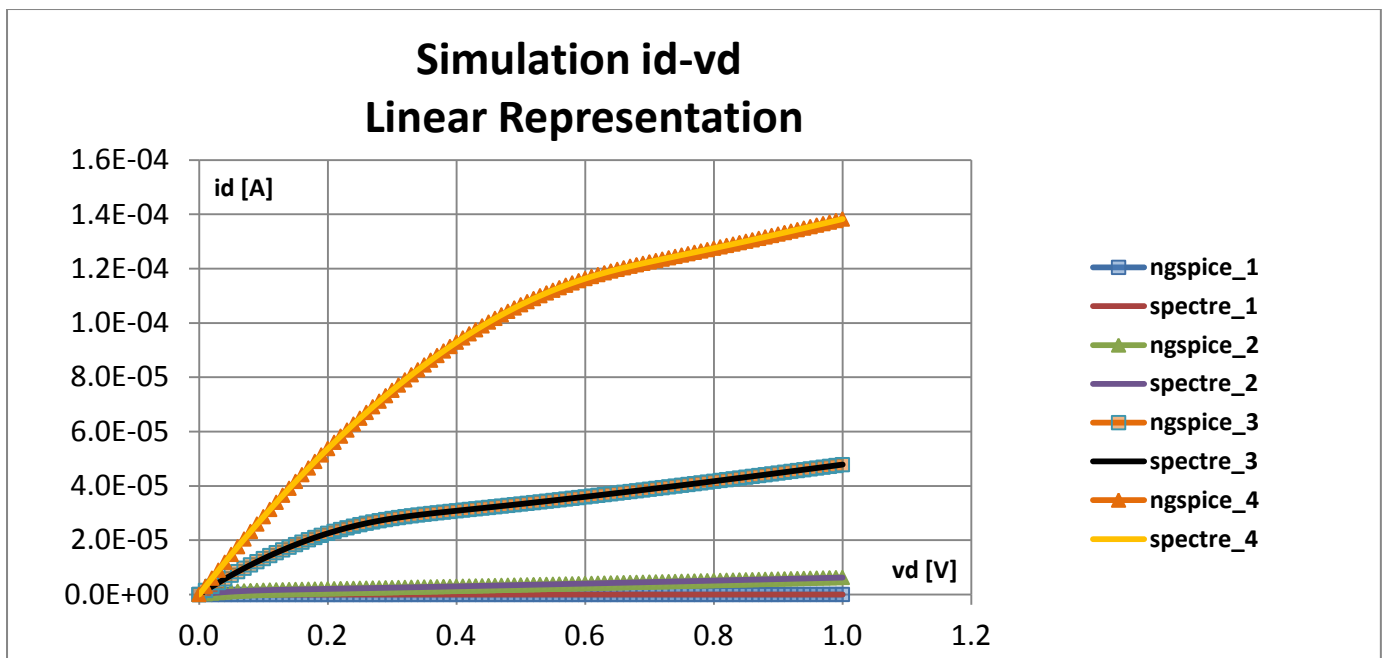
Απόκλιση μεταξύ προσομοιωτών ngspice, spectre για τις προσομοιώσεις:

ngspice_1, spectre_1: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=0.05v|vs=0v|vb=0v

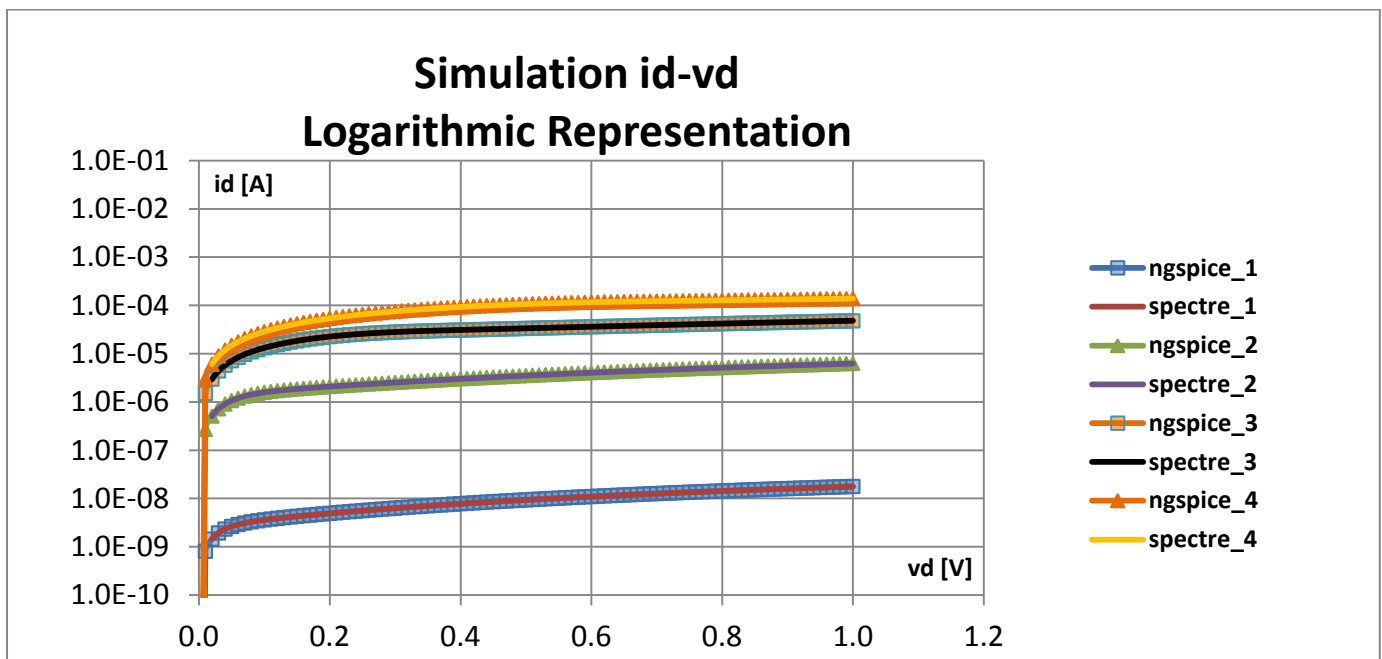
ngspice_2, spectre_2: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=0.3v|vs=0v|vb=0v

ngspice_3, spectre_3: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=0.6v|vs=0v|vb=0v

ngspice_4, spectre_4: id-vd=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=1v|vs=0v|vb=0v



Εικόνα 5.16: Διάγραμμα linear id-vd/ $L=10\mu$.



Εικόνα 5.17: Διάγραμμα logatirhmic id-vd/ $L=10\mu$.

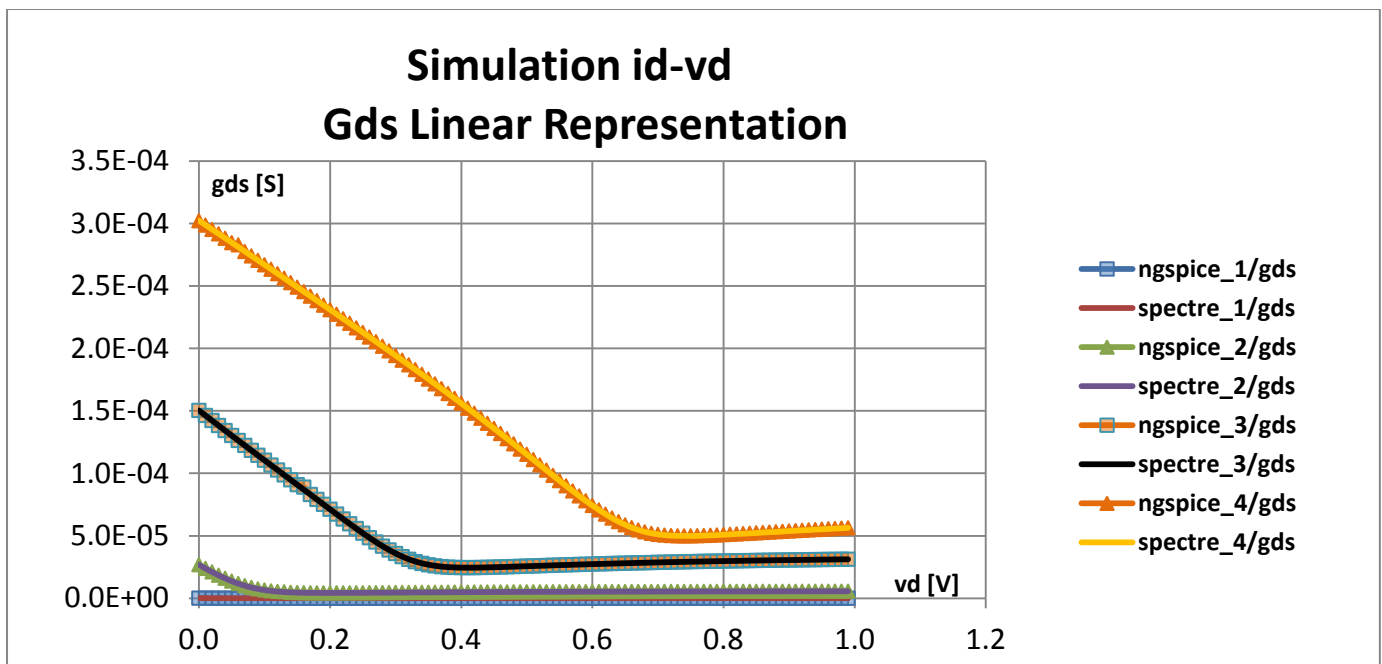
Προσομοιώσεις:

ngspice_1, spectre_1: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=0.05v|vs=0v|vb=0v

