

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ



ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Υλοποίηση Βιβλιοθηκών Τυποποιημένων Κυττάρων για  
Σύγχρονες Τεχνολογίες CMOS VLSI**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Ι. ΣΤΑΜΕΛΟΣ

**Επιβλέπων:** ΔΗΜΗΤΡΙΟΣ ΣΟΥΝΤΡΗΣ

Επίκουρος Καθηγητής

Αθήνα, Οκτώβριος 2010



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ



ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Υλοποίηση Βιβλιοθηκών Τυποποιημένων Κυττάρων για  
Σύγχρονες Τεχνολογίες CMOS VLSI**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Ι. ΣΤΑΜΕΛΟΣ

**Επιβλέπων:** ΔΗΜΗΤΡΙΟΣ ΣΟΥΝΤΡΗΣ

Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29<sup>η</sup> Οκτωβρίου 2010

-----  
ΔΗΜΗΤΡΙΟΣ ΣΟΥΝΤΡΗΣ

Επίκουρος Καθηγητής

-----  
ΚΙΑΜΑΛ ΠΕΚΜΕΣΤΖΗ

Καθηγητής

-----  
ΓΕΩΡΓΙΟΣ ΟΙΚΟΝΟΜΑΚΟΣ

Λέκτορας

Αθήνα, Οκτώβριος 2010

.....  
ΓΕΩΡΓΙΟΣ Ι. ΣΤΑΜΕΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © ΓΕΩΡΓΙΟΣ Ι. ΣΤΑΜΕΛΟΣ 2010

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



# Ευχαριστήρια

Θα ήθελα να ευχαριστήσω για την εκπόνηση της παρούσης εργασίας, τον εισηγητή και επιβλέποντα επίκουρο καθηγητή, κ. Σούντρη Δημήτριο, για την βοήθεια και την υποστήριξη που μου παρείχε. Θα ήθελα επίσης να ευχαριστήσω τον Μπεκιάρη Δημήτριο, για την υποστήριξη που μου παρείχε, ιδιαίτερα στο τεχνικό μέρος της εργασίας, καθώς και τον Παπανικολάου Αντώνιο για τη συνεισφορά του. Για τη συμμετοχή τους στην τριμελή επιτροπή, θα ήθελα να ευχαριστήσω τους κ.κ. Πεκμεστζή Κιαμάλ, Οικονομάκο Γεώργιο. Ένα μεγάλο ευχαριστώ στην οικογένειά μου, για την αμέριστη συμπαράσταση και κατανόηση που μου έδειξαν καθ' όλη τη διάρκεια των σπουδών. Τέλος αφιερώνω την παρούσα στην μνήμη του πατέρα μου, που έφυγε κατά τη διάρκεια της εκπόνησής της.

---

# Περιεχόμενα

Ευχαριστήρια .....	5
Κατάλογος Σχημάτων .....	9
Κατάλογος Πινάκων .....	15
ΠΕΡΙΛΗΨΗ .....	17
ABSTRACT.....	19
Κεφάλαιο 1 .....	21
1. Εισαγωγή .....	21
1.1. Απαιτήσεις σύγχρονων συστημάτων .....	21
1.2. Κλιμάκωση (scaling).....	22
1.3. Ιεραρχία σχεδίασης .....	24
1.4. Περιγραφές των cells .....	28
1.5. Στόχος της εργασίας.....	29
Κεφάλαιο 2 .....	33
2. Design flow.....	33
2.1. Βήματα που ακολουθούνται κατά την ψηφιακή σχεδίαση .....	33
2.2. Περιγραφή των απαραίτητων αρχείων της ροής σχεδίασης (design flow)...	37
2.2.1. Αρχείο Lib .....	37
2.2.2. Αρχείο Verilog (.v).....	44
2.2.3. Αρχείο LEF.....	46
2.2.4. Συγκέντρωση των αρχείων LIB, Verilog και LEF για τη σύνθεση της βιβλιοθήκης.....	55
Κεφάλαιο 3 .....	57
3. Κλιμάκωση (Scaling).....	57
3.1. Κανόνες Κλιμάκωσης (Scaling rules).....	57
3.2. Μοντέλα κλιμάκωσης .....	59
3.3. Δημιουργία LIB για μελλοντικές τεχνολογίες .....	61

---

3.3.1.	Δημιουργία του αρχείου Spice .....	61
3.3.2.	Επιλογή κατάλληλου μοντέλου τρανζίστορ .....	66
3.4.	Δημιουργία αρχείων LEF .....	67
3.5.	Δημιουργία Verilog .....	75
Κεφάλαιο 4	.....	77
4.	Χαρακτηρισμός βιβλιοθήκης .....	77
4.1.	Σκοπός του χαρακτηρισμού .....	77
4.2.	Περιγραφή βιβλιοθήκης standard cells που θα χαρακτηριστούν .....	78
4.3.	Συνθήκες χαρακτηρισμού .....	80
4.3.1.	Συνθήκες περιβάλλοντος λειτουργίας και διαδικασία παραγωγής των τρανζίστορ .....	80
4.3.2.	Επίδραση σημάτων εισόδου και χωρητικών φορτίων εξόδου .....	83
4.4.	Περιγραφή της διαδικασίας χαρακτηρισμού .....	86
4.4.1.	Εξαγωγή των αρχείων LIB και Verilog .....	87
4.4.2.	Εξαγωγή του LEF .....	89
4.5.	Εκτέλεση του χαρακτηρισμού .....	89
4.5.1.	Εκτέλεση του χαρακτηρισμού με τον Encounter Library Characterizer .....	89
4.5.2.	Χαρακτηρισμός των cells σε σχέση με την τοποθέτηση και τη διασύνδεσή τους μέσω του αρχείου LEF .....	92
Κεφάλαιο 5	.....	95
5.	Αποτελέσματα .....	95
5.1.	Αποτελέσματα σύνθεσης των designs, ανάλυσης του χρονισμού και της κατανάλωσής τους .....	95
5.1.1.	Σύνθεση .....	95
5.1.2.	Αποτελέσματα χρονισμού .....	98
5.1.2.1.	Αποτελέσματα χρονισμού σε επίπεδο πύλης .....	98
5.1.2.2.	Αποτελέσματα χρονισμού για designs .....	103
5.1.3.	Αποτελέσματα κατανάλωσης .....	106
5.1.3.1.	Αποτελέσματα δυναμικής κατανάλωσης .....	107
5.1.3.1.1.	Αποτελέσματα δυναμικής κατανάλωσης σε επίπεδο πύλης .....	107
5.1.3.1.2.	Αποτελέσματα δυναμικής κατανάλωσης σε επίπεδο κυκλώματος .....	110

---

5.1.3.2.	Αποτελέσματα στατικής κατανάλωσης.....	116
5.1.3.2.1.	Αποτελέσματα στατικής κατανάλωσης σε επίπεδο πύλης .....	117
5.1.3.2.2.	Αποτελέσματα στατικής κατανάλωσης σε επίπεδο design .....	120
5.1.3.3.	Ποσοστιαία συνεισφορά στατικής κατανάλωσης στην ολική κατανάλωση 123	
5.2.	Αποτελέσματα Placement και routing.....	126
5.2.1.	Τοποθέτηση και διασύνδεση σε διάφορα designs.....	128
5.3.	Αποτελέσματα σε επίπεδο συστήματος .....	138
Κεφάλαιο 6	.....	141
6.	Συμπεράσματα – Προτάσεις .....	141
6.1.	Συμπεράσματα .....	141
6.2.	Προτάσεις και μελλοντικές προεκτάσεις .....	142
ΠΑΡΑΡΤΗΜΑ	.....	145
7.	ΠΑΡΑΡΤΗΜΑ 1.....	145
7.1.	ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ ΚΛΙΜΑΚΩΣΗΣ (SCALING) ΤΟΥ SPICE NETLIST .....	145
7.2.	ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ ΚΛΙΜΑΚΩΣΗΣ ΤΟΥ ΑΡΧΕΙΟΥ LEF.....	148
8.	} .....	159
ΠΑΡΑΡΤΗΜΑ 2	.....	161
8.1.	SIMULATION SETUP FILE .....	161
8.2.	Configuration File (elccfg).....	171
8.3.	Command file (cmd_file).....	172
8.4.	Property file (property_file) .....	173
8.5.	Εκκίνηση της εφαρμογής Encounter Library Characterizer .....	173
8.6.	Μεταγλώττιση του αρχείου LIB σε .db.....	174
8.7.	Σύνθεση με τη βοήθεια του Design Compiler .....	175
Αναφορές-βιβλιογραφία	.....	177

---

# Κατάλογος Σχημάτων

Εικόνα 1.1.1 Η συνεισφορά της ανάπτυξης της ηλεκτρονικής στις απαιτήσεις της αγοράς. ....	21
Εικόνα 1.2.1 Μεταβολή των ελάχιστων διαστάσεων των ημιαγωγών, ανά δεκαετία. 23	
Εικόνα 1.2.2 Προβλέψεις για την εξέλιξη των ημιαγωγών. ....	23
Εικόνα 1.3.1 Γραφική απεικόνιση της από πάνω προς τα κάτω (top down) και της από κάτω προς τα πάνω (bottom up) προσέγγισης της διαδικασίας ψηφιακής σχεδίασης. 25	
Εικόνα 1.3.2 Τοποθέτηση cells. ....	26
Εικόνα 1.3.3 Απεικόνιση ενός standard cell (inverter και πύλη and) σε απλή κάτοψη και στις φάσεις ανάπτυξής του. ....	27
Εικόνα 1.3.4 Layout ενός standard cell. Εδώ διακρίνονται κάποιες χαρακτηριστικές διαστάσεις του. ....	28
Εικόνα 1.5.1 Περίγραμμα των βημάτων της εργασίας συμπεριλαμβανομένων και των εργαλείων και των αρχείων που χρησιμοποιούνται σε κάθε βήμα. ....	30
Εικόνα 2.1.1 Παρουσίαση των βημάτων ψηφιακής σχεδίασης (ASIC). ....	36
Εικόνα 2.2.1.1 Τυπική δομή του lib αρχείου. ....	38
Εικόνα 2.2.1.2 Γραφική παράσταση μέτρησης του Slew. ....	41
Εικόνα 2.2.1.3 Γραφική παράσταση μέτρησης της καθυστέρησης. ....	41
Εικόνα 2.2.1.4 Τρόπος υπολογισμού του delay (CMOS Non-linear Delay Model) με lookup table. ....	42
Εικόνα 2.2.2.1 Εσωτερική δομή του αρχείου Verilog με τα cells της βιβλιοθήκης. ....	45
Εικόνα 2.2.3.1 Εσωτερική δομή του αρχείου LEF. ....	47
Εικόνα 2.2.3.2 αφαιρετική επισκόπηση ενός κυττάρου αναστροφέα. ....	48
Εικόνα 2.2.3.3 Αφαιρετική επισκόπηση ενός κυττάρου πύλης NAND2X4. ....	48
Εικόνα 2.2.3.4 Αναλυτική περιγραφή layout του cell ενός αναστροφέα. ....	49
Εικόνα 2.2.3.5 Αναλυτική περιγραφή layout του cell πύλης NAND2x4. ....	49
Εικόνα 2.2.3.6 Δημιουργία πλέγματος για τοποθέτηση και διασύνδεση. ....	50
Εικόνα 2.2.3.7 Παράσταση της ιεραρχίας των επιπέδων. ....	51
Εικόνα 2.2.3.8 Περιγραφή του pitch των επιπέδων διασύνδεσης. ....	52
Εικόνα 2.2.3.9 Παράδειγμα υπολογισμού overhang. ....	52

---

Εικόνα 2.2.3.10 Τρόπος τοποθέτησης των γραμμών διασύνδεσης και των vias πάνω στο πλέγμα. Επίσης διακρίνονται οι line to line, line on via και via on via αποστάσεις.	53
Εικόνα 2.2.3.11 Τρόπος τοποθέτησης των cells σε γραμμές. Κάθε cell καταλαμβάνει έναν αριθμό sites.	54
Εικόνα 2.2.3.12 Λειτουργία της παραμέτρου symmetry κατά την τοποθέτηση του cell επάνω στη γραμμή.	54
Εικόνα 2.2.3.13 Περιγραφή εξαγωγής του αρχείου LEF από GDS II αρχείο (bottom - up προσέγγιση).	55
Εικόνα 2.2.4.1 Βιβλιοθήκη με τα απαραίτητα αρχεία.	56
Εικόνα 3.1.1 Η συχνότητα των επεξεργαστών διπλασιάζεται σε κάθε γενιά.	58
Εικόνα 3.1.2 Εξέλιξη της πυκνότητας των τρανζίστορ, ανάλογα με την τεχνολογία (ελάχιστη διάσταση).	59
Εικόνα 3.3.1 Τρόπος μέτρησης των γεωμετρικών χαρακτηριστικών W, L, ad, pd, as, ps ενός τρανζίστορ.	62
Εικόνα 3.3.2 Περιγραφή ενός αναστροφέα με spice netlist.	62
Εικόνα 3.3.3 Παράσταση αλγόριθμου κλιμάκωσης (scaling) διαστάσεων των τρανζίστορ του αρχείου spice netlist.	64
Εικόνα 3.3.4 Γραφικό περιβάλλον της εφαρμογής που εκτελεί τον αλγόριθμο κλιμάκωσης (scaling) του spice netlist των cells.	65
Εικόνα 3.3.5 Σχηματική παράσταση ενός NMOS τρανζίστορ.	66
Εικόνα 3.3.6 Σύγκριση I-V χαρακτηριστικών των μοντέλων BSIM1 και BSIM4.	67
Εικόνα 3.4.1 Τρόπος παράστασης διαστάσεων και θέσης.	68
Εικόνα 3.4.2 Σημεία τοποθέτησης συντεταγμένων.	68
Εικόνα 3.4.3 Τρόπος αυξομείωσης διαστάσεων, ανάλογα με τον προσανατολισμό του επιπέδου.	69
Εικόνα 3.4.4 Αλγόριθμος για την κλιμάκωση (scaling) των γεωμετρικών και ηλεκτρικών διαστάσεων των επιπέδων μετάλλου και των cells του αρχείου lef.	72
Εικόνα 3.4.5 Παράδειγμα τοποθέτησης και διασύνδεσης ενός macro cell εφαρμόζοντας τους κανόνες που αναφέρονται στο κείμενο.	75
Εικόνα 4.3.1 Χρονισμός ακολουθιακών κυκλωμάτων.	86
Εικόνα 4.4.1 Γραφική απεικόνιση των βημάτων χαρακτηρισμού.	87

---

Εικόνα 4.5.1 Αριθμός εξομοιώσεων που γίνονται για την δημιουργία του LIB αρχείου.....	90
Εικόνα 5.1.1 Αθροιστής 32x32.....	96
Εικόνα 5.1.2 Πολλαπλασιαστής 16x16. ....	96
Εικόνα 5.1.3 Πολλαπλασιαστής 16x16 με καταχωρητές στις εισόδους και την έξοδο. ....	96
Εικόνα 5.1.4 Πλήρης αθροιστής 1bit.....	97
Εικόνα 5.1.5 Σύγκριση αποτελεσμάτων χρονισμού για κάθε αντιπροσωπευτικό cell, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance). ....	99
Εικόνα 5.1.6 Σύγκριση αποτελεσμάτων χρονισμού για κάθε αντιπροσωπευτικό cell, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power). ....	99
Εικόνα 5.1.7 Σύγκριση αποτελεσμάτων χρονισμού για το cell NANDx2, για general purpose model και process corner tt, ss και ff. ....	100
Εικόνα 5.1.8 Σύγκριση αποτελεσμάτων χρονισμού για το cell NORx2, για general purpose model και για process corner tt, ss και ff. ....	100
Εικόνα 5.1.9 Σύγκριση αποτελεσμάτων χρονισμού για το cell INVX1, για general purpose model και για process corner tt, ss και ff. ....	101
Εικόνα 5.1.10 Διαγράμματα της διαφοράς $V_{dd}-V_t$ των τρανζίστορ σε κάθε τεχνολογία.....	102
Εικόνα 5.1.11 Παράσταση της ποσότητας $1/(V_{dd}-V_t)^{\alpha}$ . ....	102
Εικόνα 5.1.12 Σύγκριση αποτελεσμάτων χρονισμού για κάθε design, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance) και process corner typical .....	103
Εικόνα 5.1.13 Σύγκριση αποτελεσμάτων χρονισμού για κάθε design, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power) και process corner typical.....	104
Εικόνα 5.1.14 Σύγκριση αποτελεσμάτων χρονισμού για το design adder 16x16, για process corner typical, slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ. ....	104
Εικόνα 5.1.15 Σύγκριση αποτελεσμάτων χρονισμού για το design multiplier 32x32, για process corner typical, slow –slow, fast –fast και γενικό μοντέλο τρανζίστορ. .	105
Εικόνα 5.1.16 Σύγκριση αποτελεσμάτων χρονισμού για το design multiplier 32x32 με καταχωρητές στις εισόδους και την έξοδο, για process corner typical, slow –slow, fast –fast και γενικό μοντέλο τρανζίστορ. ....	105
Εικόνα 5.1.17 Σύγκριση τιμών του μεγέθους $V_{dd}-V_t$ για process corner ss, ff και typical και τεχνολογίες 45 και 32nm. ....	106

---

Εικόνα 5.1.3.1 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).....	107
Εικόνα 5.1.3.2 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power). .....	109
Εικόνα 5.1.3.3 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για το cell NANDx2, για process corners tt, ss και ff. ....	109
Εικόνα 5.1.3.4 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για το cell NORx2, για process corners tt, ss και ff. ....	110
Εικόνα 5.1.3.5 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για το cell INVX1, για process corners tt, ss και ff.....	110
Εικόνα 5.1.3.6 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance) και process corner typical. ....	111
Εικόνα 5.1.3.7 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power) και process corner typical. ....	112
Εικόνα 5.1.3.8 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης, για το design adder 32x32 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.....	113
Εικόνα 5.1.3.9 Ίδιο με την προηγούμενη εικόνα, με αποτελέσματα και για τα 180nm. .....	113
Εικόνα 5.1.3.10 Σύγκριση δυναμικής κατανάλωσης, για το design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ. .....	114
Εικόνα 5.1.3.11 Ίδιο με την προηγούμενη εικόνα, με αποτελέσματα και για τα 180nm. ....	114
Εικόνα 5.1.3.12 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης, για το design multiplier 16x16 με καταχωρητές στις εισόδους και στην έξοδο, για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ. ....	115
Εικόνα 5.1.3.13 Ίδιο με την προηγούμενη εικόνα, με αποτελέσματα και για τα 180nm. ....	115

---



Εικόνα 5.1.3.14 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).....	117
Εικόνα 5.1.3.15 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power). .....	118
Εικόνα 5.1.3.16 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για το cell NANDx2, για process corners tt, ss και ff. ....	118
Εικόνα 5.1.3.17 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για το cell NORx2, για process corners tt, ss και ff. ....	119
Εικόνα 5.1.3.18 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για το cell INVx1, για process corners tt, ss και ff.....	119
Εικόνα 5.1.3.19 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance). ....	120
Εικόνα 5.1.3.20 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).....	121
Εικόνα 5.1.3.21 Σύγκριση στατικής κατανάλωσης, για το design adder 32x32 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.	121
Εικόνα 5.1.3.22 Σύγκριση στατικής κατανάλωσης, για το design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.	122
Εικόνα 5.1.3.23 Σύγκριση στατικής κατανάλωσης, για το design της ακολουθιακής εκδοχής του multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.....	122
Εικόνα 5.1.3.24 Σύγκριση λόγου στατικής κατανάλωσης/συνολικής για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance). ....	123
Εικόνα 5.1.3.25 Σύγκριση λόγου στατικής κατανάλωσης/συνολικής για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).....	124
Εικόνα 5.1.3.26 Σύγκριση λόγου στατικής κατανάλωσης/ολικής, για το design adder 32x32 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.....	124
Εικόνα 5.1.3.27 Σύγκριση λόγου στατικής κατανάλωσης/ολικής, για το design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ. ....	125

---

Εικόνα 5.1.3.28 Σύγκριση λόγου στατικής κατανάλωσης/ολικής, για την ακολουθιακή εκδοχή του design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.....	125
Εικόνα 5.2.1 Περιβάλλον του SoC Encounter για την δήλωση του design και των βιβλιοθηκών.....	126
Εικόνα 5.2.2 Ορισμός floor plan.....	127
Εικόνα 5.2.3 Τρόπος τοποθέτησης power rings. ....	127
Εικόνα 5.2.4 Τοποθέτηση και διασύνδεση των cells.....	128
Εικόνα 5.2.5 τοποθέτηση και διασύνδεση των cells που συνθέτουν έναν πλήρη αθροιστή.....	129
Εικόνα 5.2.6 Διάταξη cells για τον αθροιστή 32x32 bit στα 45nm. ....	129
Εικόνα 5.2.7 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 45nm. ....	130
Εικόνα 5.2.8 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 32nm. ....	131
Εικόνα 5.2.9 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 22nm. ....	131
Εικόνα 5.2.10 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 16nm. ....	132
Εικόνα 5.2.11 χωρητικότητα καλωδίου/μονάδα μήκους συναρτήσσει της τεχνολογίας για τον αθροιστή 32x32, που έδωσε το εργαλείο SoC Encounter. ....	133
Εικόνα 5.2.12 Ολική χωρητικότητα (καλωδίου + ακίδας) συναρτήσσει της τεχνολογίας για τον αθροιστή 32x32 που έδωσε το εργαλείο SoC Encounter. ....	133
Εικόνα 5.2.13 Τοποθέτηση cells για την υλοποίηση του πολλαπλασιαστή 16x16 στα 45nm. ....	134
Εικόνα 5.2.14 Layout για τον πολλαπλασιαστή 16x16 στην τεχνολογία 45nm.....	134
Εικόνα 5.2.15 χωρητικότητα καλωδίου/μονάδα μήκους συναρτήσσει της τεχνολογίας για τον πολλαπλασιαστή 16x16 που έδωσε το εργαλείο SoC Encounter.....	135
Εικόνα 5.2.16 Ολική χωρητικότητα (καλωδίου + ακίδας) συναρτήσσει της τεχνολογίας για τον πολλαπλασιαστή 16x16 που έδωσε το εργαλείο SoC Encounter.....	136
Εικόνα 5.2.17 Τοποθέτηση των cells που απαρτίζουν την ακολουθιακή εκδοχή του πολλαπλασιαστή 16x16 .....	137

---

Εικόνα 5.2.18 Γραφική παράσταση της περιοχής που καταλαμβάνει κάθε design, σε σχέση με τη χρησιμοποιούμενη τεχνολογία. ....	138
Εικόνα 5.3.1 1. Το 1940x1896 um <sup>2</sup> layout του LEON3-based MP-SoC με δύο πυρήνες στο SoC Encounter. ....	140
Εικόνα 8.1.1 Τρόπος υπολογισμού της μέσης τιμής του Hold time για ανερχόμενο παλμό. ....	163
Εικόνα 8.1.2 Τρόπος υπολογισμού της μέσης τιμής του Hold time για κατερχόμενο παλμό. ....	163

## Κατάλογος Πινάκων

2.2.1.1 Παρουσίαση αποτελεσμάτων, με lookup table. ....	43
Πίνακας 3.4.1 Παράδειγμα αντιμετώπισης του προβλήματος εξασφάλισης ακεραίας πολλαπλότητας δύο τιμών. ....	73
Πίνακας 4.2.1 Περιγραφή της βιβλιοθήκης. ....	78
4.3.1 Συνοπτικός πίνακας με τις συνθήκες χαρακτηρισμού ανά τεχνολογία. ....	83
Πίνακας 4.3.2 Επιλογή περιοχής τιμών Slew και Cload για τον χαρακτηρισμό σε κάθε τεχνολογία. ....	85
Πίνακας 4.5.1 Είδη διανυσμάτων εξομοίωσης. ....	90
Πίνακας 4.5.2 Σύνολο εξομοιώσεων που γίνονται ανά cell. ....	91
Πίνακας 4.5.3 Εμβαδό που καταλαμβάνει κάθε cell. ....	93
Πίνακας 5.1.1 Χωρητικά φορτία και slew σήματος εισόδου στα οποία αντιστοιχούν τα αποτελέσματα χρονισμού και κατανάλωσης των τριών βασικών πυλών που επιλέξαμε. ....	98
Πίνακας 5.2.1 Μεσαίες τιμές συνολικής χωρητικότητας καλωδίων και χωρητικότητας καλωδίου/micron που έδωσε το εργαλείο SoC Encounter. ....	132
Πίνακας 5.2.2 Πίνακας 5.2.2 Ολική χωρητικότητα (καλωδίου + ακίδας) συναρτήσει της τεχνολογίας για τον πολλαπλασιαστή 16x16 που έδωσε το εργαλείο SoC Encounter. ....	135
Πίνακας 5.2.3 Αποτελέσματα για το εμβαδό που καταλαμβάνει κάθε design. ....	137
Πίνακας 5.3.1 Αποτελέσματα για σύστημα με δύο πυρήνες LEON3 SPARC, σε τεχνολογία 32nm (General Purpose model, typical corner). ....	139

---

Πίνακας 8.1.1 Κυριότερες παράμετροι του setup αρχείου. ....	162
Πίνακας 8.1.2 Δηλώσεις για τις Nominal τιμές διαφόρων μεγεθών. ....	164
Πίνακας 8.1.3 Παράμετροι που αφορούν την ενότητα Process του Setup αρχείου. .	164
Πίνακας 8.1.4 Παράμετροι που καθορίζουν τη μορφή και τις στάθμες του σήματος εισόδου ή εξόδου. ....	168
Πίνακας 8.1.5 Δηλώσεις που αφορούν το είδος της εξομοίωσης, την δυαδική αναζήτηση, την τιμή της αντίστασης Pulling και το όνομα του προκαθορισμένου κυκλώματος, που παράγει μη γραμμικό Slew στην είσοδο. ....	169

---

## ΠΕΡΙΛΗΨΗ

Οι αυξανόμενες απαιτήσεις για υλοποίηση όλο και περισσότερων λειτουργιών από ένα κύκλωμα, έχει οδηγήσει στον υψηλό βαθμό ολοκλήρωσης. Η αύξηση του βαθμού ολοκλήρωσης οδηγεί την τεχνολογία και την έρευνα σε αναζήτηση νέων μεθόδων κατασκευής και υλοποίησης των ολοκληρωμένων κυκλωμάτων. Η ανάγκη αυτή προέκυψε απ' την ελάττωση των διαστάσεων της τεχνολογίας που απαιτείται σε κάθε φάση εξέλιξης. Σκοπός της παρούσας εργασίας είναι η υλοποίηση νέων βιβλιοθηκών, απαραίτητων για τα εργαλεία cad ψηφιακής σχεδίασης, που θα χρησιμοποιηθούν στη σχεδίαση εφαρμογών σε τεχνολογίες που βρίσκονται ακόμα σε πρώιμο στάδιο ανάπτυξης (45, 32, 22 και 16nm). Η υλοποίησή τους κρίνεται απαραίτητη γιατί θα επιτρέψει τη μελέτη διαφόρων εφαρμογών και αρχιτεκτονικών για αυτές τις τεχνολογίες, χωρίς κατά ανάγκη να υποστηρίζονται ακόμα από τη βιομηχανία κατασκευής ολοκληρωμένων κυκλωμάτων, ενώ παράλληλα γίνεται ανεμπόδιστα η απαιτούμενη έρευνα και ανάπτυξη των εν λόγω τεχνολογιών. Οι βιβλιοθήκες αυτές περιλαμβάνουν τυποποιημένα κύτταρα (cells) βασικών λογικών συνδυαστικών και ακολουθιακών κυκλωμάτων, από τα οποία μπορούν να προκύψουν με τη διαδικασία της σύνθεσης πολύπλοκες εφαρμογές. Η διαδικασία υλοποίησης της κάθε βιβλιοθήκης βασίζεται στην κλιμάκωση (scaling) μιας αντίστοιχης, ήδη υπάρχουσας στην τεχνολογία των 180nm. Βάση αποτελούν το αρχείο netlist spice που περιέχει υπό μορφή sub circuits, όλα τα τυποποιημένα κύτταρα και το αρχείο .LEF που δίνει τις απαραίτητες πληροφορίες για την κατασκευή του layout του κάθε κελιού και των συνδέσεων. Με τη βοήθεια συγκεκριμένων εργαλείων αλλά και κανόνων, θα εκτελεστούν όλα τα στάδια της υλοποίησης. Η διαδικασία χωρίζεται σε δύο σκέλη. Το πρώτο σκέλος περιλαμβάνει τα στάδια παραγωγής του αρχείου .LIB το οποίο θα περιλαμβάνει πληροφορίες χρονισμού και κατανάλωσης για την σύνθεση των κυκλωμάτων. Το δεύτερο σκέλος περιλαμβάνει τα στάδια παραγωγής του .LEF αρχείου, το οποίο είναι απαραίτητο για την υλοποίηση του layout των κυκλωμάτων σε επίπεδο cells. Τα στάδια του πρώτου σκέλους περιλαμβάνουν την αναζήτηση του σωστού μοντέλου των τρανζίστορ που θα χρησιμοποιήσουμε για την επιθυμητή τεχνολογία, την κλιμάκωση των διαστάσεων των τρανζίστορ που δομούν τα cells της δεδομένης βιβλιοθήκης, τον χαρακτηρισμό των cells της βιβλιοθήκης με σκοπό την

εξαγωγή πληροφορίας κατανάλωσης ισχύος και καθυστερήσεων για το κάθε cell. Το δεύτερο σκέλος βασίζεται στην ύπαρξη του αρχείου .LEF για την τεχνολογία 180nm. Τα στάδια υλοποίησης εδώ περιλαμβάνουν την κλιμάκωση των φυσικών διαστάσεων τόσο των καλωδιώσεων σύνδεσης σε κάθε επίπεδο μετάλλου, όσο και των αντίστοιχων για τις διασυνδέσεις των cells. Για την αυτοματοποίηση της διαδικασίας είναι απαραίτητη η υλοποίηση βοηθητικών προγραμμάτων, που θα εφαρμόσουν τους κανόνες scaling στα δεδομένα αρχεία της τεχνολογίας 180nm και θα αυτοματοποιήσουν τη διαδικασία. Το αποτέλεσμα θα επαληθευθεί με την υλοποίηση κυκλωμάτων που θα συντεθούν από cells των νέων βιβλιοθηκών για τεχνολογίες 45nm, 32nm, 22nm, 16nm, με εξαγωγή αποτελεσμάτων χρονισμού, κατανάλωσης και τοποθέτησης και διασύνδεσης.