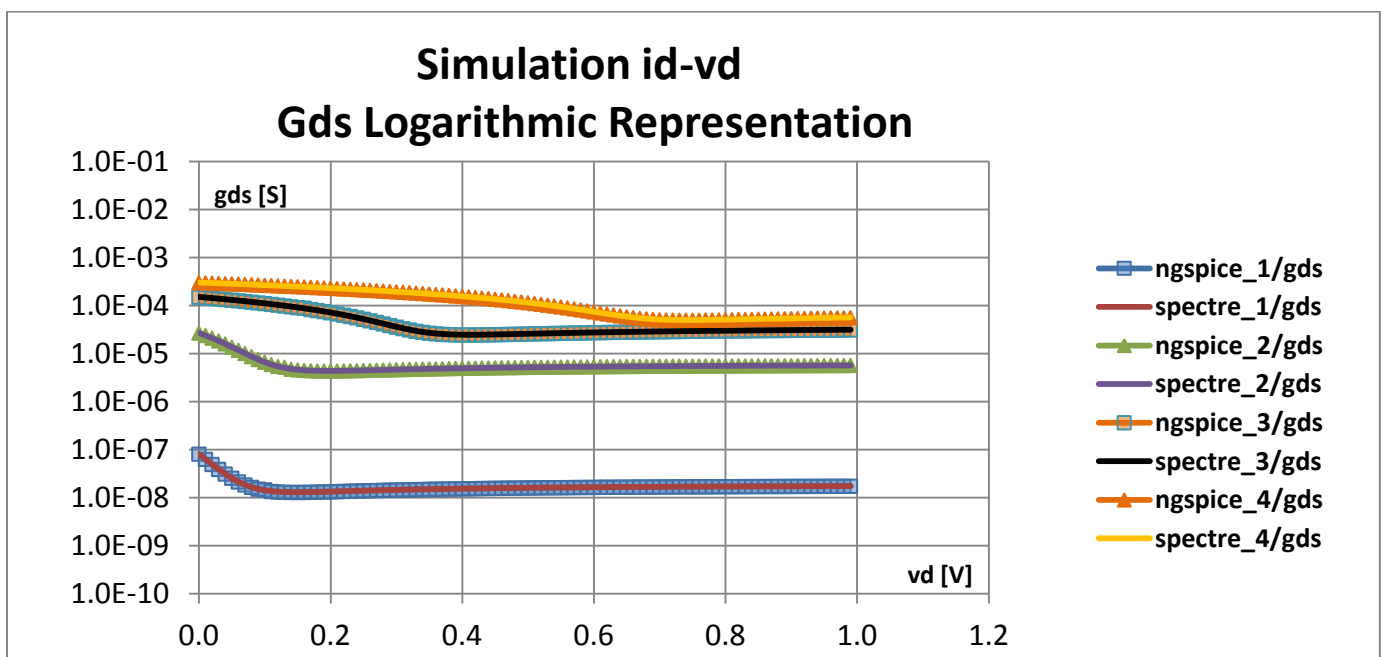
ngspice_2, spectre_2: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=0.3v|vs=0v|vb=0v

ngspice_3, spectre_3: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=0.6v|vs=0v|vb=0v

ngspice_4, spectre_4: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=1v|vs=0v|vb=0v



Εικόνα 5.18: Διάγραμμα linear g_{ds} - v_d / $L=10\mu$.



Εικόνα 5.19: Διάγραμμα logarithmic g_{ds} - v_d / $L=10\mu$.

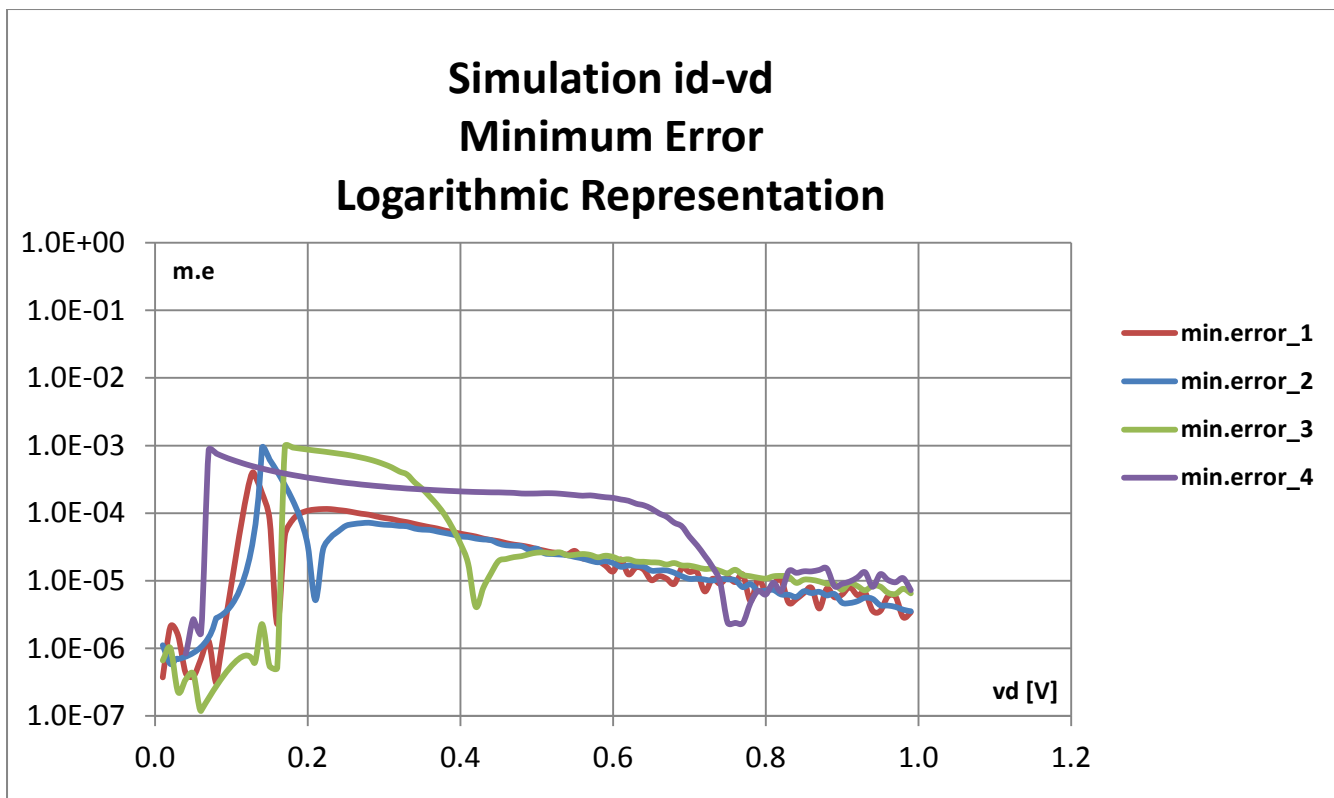
g_{ds} = Αναπαράσταση της διαγωγιμότητας για τις προσομοιώσεις:

ngspice_1, spectre_1: $id-v_d \Rightarrow 0mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|v_g=0.05v|v_s=0v|v_b=0v$

ngspice_2, spectre_2: $id-v_d \Rightarrow 0mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|v_g=0.3v|v_s=0v|v_b=0v$

ngspice_3, spectre_3: $id-v_d \Rightarrow 0mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|v_g=0.6v|v_s=0v|v_b=0v$

ngspice_4, spectre_4: $id-v_d \Rightarrow 0mv \Rightarrow 1v/10mv|w=10\mu|L=10\mu|v_g=1v|v_s=0v|v_b=0v$



Εικόνα 5.20: Διάγραμμα logarithmic min.error-vd/ $L=10\mu$.

Min.error[1,2,3,4]=

Απόκλιση μεταξύ προσομοιωτών ngspice, spectre για τις προσομοιώσεις:

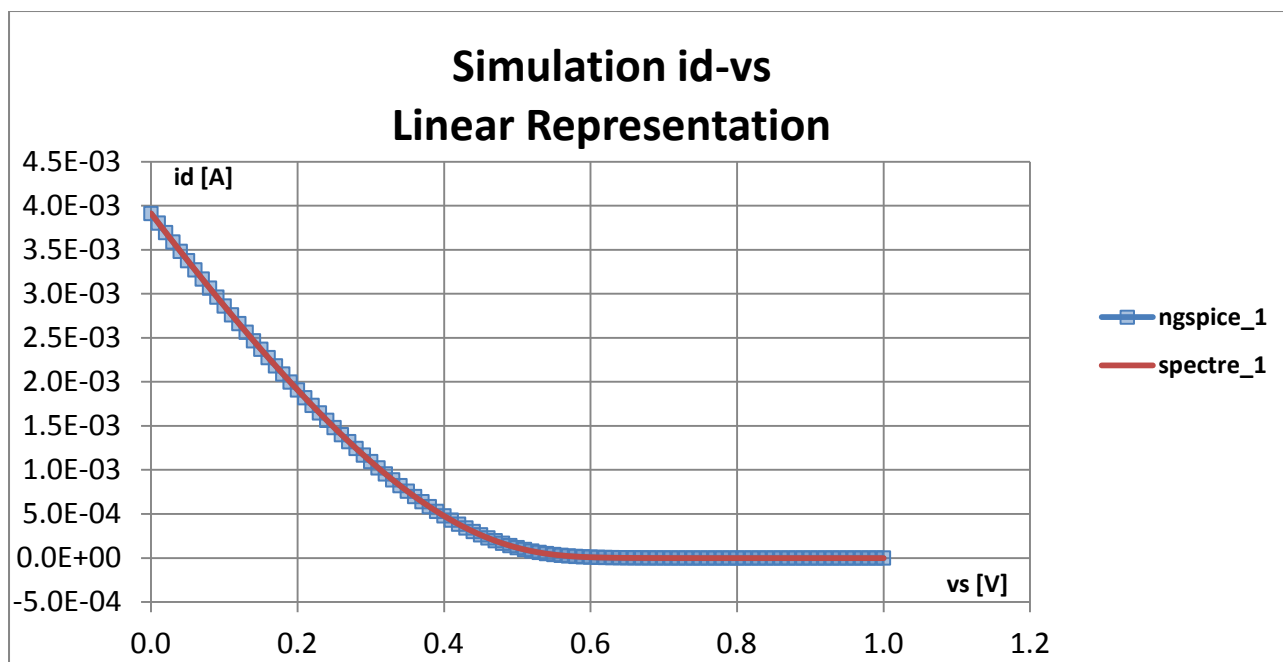
ngspice_1, spectre_1: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=0.05v|vs=0v|vb=0v

ngspice_2, spectre_2: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=0.3v|vs=0v|vb=0v