## ABSTRACT

The growing demands for the achievement of more and more processes of one circuit have led to a high level of integration. Moore's rule stands for the increase of the level of completion, thus leading technology and research to the search of new integrated circuits' construction and materialization methods. This need emerged from the decrease of technology's dimensions, which is demanded on every phase of evolution. The purpose of this task is the materialization of new libraries which are going to be used on the development of applications on technologies of 45, 32, 22, and 16 nm. These libraries include standardized cells of basic logical combining and sequential circuits, from which any applications can derive with the procedure of composition. The procedure of materialization of every library is based on the escalation of an already existing one in the technology of 180 nm. The netlist spice document, which contains all the standardized cells in the form of sub circuits, and the lef document, which gives the necessary information for the construction of each cell's and connections' layout, consist the base for this procedure. All the materialization points are going to be executed with the aid of specific tools and rules. The procedure is divided in two parts. The first part includes the production steps of lib document, which will include time delay and consumption information for the circuits' composition. The second part includes the production steps of lef document, which is necessary for the circuits' layout materialization. The first part's steps include the inquire of the right transistor model that we are going to use for the desirable technology, the escalation of the specific library's cells dimensions, the characterization of the library's cells which intends to the extract of the voltage consumption and time delay information for each cell. The second part is based on lef document's existence for the 180nm technology. In this case, the materialization points include the escalation of the physical dimensions of connection wirings on every metal level, as well as the equivalent for the cells interconnections. For the automatization of the procedure, the materialization of ancillary applications which are going to apply the escalation rules at the documents of 180nm technology is necessary. Confirmation with the materialization of circuits which derive from the composition of cells from the new library is made at each part, for each one of the

technologies of 45nm, 32nm, 22nm, 16nm, with the extract of tune, consumption, and parasitic capacity, resistance and auto induction results.

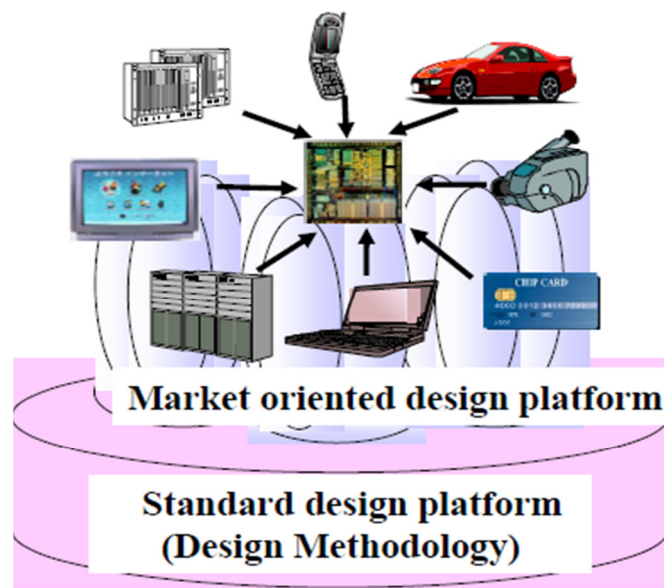


# Κεφάλαιο 1

## Εισαγωγή

### 1.1. Απαιτήσεις σύγχρονων συστημάτων

Η ανάγκη για δημιουργία όλο και μικρότερων διαστάσεων τεχνολογίας, προκύπτει από τις ανάγκες και υψηλές απαιτήσεις των σύγχρονων ηλεκτρονικών συστημάτων. Ηλεκτρονικά συστήματα κατακλύζουν την καθημερινότητά μας και οι απαιτήσεις μας οδηγούν στον σχεδιασμό και την κατασκευή προϊόντων που η λειτουργία τους και οι δυνατότητές τους περιορίζονται μόνο από τη φαντασία. Ποιος θα μπορούσε να φανταστεί πως στο μέγεθος της παλάμης μας θα χωρά ένας ισχυρός υπολογιστής πριν μερικές δεκαετίες. Συστήματα που θα είχαν ενσωματωμένη τη λειτουργία ενός κινητού τηλεφώνου, ενός πλοηγού ιστοσελίδων, ενός video player, μιας παιχνιδομηχανής, ή ενός εντοπιστή θέσεως και πολλών άλλων.



Εικόνα 1.1.1 Η συνεισφορά της ανάπτυξης της ηλεκτρονικής στις απαιτήσεις της αγοράς.

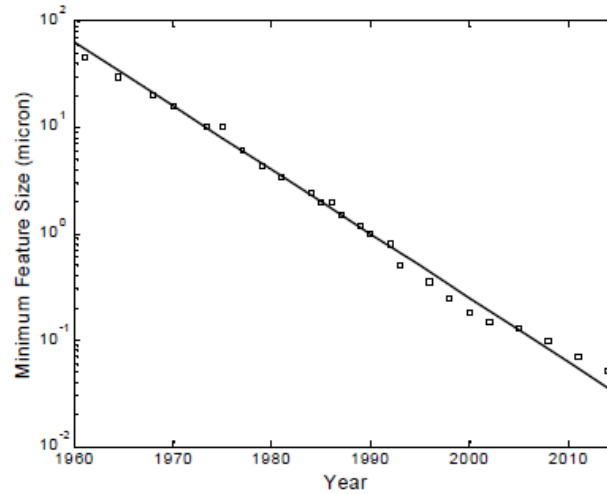
Εδώ είναι όμως που η σύγχρονη τεχνολογία συνέβαλε ώστε να είναι δυνατή η ανάπτυξη τέτοιων εφαρμογών. Οι απαιτήσεις που θέτουν οι αυστηρές προδιαγραφές συστημάτων παρόμοιων με αυτά που αναφέρθηκαν, είναι πολύ μεγάλες τόσο για μικρό μέγεθος, όσο και για μειωμένη κατανάλωση και γρήγορη απόκριση. Οι τρεις αυτές βασικές απαιτήσεις είναι προφανείς αν κανείς αναλογιστεί και μόνο ότι δε θα είχε καμία σημασία η κατασκευή ενός συστήματος μεγάλων διαστάσεων

συγκρίσιμων με αυτές ενός υπολογιστή desktop, ή η κατασκευή ενός συστήματος που η μπαταρία του θα διαρκούσε για λίγες μόνο ώρες, και ακόμα θα ήταν αδύνατο να δούμε μια ταινία αν το σύστημά μας δεν ήταν αρκετά γρήγορο για να κατεβάσει και να αποκωδικοποιήσει ένα video stream. Για να είναι δυνατή λοιπόν η πραγματοποίηση των προηγούμενων, πρέπει οι βασικές δομικές μονάδες των συστημάτων αυτών να καταλαμβάνουν όσο το δυνατό μικρότερη επιφάνεια, να τροφοδοτούνται με χαμηλότερες τάσεις λειτουργίας, να καταναλώνουν χαμηλά ποσά ενέργειας και τα ψηφιακά τους μέρη να μπορούν να δουλέψουν σε υψηλούς ρυθμούς ρολογιού. Όσο εύκολα κι αν ακούγονται όμως όλα αυτά, τόσο δύσκολα επιτυγχάνονται. Αν αναλογιστούμε ειδικότερα πως η βελτιστοποίηση κάποιων εκ των προηγούμενων ιδιοτήτων εγκυμονεί κίνδυνο επιδείνωσης κάποιας άλλης, καταλαβαίνουμε τη δυσκολία που πρέπει να αντιμετωπίσουμε. Η αύξηση της συχνότητας λειτουργίας π.χ ενός ψηφιακού συστήματος αυξάνει αναλογικά την κατανάλωση ενέργειας. Η μείωση των διαστάσεων των αγωγών διασύνδεσης και των στοιχείων των ημιαγωγών αυξάνουν τις εσωτερικές αντιστάσεις και επομένως τους χρόνους φόρτισης – εκφόρτισης των παρασιτικών χωρητικοτήτων και των χωρητικοτήτων που βλέπουν οι έξοδοι των ψηφιακών κυκλωμάτων.

## 1.2. Κλιμάκωση (scaling)

Η σταδιακή κλιμάκωση (scaling) των διαστάσεων των δομικών μονάδων μιας παλαιότερης τεχνολογίας είναι ίσως μια απλοϊκή σκέψη, ώστε να προκύψουν αντίστοιχα συστήματα μικρότερων διαστάσεων της τάξεως μεγέθους που επιδιώκουμε. Αυτό ουσιαστικά γίνεται και στην πράξη. Γίνεται όμως σταδιακά και με συγκεκριμένους κανόνες που θα δούμε παρακάτω. Ο λόγος που δεν μπορούμε να μειώσουμε αυθαίρετα τις διαστάσεις των βασικών δομικών μονάδων είναι ότι δεν έχουμε εξαρχής τα εργαλεία σχεδίασης, εξομοίωσης και κατασκευής που υποστηρίζουν τεχνολογίες πολύ μικρών διαστάσεων. Εδώ θα ήταν χρήσιμο να δούμε και μερικά στοιχεία από την εξέλιξη των διαστάσεων της τεχνολογίας στο πέρασμα των χρόνων.

## Technology Scaling (1)



### Minimum Feature Size

EECS141-S04

Εικόνα 1.2.1 Μεταβολή των ελάχιστων διαστάσεων των ημιαγωγών, ανά δεκαετία.<sup>1</sup>

## Technology Roadmap

International Technology Roadmap for Semiconductors  
2002 data

Year	2001	2003	2005	2007	2010	2013	2016
DRAM ½ pitch [nm]	130	100	80	65	45	32	22
MPU transistors/chip	97M	153M	243M	386M	773M	1.55G	3.09G
Wiring levels	8	8	10	10	10	11	11
High-perf. phys. gate [nm]	65	45	32	25	18	13	9
High-perf. VDD [V]	1.2	1.0	0.9	0.7	0.6	0.5	0.4
Local clock [GHz]	1.7	3.1	5.2	6.7	11.5	19.3	28.8
High-perf. power [W]	130	150	170	190	218	251	288
Low-power phys. gate [nm]	90	65	45	32	22	16	11
Low-power VDD [V]	1.2	1.1	1.0	0.9	0.8	0.7	0.6
Low-power power [W]	2.4	2.8	3.2	3.5	3.0	3.0	3.0

Node years: 2007/65nm, 2010/45nm, 2013/32nm, 2016/22nm

EECS141-S04

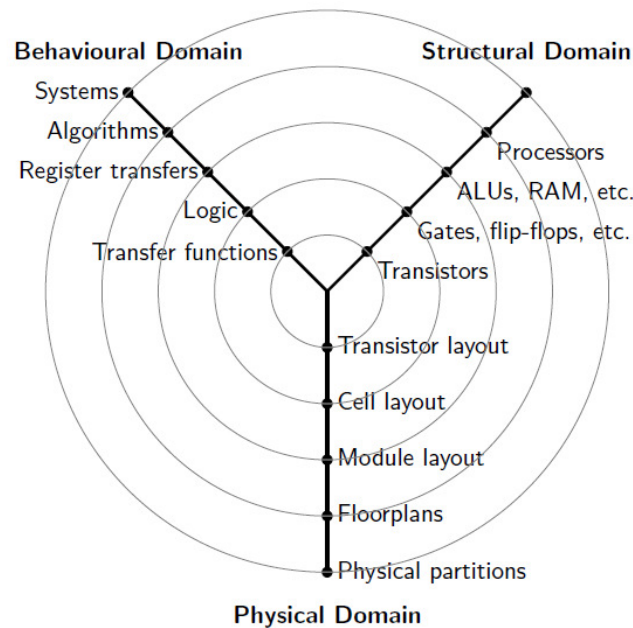
Εικόνα 1.2.2 Προβλέψεις για την εξέλιξη των ημιαγωγών.<sup>1</sup>

<sup>1</sup> Στοιχεία από Berkeley University

Έτσι από τις τεχνολογίες ελάχιστης διάστασης 12μm, περάσαμε στην τεχνολογία 8.5μm, 6μm, 4.5μm, 3.2μm, 2μm, 1μm, 0.6 μm, 0.8 μm, 0.25μm κλπ. Οι μεταβάσεις αυτές παίρνουν συνήθως αρκετό χρόνο ώστε να διεξαχθεί και να τεκμηριωθεί η αντίστοιχη επιστημονική έρευνα και εν συνεχεία να ενσωματωθεί στη διαδικασία παραγωγής. Επίσης η τεχνολογία που μπορούν να υποστηρίξουν οι εταιρείες παραγωγής ολοκληρωμένων κυκλωμάτων ASIC δεν ακολουθεί την ίδια πορεία με τα αντίστοιχα εργαλεία σχεδίασης. Πάντα χρειάζεται ένα χρονικό περιθώριο για την κάλυψη αυτής της υστέρησης. Καταλαβαίνουμε λοιπόν την σπουδαιότητα του scaling αλλά και τις τεχνικές και τεχνολογικές δυσκολίες που πρέπει να υπερνικήσουμε. Μπορεί κάποιος διαθέτοντας το κατάλληλο εργαλείο (σχεδιασμένο για υποστήριξη τέτοιων τεχνολογιών) να σχεδιάσει εξ αρχής όλα τα δομικά στοιχεία που θα χρειαστούν κατά την κατασκευή και σύνθεση των συστημάτων, ή να πάρει τα ίδια δομικά στοιχεία που χρησιμοποιήθηκαν σε παλαιότερη τεχνολογία (μεγαλύτερης ελάχιστης διάστασης) και εφαρμόζοντας κανόνες scaling, να καταλήξει σε νέα δομικά στοιχεία με χαρακτηριστικά φυσικών διαστάσεων, κατανάλωσης και χρονισμού τέτοια που θα οδηγήσουν σε ανάπτυξη κυκλωμάτων υψηλότερης ολοκλήρωσης, μικρότερης κατανάλωσης και ταχύτερης απόκρισης. Στην παρούσα εργασία θα χρησιμοποιήσουμε τη δεύτερη μέθοδο την οποία και θα αναλύσουμε με λεπτομέρεια.

### 1.3. Ιεραρχία σχεδίασης

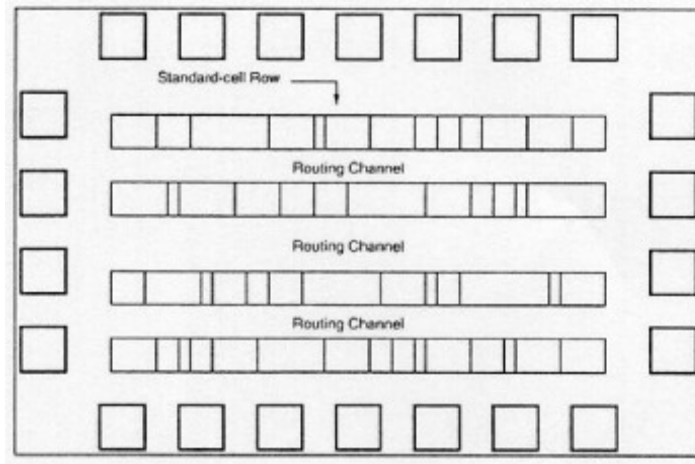
Για να δούμε όμως πως χρησιμοποιούμε όλα αυτά που αναφέραμε στην προηγούμενη ενότητα, θα πρέπει να δούμε λίγο την ιεραρχία σχεδίασης. Η ιεραρχική σχεδίαση είναι το εργαλείο εκείνο, που μας επιτρέπει τη σχεδίαση πολύπλοκων συστημάτων. Η δυνατότητα αυτή προκύπτει από την απόκρυψη “περιττής” πληροφορίας από το εκάστοτε στάδιο σχεδίασης. Αφαίρεση δηλαδή της πολυπλοκότητας είναι ο βασικός στόχος, ώστε η κάθε εμπλεκόμενη διαδικασία να απασχολείται λεπτομερώς με το επίπεδο ιεραρχίας στο οποίο ανήκει. Έτσι γίνεται καταμερισμός και συστηματοποίηση ολόκληρης της διαδικασίας σχεδίασης. Στην ψηφιακή σχεδίαση ακολουθείται το μοντέλο του “Y” chart.



Εικόνα 1.3.1 Γραφική απεικόνιση της από πάνω προς τα κάτω (top down) και της από κάτω προς τα πάνω (bottom up) προσέγγισης της διαδικασίας ψηφιακής σχεδίασης.

Σύμφωνα με το μοντέλο αυτό η σχεδίαση ενός ψηφιακού συστήματος κινείται σε τρεις άξονες. Ο πρώτος άξονας αφορά την αρχιτεκτονική σχεδίαση του συστήματος, ο δεύτερος την ανάπτυξη του λογισμικού και ο τρίτος την σχεδίαση του hardware. Ο κάθε ομόκεντρος κύκλος, παριστάνει ένα επίπεδο ιεραρχίας. Πλησιάζοντας το κέντρο των κύκλων κατεβαίνουμε προς το φυσικό επίπεδο, ενώ όσο πιο μακριά κινούμαστε από το κέντρο, τόσο πιο αφαιρετική γίνεται η προσέγγιση του συστήματος. Ο τελευταίος άξονας είναι εκείνος που θα μας απασχολήσει. Κάποια απ' τα εργαλεία CAD που εμπλέκονται στο επίπεδο “cell layout” της ιεραρχίας, θα χρησιμοποιήσουν τις βιβλιοθήκες που θα δημιουργήσουμε. Έχοντας κατά νου το μοντέλο αυτό η σχεδίαση γίνεται στα εξής βήματα:

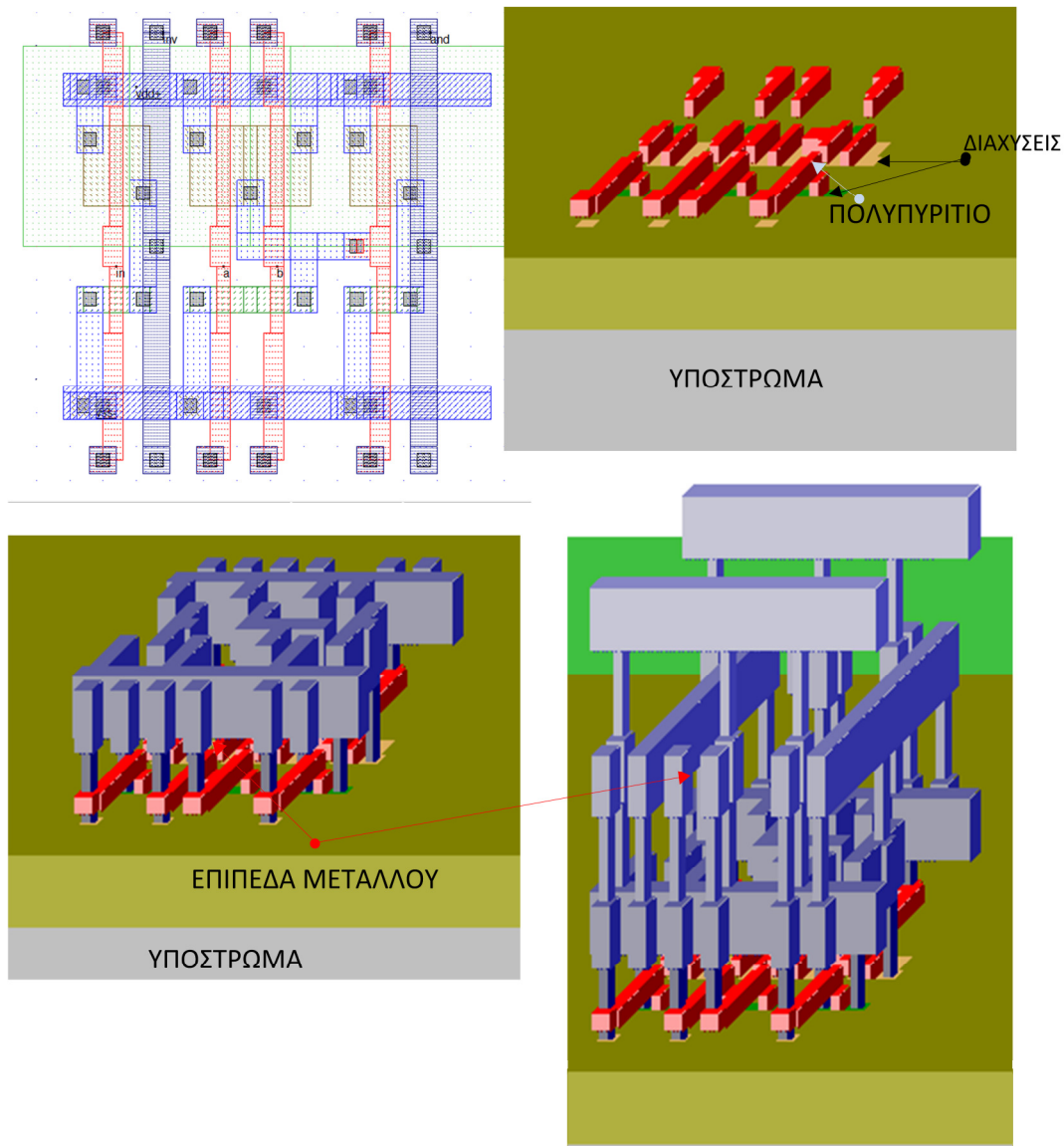
- Το υλικό (hardware) περιγράφεται με τη βοήθεια μιας γλώσσας σε υψηλό επίπεδο (RTL register transfer level για παράδειγμα).
- Η περιγραφή αυτή μεταφράζεται σε Boolean λογική που υλοποιείται από συνδυαστικά και ακολουθιακά στοιχεία.
- Η πιο πάνω λογική αντιστοιχίζεται σε τυποποιημένα κύτταρα (std cells).
- Τα cells τοποθετούνται σε γραμμές.
- Τέλος συνδέονται μεταξύ τους μέσω των καναλιών καλωδίωσης.



**Εικόνα 1.3.2 Τοποθέτηση cells.**

Η υλοποίηση designs ως αυτό το επίπεδο, μας επιτρέπει να μελετήσουμε τη συμπεριφορά συστημάτων για διάφορες τεχνολογίες (45nm, 32nm, 22,nm, 16nm) που ακόμα δεν είμαστε σε θέση να κατασκευάσουμε. Δηλαδή η ύπαρξη των standard cells σε τεχνολογίες νεότερες, μας αρκεί για να μελετήσουμε διάφορες αρχιτεκτονικές σχετικά με την κατανάλωση την ταχύτητα και τον χώρο που καταλαμβάνουν.

Η ελάχιστη δομική μονάδα δηλαδή είναι το cell. Το cell είναι δομική μονάδα συγκεκριμένων διαστάσεων, που περιέχει κάποιο συνδυαστικό ή ακολουθιακό κύκλωμα. Υλοποιεί δηλαδή κάποια ή κάποιες λογικές πράξεις.

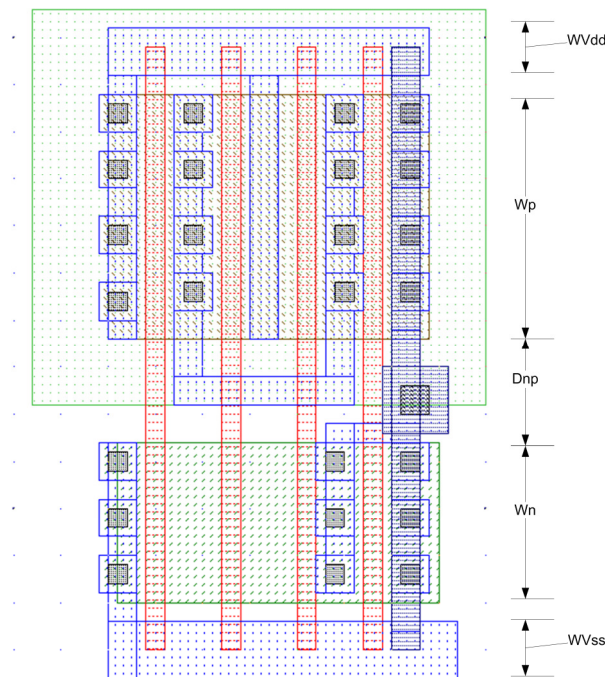


Εικόνα 1.3.3 Απεικόνιση ενός standard cell (inverter και πύλη and) σε απλή κάτοψη και στις φάσεις ανάπτυξής του.

Η οργάνωση των cells γίνεται σε βιβλιοθήκες, οι οποίες περιέχουν σύνολα τέτοιων μονάδων. Ο αριθμός των κυττάρων που θα περιέχει η βιβλιοθήκη δεν είναι κάτι που ορίζεται ρητά, αλλά εξαρτάται από τη λειτουργικότητα, το drive strength, καθώς και αν προσανατολιζόμαστε για designs που θα καταλαμβάνουν μικρότερη επιφάνεια ή θα είναι ταχύτερα. Θεωρητικά μπορεί για παράδειγμα να γίνει σύνθεση οποιουδήποτε ψηφιακού συστήματος με τη χρήση μιας πύλης NAND. Όσο πιο πολλά είναι τα διαθέσιμα κελιά, τόσο λιγότερος είναι ο αριθμός των χρησιμοποιούμενων για την υλοποίηση ενός design, ειδικά όταν οι περιορισμοί χρόνισμού είναι αυστηροί.

Κατά συνέπεια θα χρειαστεί λιγότερος χώρος για την κατασκευή του chip.

Για λόγους ευκολίας κατά την σχεδίαση και την κατασκευή, κρατάμε σταθερό το ύψος όλων των cells, ενώ το πλάτος μεταβάλλεται ανάλογα με την λειτουργία του κυττάρου η οποία καθορίζει και την πολυπλοκότητα του κυκλώματος, άρα και τις διαστάσεις του. Δηλαδή τον αριθμό των τρανζίστορ που θα δομεί εσωτερικά κάθε cell. Λόγω της τεχνολογίας CMOS που χρησιμοποιείται, κάθε standard cell αποτελείται από μια σειρά  $p$  τρανζίστορ και μια σειρά  $n$  τρανζίστορ με μέγιστο πλάτος  $W_n$  και  $W_p$  αντίστοιχα. Το σχήμα που ακολουθεί περιγράφει ένα τυποποιημένο κύτταρο CMOS και τις χαρακτηριστικές διαστάσεις του.



Εικόνα 1.3.4 Layout ενός standard cell. Εδώ διακρίνονται κάποιες χαρακτηριστικές διαστάσεις του.

## 1.4. Περιγραφές των cells

Το ζητούμενο εδώ είναι πως πρέπει να περιγραφεί το κάθε cell, ώστε να μπορέσει να χρησιμοποιηθεί από τα εργαλεία σχεδίασης. Οι περιγραφές αυτές, όπως είναι αναμενόμενο, θα πρέπει να πληροφορούν για την απόδοση, την λειτουργικότητα καθώς και για τις φυσικές διαστάσεις του και τις διασυνδέσεις του.

- Περιγραφή χαρακτηριστικών επιδόσεων, όπως κατανάλωση και χρονισμός
- Περιγραφή λειτουργικότητας
- Περιγραφή γεωμετρικών διαστάσεων και διασυνδέσεων

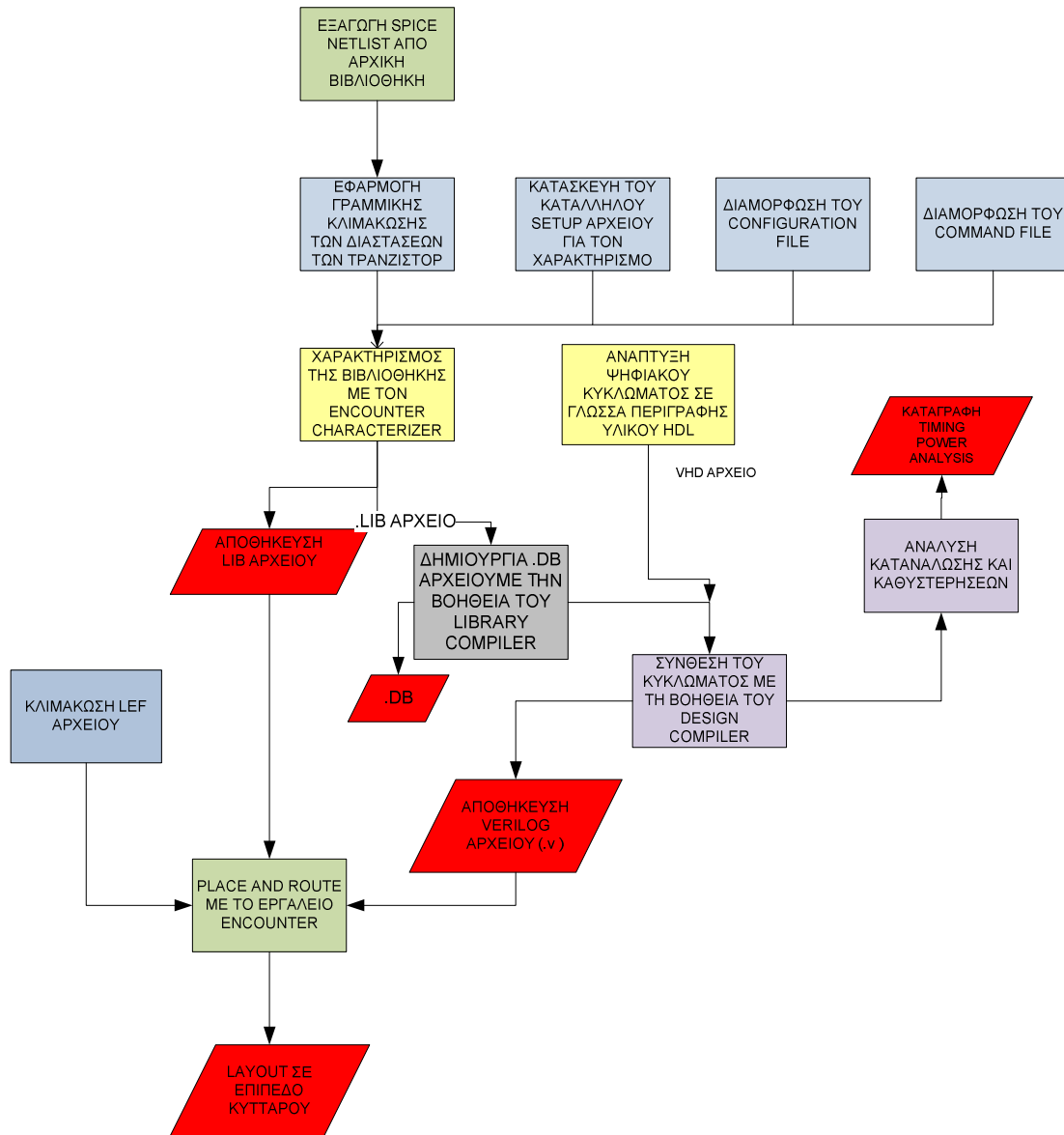


Σε κάθε μία από τις πιο πάνω περιγραφές αντιστοιχεί ένα αρχείο κατάλληλα δομημένο. Για την περιγραφή των χαρακτηριστικών επιδόσεων (κατανάλωση – χρονισμός) δομείται βάσει προδιαγραφής το .LIB αρχείο, το οποίο φέρει την απαραίτητη πληροφορία κατανάλωσης και χρονισμού για κάθε cell. Η λειτουργικότητα περιγράφεται σε ένα Verilog αρχείο, όπου ταυτόχρονα υπάρχει πληροφορία για ελάχιστη, τυπική και μέγιστη καθυστέρηση κάθε cell, για συγκεκριμένη χωρητικότητα φορτίου και slew σήματος εισόδου. Τέλος η περιγραφή των γεωμετρικών διαστάσεων κάθε cell και οι κανόνες σχεδίασης που αφορούν στα interconnects για τη συγκεκριμένη τεχνολογία, υπάρχει καταγεγραμμένη σε ένα αρχείο .LEF. Η γεωμετρία των cells θα μπορούσε να δίνεται και σε ανεξάρτητο αρχείο, διαφορετικό από αυτό που ορίζει τις διαστάσεις των διασυνδέσεων.

## 1.5. Στόχος της εργασίας

Για να σχεδιάζουμε συστήματα σε τεχνολογίες που δεν έχουν ακόμα αναπτυχτεί πλήρως, ή που δεν έχουν γίνει παραγωγικό στάνταρ, χρειαζόμαστε σήμερα βιβλιοθήκες που να προσομοιώνουν τις πραγματικές βιβλιοθήκες που θα εμφανισθούν στο μέλλον. Έτσι θα είναι δυνατή η έγκαιρη ανάπτυξη των απαραίτητων αρχιτεκτονικών και συστημάτων. Όταν μια νέα τεχνολογία είναι έτοιμη προς χρήση θα πρέπει και οι αντίστοιχες αρχιτεκτονικές συστημάτων να είναι έτοιμες προς κατασκευή. Ο μόνος τρόπος να παραλληλίσουμε την ανάπτυξη της τεχνολογίας κατασκευής με την ανάπτυξη των αρχιτεκτονικών και εργαλείων σχεδίασης είναι με τη χρήση βιβλιοθηκών που να “προβλέπουν” τη συμπεριφορά των τρανζίστορ και των cells σε τεχνολογίες που δεν έχουν αναπτυχθεί ακόμη πλήρως. Επομένως στόχος εδώ είναι η υλοποίηση βιβλιοθηκών standard cells σε deep-deep submicron τεχνολογίες που δεν έχουν ακόμα χρησιμοποιηθεί για την κατασκευή ολοκληρωμένων κυκλωμάτων, και μάλιστα με αυτόματο τρόπο.

Αν και δεν έχουν εξετασθεί όλες οι πλευρές, που εμπλέκονται στην εργασία, θα βοηθούσε πολύ στην κατανόηση των βημάτων, αν δοθεί σ’ αυτό το σημείο, το περίγραμμα της εργασίας.



Εικόνα 1.5.1 Περίγραμμα των βημάτων της εργασίας συμπεριλαμβανομένων και των εργαλείων και των αρχείων που χρησιμοποιούνται σε κάθε βήμα.

Όπως παρατηρούμε στο σχήμα λοιπόν, το έργο που πρέπει να γίνει, αποτελείται από:

- κάποιες διαδικασίες κατασκευής απαραίτητων αρχείων, που χρειάζονται οι CAD εφαρμογές και

- από κάποια tests των παραγόμενων αυτών αρχείων για την επαλήθευση της σωστής λειτουργίας των εργαλείων.

Στη γενικότερη έννοια του χαρακτηρισμού, περιλαμβάνονται στο σύνολό τους οι διαδικασίες παραγωγής των απαραίτητων αυτών αρχείων (πχ. LIB, .db, LEF). Οι έλεγχοι και οι δοκιμές αποτελούν επίσης μέρος του έργου που πρέπει να γίνει με τη βοήθεια αντίστοιχων CAD εργαλείων. Βάσει της προηγούμενης πρότασης, για το μέρος του χαρακτηρισμού, παρατηρούμε πως πρέπει να παραχθούν τα αρχεία LIB, .db και LEF. Ενώ για το μέρος της επαλήθευσης, πρέπει να αναπτύξουμε τα κατάλληλα designs, όπου με τη βοήθεια των κατάλληλων εργαλείων, αλλά και των παραπάνω αρχείων που θα δώσει ο χαρακτηρισμός, θα επαληθεύσουμε την ορθότητα του αποτελέσματος. Θα δοκιμαστεί δηλαδή η δυνατότητα σύνθεσης των designs, η δυνατότητα εξαγωγής αξιόπιστων αποτελεσμάτων κατανάλωσης και χρονισμού και η δυνατότητα εξαγωγής του layout των διασυνδέσεων των designs σε επίπεδο cells. Πριν γίνει οτιδήποτε όμως, όπως φαίνεται και στο σχήμα, πρέπει να εφαρμόσουμε κλιμάκωση (scaling) σε δύο στάδια του έργου. Αυτό πρέπει να γίνει κατά την προετοιμασία του spice netlist των cells και κατά την τροποποίηση του LEF (Layout, Exchange, Format) αρχείου των cells. Για τα δύο αυτά αρχεία, που θα δούμε λεπτομέρειες στο αντίστοιχο κεφάλαιο, θα πούμε μόνο πως το πρώτο είναι ένα αρχείο που περιγράφει σε χαμηλό επίπεδο τις διασυνδέσεις των τρανζίστορ και τις διαστάσεις τους, ενώ το δεύτερο είναι ένα αρχείο που μεταφέρει πληροφορία, για τις γεωμετρικές διαστάσεις και τα ηλεκτρικά χαρακτηριστικά των αγωγών διασύνδεσης και των cells. Τα υπόλοιπα αρχεία που εμφανίζονται στην είσοδο του characterizer (setup, elccfg κλπ), είναι κάποια απαραίτητα για το εργαλείο αρχεία, που περιέχουν δηλώσεις, ορισμούς και εντολές. Καταλήγοντας βλέπουμε πως τα εργαλεία που θα χρειαστούμε για την ολοκλήρωση του έργου, είναι:

- O library characterizer
- O library compiler
- O design compiler
- Το εργαλείο placement and routing



# Κεφάλαιο 2

## Design flow

### 2.1. Βήματα που ακολουθούνται κατά την ψηφιακή σχεδίαση

Όπως αναφέρθηκε και στην §1.3 παραπάνω, κατά την ψηφιακή σχεδίαση υπάρχει ιεράρχηση των διαδικασιών σε διάφορα επίπεδα (abstract layers). Στην πράξη για να ακολουθηθεί αυτό το μοντέλο, πρέπει να υπάρξουν τα κατάλληλα εργαλεία σχεδίασης, τα οποία έχουν αναπτυχθεί βάσει αυτής της λογικής. Το πιο κάτω σχήμα (Εικόνα 2.1.1) περιγράφει συνοπτικά με έναν αλγοριθμικό τρόπο τα βήματα που ακολουθούνται για τη σχεδίαση. Πριν ξεκινήσει η περιγραφή των βημάτων σχεδίασης, πρέπει να τονισθεί πως τα αρχεία που βρίσκονται στο κίτρινο πλαίσιο, είναι εκείνα τα αρχεία που **πρέπει απαραίτητως να υλοποιήσουμε**, ώστε να μπορούμε να εκτελεστούν τα βήματα αυτά από τα αντίστοιχα εργαλεία.

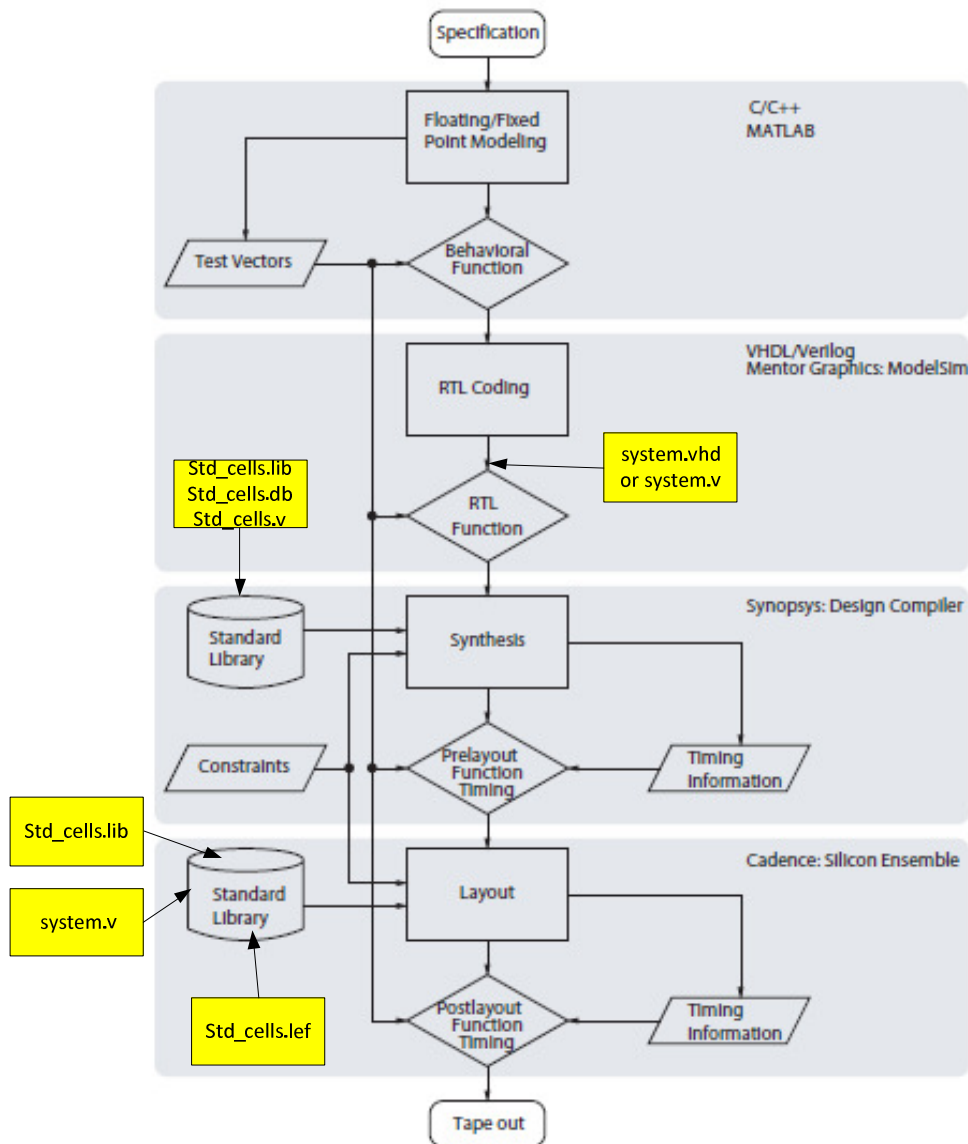
Περιγράφοντας έτσι τα βήματα που βλέπουμε στην Εικόνα 2.1.1, σαν πρώτο βήμα διακρίνουμε την σύλληψη της ιδέας από την οποία ξεκινάει και η σχεδίαση. Η ιδέα αυτή ξεκινά από τις προδιαγραφές που έχουν ορισθεί, και η επεξεργασία της οδηγεί στην ανάπτυξη κοινά αποδεκτών από την ομάδα σχεδίασης λειτουργικών μπλοκ. Τα μπλοκ αυτά συνθέτουν ένα μπλοκ διάγραμμα που περιγράφει τη λειτουργία του συστήματος.

Το επόμενο βήμα στη διαδικασία είναι η περιγραφή συμπεριφοράς ή της δομής του συστήματος (behavioral, structural, or data flow description) με τη βοήθεια μιας γλώσσας περιγραφής υλικού HDL. Λεπτομέρειες που αφορούν τα επιμέρους κυκλώματα ή τα στοιχεία που συνθέτουν το σύστημα, δεν μας ενδιαφέρουν σ' αυτό το στάδιο. Οι λεπτομέρειες αυτές υποκρύπτονται βάσει του επιπέδου αφαίρεσης που προσφέρεται σε αυτό το βήμα. Μοντελοποιείται μόνο η συμπεριφορά του συστήματος, η οποία είναι απαραίτητη στο παρόν στάδιο σχεδίασης. Η έξοδος (output) που δίνει το βήμα αυτό είναι ένα system.vhd ή ένα system.v αρχείο (όπου system είναι το όνομα του design που έχουμε δημιουργήσει), το οποίο είναι απαραίτητο όπως βλέπουμε για τα επόμενα βήματα.

Ακολουθεί η σύνθεση, ώστε συνδυάζοντας στιγμιότυπα βασικών δομικών μονάδων της βασικής βιβλιοθήκης με τα cells, να προκύψει το τελικό σύστημα. Έτσι η δομική περιγραφή κατεβαίνει σε πιο χαμηλό επίπεδο αφαίρεσης, προσθέτοντας περισσότερη λεπτομέρεια στη λειτουργία του συστήματος. Απαραίτητα αρχεία για τη σύνθεση είναι το αρχείο HDL που προέκυψε στο προηγούμενο βήμα, και το αρχείο (std\_cells.db) το οποίο παράγεται από μεταγλώττιση του αρχείου (std\_cells.lib). Ειδικά το τελευταίο αρχείο είναι αυτό που θα προκύψει από τη διαδικασία χαρακτηρισμού που θα αναλύσουμε πιο κάτω. Για το κάθε μπλοκ που δομεί το σύστημα, γίνεται εξομοίωση της λειτουργίας του χωριστά, και αφού ικανοποιηθούν οι προδιαγραφές, γίνεται εξομοίωση του συστήματος συνολικά με όλα τα μπλοκ συνδεδεμένα, συγκρίνοντας τα αποτελέσματα με την αναμενόμενη λειτουργία του συστήματος. Επιτυχής εξομοίωση σημαίνει πως το κύκλωμα είναι απολύτως συμβατό με την περιγραφή συμπεριφοράς που προηγήθηκε.

Η σχηματική περιγραφή είναι κι αυτή μια δομική περιγραφή που εστιάζει στην τοπολογία του συστήματος. Η περιγραφή αυτή μπορεί να γίνει αυστηρότερη δίνοντας περισσότερες κυκλωματικές λεπτομέρειες αν αυτή γίνεται σε χαμηλότερο επίπεδο αφαίρεσης. Συνήθως η περιγραφή αυτή είναι σε γραφικό περιβάλλον, όπου με τη βοήθεια των CAD εργαλείων μπορεί να γίνει η συνδεσμολογία των βασικών δομικών μονάδων (standard cells). Κατεβαίνοντας σε χαμηλότερο επίπεδο αφαίρεσης προσεγγίζουμε το φυσικό επίπεδο περισσότερο. Έτσι στην επόμενη φάση ακολουθεί το floor planning του chip όπου στη συνέχεια γίνεται τοποθέτηση και διασύνδεση όλων των βασικών δομικών μονάδων (standard cells) που συνθέτουν το σύστημα. Τοποθέτηση είναι η διαδικασία κατά την οποία όλες οι μονάδες που συνθέτουν το σύστημά μας, μπαίνουν γειτονικά ή μία στην άλλη με κριτήριο την ελαχιστοποίηση της επιφάνειας ή του χρόνου κύκλου. Για τον σκοπό αυτό υπάρχουν διάφοροι αλγόριθμοι όπως Min-cut και thermal annealing. Διασύνδεση είναι η διαδικασία σύνδεσης των κυττάρων με κατάλληλες καλωδιώσεις μεταξύ τους με σκοπό την αποκατάσταση του αρχικού σχεδίου. Ο router λαμβάνει ως δεδομένα την θέση μιας μονάδας και έναν κατάλογο από συνδέσεις. Απαραίτητο αρχείο εδώ είναι το std\_cells.lef, το οποίο παρέχει πληροφορία για τα προηγούμενα, στο αντίστοιχο εργαλείο CAD. Η πληροφορία αυτή περιγράφει κάποια χαρακτηριστικά (χωρητικότητα και αντίσταση ανά τετράγωνο) τις γεωμετρικές διαστάσεις των cells καθώς και λεπτομερή περιγραφή των ιδιοτήτων των αγωγών διασύνδεσης.

Καταλήγοντας, σ' αυτό το στάδιο μπορεί να γίνει εξαγωγή των παρασιτικών του κυκλώματος όπως χωρητικότητας / μονάδα μήκους, ή αντίστασης / μονάδα μήκους. Μέχρι εδώ έχει τελειώσει η μελέτη της αρχιτεκτονικής, της δομής και της λειτουργίας του κυκλώματος. Απομένουν λίγα ακόμα βήματα για να “κατέβουμε” στο τελευταίο επίπεδο της ιεραρχικής σχεδίασης, που δεν είναι άλλο από το φυσικό επίπεδο. Εδώ περιγράφεται πια το σύστημα σε επίπεδο τρανζίστορ. Τα αρχεία που παράγονται εδώ είναι τύπου GDS II ή CIF και είναι εκείνα που δίνονται στις μονάδες παραγωγής chip. Κατά την παρούσα φάση, γίνεται πάντα έλεγχος τήρησης των κανόνων σχεδίασης **Design Rule Check (DRC)** και επιπλέον **LAYOUT vs SCHEMATIC (LVS)** έλεγχος. με το DRC ελέγχεται αν οι κανόνες σχεδίασης που καθορίζονται από την τεχνολογία επεξεργασίας και κατασκευής του chip, δεν παραβιάζονται, ενώ με τον τελευταίο έλεγχο πιστοποιείται ότι το layout είναι σε συμβατότητα με τη σχηματική δομή.



Εικόνα 2.1.1 Παρουσίαση των βημάτων ψηφιακής σχεδίασης (ASIC)<sup>2</sup>.

Αυτό που συμπεραίνουμε απ' την παρουσίαση αυτή, είναι πως για να εξασφαλιστεί η ορθή λειτουργία των συστημάτων που σχεδιάζουμε, σύμφωνα με τις προδιαγραφές και τους περιορισμούς που έχουν τεθεί εξ' αρχής, απαιτείται η συνδυασμένη χρήση διαφορετικών εργαλείων σε όλο το design flow της ψηφιακής σχεδίασης. Τα εργαλεία αυτά απαιτούν συγκεκριμένες πληροφορίες σε διάφορα formats για κάθε κύτταρο της τυποποιημένης βιβλιοθήκης που χρησιμοποιούμε όπως είδαμε και στην περιγραφή των βημάτων σχεδίασης. Τα formats που αναφέρονται στη συνέχεια

<sup>2</sup> Digital ASIC Design Flow. Digital ASIC Group October 2005 Lund University (11)



αφορούν τα εργαλεία σχεδίασης της Cadence αλλά παρόμοια ισχύουν και για τις πληροφορίες που απαιτούν εργαλεία σχεδίασης άλλων εταιρειών.

- Physical Layout (gdsII, Virtuoso Layout Editor) πρέπει να ακολουθούνται συγκεκριμένα standards όπως για παράδειγμα σταθερό ύψος κελιών.
- Logical View (verilog description or TLF or LIB) Η Verilog απαιτείται για δυναμική εξομοίωση. Στη τοποθέτηση και τη δρομολόγηση των κελιών συνήθως χρησιμοποιείται TLF ή LIB format σε συνδυασμό με το αρχείο LEF. Η Verilog περιγραφή πρέπει να υποστηρίζει κατά προτίμηση back annotation για την πληροφορία που αφορά τις καθυστερήσεις και τον χρονισμό.
- Abstract View (Cadence Abstract Generator, LEF). Τα LEF αρχεία περιέχουν πληροφορία για τις διαστάσεις των κελιών, για παρασιτικές χωρητικότητες, αντιστάσεις και αυτεπαγωγές, για τις διαστάσεις των καλωδίων στα διάφορα επίπεδα μετάλλου καθώς και για την τεχνολογία αλλά και για τον τρόπο διασύνδεσης τόσο των κελιών όσο και των επιπέδων μετάλλου.

Timing, power and parasitics (TLF or LIB). Τα παρασιτικά των τρανζίστορς και των συνδέσεων, προκύπτουν με την βοήθεια των εργαλείων Cadence. Στη συνέχεια με την παραγωγή Spice ή Spectre netlist μπορούν να γίνουν ακριβείς εξομοιώσεις για την καθυστέρηση και την κατανάλωση. Τα αρχεία τύπου LIB περιέχουν επίσης, πληροφορία που αφορούν την επεξεργασία και τις μεταβολές της θερμοκρασίας και της τάσης τροφοδοσίας καθώς και πληροφορία για την λογική συμπεριφορά των κελιών.

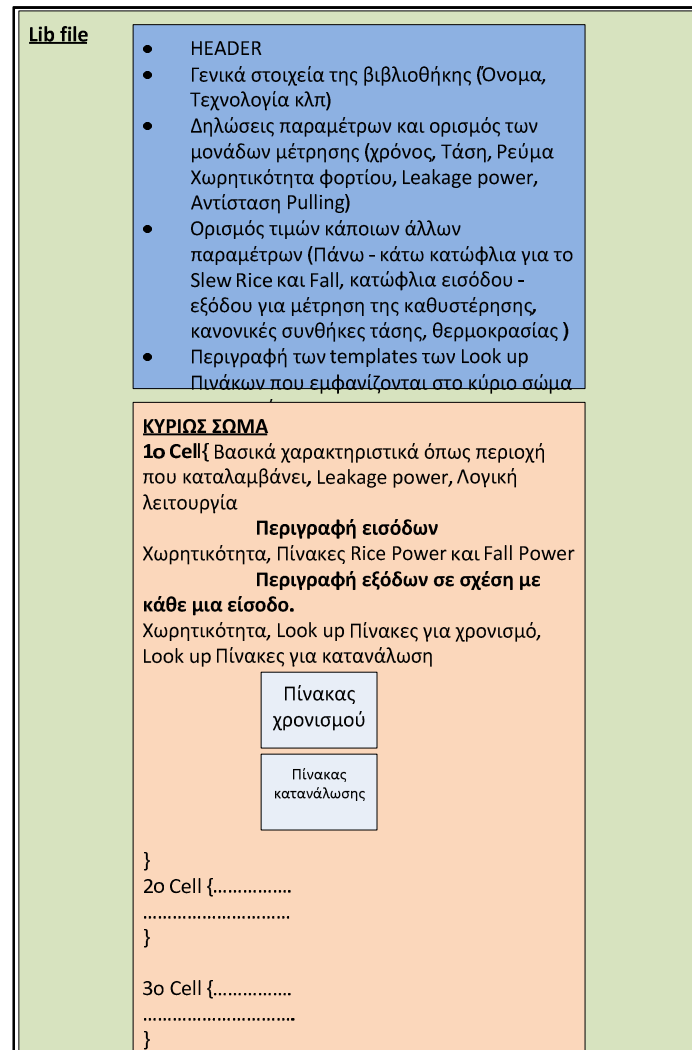
## 2.2. Περιγραφή των απαραίτητων αρχείων της ροής σχεδίασης (design flow)

Ας δούμε εδώ αναλυτικότερα την πληροφορία που φέρει το καθένα από αυτά τα αρχεία και τον τρόπο που είναι δομημένη.

### 2.2.1. Αρχείο Lib

Τα αρχεία LIB (Liberty format) χρησιμοποιούνται από τα εργαλεία σύνθεσης και μεταφέρουν πληροφορία για καθυστέρηση, κατανάλωση και λειτουργία των cells. Για να μπορούμε να συνθέτουμε σωστά τα ψηφιακά κυκλώματα που θα προκύψουν από

την βιβλιοθήκη τυποποιημένων κυττάρων, χρειαζόμαστε πληροφορία για την κατανάλωση και τον χρονισμό του κάθε κυττάρου όπως αναφέρθηκε και πριν. Αυτή η πληροφορία προκύπτει ως έξοδος των εργαλείων χαρακτηρισμού, όπως θα αναλύσουμε πιο κάτω, και δομείται καταλλήλως για χρησιμοποίηση από τα εργαλεία εξομοίωσης. Η μορφή του φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 2.2.1.1 Τυπική δομή του lib αρχείου.

Διακρίνουμε δύο βασικές περιοχές. Η πρώτη περιοχή που φαίνεται στο σχήμα ως Header, ορίζει ουσιαστικά την βιβλιοθήκη, δίνοντας γενική πληροφορία τόσο για τις συνθήκες εξομοίωσης, όσο και για την σωστή ανάγνωση και ερμηνεία των αποτελεσμάτων που περιλαμβάνει η δεύτερη περιοχή. Έτσι στην πρώτη περιοχή θα

βρούμε στοιχεία για την ταυτότητα της βιβλιοθήκης όπως το όνομα, την υποστηριζόμενη τεχνολογία, το μοντέλο καθυστέρησης που χρησιμοποιεί κλπ. Θα βρούμε στοιχεία για δηλώσεις παραμέτρων και για τον ορισμό των μονάδων μέτρησης (χρόνος, Τάση, Ρεύμα, χωρητικότητα φορτίου, leakage power, αντίσταση pulling), για τον ορισμό τιμών κάποιων άλλων παραμέτρων (Πάνω - κάτω κατώφλια για το Slew Rise και Fall, κατώφλια εισόδου - εξόδου για μέτρηση της καθυστέρησης, κανονικές συνθήκες τάσης, θερμοκρασίας) και περιγραφές των templates των look up πινάκων που εμφανίζονται στο υπόλοιπο αρχείο. Προσπαθώντας να δούμε με λίγο μεγαλύτερη λεπτομέρεια τα προηγούμενα παρατίθεται ένα μικρό δείγμα ενός LIB αρχείου που περιέχει ένα μόνο cell αναστροφέα. Ας ρίξουμε μια ματιά στη δομή του LIB αρχείου, επικεντρώνοντας σε κάποια βασικά σημεία<sup>3</sup>.

```

library (std_cells_45nm ) {
/*=== Library Level Attributes ===*/
/* technology family */
technology( cmos );
/* delay model */
delay_model : table_lookup;
/* documentation details */
/* library units */
voltage_unit : "1V";
current_unit : "1mA";
pulling_resistance_unit : "1kohm";
time_unit : "1ns";
/* reresetn capacitance in terms of the standard load
unit of an inverter */

capacitive_load_unit(1,pf);
/* nominal operating condition */
nom_process : 1.0;
nom_temperature : 25.0;
nom_voltage : 1.0;
/* Delay threshold points */
output_threshold_pct_fall : 50 ;
input_threshold_pct_fall : 50 ;
output_threshold_pct_rise : 50 ;
input_threshold_pct_rise : 50 ;
/* Slew threshold points */
slew_derate_from_library : 1.0;
slew_lower_threshold_pct_fall : 20 ;
slew_upper_threshold_pct_fall : 80 ;

```

<sup>3</sup> Από paper Characterizing Cells and Writing a Technology Library File by Don Wichern, February 1-2005 (12)

```
slew_lower_threshold_pct_rise : 20 ;
slew_upper_threshold_pct_rise : 80 ;
/* Cell swapping criteria */
in_place_swap_mode : match_footprint;
```

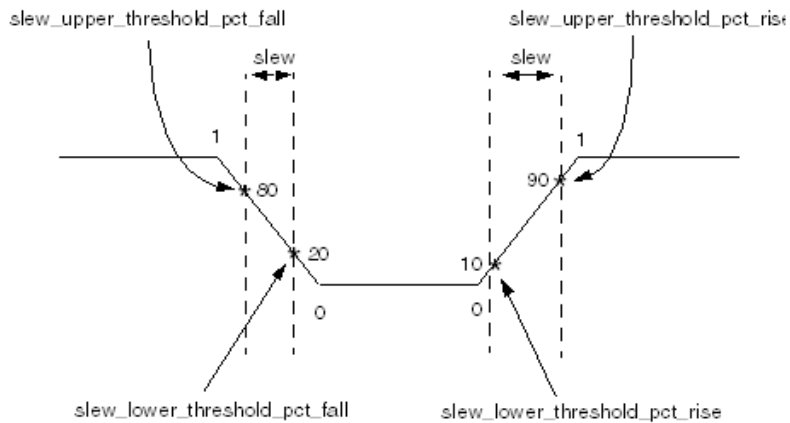
**Library Name:** Το όνομα της βιβλιοθήκης των τυποποιημένων κυττάρων είναι η τιμή αυτής της παραμέτρου.