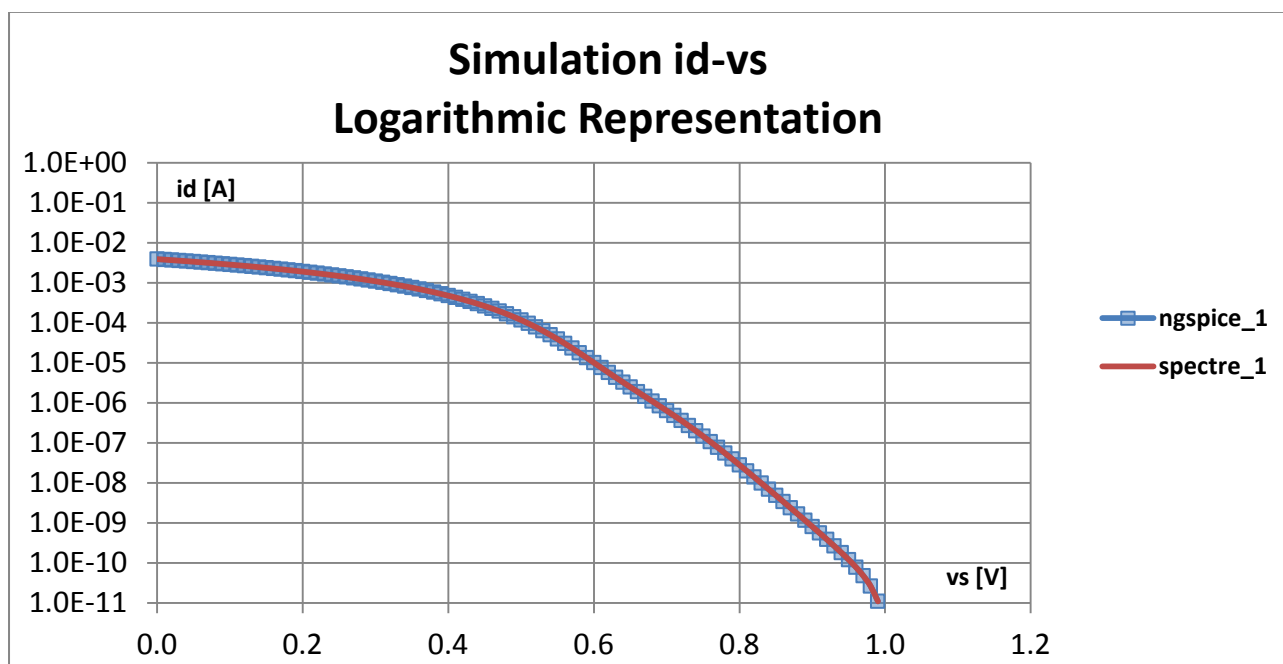
ngspice_3, spectre_3: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=0.6v|vs=0v|vb=0v

ngspice_4, spectre_4: id-vd=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=1v|vs=0v|vb=0v

Παρατηρούμε ότι το σχετικό σφάλμα (Minimum Error) στις id-vd προσομοιώσεις μειώνεται όσο απομακρυνόμαστε από την γραμμική περιοχή και μεταφερόμαστε προς την περιοχή κορεσμού, όπως φανερώνουν τα διαγράμματα 5.15 και 5.20.



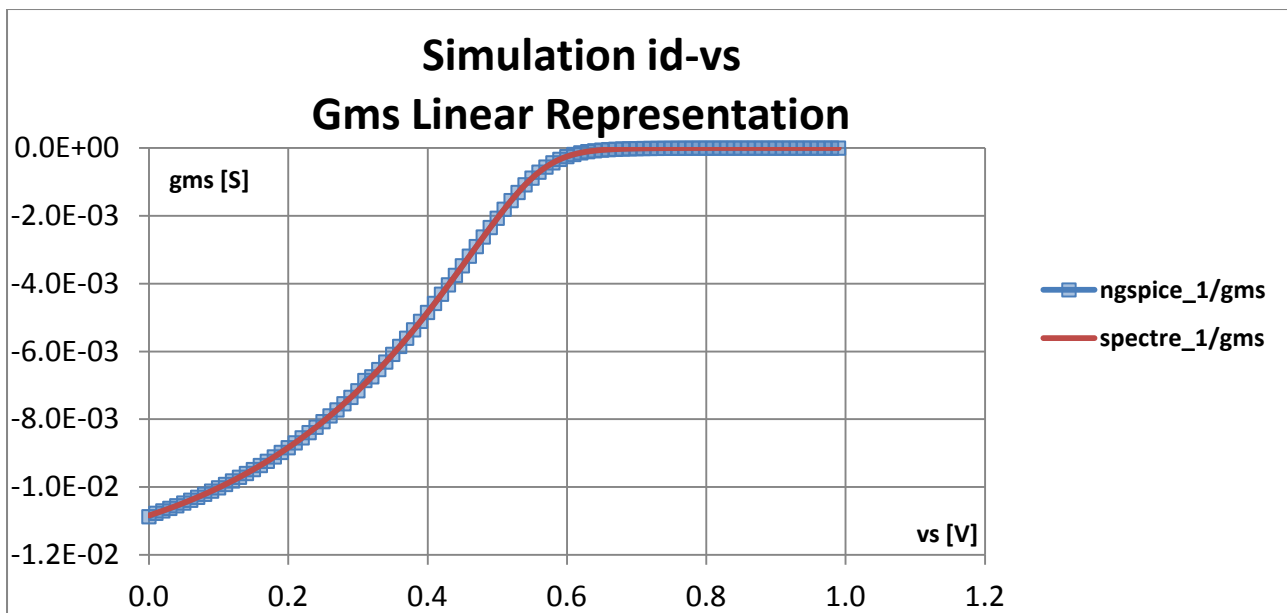
Εικόνα 5.21: Διάγραμμα linear i_d - v_s / $L=0.1\mu$.



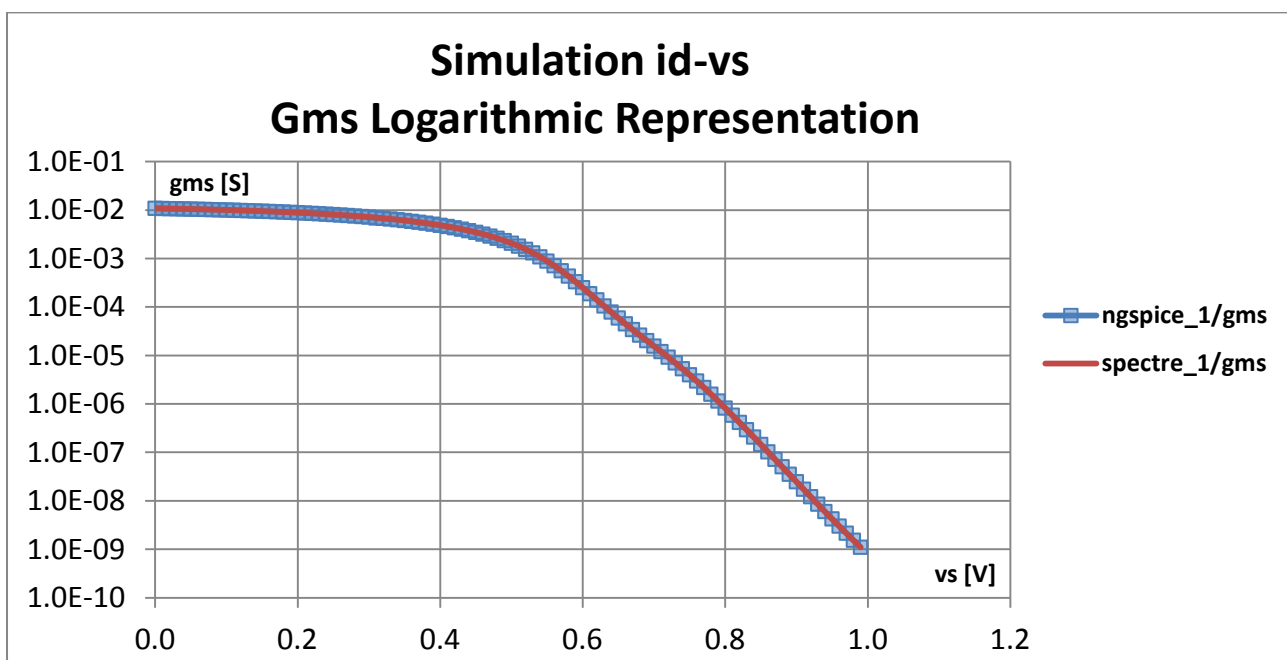
Εικόνα 5.22: Διάγραμμα logarithmic i_d - v_s / $L=0.1\mu$.