**Technology Family:** Εδώ δηλώνεται η τεχνολογία που χρησιμοποιούμε στη βιβλιοθήκη.

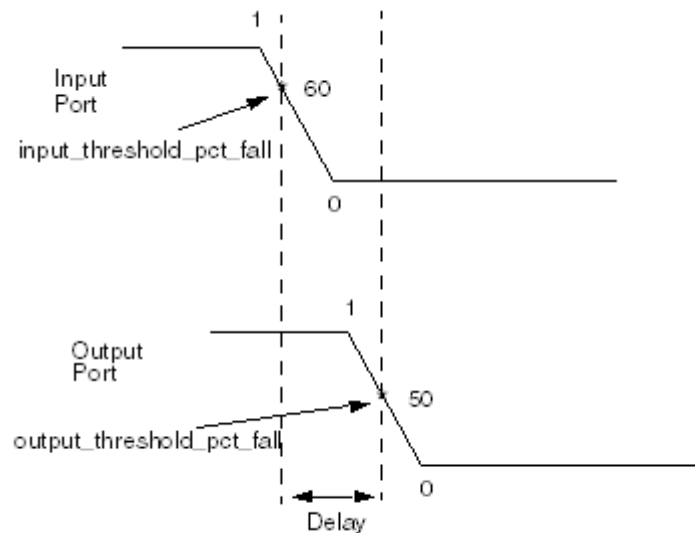
**Delay Model:** Εδώ δηλώνεται το μοντέλο υπολογισμού της καθυστέρησης, που είναι lookup table. Οι εναλλακτικές επιλογές είναι generic cmos, lookup table, piecewise cmos, cmos2, και dcm.

**Capacitive Unit Load:** Η μονάδα μέτρησης χωρητικού φορτίου, επιτρέπει τον προσδιορισμό των χωρητικοτήτων του κυκλώματος συνήθως σε σχέση με το τυποποιημένο χωρητικό φορτίο, που είναι αυτό που συνθέτουν τέσσερις αναστροφείς. Η χωρητικότητα κάθε κυττάρου στην περίπτωση αυτή, προσδιορίζεται σε σχέση μ' αυτό το φορτίο.

**Rise and Fall Timing Measurements:** Το σημείο του κατωφλίου καθυστέρησης προσδιορίζει το ποσοστό του χρόνου ανόδου και καθόδου που λαμβάνεται υπόψη για τις μετρήσεις καθυστέρησης μετάδοσης. Το σημείο του κατωφλίου slew προσδιορίζει το ποσοστό του χρόνου ανόδου και καθόδου που χρησιμοποιείται για τις μετρήσεις της αντίστασης καθυστέρησης. Η καθυστέρηση διάδοσης όπως θα δούμε πιο κάτω, μετράται συνήθως ως ο χρόνος μεταξύ του σημείου που αντιστοιχεί στο 50% του χρόνου ανόδου του σήματος εισόδου και του 50% του χρόνου ανόδου του σήματος εξόδου (αυτές οι τιμές θα χρησιμοποιηθούν και στις νέες βιβλιοθήκες). Για την αποφυγή πιθανής τιμής καθυστέρησης με αρνητική τιμή, εξαιτίας της αργής μεταβολής των εξόδων, γίνεται επιλογή σημείου κατωφλίου 30% του χρόνου ανόδου για το σήμα εισόδου και 70% για το σήμα εξόδου. Οι μετρήσεις των καθυστερήσεων διάδοσης, γίνονται χωρίς οδηγούμενο φορτίο (επειδή δεν γίνεται εξαγωγή netlist με παρασιτικές στην παρούσα εργασία, αλλά scaling μιας ήδη υπάρχουσας). Απ' την άλλη πλευρά, η μέτρηση του slew του σήματος εισόδου, γίνεται από το 20% του σήματος στο 80%, τόσο για την άνοδο, όσο και για την πτώση του παλμού. Σ' αυτό το σημείο κρίνεται απαραίτητο να αποδοθεί σχηματικά η πιο πάνω περιγραφή, μέσω των σχημάτων που ακολουθούν.



Εικόνα 2.2.1.2 Γραφική παράσταση μέτρησης του Slew<sup>4</sup>.



Εικόνα 2.2.1.3 Γραφική παράσταση μέτρησης της καθυστέρησης<sup>4</sup>.

Τα εργαλεία σύνθεσης, συνδυάζουν αυτούς τους χρόνους, με τις χωρητικότητες (εκφρασμένες συνήθως σε τυποποιημένο φορτίο 4x αναστροφή)

**Nominal Operating Condition:** Εδώ περιγράφονται οι κανονικές συνθήκες κάτω από τις οποίες γίνεται ο χαρακτηρισμός. Αυτές αφορούν τη θερμοκρασία την τάση λειτουργίας και την επεξεργασία.

<sup>4</sup> Liberty User Guide, Vol. 1 Version 2009.06 (13)

```

/* lu_table_templates */

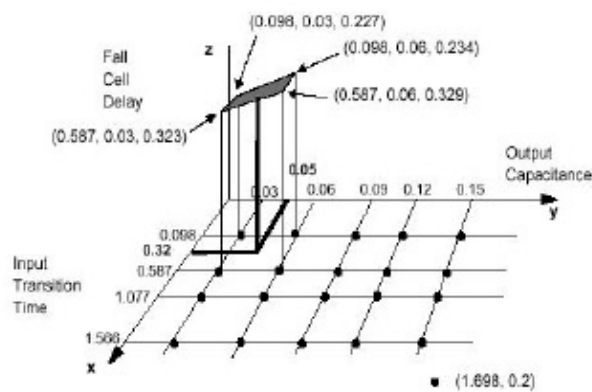
lu_table_template(delay_template_7x7) {
    variable_1 : input_net_transition;
    variable_2 : total_output_net_capacitance;

    index_1 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
    index_2 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
}

power_lut_template(energy_template_7x7) {
    variable_1 : input_transition_time;
    variable_2 : total_output_net_capacitance;
    index_1 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
    index_2 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
}

```

**Lookup\_table templates:** Εδώ περιγράφεται ο τρόπος ανάγνωσης των πινάκων, που περιέχουν τιμές καθυστερήσεων και κατανάλωσης. Για τις καθυστερήσεις και τις καταναλώσεις, σύμφωνα με το μοντέλο αυτό (CMOS Non-linear Delay Model), ο προσδιορισμός της τιμής που θα διαβαστεί γίνεται με τη βοήθεια δύο διανυσμάτων. Το ένα διάνυσμα έχει καταχωρημένες τις τιμές του φορτίου εξόδου Cload, ενώ το άλλο έχει καταχωρημένες τιμές Slew σήματος εισόδου. Η λειτουργία του μοντέλου παριστάνεται στο επόμενο σχήμα.



Εικόνα 2.2.1.4 Τρόπος υπολογισμού του delay (CMOS Non-linear Delay Model) με lookup table<sup>5</sup>.

<sup>5</sup> Περιγραφή από Synopsys; Library Compiler User Guide

Σύμφωνα με το σχήμα, από τους σχετικούς πίνακες της καθυστέρησης, ή της κατανάλωσης, ο συνδυασμός ενός slew και ενός Cload, από τα αντίστοιχα διανύσματα, δίνει μια γραμμή και μια στήλη των lookup tables.

#### 2.2.1.1 Παρουσίαση αποτελεσμάτων, με lookup table.

	CL1	CL2	CL3	CL4	CL5	CL6	CL7
SLEW1	Τιμή11	Τιμή12	Τιμή13	Τιμή14	Τιμή15	Τιμή16	Τιμή17
SLEW2	Τιμή21	Τιμή22	Τιμή23	Τιμή24	Τιμή25	Τιμή26	Τιμή27
SLEW3	Τιμή31	Τιμή32	Τιμή33	Τιμή34	Τιμή35	Τιμή36	Τιμή37
SLEW4	Τιμή41	Τιμή42	Τιμή43	Τιμή44	Τιμή45	Τιμή46	Τιμή47
SLEW5	Τιμή51	Τιμή52	Τιμή53	Τιμή54	Τιμή55	Τιμή56	Τιμή57
SLEW6	Τιμή61	Τιμή62	Τιμή63	Τιμή64	Τιμή65	Τιμή66	Τιμή67
SLEW7	Τιμή71	Τιμή72	Τιμή73	Τιμή74	Τιμή75	Τιμή76	Τιμή77

Η τομή της γραμμής και της στήλης δίνουν μια τιμή του μετρούμενου μεγέθους. Η τιμή αυτή παριστάνεται στον άξονα Z (βλ. Εικόνα 2.2.1.4). Για τιμές slew και Cload μεταξύ αυτών του πίνακα, γίνεται interpolation, ώστε να αναδειχτεί η μετρούμενη τιμή του μεγέθους.

Οι τιμές των Cload και των Slews υπάρχουν υπό μορφή διανυσμάτων πριν από κάθε τέτοιο πίνακα. Συγκεκριμένα το πρώτο διάνυσμα δέχεται τις τιμές του Slew, ενώ το δεύτερο τις τιμές του Cload.

```
cell (INVX1) {
  area : 10.0;
  cell_footprint : INV;
  pin( A ) {
    direction : input;
    capacitance : 0.00123823;
  }
  pin( Y ) {
    direction : output;
  }
  function : "!A";
  timing() {
    cell_rise(delay_template_7x7) {
      index_1 ("0.0231, 0.0282, 0.048, 0.06, 0.12, 0.48, 0.6216");
      index_2 ("0.00082, 0.0033, 0.00842, 0.01848, 0.03861, 0.0787,
0.18975");
      values ( \
        "0.01846, , , , , , , , , , , , , , , ,
, , , , , , , , , , , , , , , , , , , , , , ,
, , , , , , , , , , , , , , , , , , , , , , ,
0.847567");
    }
  }
}
```





εισόδων και εξόδων τους, δίνοντας ταυτόχρονα και πληροφορία χρονισμού. Η βασική μορφή του Verilog αρχείου της βιβλιοθήκης, απεικονίζεται στην Εικόνα 2.2.2.1. Όπως διακρίνουμε η παρουσίαση των cells είναι πολύ πιο συνοπτική σε σχέση με το LIB αρχείο. Υπάρχει ωστόσο η περιγραφή της διάταξης των εισόδων και εξόδων και η λειτουργία που τις συνδέει, αλλά και οι καθυστερήσεις από την κάθε είσοδο, στην έξοδο. Αυτό έχει σαν αποτέλεσμα τη δυνατότητα της δυναμικής εξομοίωσης από αντίστοιχα εργαλεία. Αυτή είναι και η μοναδική χρήση του συγκεκριμένου αρχείου.

```

Verilog file

module cell1_name (out1, out2, ... , in1, in2, ...);
output out1, out2, ...;
input in1, in2, ...;
Λειτουργία κελιού;
Πληροφορία χρονισμού;
Endmodule

module cell2_name (out1, out2, ... , in1, in2, ...);
output out1, out2, ...;
input in1, in2, ...;
Λειτουργία κελιού;
Πληροφορία χρονισμού;
endmodule
:
:
module celln_name (out1, out2, ... , in1, in2, ...);
output out1, out2, ...;
input in1, in2, ...;
Λειτουργία κελιού;
Πληροφορία χρονισμού;
endmodule

```

Εικόνα 2.2.2.1 Εσωτερική δομή του αρχείου Verilog με τα cells της βιβλιοθήκης.

Για λόγους πληρότητας παρατίθεται σ' αυτό το σημείο το αρχείο Verilog που περιγράφει τον αναστροφέα INVX1.

```

module INVX1 (A, Y);
input A ;
output Y ;

not (Y, A);

```

```

specify
  // delay parameters
  specparam
    tplhl$A$Y = 0.014:0.014:0.014,
    tphlh$A$Y = 0.018:0.018:0.018;

  // path delays
  (A *> Y) = (tphlh$A$Y, tplhl$A$Y);

endspecify

endmodule
`endcelldefine

```

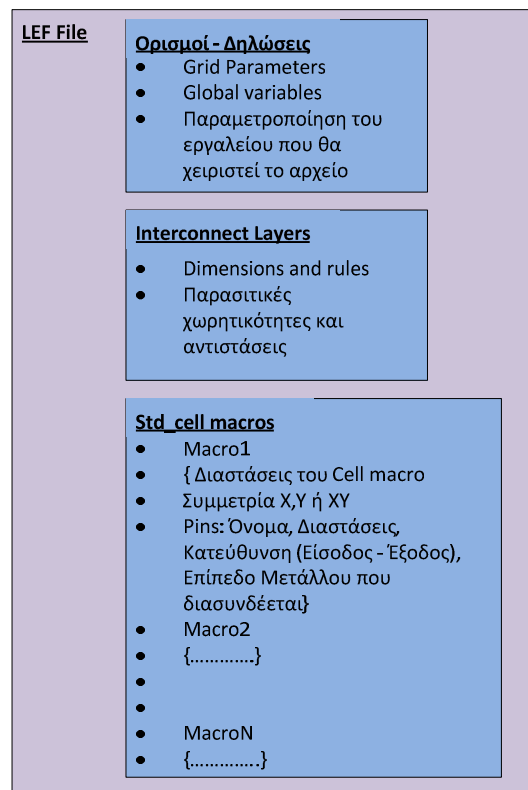
### 2.2.3. Αρχείο LEF

Στις προηγούμενες ενότητες είδαμε, πως η πληροφορία λειτουργίας, χρονισμού και κατανάλωσης, “περνάει” μέσω του αρχείου LIB στα εργαλεία σχεδίασης (EDA). Ακόμα πως με το Verilog αρχείο είναι εφικτή η δυναμική εξομοίωση των cells. Ωστόσο τα εργαλεία τοποθέτησης και διασύνδεσης σε επίπεδο cell, δεν χρειάζονται την λεπτομέρεια που εσωκλείει το LIB αρχείο, ούτε το εσωτερικό layout κάθε κελιού σε επίπεδο τρανζίστορ, που δίνουν τα αρχεία GDS. Χρειάζεται δηλαδή μια αφαιρετική επισκόπηση των κυττάρων της βιβλιοθήκης σε επίπεδο block. Η αφαιρετική αυτή επισκόπηση δίνει πληροφορίες όπως:

- Όνομα του κυττάρου, ορισμός των διαστάσεών του καθώς και προσανατολισμός του κυττάρου.
- Ορισμός των συνόρων που εκτελείται η τοποθέτηση και διασύνδεση των κυττάρων.
- Όνομα ακροδεκτών, επίπεδο μετάλλου όπου συνδέονται, τύπος και κατεύθυνση (είσοδος/έξοδος).
- Παρέχει επίσης πληροφορία για όλα τα καλώδια διασύνδεσης και των vias στο layout.

Αυτές όλες οι πληροφορίες περνούν στα εργαλεία τοποθέτησης και διασύνδεσης μέσω του Library Exchange Format (LEF). Μπορούμε δηλαδή να πούμε πως το LEF αρχείο εσωκλείει πληροφορία για την τεχνολογία μαζί με την αφαιρετική περιγραφή όλων των κελιών.

Προσπαθώντας να παραστήσουμε τη γενική δομή του αρχείου LEF καταλήγουμε στο επόμενο σχήμα.

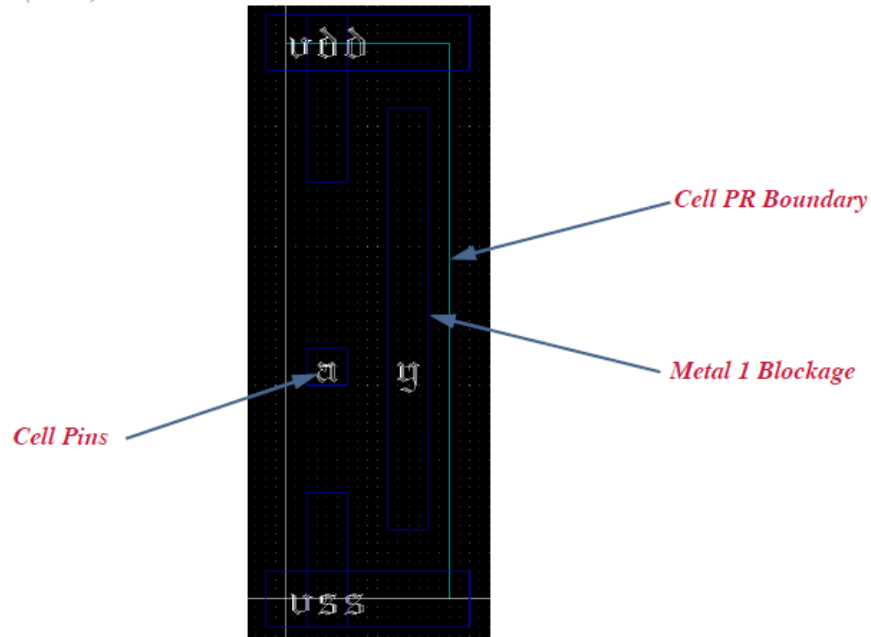


Εικόνα 2.2.3.1 Εσωτερική δομή του αρχείου LEF.

Σύμφωνα με το σχήμα λοιπόν, μπορούμε να διακρίνουμε τρεις ενότητες. Η πρώτη ενότητα περιέχει δηλώσεις και παραμετροποιήσεις που αφορούν το εργαλείο τοποθέτησης και δρομολόγησης. Μια από τις βασικότερες παραμέτρους που ορίζεται εδώ, είναι οι διαστάσεις του Manufacturing Grid. Πρόκειται για ένα πλέγμα “αναφοράς”, όπου η διάσταση των τετραγώνων που το συνθέτουν, ποικίλει ανάλογα με την τεχνολογία για την οποία είναι κατασκευασμένο. Αυτό είναι η βάση πάνω στην οποία “κτίζεται” ολόκληρο το σύστημα και η αρχιτεκτονική, ως προς την τοποθέτηση και τις διασυνδέσεις, σε επίπεδο cells. Περαιτέρω παραμετροποίηση αφορά για παράδειγμα αν οι χαρακτήρες είναι case sensitive, πως παριστάνονται τα busbits, μονάδες μέτρησης κλπ. Η δεύτερη ενότητα δομεί πληροφορία για τα διάφορα επίπεδα ημιαγωγού και διασυνδέσεων, όπως nwell layer, poly layer, metal layers, vias κλπ. Εδώ πρέπει να οριστούν με αυστηρό τρόπο, παράμετροι όπως width, spacing, pitch κλπ. Τελευταία ενότητα, είναι εκείνη που περιέχει τα cell macros και δεν είναι τίποτα περισσότερο από εκείνη την ενότητα, που δομεί καταλλήλως την πληροφορία για τα γεωμετρικά χαρακτηριστικά, και τον τρόπο διασύνδεσης κάθε cell. Για να

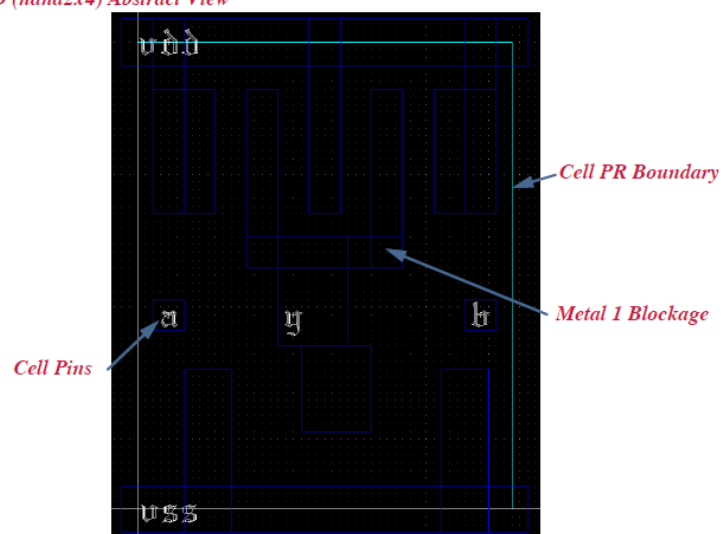
γίνει όμως απόλυτα κατανοητός ο ρόλος και η δομή του LEF αρχείου, δίνεται το σχήμα που ακολουθεί. Εκεί μπορούμε να διακρίνουμε δύο κύτταρα (ενός αναστροφέα και μιας πύλης NAND) σε αφαιρετική επισκόπηση όπως συμβαίνει με το LEF αρχείο. Σε συνέχεια των δύο προηγούμενων σχημάτων, δίνεται η λεπτομερής περιγραφή του αναστροφέα και της πύλης NAND, όπου διακρίνεται το layout με λεπτομέρεια σε επίπεδο τρανζίστορ και είναι εμφανής η διαφορά μεταξύ των δύο layouts.

*Inverter (invx1) Abstract View*

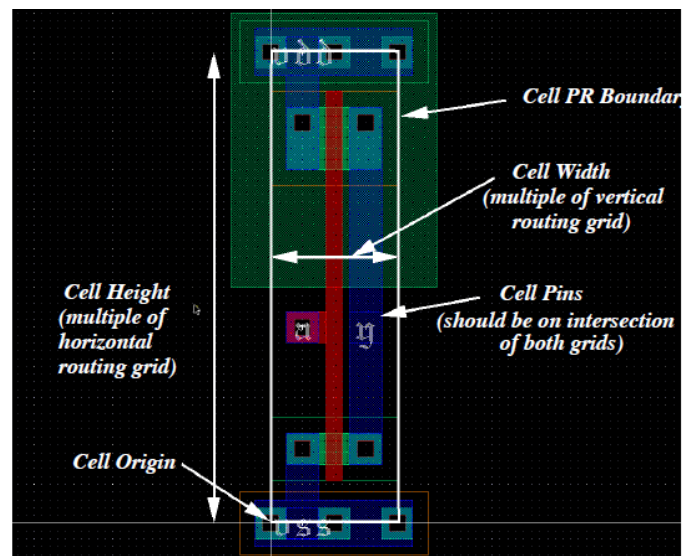


Εικόνα 2.2.3.2 αφαιρετική επισκόπηση ενός κυττάρου αναστροφέα.

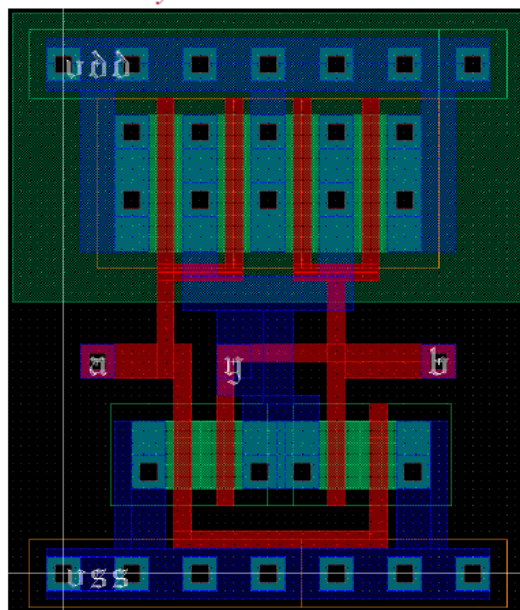
*NAND (nand2x4) Abstract View*



Εικόνα 2.2.3.3 Αφαιρετική επισκόπηση ενός κυττάρου πύλης NAND2X4.



Εικόνα 2.2.3.4 Αναλυτική περιγραφή layout του cell ενός αναστροφέα.

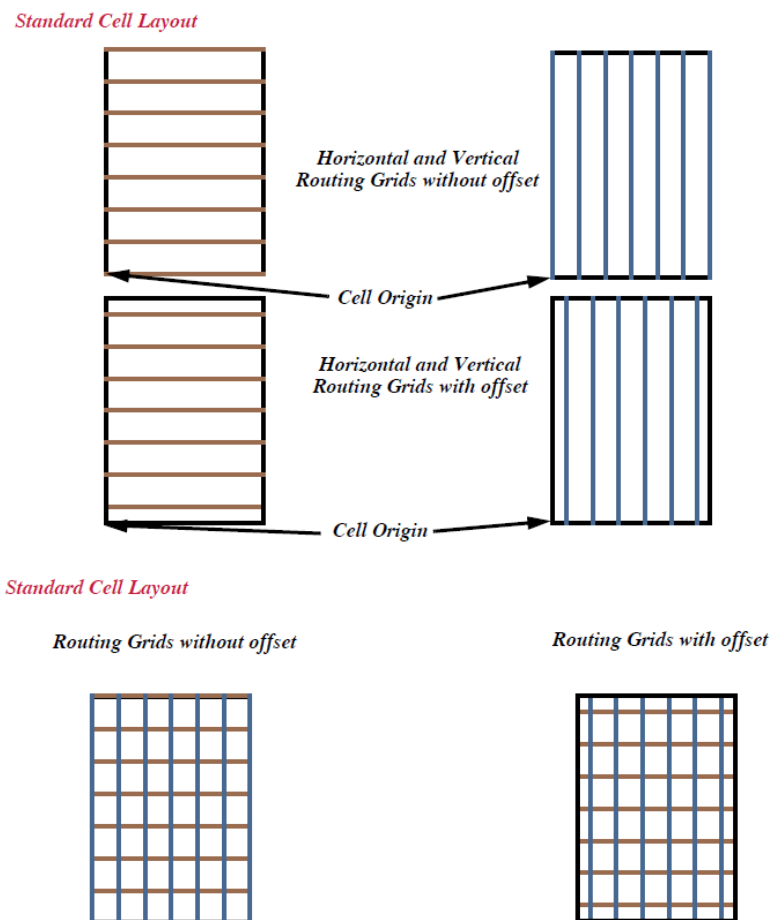


Εικόνα 2.2.3.5 Αναλυτική περιγραφή layout του cell πύλης NAND2x4.

Το LEF αρχείο λοιπόν περιλαμβάνει κανόνες σχεδίασης και διασύνδεσης των μπλοκ των τυποποιημένων cells και σε καμία περίπτωση δεν φέρει πληροφορία για τις εσωτερικές διασυνδέσεις τους (φυσικό επίπεδο).

Ως εδώ έχει γίνει κατανοητός ο ρόλος του LEF αρχείου, στη διαδικασία σχεδίασης. Έτσι εδώ θα πρέπει να εξετάσουμε τις παραμέτρους που συναντάμε, στην εσωτερική δομή του αρχείου.

Βάση όλης της αρχιτεκτονικής που θα δημιουργήσει το εργαλείο, όπως αναφέρθηκε, είναι το πλέγμα διασύνδεσης. Επάνω σ' αυτό τοποθετούνται τα cells. Μέσα στα routing tracks που ορίζει το πλέγμα, πραγματοποιούνται οι οριζόντιες και κάθετες διασυνδέσεις, εναλλάσσοντας τον προσανατολισμό από επίπεδο μετάλλου σε επίπεδο μετάλλου (περιττά επίπεδα οριζόντιος προσανατολισμός και άρτια επίπεδα κάθετος προσανατολισμός). Οι ακροδέκτες των τυποποιημένων cells τοποθετούνται στις διασταυρώσεις των οριζόντιων και κάθετων γραμμών του πλέγματος. Εξαιρέση αποτελούν οι ακροδέκτες στήριξης των γραμμών  $V_{DD}$  και  $GND$ .



Εικόνα 2.2.3.6 Δημιουργία πλέγματος για τοποθέτηση και διασύνδεση.

Το κάθε interconnect επίπεδο, το χαρακτηρίζουν διάφορες παράμετροι όπως δείξαμε στην Εικόνα 2.2.3.1, τις οποίες πρέπει να εξετάσουμε. Σκοπός είναι η απόλυτη κατανόησή τους, γιατί αντίθετα με τα αρχεία LIB και Verilog που είδαμε όπου η παραγωγή τους θα γίνει αυτόματα από το εργαλείο χαρακτηρισμού, το LEF αρχείο θα

διαμορφωθεί κατ' ευθείαν για την τεχνολογία που θέλουμε, τροποποιώντας τις τιμές των παραμέτρων του από ένα αρχικό αρχείο LEF παλαιότερης τεχνολογίας.

Οι παράμετροι που διακρίνουμε σε κάθε επίπεδο είναι:

- Type: ο τύπος του επιπέδου μπορεί να είναι διασύνδεσης, overlapping, επαφής, cut (contact), masterslice (poly, active).
- width/pitch/spacing rules
- direction
- resistance and capacitance per unit square

Ο τύπος του επιπέδου, ορίζει τον τρόπο που θα χρησιμοποιηθεί από το εργαλείο τοποθέτησης και διασύνδεσης, στο αντίστοιχο επίπεδο σχεδίασης. Έτσι μπορεί το κάθε επίπεδο να είναι διασύνδεσης, οπότε σ' αυτό θα τρέξουν οι αγωγοί σύνδεσης, ή overlapping, το οποίο θα χρησιμοποιηθεί από το εργαλείο για έλεγχο επικάλυψης ευθύγραμμων block.

Σ' αυτό το επίπεδο ανατρέχουν οι περιοριστικές περιγραφές, που υπάρχουν στα macros, ώστε να προκύψουν κατάλληλα ευθύγραμμα cells και μπλοκ (τέτοια είναι μπλοκ σε σχήμα L).

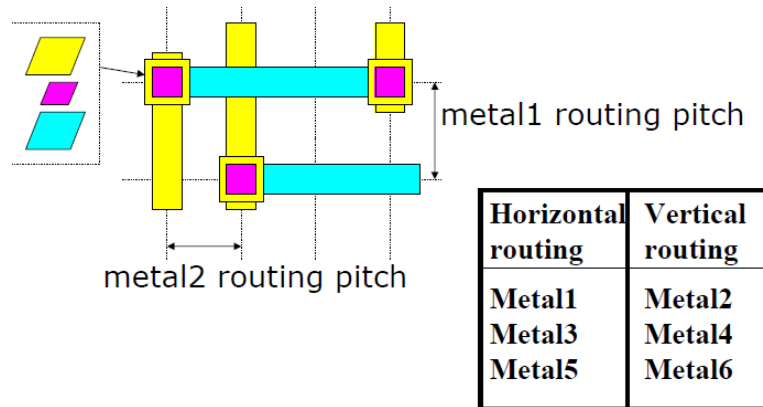
Άλλος τύπος επιπέδου μπορεί να είναι cut. Αυτό το layer χρησιμοποιείται για την μοντελοποίηση των vias. Τα vias χρησιμοποιούνται για τη σύνδεση ενός επιπέδου με ένα άλλο, τα οποία είναι ιεραρχημένα όπως στο σχήμα που ακολουθεί.