Προσομοίωση:

ngspice_1, spectre_1: i_d - $v_s \Rightarrow 0\text{mv} \Rightarrow 1\text{v}/10\text{mv}|w=10\mu|L=0.1\mu|v_g=1\text{v}|v_s=1\text{v}|v_b=0\text{v}$



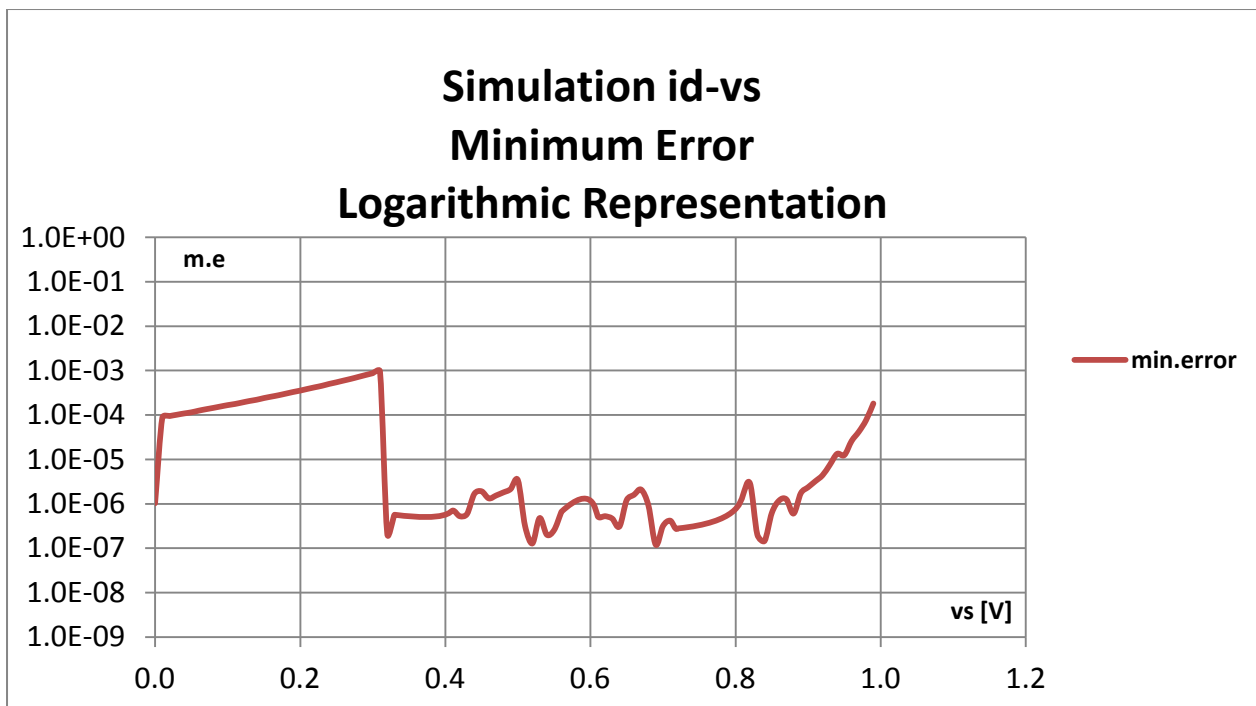
Εικόνα 5.23: Διάγραμμα linear gms-vs/ $L=0.1\mu$.



Εικόνα 5.24: Διάγραμμα logarithmic gms-vs/ $L=0.1\mu$.

gms = Αναπαράσταση της διαγωγιμότητας για την προσομοίωση:

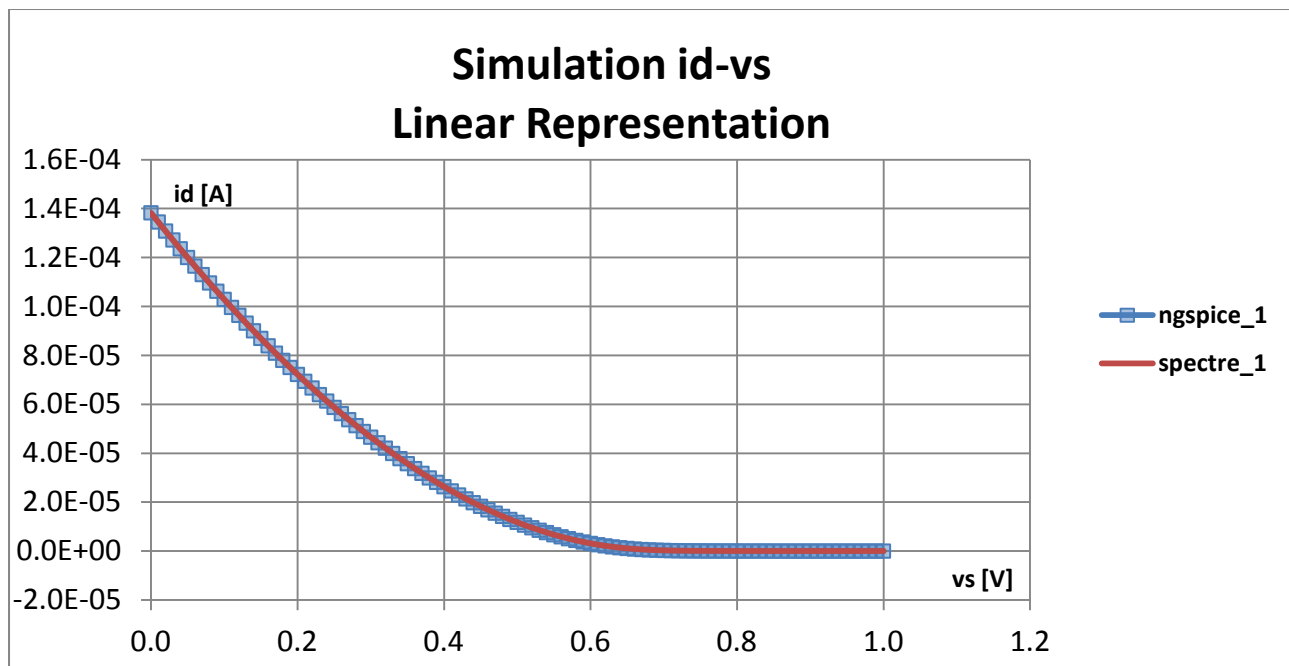
ngspice_1, spectre_1: id-vs=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=1v|vs=1v|vb=0v



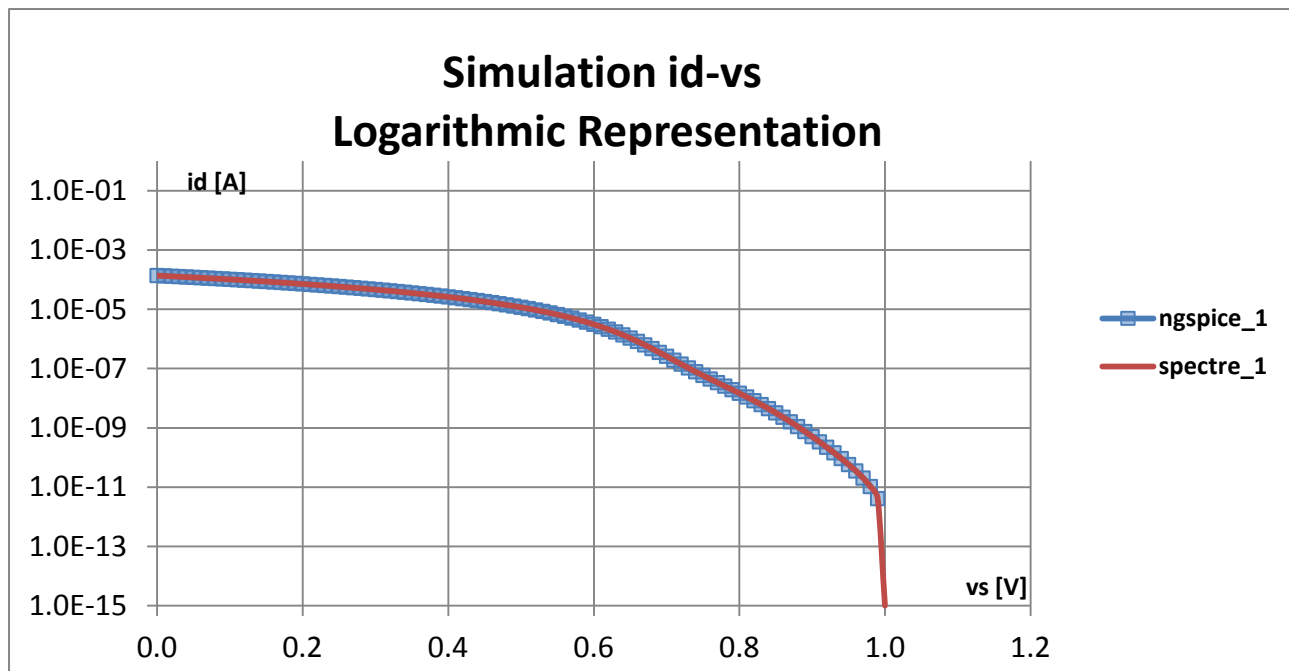
Εικόνα 5.25: Διάγραμμα *logarithmic min.error-vs/* $L=0.1\mu$.

Min.error[1] = Απόκλιση μεταξύ προσομοιωτών ngspice, spectre για την προσομοίωση:

ngspice_1, spectre_1: id-vs=> 0mv =>1v/10mv|w=10μ|L=0.1μ|vg=1v|vs=1v|vb=0v



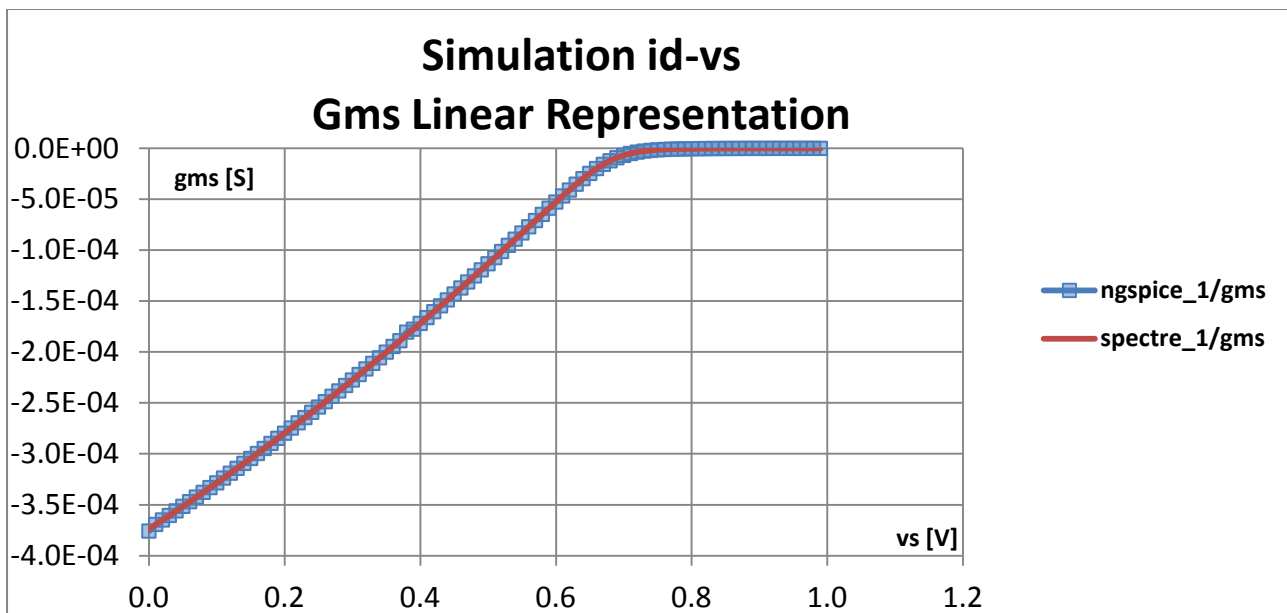
Εικόνα 5.26: Διάγραμμα *linear id-vs/ $L=10\mu$* .