Layers	Design Rule	Parasitic
<b>POLY</b>	Net width	Resistance Capacitance
◆ Contact	Net spacing	
<b>Metal1</b>	Area	
◆ Via1	Enclosure	
<b>Metal2</b>	Wide metal slot	
	Antenna	
	Current density	

Εικόνα 2.2.3.7 Παράσταση της ιεραρχίας των επιπέδων.

Τέλος ο τύπος masterslice είναι τυπικά επίπεδο πολυπυριτίου, και χρειάζεται μόνο όταν τα cell macros έχουν pins πάνω στο πολυπυρίτιο. Οι τύποι των επιπέδων καθορίζονται στην σειρά επεξεργασίας, από κάτω προς τα επάνω (βλ Εικόνα 2.2.3.7).

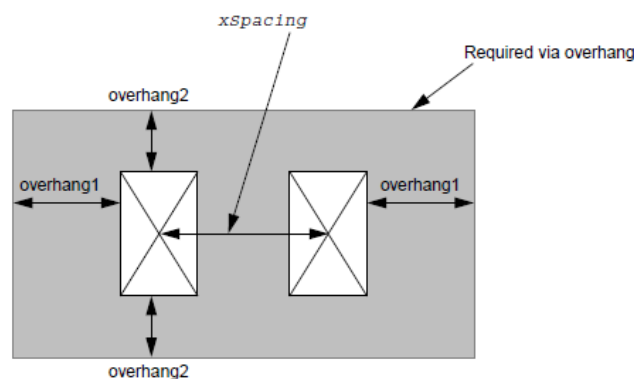
Άλλα χαρακτηριστικά των επιπέδων, είναι το ελάχιστο πλάτος (width) και το pitch. Το pitch ορίζει την απόσταση μεταξύ των αγωγών διασύνδεσης κάθε επιπέδου διασύνδεσης, μετρώντας την απόσταση από τους άξονες που διέρχονται από το κέντρο των αγωγών ως το κέντρο των vias, όπως δείχνει και το επόμενο σχήμα.



Εικόνα 2.2.3.8 Περιγραφή του pitch των επιπέδων διασύνδεσης.

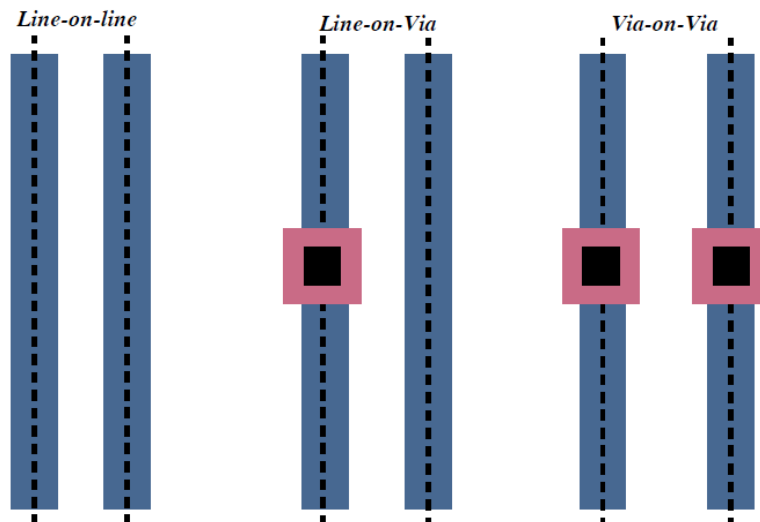
Σε κάθε layer ο προσανατολισμός είναι δεδομένος, ορίζεται από την παράμετρο direction, και έτσι με τη βοήθεια του pitch δημιουργούνται τα routing tracks (πίστες δρομολόγησης διασυνδέσεων). Η ένωση των tracks όλων των επιπέδων διασύνδεσης, σχηματίζει το πλέγμα πάνω στο οποίο γίνονται οι διασυνδέσεις.

Η ελάχιστη απόσταση μεταξύ καλωδίων, ή μεταξύ καλωδίου – via, καλωδίου – pin, ορίζεται από την παράμετρο spacing (Εικόνα 2.2.3.10). Ενώ η απόσταση του via από τα όρια του μετάλλου των αγωγών διασύνδεσης ορίζεται από την παράμετρο overhang.



Εικόνα 2.2.3.9 Παράδειγμα υπολογισμού overhang.



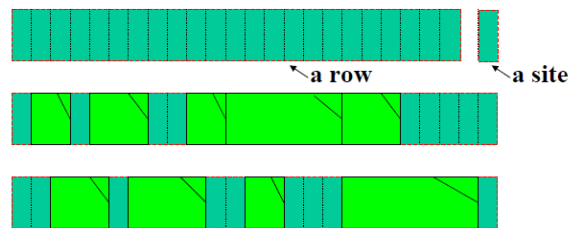
*Standard Cell Layout**Routing Grid Spacing*

Εικόνα 2.2.3.10 Τρόπος τοποθέτησης των γραμμών διασύνδεσης και των vias πάνω στο πλέγμα. Επίσης διακρίνονται οι line to line, line on via και via on via αποστάσεις.

Η αντίσταση ανά τετράγωνο και η χωρητικότητα ανά τετράγωνο, δίνουν τα χαρακτηριστικά των αγωγών διασύνδεσης. Από την τιμή τους θα υπολογίσει το εργαλείο, τις παρασιτικές χωρητικότητες και αντιστάσεις<sup>6</sup>.

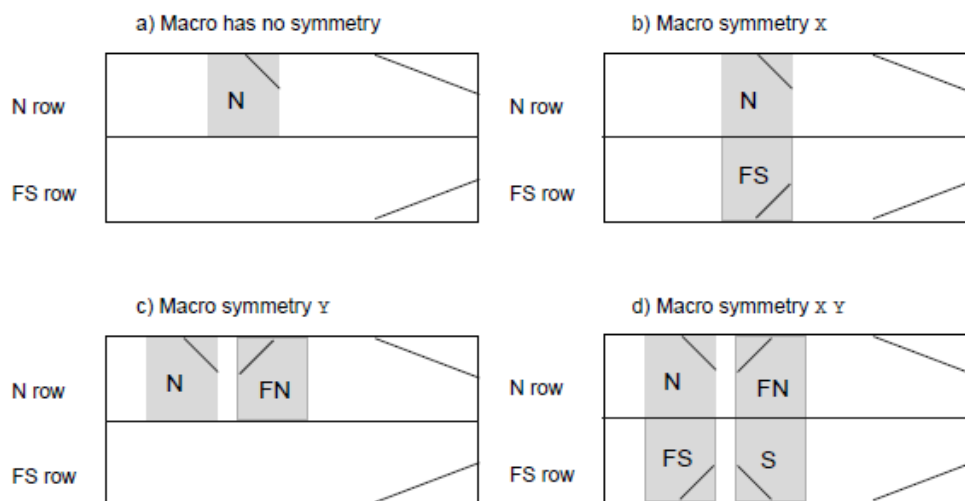
Ακολουθώντας την εσωτερική δομή του LEF σύμφωνα με την Εικόνα 2.2.3.1, προχωράμε στην περιγραφή των cell macros. Το ύψος των cells είναι σταθερό και πολλαπλάσιο της οριζόντιας απόστασης του πλέγματος, με εξαίρεση κάποια σύνθετα κύτταρα που μπορεί να έχουν διπλάσιο ύψος σε σχέση με τα υπόλοιπα. Το πλάτος των cells είναι πολλαπλάσιο της κατακόρυφης απόστασης του πλέγματος. Η περιγραφή κάθε cell εδώ ξεκινά με την δήλωση macro. Μέσα σε κάθε δήλωση macro περιλαμβάνεται η περιγραφή των cells, οι διαστάσεις τους, layout των ακροδεκτών, των μπλοκ και ακριβείς περιγραφές σχετικές με τη διασύνδεση των pin του cell με τα διάφορα επίπεδα. Λόγω του σταθερού ύψους των cells, η τοποθέτησή τους μπορεί να οργανωθεί εύκολα σε γραμμές όπως στο σχήμα που ακολουθεί, ενώ το ελάχιστο τμήμα μιας γραμμής (row), που μπορεί να περιλάβει ένα τμήμα του standard-cell είναι το site.

<sup>6</sup> Για περισσότερες πληροφορίες σχετικά με τη δομή και τις παραμέτρους του LEF αρχείου μπορείτε να ανατρέξετε στο εγχειρίδιο LEF/DEF Language Reference Product Version 5.6 September 2004 της cadence (10)



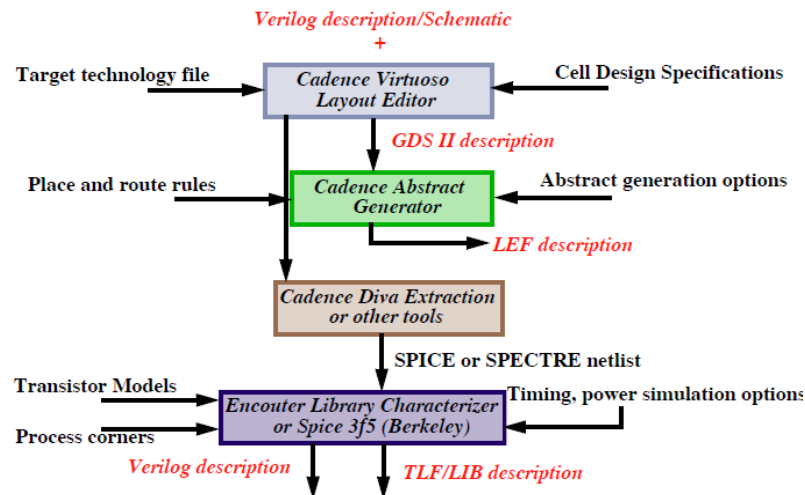
Εικόνα 2.2.3.11 Τρόπος τοποθέτησης των cells σε γραμμές. Κάθε cell καταλαμβάνει έναν αριθμό sites.

Η παράμετρος symmetry δηλώνει αν επιτρέπεται η στρέψη του cell κατά τον άξονα X, Y ή XY, κατά τη διαδικασία τοποθέτησής του (place) επάνω στη γραμμή.



Εικόνα 2.2.3.12 Λειτουργία της παραμέτρου symmetry κατά την τοποθέτηση του cell επάνω στη γραμμή.

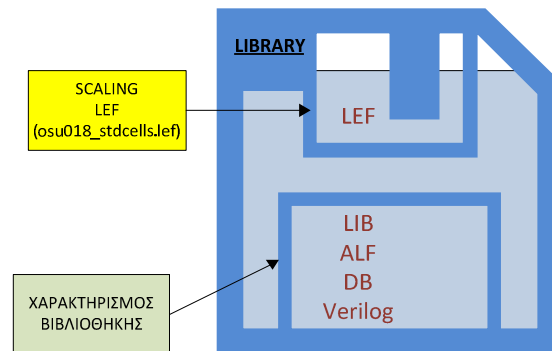
Το εργαλείο Cadence Abstract Generator είναι ένα εργαλείο που παράγει τα αρχεία LEF παίρνοντας είσοδο από τα αρχεία που περιγράφουν με λεπτομέρεια το layout (bottom up προσέγγιση). Το επόμενο σχήμα περιγράφει την διαδικασία αυτής της προσέγγισης.

*Standard Cell Library Formats*

Εικόνα 2.2.3.13 Περιγραφή εξαγωγής του αρχείου LEF από GDS II αρχείο (bottom - up προσέγγιση).

## 2.2.4. Συγκέντρωση των αρχείων LIB, Verilog και LEF για τη σύνθεση της βιβλιοθήκης

Ως εδώ έχουν γίνει ξεκάθαρα τα βήματα που ακολουθούνται για την ψηφιακή σχεδίαση, και έχει διαφανεί, ποιά είναι τα απαραίτητα αρχεία που πρέπει να έχει κάποιος διαθέσιμα, ώστε να ξεκινήσει την ανάπτυξη συστημάτων και αρχιτεκτονικών σε δεδομένη τεχνολογία. Για την ολοκλήρωση κάθε μίας απ' τις βιβλιοθήκες στις τεχνολογίες των 45nm, 32nm, 22nm, 16nm, πρέπει να συγκεντρώσουμε τα αρχεία LIB, Verilog, DB, LEF. Αυτά πρέπει να είναι απόλυτα συμβατά μεταξύ τους. Δηλαδή θα πρέπει τα εργαλεία σχεδίασης, σε κάθε βήμα της σχεδίασης να χρησιμοποιούν ένα απ' αυτά τα αρχεία για κάθε τεχνολογία.



Εικόνα 2.2.4.1 Βιβλιοθήκη με τα απαραίτητα αρχεία.

Ο επιτυχής χαρακτηρισμός βιβλιοθήκης που λαμβάνει χώρα στην παρούσα εργασία, για κάθε τεχνολογία, θα δώσει σαν αποτέλεσμα:

- Το αρχεία LIB, ALF (μια μορφή εναλλακτική του LIB) και το αρχείο Verilog. Εδώ είναι χρήσιμο να αναφέρουμε πως απαιτούνται δυο Spice αρχεία. Ένα για τα cells (Spice netlist) και ένα για τα μοντέλα των τρανζίστορ (Spice model). Το αρχείο Spice netlist είναι τελικά που θα επεξεργαστούμε κατάλληλα, έχοντας παράλληλα εξασφαλίσει το αντίστοιχο τρανζίστορ Spice model, για να πετύχουμε scaling των διαστάσεων, και πέρασμα σε επόμενη τεχνολογία. Με την βοήθεια εργαλείων όπως ο Library compiler, μπορούμε να παράγουμε το αρχείο db, μια εκδοχή του LIB αρχείου σε δεκαεξαδική μορφή (βάση δεδομένων) απαραίτητη για τη διαδικασία σύνθεσης.
- Το αρχείο LEF θα παραχθεί για κάθε τεχνολογία από ένα ήδη υπάρχον στην τεχνολογία 180nm, με διαδικασία κλιμάκωσης (scaling) των παραμέτρων που περιγράφει το αρχικό.

## Κεφάλαιο 3

### Κλιμάκωση (Scaling)

#### 3.1. Κανόνες Κλιμάκωσης (Scaling rules)

Με τον όρο κλιμάκωση εννοούμε τη δυνατότητα να μειώσουμε τις διαστάσεις των τρανζίστορ με σκοπό να μεταβούμε από μια δεδομένη τεχνολογία σχεδίασης και κατασκευής σε μία άλλη, μικρότερων διαστάσεων. Στόχος είναι η αύξηση της απόδοσης και της πυκνότητας των τρανζίστορ και η μείωση της κατανάλωσης. Η διαδικασία αυτή δεν γίνεται αυθαίρετα αλλά διέπεται από συγκεκριμένους κανόνες. Με την κλιμάκωση από μια τεχνολογία σε μια άλλη μικρότερης διάστασης (από την προηγούμενη γενιά στην επόμενη) για να ισχύει ο νόμος του Moore<sup>7</sup>, πρέπει το εμβαδό που καταλαμβάνει ένα τρανζίστορ στην τεχνολογία της προηγούμενης γενιάς, να υποδιπλασιάζεται στην επόμενη. Άρα οι εξωτερικές του διαστάσεις θα πρέπει να μειώνονται κατά 30% (επειδή  $0.7$  των αρχικών  $\cdot 0.7$  των αρχικών, όπου γίνονται τελικά οι διαστάσεις, δίνουν εμβαδό  $0.50$  του αρχικού). Εκτιμείται η μείωση των καθυστερήσεων των πυλών κατά 30%, και μείωση της κατανάλωσης ανά τρανζίστορ κατά 65%. Σύμφωνα με την θεωρία της κλιμάκωσης, με μείωση των πλευρικών και κατακόρυφων διαστάσεων κατά 30%, αναμένεται ταυτόχρονα μείωση της χωρητικότητας και της καθυστέρησης κατά 30%. Οπότε κανονικοποιώντας στην μονάδα τα πιο πάνω μεγέθη, θα έχουμε ως αποτέλεσμα να πάμε στο  $0.7$  της τιμής τους.

$$New\_Delay = 0.7Delay, \quad Frequency = 1/New\_Delay = 1.43, \quad New\_W = 0.7W, \\ New\_L = 0.7L, \quad New\_tox = 0.7tox.$$

$$Area\ capacitance = New\_CA = (0.7 * 0.7)/0.7 = 0.7 CA.$$

$$Fringing\ capacitance = C_F \sim L \Rightarrow New\_C_F = 0.7C_F.$$

$$New\_Total\ Capacitance = 0.7Total\ Capacitance.$$

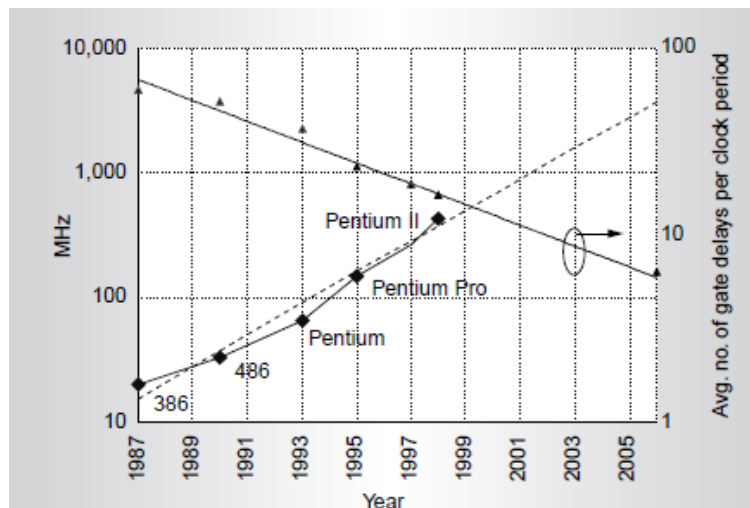
---

<sup>7</sup> Διπλασιασμός της πυκνότητας των τρανζίστορ σε ένα chip, κάθε δύο χρόνια (14)

Δεδομένου ότι οι διαστάσεις γίνονται 0,7 των αρχικών, η επιφάνεια του die θα είναι  $X*Y=0,7*0,7\cong 0,5$ . Έτσι η χωρητικότητα ανά μονάδα επιφάνειας θα είναι  $\frac{C}{Area} =$

$$\frac{0,7}{0,7*0,7} = \frac{1}{0,7}$$

Συνοψίζοντας λοιπόν, με την κλιμάκωση των διαστάσεων κατά 30% έχουμε μείωση της επιφάνειας του die κατά 50% και αύξηση της χωρητικότητας ανά μονάδα κατά 43%<sup>8</sup>. Είναι χρήσιμο να παραθέσουμε κάποια στοιχεία, που παρουσιάζουν την εξέλιξη της συχνότητας του ρολογιού για τους επεξεργαστές της Intel, σε συνάρτηση με το χρόνο και την εξέλιξη του μεγέθους των τρανζίστορ, σε σχέση με την τεχνολογία.

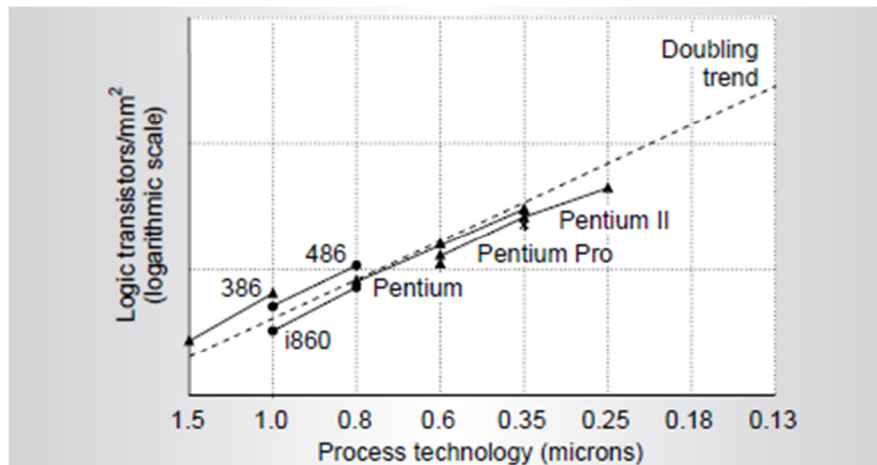


Εικόνα 3.1.1 Η συχνότητα των επεξεργαστών διπλασιάζεται σε κάθε γενιά<sup>8</sup>.

Όπως βλέπουμε λοιπόν η συχνότητα του ρολογιού διπλασιάζεται περίπου, κάθε δυο χρόνια. Αντίστοιχα η μέση τιμή της καθυστέρησης πύλης, υποδιπλασιάζεται κάθε δυο χρόνια. Αυτά βέβαια ισχύουν μέχρι το έτος που περιγράφει το διάγραμμα, που αντιστοιχεί σε τεχνολογία 180nm περίπου. Από εκεί και μετά επειδή οι ελάχιστες διαστάσεις της τεχνολογίας προσεγγίζουν τις φυσικές ιδιότητες των ημιαγωγών, η συμπεριφορά περιμένουμε να διαφοροποιείται, σχετικά με τις καθυστερήσεις. Το επόμενο σχήμα (Εικόνα 3.1.2) δείχνει πως μεταβάλλεται η πυκνότητα των τρανζίστορ, σε σχέση με την ελάχιστη διάσταση της τεχνολογίας. Η διακεκομμένη

<sup>8</sup> Στοιχεία που προκύπτουν από το paper Design Challenges of Technology Scaling του Shekhar Borkar (9)

γραμμή, παριστάνει την αναμενόμενη πορεία, για διπλασιασμό της πυκνότητας κάθε δύο χρόνια, σύμφωνα πάντα με το νόμο του Moore.



Εικόνα 3.1.2 Εξέλιξη της πυκνότητας των τρανζίστορ, ανάλογα με την τεχνολογία (ελάχιστη διάσταση)<sup>8</sup>.

## 3.2. Μοντέλα κλιμάκωσης

Έχουν επικρατήσει τρία βασικά μοντέλα κλιμάκωσης.

- **Το μοντέλο σταθερού πεδίου** ή πλήρους κλιμάκωσης όπως λέγεται. Στο μοντέλο σταθερού πεδίου όπως αναφέρεται και στον τίτλο του, προκειμένου να διατηρηθεί σταθερό το ηλεκτρικό πεδίο στο εσωτερικό του MOS θα πρέπει οι εφαρμοζόμενες τάσεις και οι διαστάσεις του τρανζίστορ να μειωθούν κατά τον ίδιο συντελεστή (scaling factor). Στην κλιμάκωση σταθερού πεδίου, οι διαστάσεις του στοιχείου, όπως το πλάτος  $W$ , το μήκος καναλιού  $L$ , το πάχος οξειδίου  $t_{ox}$ , το βάθος επαφής  $X_j$ , οι εφαρμοζόμενες τάσεις και η πυκνότητα συγκέντρωσης υποστρώματος  $N$ , αλλάζουν κλίμακα κατά την ίδια σταθερή παράμετρο  $S$ . Κλιμάκωση υπάρχει επίσης στο πάχος αραίωσης  $d$ , στην τάση κατωφλίου  $V_t$  και στο ρεύμα εκροής  $I_{ds}$ . Οι περιοχές απογύμνωσης στις επαφές  $p_n$  που δημιουργούνται στις επαφές της πηγής και της εκροής, θέτουν ένα κάτω όριο για το μήκος του καναλιού. Θα πρέπει δηλαδή το μήκος του καναλιού να είναι μεγαλύτερο από το άθροισμα των μηκών των περιοχών απογύμνωσης των εν λόγω περιοχών, ώστε κάτω από την πύλη να υπάρχει μη απογυμνωμένη περιοχή και έτσι να υπάρχει δυνατότητα ελέγχου της αγωγιμότητας του καναλιού από την πύλη. Για το ρεύμα πηγής – εκροής  $I_{ds}$ ,

στην περίπτωση κλιμάκωσης σταθερού πεδίου, το ρεύμα ανά τρανζίστορ όπως και οι διαστάσεις του στοιχείου πολλαπλασιάζονται με τον συντελεστή  $1/S$  (όπου  $S$  ο παράγοντας κλιμάκωσης). Η πυκνότητα του κυκλώματος όμως αυξάνεται κατά  $S^2$ . Οπότε η πυκνότητα ρεύματος θα είναι  $J_{scaled} = J \frac{1}{S} S^2 = J * S$ . Επίσης για τη δυναμική ισχύ, επειδή έχουμε μείωση της τάσης κατά 30% τότε  $Power = C * V^2 * f = 0.7 * 0.7^2 * 1/0.7) = 0.5$ . Άρα εδώ έχουμε μείωση κατά 50%.

- **Το μοντέλο σταθερής τάσης.** Στο μοντέλο αυτό, οι εφαρμοζόμενες τάσεις παραμένουν σταθερές και γίνεται κλιμάκωση μόνο στις διαστάσεις των τρανζίστορ. Αυτό σημαίνει πως η τάση τροφοδοσίας και οι τάσεις κατωφλίου των τρανζίστορ παραμένουν σταθερές. Στην κλιμάκωση σταθερής τάσης θα έχουμε αύξηση της πυκνότητας του ρεύματος κατά  $S^3$ . Έτσι απαιτείται η χρήση φαρδύτερων αγωγών μετάλλου με επιπτώσεις στις παρασιτικές χωρητικότητες που θα εξετάσουμε αργότερα. Για τον λόγο αυτό χρησιμοποιούνται γραμμές μετάλλου αποκλειστικά για την τάση τροφοδοσίας Vdd και τη γείωση GND. Για την δυναμική ισχύ μετά το scaling στην επόμενη γενιά, εφόσον έχουμε αύξηση της συχνότητας κατά 43%, μείωση της χωρητικότητας κατά 30%, με σταθερή τάση τροφοδοσίας Vdd (scaling σταθερής τάσης) θα έχουμε  $Power = C * V^2 * f = 0.7 * 1 * (1/0.7) = 1$  άρα περίπου την ίδια κατανάλωση.
- **Το μοντέλο γενικής κλιμάκωσης.** Το μοντέλο γενικής κλιμάκωσης είναι το πλέον ρεαλιστικό σήμερα και σύμφωνα με αυτό, οι κλιμακώσεις των διαστάσεων των τρανζίστορ και των τάσεων, γίνονται με διαφορετικούς συντελεστές. Η κλιμάκωση (scaling) των τάσεων περιορίζεται από τις φυσικές ιδιότητες των ημιαγωγών και δεν μπορεί να ακολουθήσει τον ρυθμό scaling των φυσικών διαστάσεων. Το μέγεθος  $kT/q$  ( $k$ : σταθερά Boltzmann,  $T$ : η θερμοκρασία και  $q$ : φορτίο) με  $kT/q=25mV=1/40V$  σε θερμοκρασία δωματίου, δεν μπορεί να μειωθεί. Άρα και το  $V_T$  έχει κάποιο όριο κλιμάκωσης (scaling). Εδώ εξηγείται και η πρόταση, ότι το μοντέλο γενικής κλιμάκωσης είναι το πλέον ρεαλιστικό.



### 3.3. Δημιουργία LIB για μελλοντικές τεχνολογίες

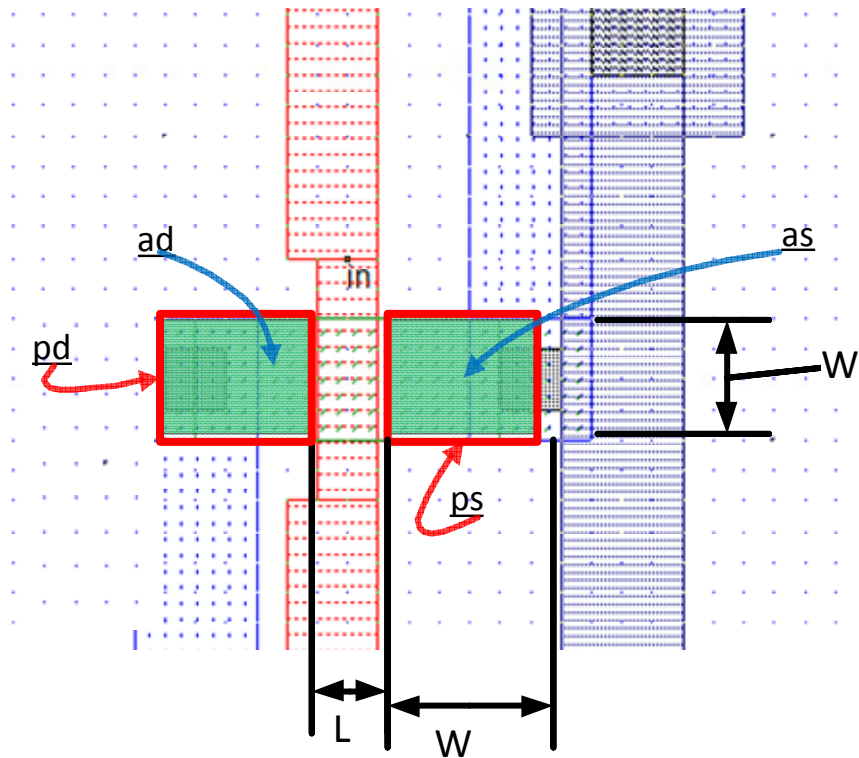
Εξετάζοντας την ροή ψηφιακής σχεδίασης (design flow), είδαμε τη σπουδαιότητα του LIB αρχείου. Είναι εκείνο το αρχείο που με μεταγλώττιση δίνει το αρχείο .db, απαραίτητο για σύνθεση του συστήματος που σχεδιάσαμε, αλλά είναι απαραίτητο και για το εργαλείο που εκτελεί place and routing, για την εξαγωγή αποτελεσμάτων χρονισμού και κατανάλωσης. Έτσι πρέπει απαραίτητως να δημιουργήσουμε από ένα τέτοιο αρχείο τουλάχιστον, για κάθε νέα τεχνολογία της βιβλιοθήκης μας.