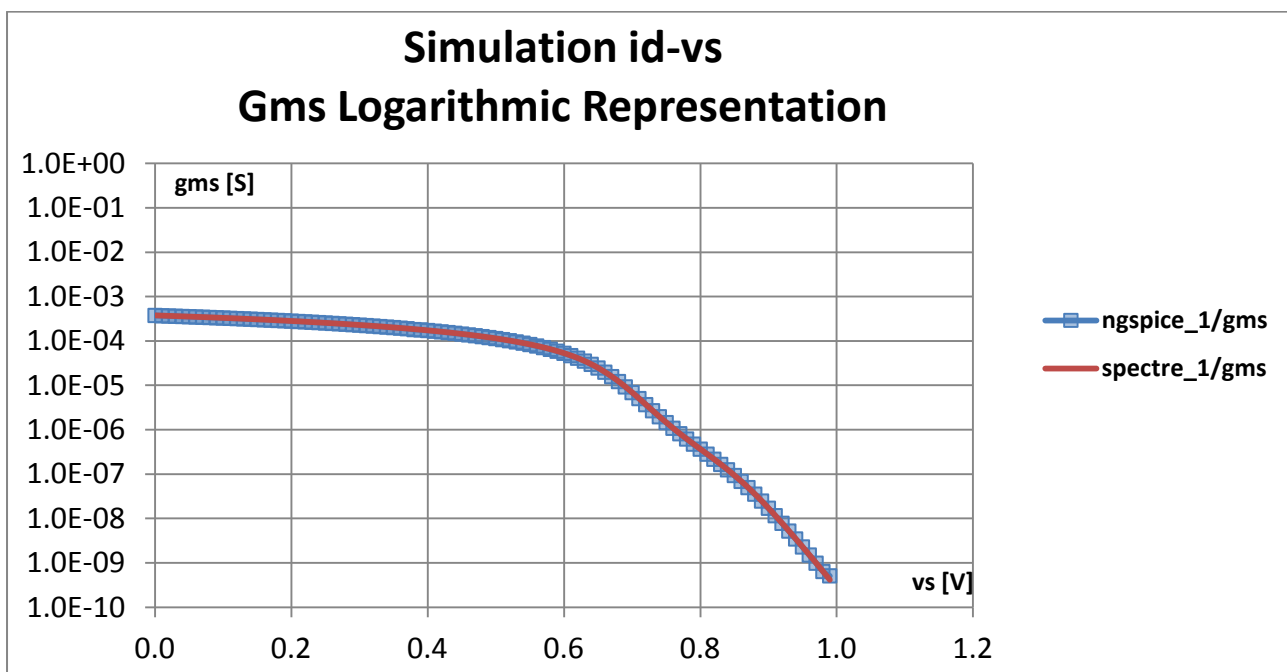
Εικόνα 5.27: Διάγραμμα *logarithmic id-vs/ $L=10\mu$* .

Προσομοίωση:

ngspice_1, spectre_1: id-vs=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=1v|vs=1v|vb=0v



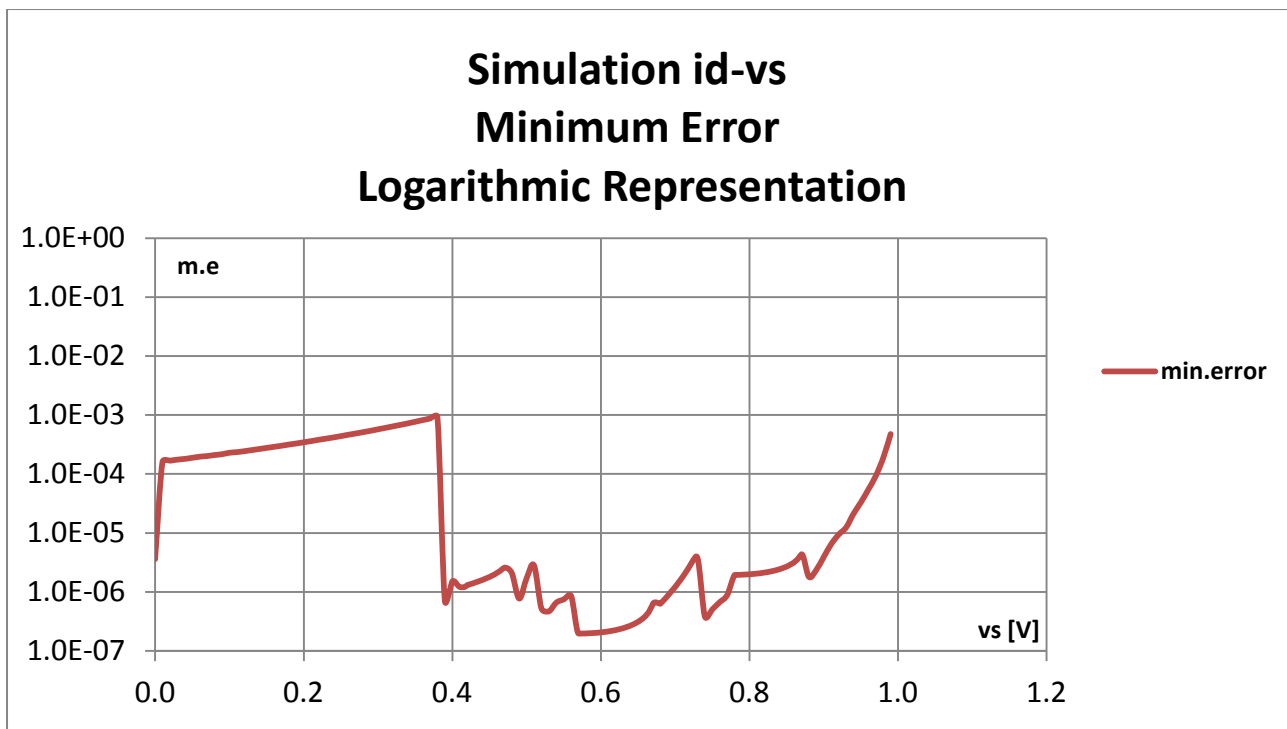
Εικόνα 5.28: Διάγραμμα linear g_{ms} -vs/ $L=10\mu$.



Εικόνα 5.29: Διάγραμμα logarithmic g_{ms} -vs/ $L=10\mu$.

g_{ms} = Αναπαράσταση της διαγωγιμότητας για την προσομοίωση:

ngspice_1, spectre_1: id-vs=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=1v|vs=1v|vb=0v



Εικόνα 5.30: Διάγραμμα *logarithmic min.error-vs/ L=10μ*.

Min.error[1] = Απόκλιση μεταξύ προσομοιωτών ngspice, spectre για την προσομοίωση:

ngspice_1, spectre_1: id-vs=> 0mv =>1v/10mv|w=10μ|L=10μ|vg=1v|vs=1v|vb=0v

Συμπεραίνουμε από την παρατήρηση των αποτελεσμάτων την ορθή λειτουργία του ενσωματωμένου στον προσομοιωτή Ngspice, μοντέλου MOSFET EKV3, καθώς το σφάλμα μεταξύ των δύο προσομοιωτών Ngspice και Spectre είναι πολύ μικρής τάξης. Αποδίδουμε το υπάρχον σφάλμα σε ιδιαιτερότητες και λεπτομέρειες της λειτουργικότητας των δύο προσομοιωτών.

5.3 Προσομοιώσεις τύπου AC

Η προσομοίωση των διαχωρητικοτήτων ενός τρανζίστορ είναι αναγκαία για την διαπίστωση της καλής AC λειτουργίας του μοντέλου στον συγκεκριμένο προσομοιωτή. Πραγματοποιήθηκαν 3 CV προσομοιώσεις για την εξαγωγή των **CGB**, **CGC** και **CGG** διαχωρητικοτήτων. Η CGB είναι η διαχωρητικότητα υποστρώματος, η CGC είναι η διαχωρητικότητα καναλιού και η CGG είναι η συνολική διαχωρητικότητα που παρουσιάζει το τρανζίστορ. Το άθροισμα των δύο πρώτων είναι περίπου ίσο με το CGG.

Ο υπολογισμός των διαχωρητικοτήτων βασίζεται στο τύπο υπολογισμού της διαγωγιμότητας όπου:

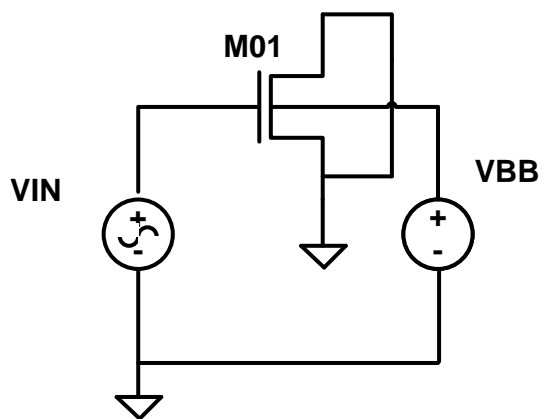
$$Y = \frac{I}{V} = \text{real}(Y) + j * \text{imag}(Y)$$

Από την στιγμή όπου το V=1 τότε προκύπτει ότι:

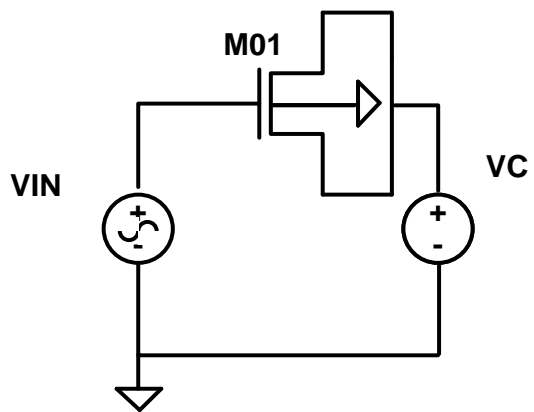
$$Y = \frac{I}{1} = I = \text{real}(I) + j * \text{imag}(I)$$

Όποτε και η διαχωρητικότητα υπολογίζεται απευθείας από το ρεύμα εξόδου χρησιμοποιώντας την εξής σχέση: $C = \text{imag}(I) / (2\pi f)$ όπου το f είναι η συχνότητα προσομοίωσης.

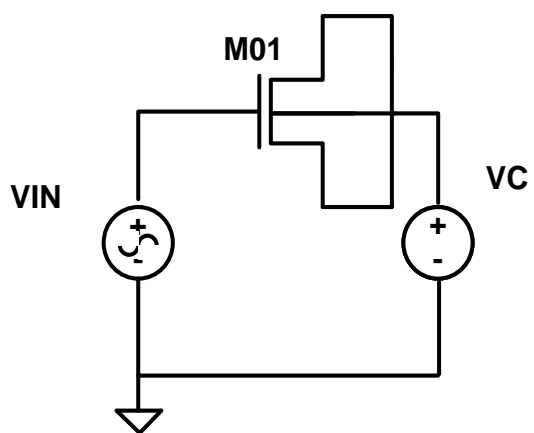
Ανάλογα με ποια διαχωρητικότητα θέλουμε να εξαγάγουμε η συνδεσμολογία διαφέρει όπως δείχνουν τα παρακάτω σχήματα. Οι πηγές VBB και VC έχουν μηδενική τάση και τις χρησιμοποιούμε για την εξαγωγή του AC ρεύματος. Για την πραγματοποίηση των προσομοιώσεων εφαρμόστηκε μία DC / AC πηγή (VIN) στη πύλη. Το DC σήμα κυμαίνεται από -1.8 έως 1.8 V και το AC είχε πλάτος 1 Volt peak και συχνότητα 1MHz.



Εικόνα 5.31: *CGB simulation-circuit.*



Εικόνα 5.32: *CGC simulation-circuit.*



Εικόνα 5.33: *CGG simulation-circuit.*

Ενδεικτική παράθεση input-netlist προσομοίωσης CGB:

```
CGB EXTRACTION
.include ekv3_018.par
.param W=660e-6
.param L=76e-6

*****STIMULUS CIRCUIT*****
VIN G 0 1.8 AC 1
M01 0 G 0 B NCH W=W L=L M=1
VBB B 0 0

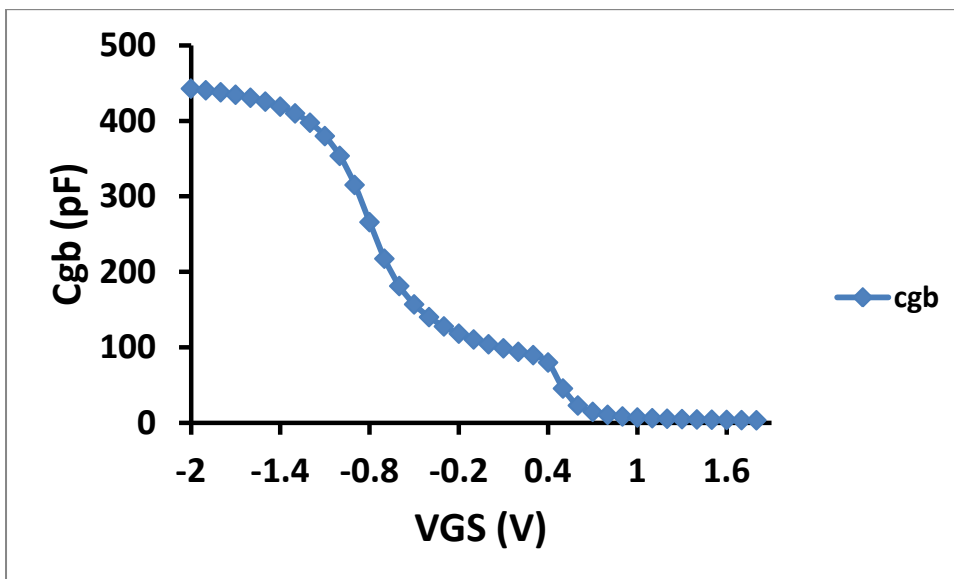
.control

set filetype=ascii
set appendwrite
foreach VGG
-2      -1.9    -1.8    -1.7    -1.6    -1.5    -1.4    -1.3    -1.2
-1.1    -1      -0.9    -0.8    -0.7    -0.6    -0.5    -0.4    -0.3
-0.2    -0.1    0.1     0.2     0.3     0.4     0.5     0.6     0.7
0.8     0.9     1       1.1     1.2     1.3     1.4     1.5     1.6
1.7     1.8
alter VIN=$VGG
ac lin 1MEG 1MEG 1MEG
wrdata cgb imag(i(VBB))/(2*3.14*1e6)

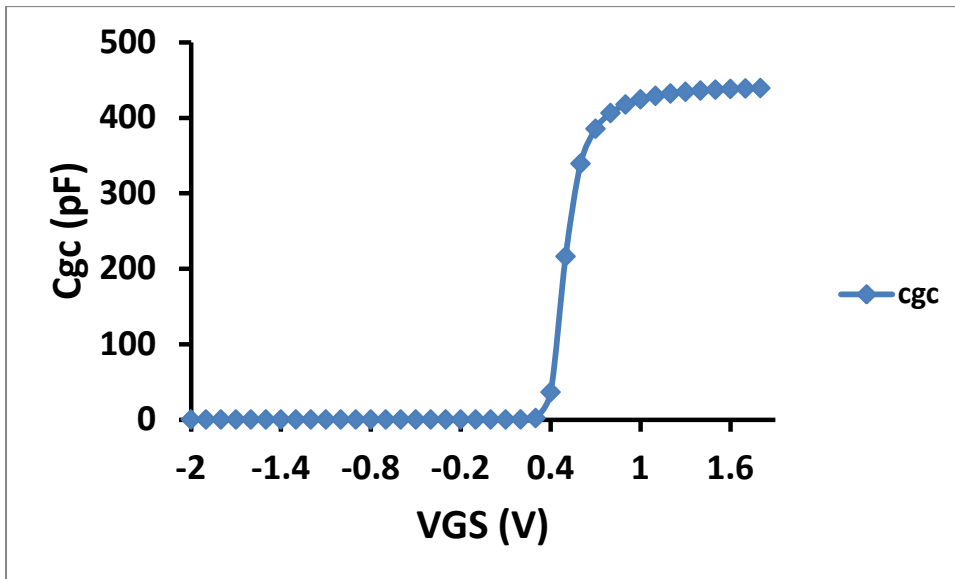
end

.endc

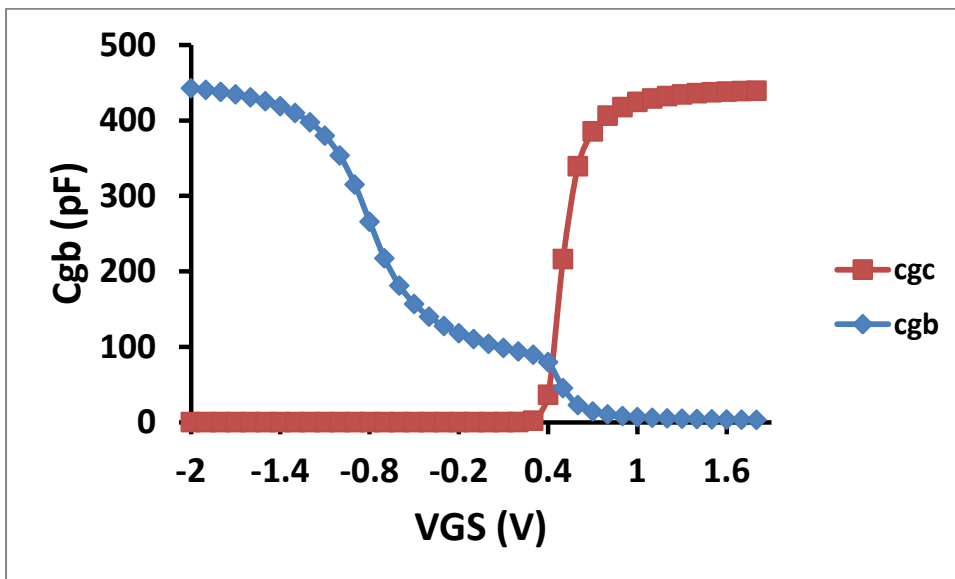
.end
```



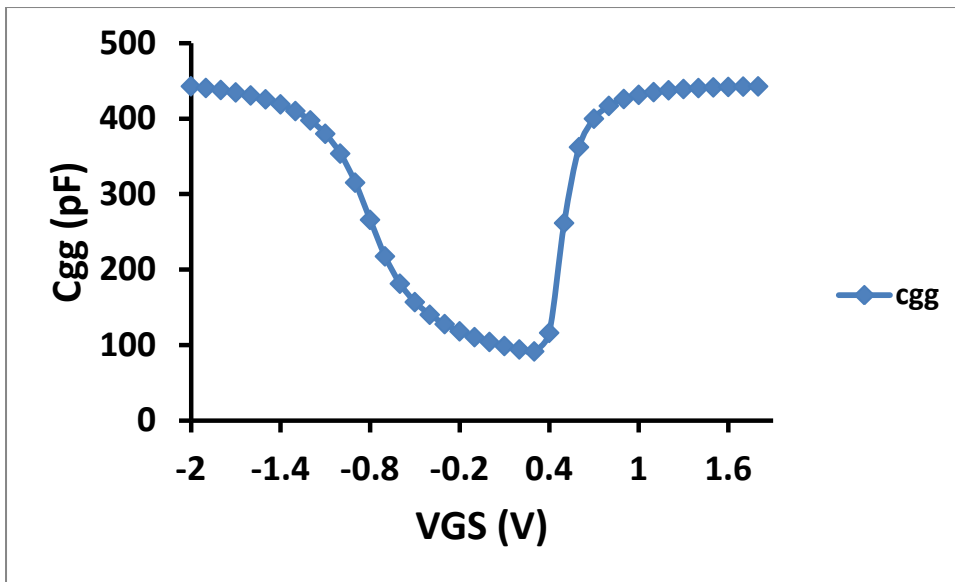
Εικόνα 5.34: CGB simulation.