#### 3.3.1. Δημιουργία του αρχείου Spice

Όπως αναφέρθηκε στο τέλος του προηγούμενου κεφαλαίου, η διαδικασία μέσω της οποίας παράγεται το αρχείο LIB, είναι μία από τις διαδικασίες που γίνονται κατά τον χαρακτηρισμό της βιβλιοθήκης. Ακόμα αναφέρθηκε πως πρέπει πριν απ' αυτό το στάδιο, να έχουμε διαθέσιμα δύο αρχεία spice.

Το ένα είναι ένα Spice netlist και περιγράφει σε πιο χαμηλό επίπεδο, τη δομή των τυποποιημένων cells της βιβλιοθήκης μας, ενώ το άλλο αρχείο είναι ένα Spice τρανζίστορ model και περιγράφει τα ηλεκτρικά χαρακτηριστικά των δομικών στοιχείων που συνθέτουν τα cells (μοντέλα NMOS και PMOS).

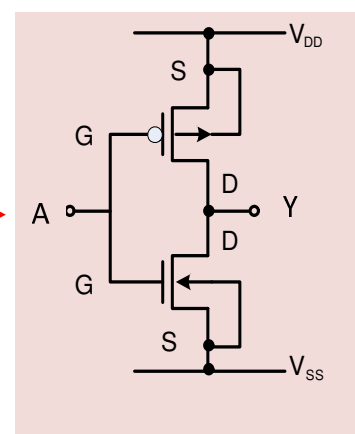
Το αρχείο spice netlist που περιγράφει τα cells της βιβλιοθήκης μας, είναι ένα αρχείο με ειδική δομή και παραμέτρους, ώστε να πληροφορήσει τα εργαλεία cad, για τον τύπο και τον αριθμό (έμμεσα) των NMOS και PMOS και τη συνδεσμολογία τους με τους εσωτερικούς αγωγούς του cell καθώς και τα pins εισόδων/εξόδων ή τροφοδοσίας (Vdd, Gnd). Στο σχήμα στην Εικόνα 3.3.2, δίνεται το παράδειγμα ενός αναστροφέα και ο τρόπος που περιγράφεται μέσω της spice netlist. Όπως παρατηρούμε το cell ορίζεται ως ένα sub circuit (subckt). Το συγκεκριμένο περιέχει δύο τρανζίστορ, το M0 και M1. Το M0 είναι PMOS με πλάτος  $w=0.5\mu\text{m}$  και  $l=0.05\mu\text{m}$ . Απ' τις τιμές αυτές προκύπτει εμβαδό drain  $ad=w*w=0.5*0.5=0.25\mu\text{m}^2$  και περίμετρος drain  $pd=4*0.5=2.0\mu\text{m}$  (υποθέτουμε πως οι περιοχές της πηγής και του drain, είναι τετράγωνα). Στην Εικόνα 3.3.1 περιγράφεται ο τρόπος μέτρησης αυτών των γεωμετρικών χαρακτηριστικών.



Εικόνα 3.3.1 Τρόπος μέτρησης των γεωμετρικών χαρακτηριστικών  $W$ ,  $L$ ,  $ad$ ,  $pd$ ,  $as$ ,  $ps$  ενός τρανζίστορ.

Τα ίδια προκύπτουν για την επιφάνεια και την περίμετρο της πηγής (source) του τρανζίστορ. Το M1 είναι NMOS με  $w=0.25 \mu\text{m}$  και  $l=0.05 \mu\text{m}$ . Απ' τις τιμές αυτές προκύπτει εμβαδό drain  $ad=w*w= 0.25*0.25 = 0.0625\mu\text{m}^2$  και περίμετρος drain  $pd=4*0.25=1.0\mu\text{m}$ . Το ίδιο ισχύει για τις διαστάσεις της πηγής (source).

```
.subckt INVX1 A Y VDD VSS
M0 Y A VDD VDD pmos w=0.5u l=0.05u
+ ad=0.25p pd=2.0u as=0.25p ps=2.0u
M1 Y A VSS VSS nmos w=0.25u l=0.05u
+ ad=0.0625p pd=1.0u as=0.0625p ps=1.0u
.ends INVX1
```



Εικόνα 3.3.2 Περιγραφή ενός αναστροφέα με spice netlist.

Τα υπόλοιπα στοιχεία που διακρίνονται στη netlist, είναι η περιγραφή της εσωτερικής διασύνδεσης. Εδώ δηλώνεται πως το drain του τρανζίστορ PMOS M0 συνδέεται με

την έξοδο Y, η πύλη (gate) με την είσοδο A και η πηγή (source) καθώς και το substrate με την τάση Vdd. Αντίστοιχα στο NMOS τρανζίστορ M1 το drain συνδέεται με την έξοδο Y, η πύλη με την είσοδο A και η πηγή και το substrate με την τάση Vss.

Η εργασία που πρέπει να γίνει σ' αυτό το αρχείο της βιβλιοθήκης μας, είναι η ανάγνωση όλων των cells ένα προς ένα και η γραμμική κλιμάκωση (scaling) των διαστάσεων w και l, με έναν παράγοντα που προκύπτει από την τεχνολογία στην οποία θέλουμε να μεταβούμε. Δηλαδή  $new\_w=w/f_n$ ,  $new\_l=l/f_n$  όπου  $f_n$  ο ηστός παράγοντας scaling ( $n=1,2,3,4$ ).

$$f1=180nm/45nm=4$$

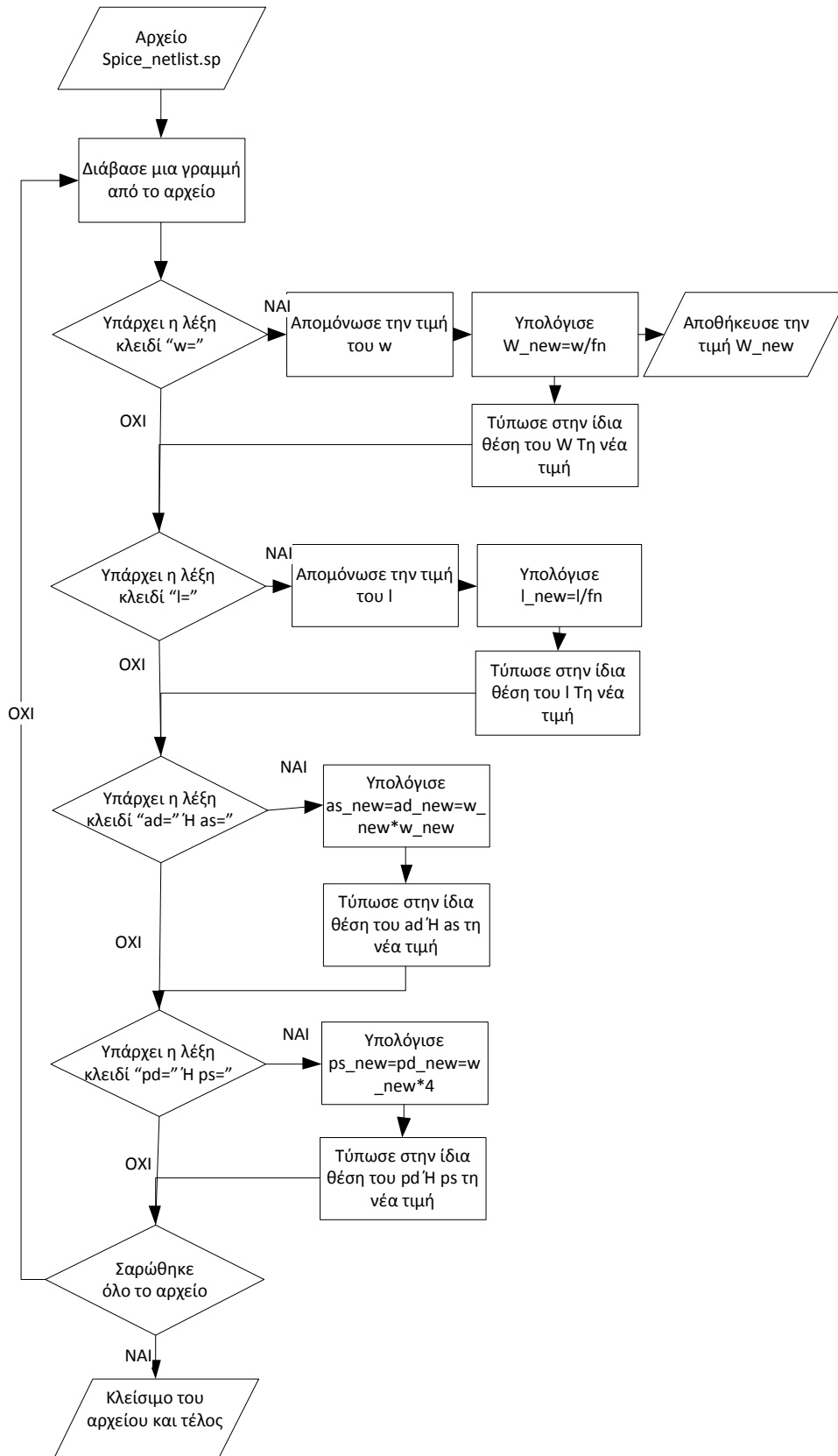
$$f2= \text{int}(180/32)=5$$

$$f3= \text{int}(180/22)=8$$

$$f4=\text{int}(180/22)=11$$

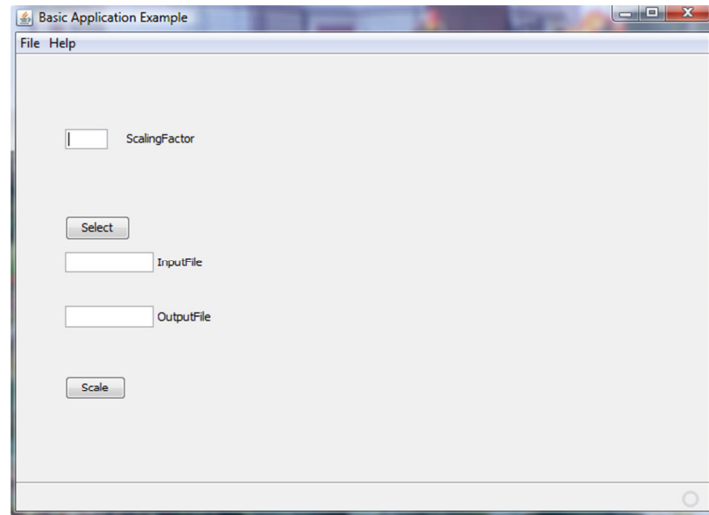
Τελικό ζητούμενο είναι ο νέος υπολογισμός των διαστάσεων της πηγής και του drain, από τις τιμές του w\_new που προκύπτει απ' την κλιμάκωση (scaling).

Προκειμένου να αυτοματοποιηθεί η εργασία αυτή, μέσω κώδικα σε κάποια γλώσσα προγραμματισμού, δίνεται η παράστασή της σε βήματα ενός αλγορίθμου.



Εικόνα 3.3.3 Παράσταση αλγόριθμου κλιμάκωσης (scaling) διαστάσεων των τρανζίστορ του αρχείου spice netlist.

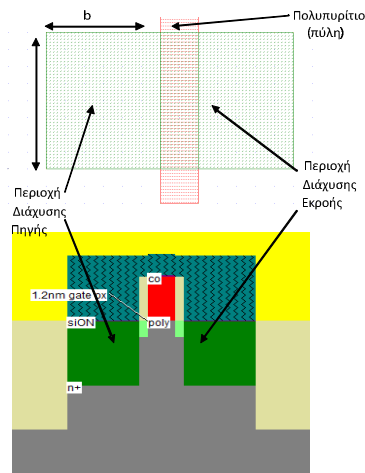
Ο αλγόριθμος αυτός πραγματοποιείται στην παρούσα εργασία, σε γλώσσα προγραμματισμού java ο κώδικας του οποίου βρίσκεται στο παράρτημα 1. Το γραφικό περιβάλλον της εφαρμογής java φαίνεται στο ακόλουθο σχήμα.



Εικόνα 3.3.4 Γραφικό περιβάλλον της εφαρμογής που εκτελεί τον αλγόριθμο κλιμάκωσης (scaling) του spice netlist των cells.

Όπως βλέπουμε υπάρχουν τρία πεδία εισαγωγής δεδομένων. Το πρώτο δέχεται τον παράγοντα κλιμάκωσης (scaling), ενώ τα άλλα δύο δέχονται τα αρχεία spice netlist εισόδου και εξόδου αντίστοιχα. Με το πάτημα του κουμπιού “Scale” τρέχει ο αλγόριθμος που σχεδιάσαμε και το αποτέλεσμα καταγράφεται στο αρχείο εξόδου.

### 3.3.2. Επιλογή κατάλληλου μοντέλου τρανζίστορ

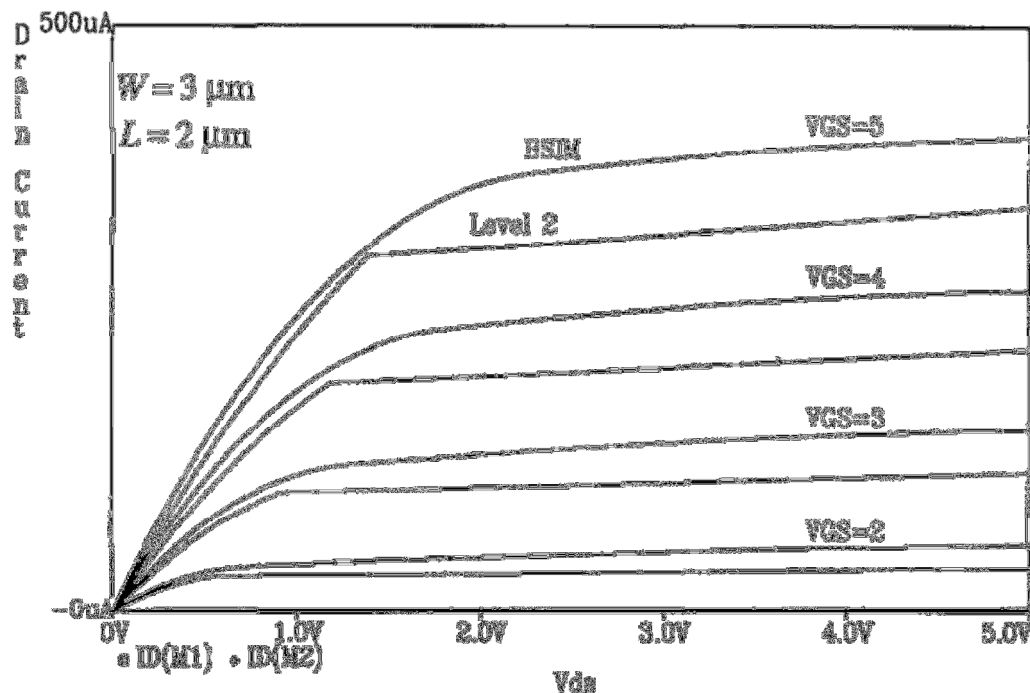


Εικόνα 3.3.5 Σχηματική παράσταση ενός NMOS τρανζίστορ.

Στο προηγούμενο σχήμα παριστάνεται ένα NMOS τρανζίστορ, το οποίο μοντελοποιείται καταλλήλως για τα εργαλεία CAD. Υπάρχει ένα spice model card για κάθε τρανζίστορ, που περιγράφει με δομημένο τρόπο τις παραμέτρους τους. Κάποιες εκ των παραμέτρων τους είναι το  $V_t$ ,  $t_{ox}$ ,  $t_{nom}$ ,  $k_1$  κλπ<sup>9</sup>. Δηλαδή η τιμή της τάσης κατωφλίου  $V_t$ , ή το πάχος του οξειδίου μεταξύ της πύλης και του ημιαγωγού και η θερμοκρασία. Το μοντέλο που χρησιμοποιείται εδώ είναι το BSIM4 (Berkeley Short Channel IGFET Model) ή level 54, που έχει προκύψει ερευνητικά από το πανεπιστήμιο Berkeley και έχει γίνει πρότυπο για τα εργαλεία εξομοίωσης. Το απλούστερο μοντέλο είναι το μοντέλο level 2 ή BSIM1. Αυτό μοντελοποιεί πολύ απλά το τρανζίστορ MOS με τις δύο περιοχές λειτουργίας, που γνωρίζουμε από τη βιβλιογραφία VLSI. Την γραμμική περιοχή ( $V_{ds} < V_{gs} - V_t$ ) και την περιοχή κορεσμού ( $V_{ds} > V_{gs} - V_t$ ). Το BSIM1 μοντέλο μπορεί να φέρει ικανοποιητικά αποτελέσματα για τεχνολογίες μέχρι 1 $\mu$ m. Η βασική διαφορά από μοντέλο σε μοντέλο είναι ο διαφορετικός τρόπος περιγραφής των τάσεων κατωφλίου, ή ο διαφορετικός τρόπος υπολογισμού της διαγωγιμότητας και της σχέσης V-I και των κανόνων που τη διέπουν (1) (2). Στην Εικόνα 3.3.6 μπορούμε να διακρίνουμε τη διαφορά μεταξύ του μοντέλου level2 και του BSIM1. Η περίπτωση του BSIM4 προσεγγίζει με

<sup>9</sup> Για πληροφορίες σχετικά με τα spice model cards των τρανζίστορ υπάρχει η σχετική ιστοσελίδα του Berkeley University <http://ptm.asu.edu/>

ρεαλιστικότερο τρόπο τις V-I χαρακτηριστικές του CMOS σε σχέση με το μοντέλο BSIM1 όπου παρουσιάζεται ένα είδος ασυνέχειας στο σημείο του ορίου των δύο περιοχών (γραμμική περιοχή – περιοχή κορεσμού) και οι δύο περιοχές είναι απολύτως διακριτές, χωρίς να υπάρχει η περιοχή τριόδου. Αντίθετα στο μοντέλο BSIM4 η μετάβαση από τη μια περιοχή στην άλλη, γίνεται ομαλότερα και περιγράφονται και οι τρεις περιοχές του τρανζίστορ (γραμμική, τριόδου και κόρου).



Εικόνα 3.3.6 Σύγκριση I-V χαρακτηριστικών των μοντέλων BSIM1 και BSIM4.

Τα μοντέλα που χρειάζονται για την παρούσα εργασία, τα λαμβάνουμε έτοιμα για τις τεχνολογίες που χρειαζόμαστε, από την ειδική web εφαρμογή που υπάρχει στην ιστοσελίδα του Berkeley university<sup>9</sup>.

### 3.4. Δημιουργία αρχείων LEF

Στο βήμα αυτό, πρέπει να δημιουργήσουμε για κάθε μία από τις τέσσερις τεχνολογίες, τα αντίστοιχα αρχεία LEF. Η βασικές διαφορές που έχουν μεταξύ τους, αφορούν τα γεωμετρικά και τα ηλεκτρικά χαρακτηριστικά τους. Έχοντας διαθέσιμο ως αρχείο LEF αναφοράς αυτό της τεχνολογίας 180nm (osu018\_stdcells.lef), θα προχωρήσουμε σε κλιμάκωση (scaling) όλων των απαιτούμενων παραμέτρων με κατάλληλο συντελεστή για κάθε τεχνολογία. Θα προσπαθήσουμε κι εδώ να

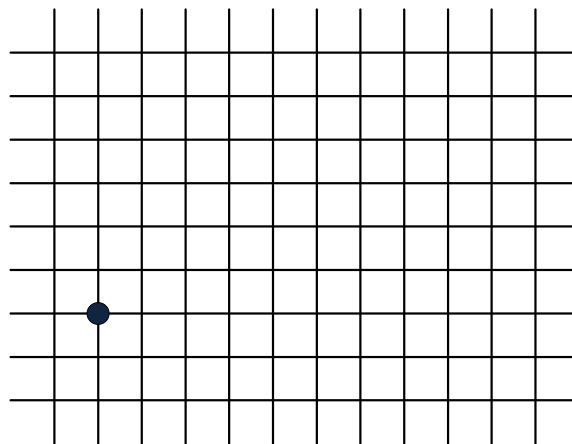
εξαγάγουμε έναν αλγόριθμο υλοποίησης της διαδικασίας, με σκοπό την αυτοματοποίησή της, με τη βοήθεια μιας γλώσσας προγραμματισμού. Ας δούμε όμως ποιες είναι οι παρεμβάσεις και πως πρέπει να γίνουν, στο αρχικό αρχείο LEF.

Ο τρόπος περιγραφής των διαστάσεων στο αρχείο LEF εικονίζεται στο επόμενο σχήμα. Συγκεκριμένα οι διαστάσεις και οι θέσεις των στοιχείων δίνονται υπό μορφή συντεταγμένων.



Εικόνα 3.4.1 Τρόπος παράστασης διαστάσεων και θέσης.

Όσο για τα σημεία που μπορούν να τοποθετηθούν τα στοιχεία του κάθε επιπέδου, είναι αυτά που προκύπτουν απ' τα σημεία τομής των οριζόντιων και κάθετων γραμμών του manufacturing grid.

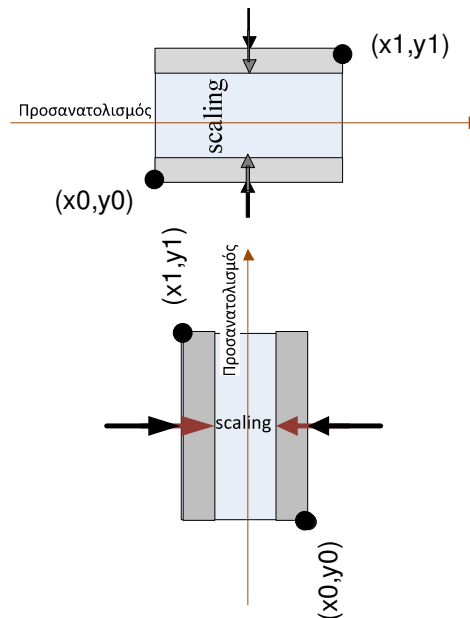


Εικόνα 3.4.2 Σημεία τοποθέτησης συντεταγμένων.

Ανάλογα με το επίπεδο που βρισκόμαστε υπάρχει διαφορετικός προσανατολισμός των καλωδιώσεων. Στην παρούσα βιβλιοθήκη, τα περιττά επίπεδα εμφανίζουν οριζόντιο προσανατολισμό ενώ τα άρτια επίπεδα εμφανίζουν κάθετο προσανατολισμό. Για τον λόγο αυτό το scaling των γραμμών μετάλλου γίνεται μόνο κατά τον άξονα x ή y αναλόγως σε ποιο επίπεδο μετάλλου ανήκει η εν λόγω γραμμή. Έτσι όταν έχουμε οριζόντιο προσανατολισμό η κλιμάκωση θα γίνει κατά τον άξονα y.



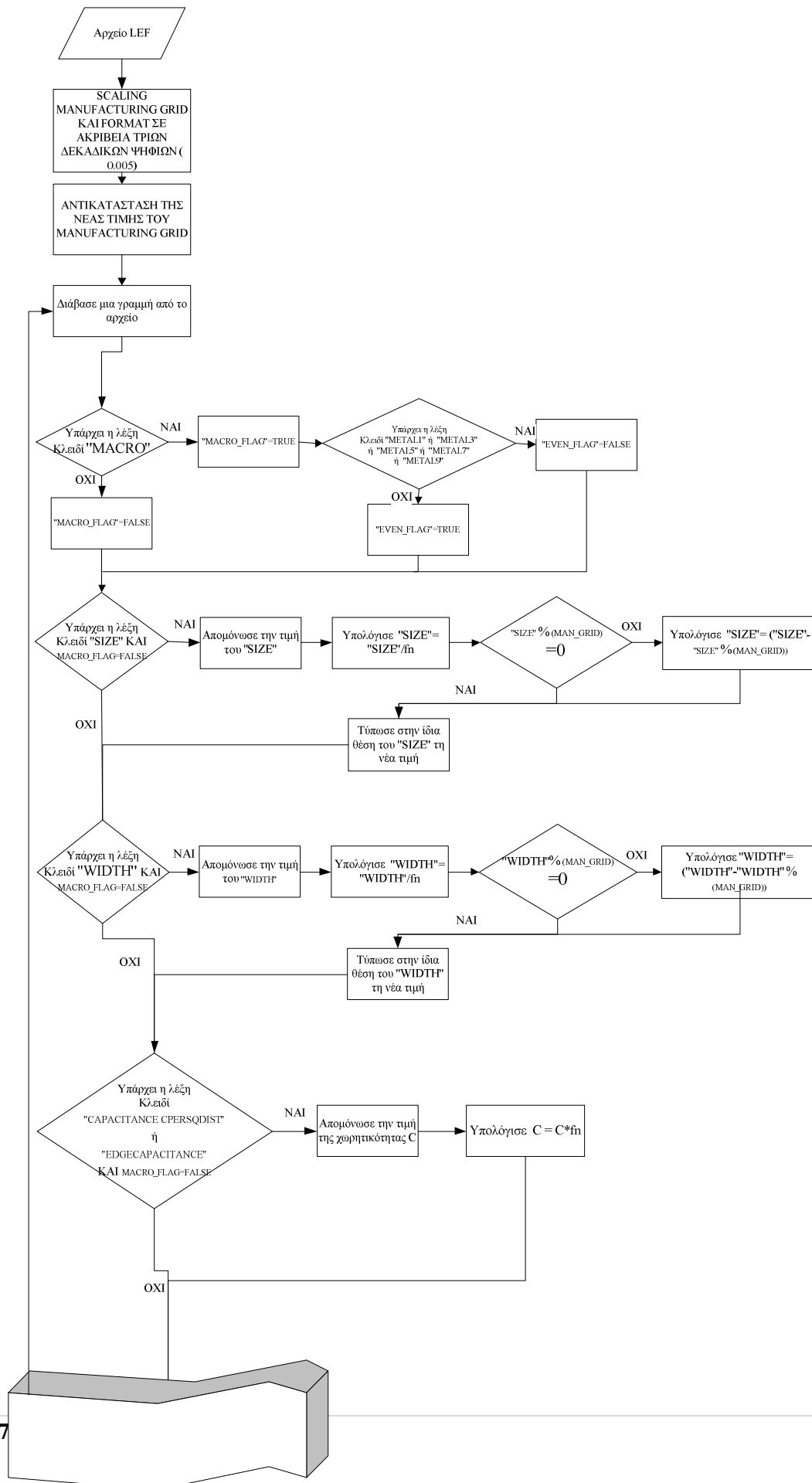
Ενώ όταν έχουμε κατακόρυφο προσανατολισμό τότε το scaling θα πρέπει να γίνει κατά τον x άξονα (βλ. Εικόνα 3.4.3)

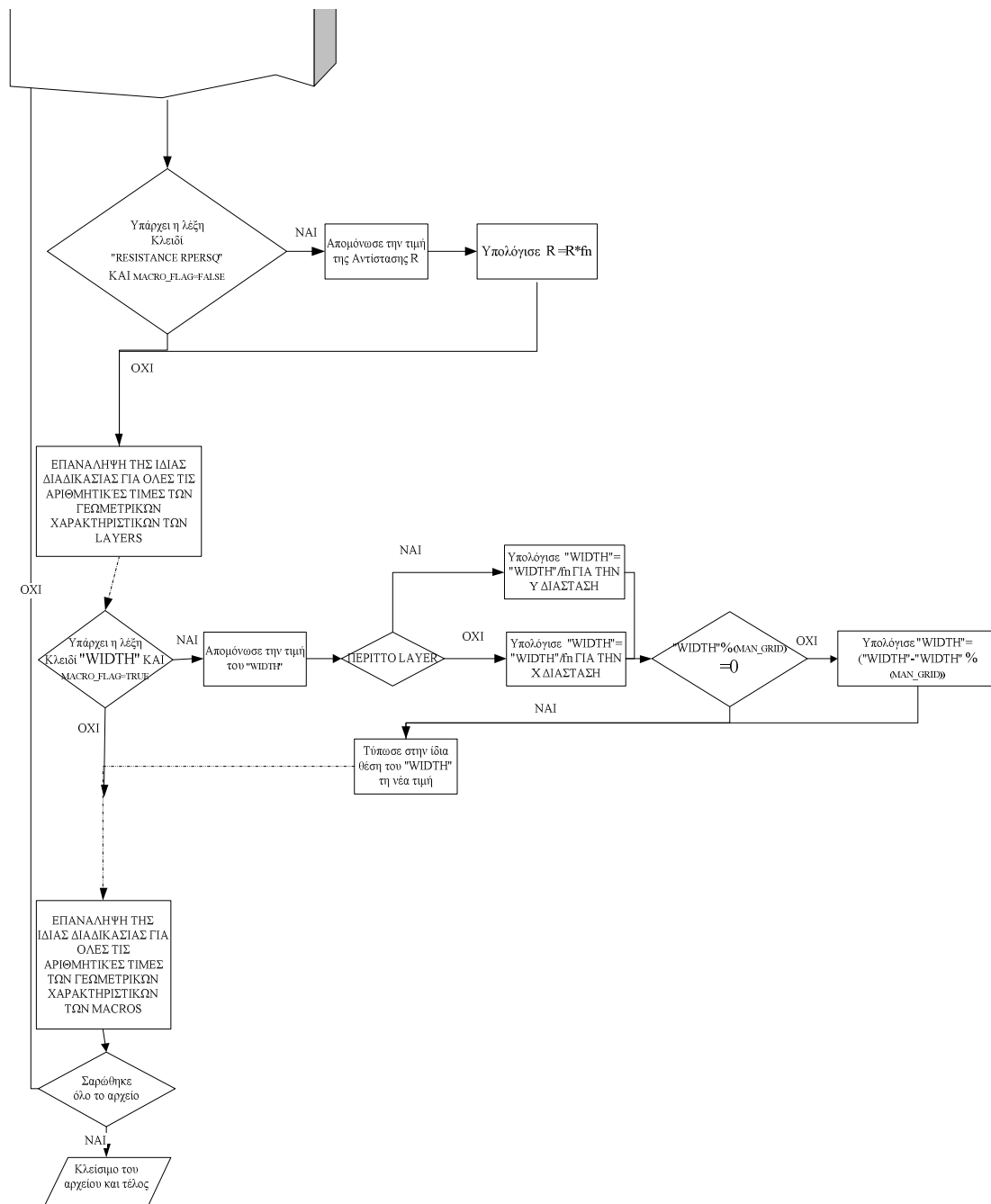


Εικόνα 3.4.3 Τρόπος αυξομείωσης διαστάσεων, ανάλογα με τον προσανατολισμό του επιπέδου.