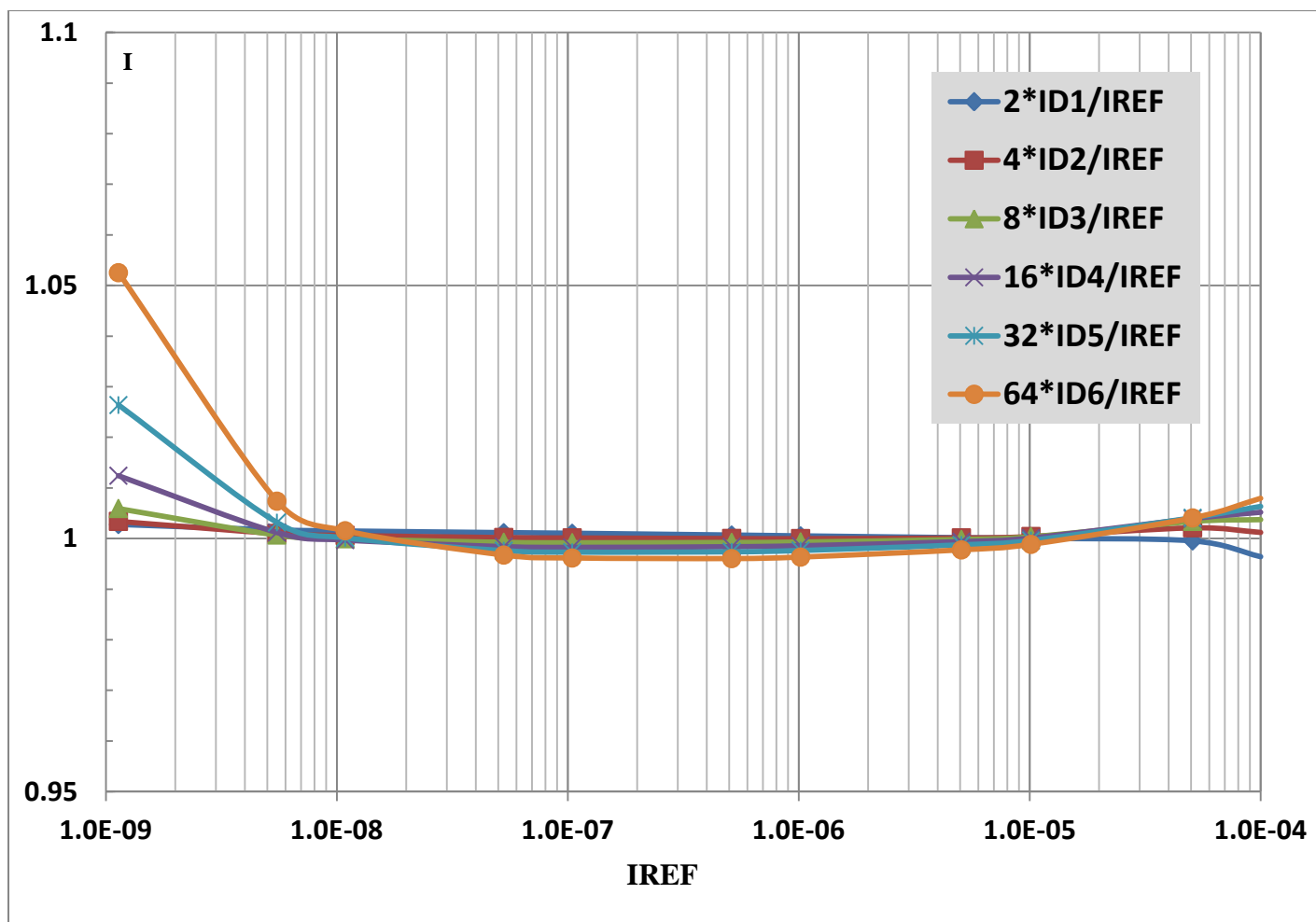
Εικόνα 5.35: *CGC simulation.*



Εικόνα 5.36: *CGB και CGC simulation.*



Εικόνα 5.37: *CGG simulation.*



Εικόνα 5.39 : Διάγραμμα I - $IREF$.

ΚΕΦΑΛΑΙΟ 6

-ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ-

6.1 Εισαγωγή

Θέμα της παρούσας διπλωματικής εργασίας είναι η ενσωμάτωση συμπαγών μοντέλων τύπου MOSFET σε κυκλώματα τύπου Spice και συγκεκριμένα η ενσωμάτωση του μοντέλου MOSFET EKV3 στον προσομοιωτή Ngspice.

Ο προσομοιωτής Ngspice είναι εργαλείο ελεύθερου λογισμικού και συνεπώς η διερεύνηση της λειτουργικότητας και οι προοπτικές ανάπτυξης του, παρουσιάζουν τεράστιο ενδιαφέρον.

Μέσω της σύγκρισης με άλλο παρεμφερές, μη ελεύθερο εργαλείο, τον προσομοιωτή Spectre και χρησιμοποιώντας ως προσομοιούμενο στοιχείο το μοντέλο MOSFET EKV3, καταλήξαμε σε ένα διαφωτιστικό σύνολο αποτελεσμάτων που επιβεβαιώνουν την εγκυρότητα της λειτουργίας του προσομοιωτή Ngspice.

6.2 Ανασκόπηση της διαδικασίας υλοποίησης

Τα στάδια της εργαστηριακής υλοποίησης είναι τα εξής:

α) Μετάφραση του κώδικα του MOSFET EKV3 από Verilog-A σε C και παράλληλη εγκατάστασή της παραγόμενης έκδοσης στον προσομοιωτή Ngspice. Η διαδικασία αυτή πραγματοποιήθηκε με την χρήση των εργαλείων ADMS και των αρχείων XML.

β) Εγκατάσταση της Verilog-A έκδοσης του MOSFET EKV3 στον προσομοιωτή Spectre.

γ) Εκτέλεση του ίδιου κύκλου DC και AC προσομοιώσεων του στοιχείου EKV3 στους δύο προσομοιωτές και σύγκριση των αποτελεσμάτων.

δ) Δημιουργία και προσομοίωση κυκλώματος (διαίρετη ρεύματος-r 2r ladder circuit) που περιλαμβάνει το MOSFET EKV3 σαν ηλεκτρικό στοιχείο.

Η υλοποίηση του εργαστηριακού τμήματος της εργασίας διεξήχθη με τα εργαλεία: προσομοιωτές ηλεκτρικών κυκλωμάτων *Ngspice* και *Spectre* και τα εργαλεία μετάφρασης *ADMS* και *XML* αρχεία. Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την μοντελοποίηση του στοιχείου MOSFET EKV3 είναι οι: γλώσσα περιγραφής μοντέλων *Verilog-A* και η γλώσσα προγραμματισμού C.

Επίσης χρησιμοποιήθηκαν τα εξής “βοηθητικά” προγράμματα: οι μεταγλωττιστές *bison* και *flex*, ο compiler *g++* και οι εξειδικευμένες για την εγκατάσταση των εργαλείων συναρτήσεις, *gawk* και *libreadline6-dev*.

6.3 Συμπεράσματα

Στους δύο προσομοιωτές *Ngspice* και *Spectre*, διεξήχθησαν πανομοιότυπες προσομοιώσεις του μοντέλου MOSFET EKV3, τύπου DC (id-vg, id-vd, id-vs), AC (Cgb, Cgc, Cgg) και CIRCUIT BASED (διαίρετης ρεύματος-r 2r ladder circuit).

Συμπεραίνουμε από την παρατήρηση των αποτελεσμάτων την ορθή λειτουργία του ενσωματωμένου στον προσομοιωτή *Ngspice*, μοντέλου MOSFET EKV3, καθώς το σφάλμα μεταξύ των προσομοιωτών *Ngspice* και *Spectre* είναι πολύ μικρής τάξης. Αποδίδουμε το υπάρχον σφάλμα σε ιδιαιτερότητες και λεπτομέρειες της λειτουργικότητας των δύο προσομοιωτών.

Επιβεβαιώνεται έτσι η ορθή λειτουργία του προσομοιωτή Ngspice και επαληθεύεται η εγκυρότητα και η αξιοπιστία του ως εργαλείο προσομοίωσης, ελεύθερου λογισμικού.

6.4 Προοπτική μελλοντικής εξέλιξης

Κατά την μετάφραση του αρχικού κώδικα του μοντέλου MOSFET EKV3 από Verilog-A σε γλώσσα C, μέρος του κώδικα που αφορά την συμπεριφορά του μοντέλου στον θόρυβο δεν μεταφράστηκε, λόγω περιορισμών του εργαλείου ADMS. Επίσης άλλα είδη ανάλυσης κυκλωμάτων, όπως *transient* και *στατιστική* ανάλυση παραμένουν προς διερεύνηση.

Στόχος της μελλοντικής εξέλιξης του μοντέλου MOSFET EKV3 είναι να υπερπηδηθούν οι περιορισμοί και να διερευνηθεί εκτενέστερα η συμπεριφορά του, παρέχοντας μία πιο ολοκληρωμένη εικόνα του φάσματος λειτουργίας του μοντέλου, μέσω της ενσωμάτωσής του στον προσομοιωτή Ngspice.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Computer History Museum, “The Silicon Engine – 1926 Field Effect Semiconductor Device Concepts Patented”.
<http://www.computerhistory.org/semiconductor/timeline/1926-field.html>
- [2] Αμπαζόγλου Γιάννης , “Αναλογικά Ηλεκτρονικά / Κεφ.1 - Εισαγωγή στην Ηλεκτρονική”.
- [3] C. Enz, F. Krummenacher, E. Vittoz. An analytical MOS transistor model valid in all regions of Operation and dedicated to low-voltage and low-current applications. Journal on Analog Integrated Circuits and Signal Processing, vol. 8, issue 1, pp.83–114, July 1995.
- [4] Matthias Bucher: “Analytical MOS transistor modelling for analog circuit simulation”, διδακτορική διατριβή No 2114 EPFL, 2000
- [5] J. M. Sallese, M. Bucher, F. Krummenacher, P. Fazan. Inversion Charge Linearization in MOSFET Modeling and Rigorous Derivation of the EKV Compact Model. Solid- State Electronics, vol. 47, No 4, pp.677- 683, 2003.
- [6] C. C. Enz and E. A. Vittoz, “Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF IC Design.” Chichester: Wiley, 2006.
- [7] Αντώνιος Στ. Μπαζίγος, “Ανάπτυξη μοντέλου MOS-FET στην γλώσσα Verilog-A βάσει του μοντέλου EKV”, Διπλωματική Εργασία, ΕΜΠ, Ιούλιος 2003.
- [8] Αντώνιος Στ. Μπαζίγος, “Μοντελοποίηση Mos τρανζίστορ σε υψηλές συχνότητες”, Διδακτορική Διατριβή, Αθήνα, Μάιος 2008.

[9] A. Bazigos, M. Bucher, e.a., EKV3 model code documentation, Technical University of Crete, 2008.

[10] Αντώνιος Παναγιώτης Ι. Μανδηλας, “Υλοποίηση Χρονοδρομολογητή με Υλικό”, Διπλωματική Εργασία, ΕΜΠ, Αθήνα, Ιούλιος 2007.

[11] “Verilog-A Language Reference Manual - Analog Extensions to Verilog HDL”, Version 1.0, Open Verilog International, August 1, 1996.

[12] NGSPICE user’s Manual.

[13] <http://adms.noovela.com/>, Paper.

ΠΑΡΑΡΤΗΜΑ

Το παρόν παράρτημα περιλαμβάνει τα input files (εντολές), των προσομοιώσεων του μοντέλου MOSFET EKV3 στον προσομοιωτή Ngspice.

[DC Simulation] id vg ngspice 1:

```
.include ekv3_090.par

vd d 0 dc 1
vg g 0 dc 1
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vg -200E-3 1 0.01

set filetype=ascii
run
echo vg " " id > a1.data
set appendwrite
wrdata a1 i(vs)
.endc
.end
```

[DC Simulation] id vg ngspice 2:

```
.include ekv3_090.par

vd d 0 dc 0.05
vg g 0 dc 1
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vg -200E-3 1 0.01

set filetype=ascii
run
echo vg " " id > a2.data
set appendwrite
wrdata a2 i(vs)
.endc
.end
```

ΠΑΡΑΡΤΗΜΑ

DC Simulation id_vd ngspice_1:

```
.include ekv3_090.par

vd d 0 dc 1
vg g 0 dc 0.05
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vd 0 1 0.01

set filetype=ascii
run
echo vd "      " id > a3.data
set appendwrite
wrdata a3 i(vs)
.endc
.end
```

[DC Simulation] id_vd ngspice_2:

```
.include ekv3_090.par

vd d 0 dc 1
vg g 0 dc 0.3
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vd 0 1 0.01

set filetype=ascii
run
echo vd "      " id > a4.data
set appendwrite
wrdata a4 i(vs)
.endc
.end
```

ΠΑΡΑΡΤΗΜΑ

[DC Simulation] id_vd ngspice 3:

```
.include ekv3_090.par

vd d 0 dc 1
vg g 0 dc 0.6
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vd 0 1 0.01

set filetype=ascii
run
echo vd "      " id > a5.data
set appendwrite
wrdata a5 i(vs)
.endc
.end
```

[DC Simulation] id_vd ngspice 4:

```
.include ekv3_090.par

vd d 0 dc 1
vg g 0 dc 1
vs s 0 dc 0.0
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vd 0 1 0.01

set filetype=ascii
run
echo vd "      " id > a6.data
set appendwrite
wrdata a6 i(vs)
.endc
.end
```

ΠΑΡΑΡΤΗΜΑ

[DC Simulation] id_vs ngspice_1:

```
.include ekv3_090.par

vd d 0 dc 1
vg g 0 dc 1
vs s 0 dc 1
vb b 0 dc 0.0
m1 d g s b nch W=10e-6 L=0.1e-6 NF=1

.control

dc vs 0 1 0.01

set filetype=ascii
run
echo vs "      " id > a7.data
set appendwrite
wrdata a7 i(vs)
.endc
.end
```

Οι ίδιες προσομοιώσεις επαναλήφθηκαν με παράμετρο $L=10e-6$.

ΠΑΡΑΡΤΗΜΑ

[AC Simulation] CGB:

CGB EXTRACTION

.include ekv3_018.par

.param W=660e-6

.param L=76e-6

*****STIMULUS CIRCUIT*****

VIN G 0 1.8 AC 1

M01 0 G 0 B NCH W=W L=L M=1

VBB B 0 0

.control

set filetype=ascii

set appendwrite

foreach VGG

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| -2 | -1.9 | -1.8 | -1.7 | -1.6 | -1.5 | -1.4 | -1.3 |
| -1.2 | -1.1 | -1 | -0.9 | -0.8 | -0.7 | -0.6 | -0.5 |
| -0.4 | -0.3 | -0.2 | -0.1 | 0.1 | 0.2 | 0.3 | 0.4 |
| 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 | 1.1 | 1.2 |
| 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | | |

alter VIN=\$VGG

ac lin 1MEG 1MEG 1MEG

wrdata cgb imag(i(VBB))/(2*3.14*1e6)

end

.endc

.end

ΠΑΡΑΡΤΗΜΑ

[AC Simulation] CGC:

CGC EXTRACTION

.include ekv3_018.par

.param W=660e-6

.param L=76e-6

*****STIMULUS CIRCUIT*****

VIN G 0 1.8 AC 1

M01 C G C 0 NCH W=W L=L M=1

VC C 0 0

.control

set filetype=ascii

set appendwrite

foreach VGG

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| -2 | -1.9 | -1.8 | -1.7 | -1.6 | -1.5 | -1.4 | -1.3 |
| -1.2 | -1.1 | -1 | -0.9 | -0.8 | -0.7 | -0.6 | -0.5 |
| -0.4 | -0.3 | -0.2 | -0.1 | 0 | 0.1 | 0.2 | 0.3 |
| 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 | 1.1 |
| 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | |

alter VIN=\$VGG

ac lin 1MEG 1MEG 1MEG

wrdata cgc imag(i(VC))/(2*pi*1e6)

end

.endc

.end

ΠΑΡΑΡΤΗΜΑ

AC Simulation CGG:

CGG EXTRACTION

.include ekv3_018.par

.param W=660e-6

.param L=76e-6

*****STIMULUS CIRCUIT*****

VIN G 0 1.8 AC 1

M01 C G C C NCH W=W L=L M=1

VC C 0 0

.control

set filetype=ascii

set appendwrite

foreach VGG

| | | | | | | | | |
|------|------|------|------|------|------|------|------|------|
| -2 | -1.9 | -1.8 | -1.7 | -1.6 | -1.5 | -1.4 | -1.3 | -1.2 |
| -1.1 | -1 | -0.9 | -0.8 | -0.7 | -0.6 | -0.5 | -0.4 | -0.3 |
| -0.2 | -0.1 | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| 0.7 | 0.8 | 0.9 | 1 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 |
| 1.6 | 1.7 | 1.8 | | | | | | |

alter VIN=\$VGG

ac lin 1MEG 1MEG 1MEG

wrdata cgg imag(i(VIN))/(2*3.14*1e6)

print imag(i(VIN))/(2*3.14*1e6)

end

.endc

.end

ΠΑΡΑΡΤΗΜΑ

Ngspice current divider circuit:

```
.include ekv3_018.par

.param W=10e-6
.param L=20e-6

*****STIMULUS CIRCUIT*****
VIN VGG 0 2
R1 VGG G 2066
*I1 VGG G 10e-6
M0 G G 0 0 NCH W=W L=L M=2
*****
Vt1 VOUT d1 0
M1 d1 G 0 0 NCH W=W L=L M=1
*****
Vt2 VOUT d2 0
M2 d2 G s2 0 NCH W=W L=L M=1
*****
Vt3 VOUT d3 0
M3 d3 G s3 0 NCH W=W L=L M=1
*****
Vt4 VOUT d4 0
M4 d4 G s4 0 NCH W=W L=L M=1
*****
Vt5 VOUT d5 0
M5 d5 G s5 0 NCH W=W L=L M=1
*****
Vt6 VOUT d6 0
M6 d6 G s6 0 NCH W=W L=L M=1
*****
Vt7 VOUT d7 0
M7 d7 G s6 0 NCH W=W L=L M=1
*****

*****
*****
M12 s2 G 0 0 NCH W=W L=L M=2
M23 s3 G s2 0 NCH W=W L=L M=2
M34 s4 G s3 0 NCH W=W L=L M=2
M45 s5 G s4 0 NCH W=W L=L M=2
M56 s6 G s5 0 NCH W=W L=L M=2

V1 VOUT 0 1.8

.control
```

ΠΑΡΑΡΤΗΜΑ

```
set filetype=ascii  
set appendwrite
```

```
foreach Rtest 1495491814.89322 149093333.333333 14765181.4893221  
1430933.33333333 128678.148932211 8309.33333333333
```

```
alter R1 = $Rtest  
dc V1 0.8 0.8 1
```

```
wrdata cur_divider abs(i(VIN)) 2*abs(i(vt1)) 4*abs(i(vt2)) 8*abs(i(vt3)) 16*abs(i(vt4))  
32*abs(i(vt5)) 64*abs(i(vt6))  
end
```

```
.endc
```

```
.end
```