Σύμφωνα με τα παραπάνω λοιπόν το scaling εδώ δεν είναι απλά μια υπόθεση εφαρμογής μιας γραμμικής σχέσης μεταξύ του παράγοντα κλιμάκωσης και των γεωμετρικών διαστάσεων που περιγράφει το LEF αρχείο, αλλά πρέπει να τηρούνται σε κάθε βήμα οι γεωμετρικοί κανόνες που αναφέραμε. Βασικός κανόνας για τη σωστή λειτουργία του place and routing είναι η σωστή θέση των κελιών και των γραμμών του μετάλλου πάνω στο manufacturing grid. Για την ικανοποίηση αυτής της συνθήκης αρκεί οι νέες συντεταγμένες που θα προκύψουν μετά το scaling, να είναι ακέραια πολλαπλάσια του manufacturing grid. Ειδικά οι τιμές του width πρέπει να είναι άρτια ακέραια πολλαπλάσια του manufacturing grid. Όλα αυτά πρέπει να ληφθούν υπόψη για την υλοποίηση του αλγόριθμου από τον οποίο θα προκύψει η εφαρμογή εκτέλεσης του scaling. Συγκεκριμένα σε κάθε βήμα όπου κάποια τιμή αλλάζει μετά την κλιμάκωση που υφίσταται, ελέγχεται αν η προκύπτουσα τιμή είναι ακέραιο πολλαπλάσιο του manufacturing grid με την πράξη modulo(manufacturing grid). Αν το αποτέλεσμα είναι μηδενικό τότε συνεχίζουμε με την επόμενη παράμετρο, διαφορετικά αφαιρούμε από την τροποποιημένη παράμετρο το αποτέλεσμα της

πράξης modulo() και έτσι προκύπτει η τελική τιμή της παραμέτρου που είναι ταυτόχρονα ακέραιο πολλαπλάσιο του manufacturing grid. Να επισημανθεί εδώ πως το manufacturing grid δεν μπορεί να πάρει τιμές μικρότερες του 0.005 γιατί δεν υποστηρίζεται από το εργαλείο. Επίσης η ακρίβεια της τιμής του δεν πρέπει να είναι μεγαλύτερη από τρία δεκαδικά ψηφία.





Εικόνα 3.4.4 Αλγόριθμος για την κλιμάκωση (scaling) των γεωμετρικών και ηλεκτρικών διαστάσεων των επιπέδων μετάλλου και των cells του αρχείου lef.

Στο σχήμα που προηγήθηκε, έχει γίνει μια προσπάθεια αποτύπωσης του αλγορίθμου, για την κλιμάκωση (scaling) των γεωμετρικών και ηλεκτρικών παραμέτρων των διαφόρων επιπέδων, καθώς και των γεωμετρικών παραμέτρων των macros, που αντιστοιχούν στα cells της βιβλιοθήκης. Εδώ πρέπει να επισημανθούν τα σημεία

εκείνα, που αντιμετωπίζουν τόσο το πρόβλημα της εγκυρότητας των αποτελεσμάτων, ως προς το ακέραιο πολλαπλάσιο του manufacturing grid, όσο και της κατεύθυνσης στην οποία γίνεται η κλιμάκωση των αγωγών διασύνδεσης μέσα στο κάθε macro. Το πρώτο αντιμετωπίζεται, όπως βλέπουμε και στο διάγραμμα ροής, με την πράξη modulo (%). Ουσιαστικά εφαρμόζουμε modulo (manufacturing\_grid), στα αποτελέσματα που προκύπτουν μετά τη διαίρεση των τιμών των παραμέτρων, με τον παράγοντα scaling fn. Αν το αποτέλεσμα είναι μηδενικό, τότε πληρείται η πρώτη συνθήκη, διαφορετικά γίνεται διόρθωση στο εγγύτερο ακέραιο πολλαπλάσιο του manufacturing\_grid, αφαιρώντας την τιμή που προέκυψε από την πράξη modulo, από το αποτέλεσμα της διαίρεσης της αρχικής τιμής της παραμέτρου με τον παράγοντα scaling. Αν π.χ θέλουμε να κάνουμε scaling της τιμής 185 με τους παράγοντες scaling 5, 4, και 8. Ακόμη έστω ότι θέλουμε το αποτέλεσμα να είναι ακέραιο πολλαπλάσιο του 3. Τότε θα πάρουμε τα ακόλουθα αποτελέσματα:

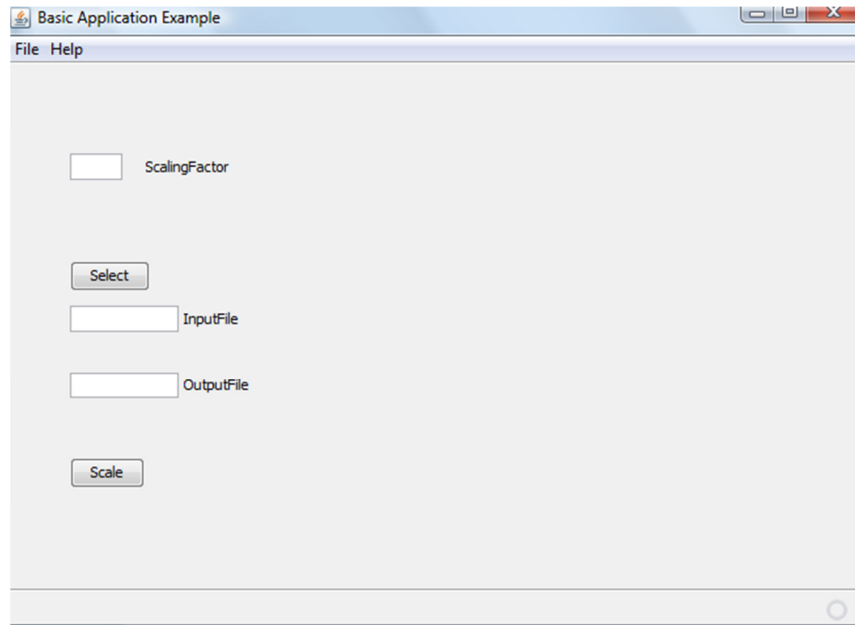
**Πίνακας 3.4.1 Παράδειγμα αντιμετώπισης του προβλήματος εξασφάλισης ακεραίας πολλαπλότητας δύο τιμών.**

185/fn	(185/fn)%3	185/fn-(185/fn)%3	(185/fn-(185/fn)%3)%3
46,25	1,25	45	0
37	1	36	0
23,125	2,125	21	0

Έτσι λοιπόν προκύπτουν αποτελέσματα, που δεν είναι ακέραια πολλαπλάσια του 3. Με την προσέγγιση όμως που κάνουμε, καταλήγουμε σε τιμές ακέραια πολλαπλάσια του 3. Οπότε η τιμή 46,25 γίνεται 45, η τιμή 37 γίνεται 36 και η τιμή 23,125 γίνεται 21.

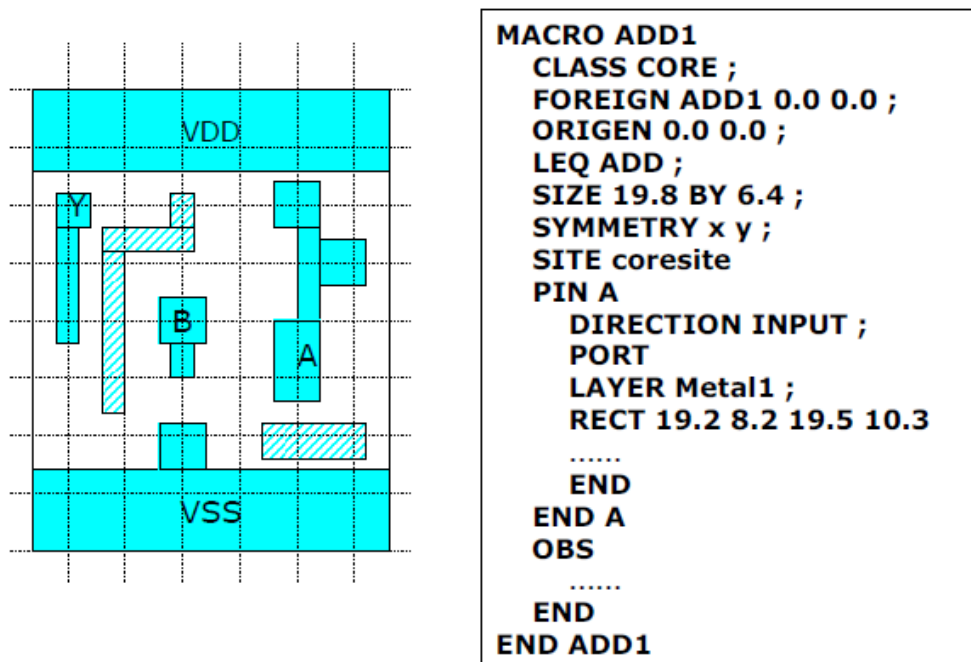
Όσο για τη δεύτερη συνθήκη της κατεύθυνσης προς την οποία γίνεται scaling, γίνεται έλεγχος σε ποιο επίπεδο μετάλλου του macro cell γίνεται η τροποποίηση, και με κατάλληλο flag (even\_flag) πληροφορούμε τον κώδικα, ώστε να εφαρμόσει τον προηγούμενο αλγόριθμο προς τον άξονα Y (περιττό επίπεδο μετάλλου) ή προς τον άξονα X (άρτιο επίπεδο μετάλλου). Ο αλγόριθμος στην παρούσα εργασία υλοποιήθηκε σε γλώσσα java, και δίνεται στο παράρτημα 1. Στο σχήμα που ακολουθεί δίνεται το γραφικό περιβάλλον, το οποίο είναι ίδιο με αυτό της προηγούμενης εφαρμογής του scaling της spice netlist. Όπως σε εκείνο το εργαλείο,

έτσι κι εδώ υπάρχουν τρία πεδία για εισαγωγή των ονομάτων των αρχείων εισόδου, εξόδου και του παράγοντα scaling. Τα αποτελέσματα καταγράφονται στο αρχείο εξόδου, το οποίο θα χρησιμοποιηθεί από το εργαλείο τοποθέτησης και διασύνδεσης (place and routing).



Εικόνα 3.4.5 Γραφικό περιβάλλον εφαρμογής για το scaling των παραμέτρων του LEF.

Ένα παράδειγμα τοποθέτησης και διασύνδεσης ενός macro cell ακολουθεί στο επόμενο σχήμα. Εδώ διακρίνεται και ο ρόλος του manufacturing grid.



Εικόνα 3.4.6 Παράδειγμα τοποθέτησης και διασύνδεσης ενός macro cell εφαρμόζοντας τους κανόνες που αναφέρονται στο κείμενο.

### 3.5. Δημιουργία Verilog

Για το αρχείο Verilog δε χρειάζεται κάποια ιδιαίτερη μέριμνα, αφού αυτό θα προκύψει μέσω του library characterizer, από το αρχείο ALF, το οποίο όπως αναφέρθηκε, είναι μια εναλλακτική μορφή του LIB αρχείου και μεταφέρει όλη την πληροφορία για timing – κατανάλωση, για καθένα cell της βιβλιοθήκης. Εφόσον λοιπόν προκύψει το σωστό LIB αρχείο, μετά την κλιμάκωση (scaling) του spice netlist και τον χαρακτηρισμό, μπορούμε να πάρουμε το αντίστοιχο Verilog για την κάθε βιβλιοθήκη.





## Κεφάλαιο 4

### Χαρακτηρισμός βιβλιοθήκης

#### 4.1. Σκοπός του χαρακτηρισμού

Χαρακτηρισμός ενός κυκλώματος, σαν γενική έννοια, είναι η διαδικασία κατά την οποία μετά από εκτελέσεις εξομοιώσεων του κυκλώματος κάτω από διάφορες συνθήκες λειτουργίας, καταγράφουμε αποτελέσματα χρονισμού και κατανάλωσης, ώστε να χρησιμοποιηθούν κατά τη χρήση του κυκλώματος σε κανονικές συνθήκες. Στην περίπτωση των βιβλιοθηκών των standard cells που χρησιμοποιούνται στην ψηφιακή σχεδίαση, γίνονται εξομοιώσεις όλων των cells της βιβλιοθήκης κάτω από διάφορες συνθήκες λειτουργίας, και καταγράφονται υποχρεωτικά τα αποτελέσματα χρονισμού και κατανάλωσης, ενώ σε ειδικές περιπτώσεις μπορούν να καταγραφούν αποτελέσματα για στατιστικά χρονισμού και στατικής κατανάλωσης.

Οι συνθήκες της εξομοίωσης αφορούν θερμοκρασία και τάση λειτουργίας (process corner), χωρητικό φορτίο που συνδέεται στην έξοδο και τιμή slew του σήματος εισόδου. Η καταγραφή των αποτελεσμάτων γίνεται με προκαθορισμένο τρόπο. Ένας τρόπος που ήδη είδαμε στην περιγραφή του LIB αρχείου είναι με look up tables.

Ο λόγος που γίνεται ο χαρακτηρισμός των cells και δεν γίνεται κάθε φορά εξομοίωση του πραγματικού κυκλώματος που προκύπτει από τη σύνθεση του συστήματος που σχεδιάζουμε, είναι ότι με τον χαρακτηρισμό μπορούμε να εργαζόμαστε σε πιο αφαιρετικό επίπεδο<sup>10</sup>. Δηλαδή γίνεται η εξομοίωση του συστήματος σε επίπεδο cell και όχι σε φυσικό επίπεδο (τρανζίστορ). Αυτό επιταχύνει σημαντικά την εξομοίωση, αφού προκειμένου να πάρουμε αποτελέσματα χρονισμού και κατανάλωσης για το σύστημα που σχεδιάζουμε, αρκεί να γίνει αναζήτηση στους σχετικούς look up πίνακες του κάθε cell, που συνθέτει το σύστημα. Επίσης η ανάδειξη της λειτουργικότητας είναι δύσκολη και μερικές φορές αδύνατο να γίνει από το φυσικό επίπεδο σχεδίασης. Ενώ είναι τεχνικά δύσκολη η αυτόματη ανίχνευση χρονικών περιορισμών (π.χ setup time).




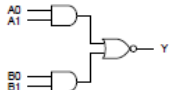
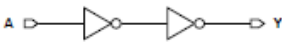
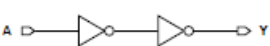
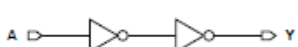
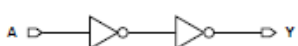
---

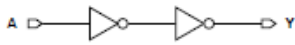
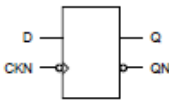
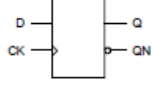
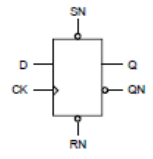
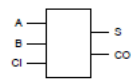
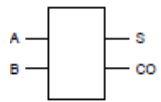
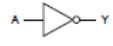
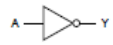
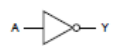
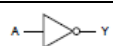
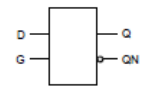
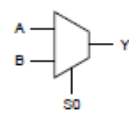
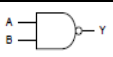
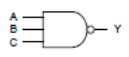
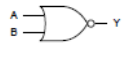
<sup>10</sup> Standard Cell Characterization by Richard Sohnus  
Computer Architecture Group  
Prof. Dr. U. Brüning  
University of Mannheim


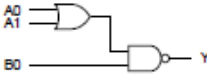
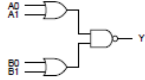




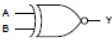

## 4.2. Περιγραφή βιβλιοθήκης standard cells που θα χαρακτηριστούν

Έχοντας εξαγάγει την netlist για τις τεχνολογίες 45nm, 32nm, 22nm και 16nm, ο χαρακτηρισμός θα κάνει εξομοίωση όλων των cells, που θα περιέχει η νέα βιβλιοθήκη και τα αποτελέσματα (χρονισμού – κατανάλωσης) θα καταγραφούν στα αντίστοιχα LIB, Verilog και ALF αρχεία. Αντίστοιχα τα νέα αρχεία LEF που δημιουργούνται, δίνουν τη δυνατότητα στο εργαλείο τοποθέτησης και διασύνδεσης, να παράγει τα αντίστοιχα layouts και να κάνει εξαγωγή παρασιτικών χωρητικότητας διασυνδέσεων και pins. Το είδος των cells που χαρακτηρίζονται προκύπτουν από την αρχική βιβλιοθήκη osu018\_stdcells. Πρόκειται για μια βιβλιοθήκη τεχνολογίας κατασκευής (fabrication) 180nm με 32 cells.

Πίνακας 4.2.1 Περιγραφή της βιβλιοθήκης.

Όνομα cell	Περιγραφή	Σύμβολο
AND2X1	Πύλη AND δύο εισόδων και δυνατότητα οδήγησης ένα	
AND2X2	Πύλη AND δύο εισόδων και δυνατότητα οδήγησης δύο	
AOI21X1	AND OR με τρεις εισόδους και δυνατότητα οδήγησης ένα	
AOI22X1	AND AND OR με τέσσερις εισόδους και δυνατότητα οδήγησης ένα	
BUFX2	Buffer με δυνατότητα οδήγησης δύο	
BUFX4	Buffer με δυνατότητα οδήγησης τεσσάρων	
CLKBUF1	Buffer με ισοσταθμισμένες καθυστερήσεις για σήματα ρολογιού και δυνατότητα οδήγησης ένα	
CLKBUF2	Buffer με ισοσταθμισμένες καθυστερήσεις για σήματα ρολογιού και δυνατότητα οδήγησης δύο	

CLKBUF3	Buffer με ισοσταθμισμένες καθυστερήσεις για σήματα ρολογιού και δυνατότητα οδήγησης τρία	
DFFNEGX1	D - Flip flop ακμοπυροδότητο στην αρνητική ακμή του ρολογιού με δυνατότητα οδήγησης ενός	
DFFPOSX1	D - Flip flop ακμοπυροδότητο στη θετική ακμή του ρολογιού με δυνατότητα οδήγησης ενός	
DFFSR	D - Flip flop που διαθέτει set- reset για ασύγχρονη θέση – επανάθεση (set- reset)	
FAX1	Πλήρης αθροιστής με δυνατότητα οδήγησης ένα	
HAX1	Ημιαθροιστής με δυνατότητα οδήγησης ένα	
INVX1	Αναστροφέας (inverter) με δυνατότητα οδήγησης ένα	
INVX2	Αναστροφέας (inverter) με δυνατότητα οδήγησης δύο	
INVX4	Αναστροφέας (inverter) με δυνατότητα οδήγησης τέσσερα	
INVX8	Αναστροφέας (inverter) με δυνατότητα οδήγησης οκτώ	
LATCH	Μανδαλωτής (latch)	
MUX2X1	Πολυπλέκτης δύο εισόδων με δυνατότητα οδήγησης ένα	
NAND2X1	Πύλη NAND δύο εισόδων με δυνατότητα οδήγησης ένα	
NAND3X1	Πύλη NAND τριών εισόδων με δυνατότητα οδήγησης ένα	
NOR2X1	Πύλη NOR δύο εισόδων με δυνατότητα οδήγησης ένα	

NOR3X1	Πύλη NOR τριών εισόδων με δυνατότητα οδήγησης ένα	
OAI21X1	OR AND τριών εισόδων με δυνατότητα οδήγησης ένα	
OAI22X1	OR AND τεσσάρων εισόδων με δυνατότητα οδήγησης ένα	
OR2X1	Πύλη OR δύο εισόδων με δυνατότητα οδήγησης ένα	
OR2X2	Πύλη OR δύο εισόδων με δυνατότητα οδήγησης ένα	
TBUF1	TRI state Buffer με δυνατότητα οδήγησης ένα	
TBUF2	TRI state Buffer με δυνατότητα οδήγησης δύο	
XNOR2X1	Πύλη XNOR δύο εισόδων με δυνατότητα οδήγησης ένα	
XOR2X1	Πύλη XOR δύο εισόδων με δυνατότητα οδήγησης ένα	

Αυτά τα cells θα περάσουν τη διαδικασία χαρακτηρισμού για διαφορετικές συνθήκες που θα μιλήσουμε στη συνέχεια.

### 4.3. Συνθήκες χαρακτηρισμού

#### 4.3.1. Συνθήκες περιβάλλοντος λειτουργίας και διαδικασία παραγωγής των τρανζίστορ

Για τις συνθήκες χαρακτηρισμού έχει γίνει ήδη μια μικρή αναφορά στην προηγούμενη παράγραφο. Η τάση λειτουργίας και η θερμοκρασία είναι δυο τέτοιες συνθήκες. Η τάση λειτουργίας (V<sub>dd</sub>) καθορίζεται από τις τάσεις κατωφλίου των τρανζίστορ και μεταβάλλεται πτωτικά από τεχνολογία σε τεχνολογία μικρότερων διαστάσεων (λόγω γενικού scaling που γίνεται από γενιά τεχνολογίας σε γενιά). Η θερμοκρασία περιβάλλοντος επηρεάζει σημαντικά τη θερμοκρασία λειτουργίας του ολοκληρωμένου. Γι αυτό πρέπει να ληφθεί υπόψη κατά τον χαρακτηρισμό. Εκτός από τη μεταβολή στις συνθήκες του περιβάλλοντος, στον χαρακτηρισμό πρέπει να ληφθεί

υπόψη η διακύμανση κάποιων χαρακτηριστικών των τρανζίστορ, που μπορεί να συμβεί κατά τη διαδικασία κατασκευής. Δηλαδή κατά την κατασκευή ομοίων τρανζίστορ, μπορεί να συμβεί εντελώς τυχαία διακύμανση σε χαρακτηριστικά τους, όπως το πλάτος των γραμμών, οι τάσεις κατωφλίου και το πάχος του οξειδίου πύλης  $\text{tox}$ . Οι μεταβολές αυτές είναι τυχαίες και περιγράφονται με στατιστικά μεγέθη, που ακολουθούν κανονική κατανομή.

Οι συνθήκες κάτω από τις οποίες γίνεται ο χαρακτηρισμός στο τρέχων έργο, αφορούν δύο γενικές περιπτώσεις.

- Η πρώτη περίπτωση αφορά μοντέλα τρανζίστορ γενικού σκοπού (general purpose), τα οποία είναι υπαρκτά για τις τεχνολογίες **45nm και 32nm** για τρία process corners (worst, typical, best)
  - Τα τρία process corners, διαφοροποιούνται μεταξύ τους στη φάση του χαρακτηρισμού, από τη θερμοκρασία λειτουργίας και την τάση κατωφλίου. Τα process corners αυτά είναι:
    - **Worst ή slow – slow** με θερμοκρασία λειτουργίας 125°C, και τάσεις κατωφλίου  $V_{tp}=0,432V$ ,  $V_{tn}=0,484V$  για τα 45nm και  $V_{tp}=0,463V$ ,  $V_{tn}=0,512V$  για τα 32nm.
    - **Typical** με θερμοκρασία λειτουργίας 27°C, και τάσεις κατωφλίου  $V_{tp}=0,418V$ ,  $V_{tn}=0,469V$  για τα 45nm και  $V_{tp}=0,452V$ ,  $V_{tn}=0,501V$  για τα 32nm.
    - **Best ή fast - fast** με θερμοκρασία λειτουργίας 0°C, και τάσεις κατωφλίου  $V_{tp}=0,372$ ,  $V_{tn}=0,424V$  για τα 45nm και  $V_{tp}=0,403V$ ,  $V_{tn}=0,452V$  για τα 32nm.
- Η δεύτερη περίπτωση αφορά μοντέλα τρανζίστορ ειδικού σκοπού για χαμηλή κατανάλωση (low power) ή υψηλή απόδοση (high performance). Για την περίπτωση αυτή είναι διαθέσιμα μοντέλα για τις τεχνολογίες 45nm, 32nm, 22nm και 16nm, για τυπικό **process corner**. Στο process corner αυτό, η θερμοκρασία είναι 27°C, και τάσεις κατωφλίου διαμορφώνονται σύμφωνα με τα δεδομένα που καταγράφονται στους συγκεντρωτικούς πίνακες που ακολουθούν.

## ○ Τεχνολογία 45nm

Process or Model	V <sub>tp</sub> [volt]	V <sub>tn</sub> [volt]	T[C]	V <sub>dd</sub> [volt]	V <sub>dd</sub> -V <sub>tp</sub> [Volt]	V <sub>dd</sub> -V <sub>tn</sub> [Volt]
Worst	0.432	0.484	125	1	0.568	0.516
Typical	0.418	0.469	27	1	0.582	0.531
Best	0.372	0.424	0	1	0.628	0.576
H_p	0.49158	0.46893	25	1	0.50842	0.53107
L_p	0.587	0.62261	25	1	0.413	0.37739

## ○ Τεχνολογία 32nm

Process or Model	V <sub>tp</sub> [volt]	V <sub>tn</sub> [volt]	T[C]	V[volt]	V <sub>dd</sub> -V <sub>tp</sub> [Volt]	V <sub>dd</sub> -V <sub>tn</sub> [Volt]
Worst	0.463	0.512	125	0.9	0.437	0.388
Typical	0.452	0.501	27	0.9	0.448	0.399
Best	0.403	0.452	0	0.9	0.497	0.448
H_p	0.49155	0.49396	25	0.9	0.40845	0.40604
L_p	0.5808	0.63	25	1	0.4192	0.37

## ○ Τεχνολογία 22nm

Model	V <sub>tp</sub> [volt]	V <sub>tn</sub> [volt]	T[C]	V[volt]	V <sub>dd</sub> -V <sub>tp</sub> [Volt]	V <sub>dd</sub> -V <sub>tn</sub> [Volt]
H_p	0.4606	0.50308	25	0.8	0.3394	0.29692
L_p	0.63745	0.68858	25	0.95	0.31255	0.26142

## ○ Τεχνολογία 16nm

Model	V <sub>tp</sub> [volt]	V <sub>tn</sub> [volt]	T[C]	V[volt]	V <sub>dd</sub> -V <sub>tp</sub> [Volt]	V <sub>dd</sub> -V <sub>tn</sub> [Volt]
H_p	0.43121	0.47965	25	0.7	0.26879	0.22035
L_p	0.6862	0.68191	25	0.9	0.2138	0.21809

Ο παρακάτω πίνακας συνοψίζει τις παραπάνω συνθήκες χαρακτηρισμού, για κάθε μία εκ' των οποίων πρέπει να προκύψει μία νέα βιβλιοθήκη.

**4.3.1 Συνοπτικός πίνακας με τις συνθήκες χαρακτηρισμού ανά τεχνολογία.**

	45nm	32nm	22nm	16nm
Worst (Slow) - (general purpose model)	*	*		
Typical - (general purpose model)	*	*		
Best (Fast) - (general purpose model)	*	*		
Typical - High performance model	*	*	*	*
Typical - Low power model	*	*	*	*

### **4.3.2. Επίδραση σημάτων εισόδου και χωρητικών φορτίων εξόδου**

Το φορτίο εξόδου και το slew του σήματος εισόδου καθιστούν τέλος συνθήκη από την οποία εξαρτάται ο χαρακτηρισμός, και από την επιλογή τους εξαρτάται η ακρίβεια των αποτελεσμάτων. Το μοντέλο υπολογισμού, είναι μέσω δισδιάστατων lookup table, όπου έχουν καταγραφεί κατά τις εξομοιώσεις του χαρακτηρισμού, τα αποτελέσματα για την καθυστέρηση και τη δυναμική κατανάλωση κάθε cell. Επειδή στους look up tables υπάρχουν, αποτελέσματα χρονισμού και κατανάλωσης, για κάποια φορτία εξόδου και slew σημάτων εισόδου, το εργαλείο εξομοίωσης, κάνει κάποιο interpolation στις τιμές ώστε να προκύψουν οι τιμές της κατανάλωσης και του χρονισμού, για τις πραγματικές τιμές του φορτίου εξόδου και slew του σήματος εισόδου. Εδώ πρέπει να δοθεί ιδιαίτερη σημασία, ώστε κατά τον χαρακτηρισμό να χρησιμοποιηθεί κατάλληλο εύρος τιμών φορτίων και slews, τέτοιο ώστε οι πραγματικές τιμές να είναι μέσα σ' αυτό. Έτσι δε θα χρειασθεί το εργαλείο να κάνει extrapolation για να υπολογίσει τιμές χρονισμού και κατανάλωσης, το οποίο θα εισήγαγε σημαντικό σφάλμα.

Για το λόγο αυτό γίνεται αρχικά χαρακτηρισμός για κάθε τεχνολογία (μόνο εξαγωγή LIB αρχείου), του απλούστερου cell. Ενός αναστροφέα, και επιλέγονται τιμές slew του σήματος εισόδου και Cload του χωρητικού φορτίου εξόδου, οι τιμές που ήδη υπάρχουν από τον χαρακτηρισμό βιβλιοθήκης σε τεχνολογία κατασκευής 90nm. Έχοντας κάνει φυσικά κάποιο scaling στις τιμές. Από τα αποτελέσματα προκύπτει για κάθε τεχνολογία η πραγματική χωρητικότητα εισόδου του αναστροφέα. Δηλαδή γνωρίζουμε τι χωρητικότητα “βλέπει” κάθε cell, με δυνατότητα οδήγησης ένα. Πολλαπλασιάζοντας το αποτέλεσμα x2, x4, x8, θα γνωρίζουμε την περιοχή τιμών του φορτίου, που πρέπει να κινηθούμε.

Για τις τιμές του Slew δεν υπάρχει εξάρτηση από τον βαθμό οδήγησης, αλλά μόνο από την τεχνολογία. Για την επιλογή τους (σε κάθε τεχνολογία) έγινε μια γραμμική κλιμάκωση αρχικώς, αλλά στη συνέχεια έγινε μια προσπάθεια βελτιστοποίησης μέσω πολλαπλών χαρακτηρισμών, με σκοπό την επιλογή των διαστημάτων εκείνων που δίνουν καλύτερα αποτελέσματα για την καθυστέρηση των cells.

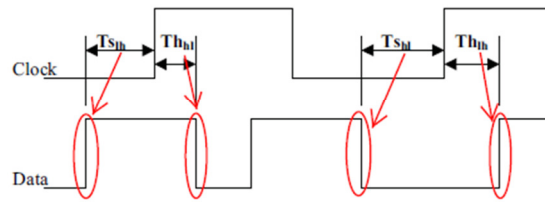
Οι τιμές τελικά που επιλέχθηκαν να χρησιμοποιηθούν για κάθε τεχνολογία και βαθμό οδήγησης (πόσα κυκλώματα μπορούν να οδηγηθούν από την έξοδο), περιγράφονται στον πίνακα που ακολουθεί. Οι μονάδες μέτρησης είναι για το slew σε nsec και για το φορτίο σε pF ή fF.



Πίνακας 4.3.2 Επιλογή περιοχής τιμών Slew και Cload για τον χαρακτηρισμό σε κάθε τεχνολογία.

<b>45nm</b>							
Slew	0.0385n	0.047n	0.08n	0.1n	0.2n	0.8n	1.0360n
Cload( X1)	0.00082p	0.00330p	0.00842p	0.01848p	0.03861p	0.07870p	0.18975p
Cload (X2)	0.00165p	0.00660p	0.01683p	0.03696p	0.07722p	0.15741p	0.37950p
Cload (X3)	0.00248p	0.00990p	0.02524p	0.05544p	0.11583p	0.23612p	0.56925p
Cload (X4)	0.00330p	0.01320p	0.03366p	0.07392p	0.15444p	0.31482p	0.75900p
Cload (X8)	0.00660p	0.02640p	0.06732p	0.14784p	0.30888p	0.62964p	1.51800p
<b>32nm</b>							
Slew	0.002n	0.004n	0.010n	0.018n	0.02n	0.028n	0.048n
Cload (X1)	0.8f	3.216f	8.208f	18.024f	37.668f	76.776f	185.12f
Cload (X2)	1.6f	6.432f	16.416f	36.048f	75.336f	153.552f	370.24f
Cload (X3)	3.2f	12.864f	32.832f	72.096f	150.672f	307.104f	740.48f
Cload (X4)	3.2f	12.864f	32.832f	72.096f	150.672f	307.104f	740.48f
<b>22nm</b>							
Slew	0.0038n	0.0044n	0.0080n	0.0100n	0.0200n	0.0500n	1.0360n
Cload (X1)	0.4f	1.608f	4.104f	9.012f	18.834f	38.388f	92.56f
Cload (X2)	0.8f	3.216f	8.208f	18.024f	37.66f	76.776f	185.12f
Cload (X3)	3.2f	12.864f	32.832f	72.096f	150.64f	307.104f	740.48f
Cload (X4)	3.2f	12.864f	32.832f	72.096f	150.64f	307.104f	740.48f
<b>16nm</b>							
Slew	0.0038n	0.008n	0.0100n	0.0200n	0.0400n	0.0600n	1.0360n
Cload (X1)	0.4f	1.608f	4.104f	9.012f	18.834f	38.388f	92.56f
Cload (X2)	0.8f	3.216f	8.208f	18.024f	37.66f	76.776f	185.12f
Cload (X3)	3.2f	12.864f	32.832f	72.096f	150.64f	307.104f	740.48f
Cload (X4)	3.2f	12.864f	32.832f	72.096f	150.64f	307.104f	740.48f

Κάτι άλλο που πρέπει να τονιστεί εδώ, είναι ο ορισμός των κατάλληλων Slews για τον υπολογισμό των setup και hold time των flip flops της βιβλιοθήκης. Οι χρόνοι αυτοί, θέτουν κάποια χρονικά όρια ασφαλείας που πρέπει να υπάρχουν ανάμεσα στην άφιξη των δεδομένων εισόδου και της άφιξης του ρολογιού.

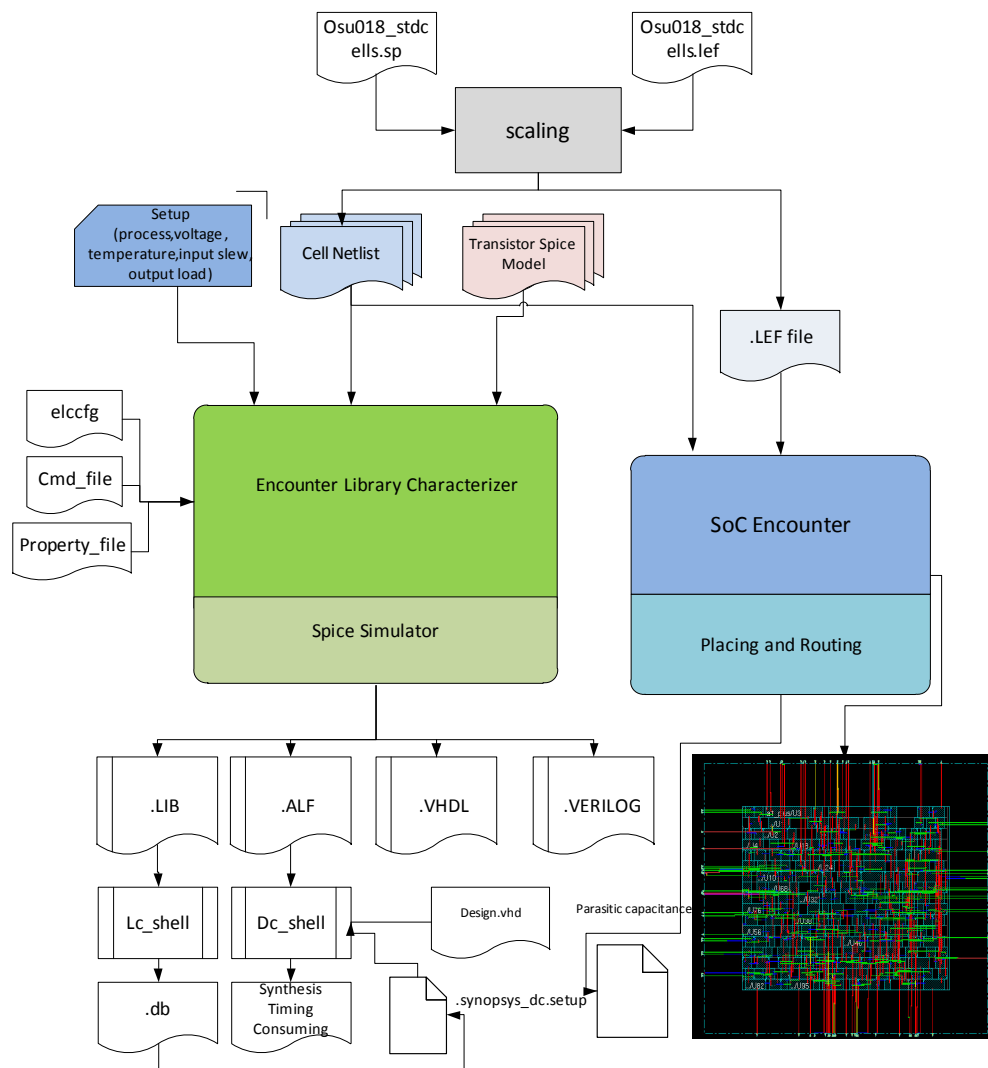


Εικόνα 4.3.1 Χρονισμός ακολουθιακών κυκλωμάτων.

Στο χρονικό διάστημα του χρόνου αποκατάστασης  $T_s$ , η είσοδος  $D$  θα πρέπει να παραμείνει σταθερή, διαφορετικά μπορεί να υπάρξει απροσδιοριστία στην τιμή που θα διαβιβαστεί προς την έξοδο. Επίσης στο διάστημα  $T_h$  δεξιά της ακμής του ρολογιού, πρέπει πάλι η είσοδος να κρατηθεί σταθερή. Οι χρόνοι αυτοί υπολογίζονται λαμβάνοντας υπόψη το SLEW, για κάθε ακολουθιακό cell, και το αποτέλεσμα καταγράφεται στο LIB αρχείο. Στην περίπτωση χαρακτηρισμού που εξετάζουμε, ορίζουμε τρεις τιμές για δυαδική αναζήτηση των τιμών του Slew. Ο χρόνος που απαιτείται μέχρι να εμφανισθεί το αποτέλεσμα στην έξοδο με σημείο αναφοράς τη θετική ακμή του ρολογιού, είναι η καθυστέρηση μέχρι την έξοδο, την οποία μετράμε και καταγράφουμε. Για τον χαρακτηρισμό διαλέγουμε τρεις τιμές για δυαδική αναζήτηση, με τιμές τέτοιες ώστε το ρολόι να καθυστερεί σε σχέση με τα δεδομένα (λόγω setup time) κατά 0.1, 0.4 και 0.7 ns.

## 4.4. Περιγραφή της διαδικασίας χαρακτηρισμού

Έχοντας αναπτύξει τις συνθήκες χαρακτηρισμού, μπορούμε να περιγράψουμε τα βήματα που πρέπει να εκτελεστούν για τον χαρακτηρισμό των βιβλιοθηκών. Το παρακάτω σχήμα στην Εικόνα 4.4.1 παριστάνει γραφικά τα βήματα αυτά. Όπως διακρίνεται, ο χαρακτηρισμός ακολουθεί δύο παράλληλες πορείες. Η μία αφορά την εξαγωγή των .LIB, Verilog αρχείων, και η άλλη την εξαγωγή του κατάλληλου LEF αρχείου, που θα χρησιμοποιηθεί για την εξαγωγή του layout και των χωρητικότητων διασύνδεσης.



Εικόνα

#### 4.4.1 Γραφική απεικόνιση των βημάτων χαρακτηρισμού.

Μπορούμε να διακρίνουμε, εκτός από τα γνωστά αρχεία που έχουμε ήδη συναντήσει, διάφορα άλλα που είναι απαραίτητα για την παραμετροποίηση και τη λειτουργία του Encounter Library Characterizer.

#### 4.4.1. Εξαγωγή των αρχείων LIB και Verilog

Ο Encounter Library Characterizer είναι το εργαλείο που στην έξοδό του θα μας δώσει τα αρχεία .LIB και Verilog, αφού πραγματοποιήσει μια σειρά εξομοιώσεων σε

όλα τα cells, για τις συνθήκες που του ορίζουμε μέσω των “βοηθητικών” αρχείων εισόδου. Απαραίτητα αρχεία για την λειτουργία του εργαλείου είναι:

- η γνωστή spice netlist των std\_cells, που έχει ήδη περιγραφεί, και έχει αναλυθεί εκτενώς ο τρόπος με τον οποίο γίνεται η κλιμάκωση (scaling) των διαστάσεων των τρανζίστορ τους.
- το spice model card των τρανζίστορ το οποίο για κάθε process corner και τεχνολογία είναι διαφορετικό.
- το αρχείο setup το οποίο είναι ένα ASCII αρχείο και περιέχει πληροφορίες για τις συνθήκες επεξεργασίας όπως θερμοκρασία, τάση τροφοδοσίας, τάσεις κατωφλίου των τρανζίστορ, process corner, χωρητικά φορτία εξόδου, slew σημάτων εισόδου και τον ορισμό του τρόπου μέτρησης των καθυστερήσεων. Εδώ θα εισαχθούν οι πίνακες με τα Slews και τα Clonds εάν το μοντέλο μέτρησης είναι με πίνακες αναζήτησης.
- Το command file που περιλαμβάνει με τη σειρά, όλες τις απαραίτητες εντολές εκτέλεσης κατά την εκκίνηση του Encounter characterizer.
- Το αρχείο elccfg περιέχει τις μεταβλητές και ντιρεκτίβες του περιβάλλοντος που θα πραγματοποιηθεί ο χαρακτηρισμός. Κατά την εκκίνηση του encounter library characterizer, γίνεται αρχικοποίηση του περιβάλλοντος φλοιού (>elc) διαβάζοντάς τις μεταβλητές περιβάλλοντος από το αρχείο αυτό.
- Το αρχείο Property file, φέρει πληροφορία για την επιφάνεια που καταλαμβάνει κάθε cell.

Τεχνικές λεπτομέρειες για τη χρησιμοποίηση και τη δομή των αρχείων setup, elccfg, cmd\_file και κάποια scripts tcl για αυτοματοποίηση κάποιων βημάτων δίνονται στο παράρτημα 2. Για την ολοκλήρωση του πρώτου σκέλους του σχήματος στην Εικόνα 4.4.1, πρέπει να γίνει η εξαγωγή του αρχείου .db. Ο Library compiler εκτελεί το βήμα αυτό, διαβάζοντας το LIB αρχείο και μεταγλωττίζοντάς το στο αντίστοιχο .db. Στη συνέχεια αυτό το αρχείο πρέπει να οριστεί ως η βιβλιοθήκη του Design compiler, για compile και link, μέσω του αρχείου .synopsys\_dc.setup. Με τον design compiler μπορούμε να κάνουμε στη συνέχεια σύνθεση οποιουδήποτε design, έχοντας σαν δομικά στοιχεία της σύνθεσης τα cells της νέας βιβλιοθήκης. Επίσης στην έξοδό του ο design compiler, θα δώσει

μια πρόβλεψη για τις τιμές της καθυστέρησης και της κατανάλωσης. Ο χειρισμός των δύο compilers, περιγράφεται στο παράρτημα 2.

#### 4.4.2. Εξαγωγή του LEF

Για την εξαγωγή του LEF, ακολουθείται η δεύτερη πορεία που περιγράφει η **Εικόνα 4.4.1**. Σύμφωνα λοιπόν με αυτήν την πορεία, μετά την επεξεργασία του αρχικού LEF που διαθέτουμε σε τεχνολογία 180nm, θα προκύψουν νέα, κατάλληλα για κάθε τεχνολογία. Αυτά θα δώσουν πληροφορίες σχετικά με τις διαστάσεις των cells και των διασυνδέσεων (εδώ δίνονται και ηλεκτρικά χαρακτηριστικά), στο εργαλείο που εμπλέκεται σ' αυτήν την πορεία, το SoC Encounter. Η εφαρμογή αυτή απαιτεί ως είσοδο, το κατάλληλα τροποποιημένο LEF αρχείο, που προκύπτει από το στάδιο της κλιμάκωσης (scaling) των ελάχιστων διαστάσεων και των ηλεκτρικών χαρακτηριστικών (σύμφωνα με τους κανόνες που αναπτύχθηκαν στο αντίστοιχο κεφάλαιο), το αρχείο LIB και το αρχείο Verilog του design, το οποίο περιγράφει τον τρόπο που θα γίνει η διασύνδεση των cells. Το νέο LEF αρχείο δηλαδή, έχει καταγεγραμμένα τα τροποποιημένα εμβαδά των cells και τις νέες ελάχιστες διαστάσεις των στοιχείων διασύνδεσης, προσαρμοσμένες στο manufacturing grid. Το αρχείο LIB έχει προέλθει επίσης από το αντίστοιχο στάδιο κλιμάκωσης, ενώ το Verilog αρχείο προκύπτει από τη σύνθεση του design.

### 4.5. Εκτέλεση του χαρακτηρισμού

Προκειμένου να υλοποιηθούν τα βήματα που περιγράφηκαν για τα δύο σκέλη, συγκεντρώνουμε όλα τα απαιτούμενα αρχεία που φαίνονται και στην **Εικόνα 4.4.1**, ανά τεχνολογία και process corner<sup>11</sup>.

#### 4.5.1. Εκτέλεση του χαρακτηρισμού με τον Encounter Library Characterizer

Αρχικά εκτελούμε το σκέλος παραγωγής του LIB. Η διαδικασία απαιτεί περίπου 15 min για να ολοκληρώσει όλες τις απαιτούμενες εξομοιώσεις, και να επαληθευθούν

<sup>11</sup> Υπάρχουν σε ηλεκτρονική μορφή σε CD που συνοδεύει την εργασία.



Για παράδειγμα το διάνυσμα `VECTOR(R0HL10DUHL) + ID(D0000) + POWER(D)`; δημιουργεί μια ακολουθία σημάτων εισόδου και αναμένει αντίστοιχα σήματα εξόδου με σκοπό την μέτρηση της ισχύος.

Αναλυτικότερα τα διανύσματα εξομοίωσης που δημιουργούνται για κάθε cell, καταγράφονται στον ακόλουθο πίνακα.

**Πίνακας 4.5.2 Σύνολο εξομοιώσεων που γίνονται ανά cell**

stimulus generation summary					
=====					
Name	#MOS	#DVEC	#RVEC		
AND2X1		6	8	0	
AND2X2		6	8	0	
AOI21X1		6	16	0	
AOI22X1		8	33	0	
BUFX2	4	2	0		
BUFX4	6	2	0		
CLKBUF1		16	2	0	
CLKBUF2		24	2	0	
CLKBUF3		32	2	0	
DFFNEGX1	22	10	4		
DFFPOSX1	22	10	4		
DFFSR	32	29	12		
FAX1	28	24	0		
HAX1	14	8	0		
INVX1	2	2	0		
INVX2	2	2	0		
INVX4	4	2	0		
INVX8	8	2	0		
LATCH	12	8	4		
MUX2X1		10	18	0	
NAND2X1		4	8	0	
NAND3X1		6	12	0	
NOR2X1		4	8	0	
NOR3X1		9	14	0	
OAI21X1		6	16	0	
OAI22X1		8	33	0	
OR2X1	6	8	0		
OR2X2	6	8	0		
TBUFX1		6	12	0	
TBUFX2		10	12	0	
XNOR2X1		12	8	0	
XOR2X1		12	8	0	
		337	24		

Όπως βλέπουμε τα διανύσματα είναι στο σύνολό τους R και D vectors, που βάσει των στοιχείων που δίνει ο Πίνακας 4.5.1 είναι επιβολή rising edge σε είσοδο, και αναμονή falling edge στην έξοδο. Επειδή λοιπόν πρέπει να καταγραφούν αποτελέσματα rise και fall χωριστά για κάθε είσοδο, για μέτρηση καθυστέρησης και δυναμικής ισχύος, αντιλαμβανόμαστε την αναγκαιότητα για μεγάλο αριθμό διανυσμάτων.

Μετά το στάδιο παραγωγής του LIB αρχείου, ακολουθεί η μεταγλώττισή του από τον Library compiler (lc\_shell), για να προκύψει το αρχείο .db. Η διαδικασία αυτή ολοκληρώθηκε άμεσα, δίνοντας το αναμενόμενο αρχείο .db, το οποίο ορίστηκε ως η βιβλιοθήκη για μεταγλώττιση και σύνδεση από τον design compiler (dc\_shell), μέσω του αρχείου .synopsys\_dc.setup.

#### **4.5.2. Χαρακτηρισμός των cells σε σχέση με την τοποθέτηση και τη διασύνδεσή τους μέσω του αρχείου LEF**

Απ' την άλλη πλευρά, ο SoC Encounter, αφού δεχτεί τα σωστά LEF αρχεία, σύμφωνα με τις προδιαγραφές που ορίστηκαν, εκτελεί τοποθέτηση και διασύνδεση μέσα στα χωρικά περιθώρια που έχουν οριστεί από το floor planning. Η χρονική καθυστέρηση για το routing είναι χωρίς ιδιαίτερη σημασία για τα κυκλώματα που δοκιμάζουμε, ενώ ο χώρος που καταλαμβάνει το κάθε cell, είναι ρητώς καθορισμένος από το LEF και σύμφωνα με τον πίνακα που ακολουθεί για κάθε τεχνολογία.



Πίνακας 4.5.3 Εμβαδό που καταλαμβάνει κάθε cell.

Όνομα macro cell	45nm	32nm	22nm	16nm
	Εμβαδό m <sup>2</sup>	Εμβαδό m <sup>2</sup>	Εμβαδό m <sup>2</sup>	Εμβαδό m <sup>2</sup>
and2x1	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
and2x2	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
AOI21X1	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
AOI22X1	7,96723E-12	7,9401E-12	4,97952E-12	3,97005E-12
BUFX2	5,96045E-12	4,7561E-12	2,98272E-12	2,37805E-12
BUFX4	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
CLKBUF1	1,79712E-11	1,4328E-11	8,9856E-12	7,164E-12
CLKBUF2	2,59384E-11	2,0696E-11	1,29792E-11	1,0348E-11
CLKBUF3	3,39356E-11	2,70441E-11	1,69603E-11	1,35221E-11
DFFNEGX1	2,39316E-11	1,90841E-11	1,19683E-11	9,54205E-12
DFFPOSX1	2,39316E-11	1,90841E-11	1,19683E-11	9,54205E-12
DFFSR	4,39096E-11	3,5024E-11	2,19648E-11	1,7512E-11
FAX1	2,9922E-11	2,38601E-11	1,49635E-11	1,19301E-11
HAX1	1,9948E-11	1,59001E-11	9,97152E-12	7,95005E-12
INVX1	3,98362E-12	3,184E-12	1,9968E-12	1,592E-12
INVX2	3,98362E-12	3,184E-12	1,9968E-12	1,592E-12
INVX4	5,96045E-12	4,7561E-12	2,98272E-12	2,37805E-12
INVX8	9,97402E-12	7,9401E-12	4,97952E-12	3,97005E-12
LATCH	1,39576E-11	1,11241E-11	6,97632E-12	5,56205E-12
MUX2X1	1,19508E-11	9,5321E-12	5,97792E-12	4,76605E-12
NAND2X1	5,96045E-12	4,7561E-12	2,98272E-12	2,37805E-12
NAND3X1	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
NOR2X1	5,96045E-12	4,7561E-12	2,98272E-12	2,37805E-12
NOR3X1	1,59644E-11	1,2736E-11	7,9872E-12	6,368E-12
OAI21X1	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
OAI22X1	9,97402E-12	7,9401E-12	4,97952E-12	3,97005E-12
OR2X1	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
OR2X2	7,96723E-12	6,368E-12	3,9936E-12	3,184E-12
TBUFX1	9,97402E-12	7,9401E-12	4,97952E-12	3,97005E-12
TBUFX2	1,39576E-11	1,11241E-11	6,97632E-12	5,56205E-12
XNOR2X1	1,39576E-11	1,11241E-11	6,97632E-12	5,56205E-12
XOR2X1	1,39576E-11	1,11241E-11	6,97632E-12	5,56205E-12

Παρατηρώντας τα στοιχεία του πίνακα, διαπιστώνουμε πως κατεβαίνοντας σε τεχνολογία, μειώνεται το εμβαδό των cells, όπως είναι αναμενόμενο άλλωστε, αλλά όχι σε αναλογία με το τετράγωνο της τιμής του συντελεστή κλιμάκωσης, όπως θα περιμέναμε, αλλά κατά προσέγγιση σε αναλογία με την τιμή του. Αυτό οφείλεται στο γεγονός ότι η κλιμάκωση (scaling), γίνεται μόνο κατά τη διεύθυνση  $Y$  αφενός, και αφετέρου, προσπαθώντας να κρατήσουμε το αποτέλεσμα ακέραιο πολλαπλάσιο του manufacturing grid, καταλήγουμε κάθε φορά στην εγγύτερη τιμή που ικανοποιεί αυτή τη συνθήκη.

# Κεφάλαιο 5

## Αποτελέσματα

Μετά τον επιτυχή χαρακτηρισμό των βιβλιοθηκών, ακολουθεί η επαλήθευση τόσο της σύνθεσης κάποιων κυκλωμάτων, όσο και της παραγωγής του layout και της εξαγωγής παρασιτικών χωρητικότητων.

Συγκεκριμένα για την επαλήθευση του πρώτου σκέλους, που ολοκληρώνεται με την σύνθεση του design και την πρόβλεψη των τιμών καθυστέρησης και κατανάλωσης, υλοποιήθηκαν τρία designs.

- Ένα design αθροιστή 32x32 bit.
- Ένα design πολλαπλασιαστή 16x16.
- Ένα design πολλαπλασιαστή 16x16 με καταχωρητές εισόδου εξόδου

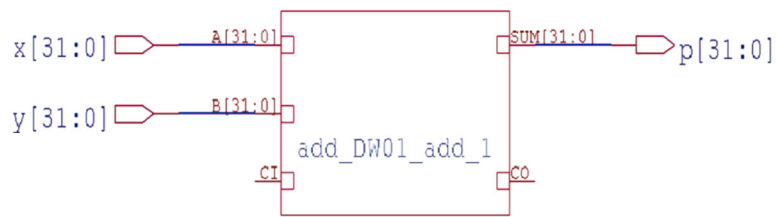
Πριν όμως δούμε τα αποτελέσματα για τα designs, θα δούμε τα αποτελέσματα χρονισμού και κατανάλωσης για τρία αντιπροσωπευτικά cells της βιβλιοθήκης. Αυτά είναι μια πύλη NAND, μια πύλη NOR και ένας αναστροφέας INVX1.

Αντίστοιχα για το σκέλος του χαρακτηρισμού που επαληθεύεται με την κατασκευή του layout σε επίπεδο πυλών, και την εξαγωγή παρασιτικών χωρητικότητων, έγινε δοκιμή στα ίδια designs και σε έναν απλό full adder.

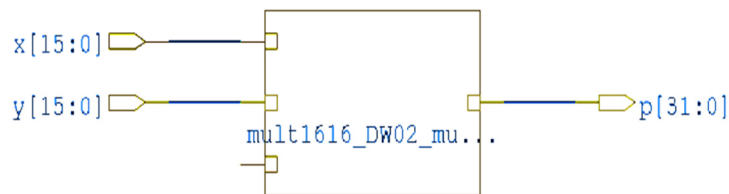
## 5.1. Αποτελέσματα σύνθεσης των designs, ανάλυσης του χρονισμού και της κατανάλωσής τους

### 5.1.1. Σύνθεση

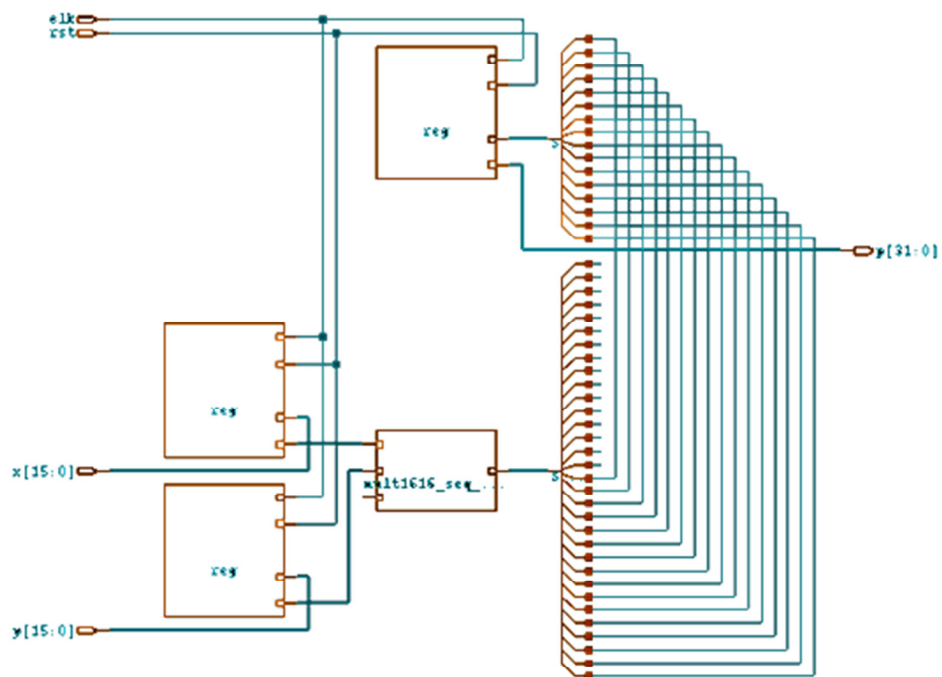
Στα σχήματα που ακολουθούν δίνονται τα designs που υλοποιήθηκαν, και βγήκαν από το εργαλείο SoC Encounter.



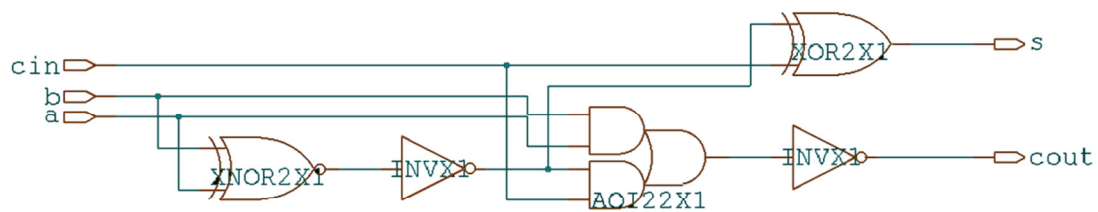
Εικόνα 5.1.1 Αθροιστής 32x32.



Εικόνα 5.1.2 Πολλαπλασιαστής 16x16.



Εικόνα 5.1.3 Πολλαπλασιαστής 16x16 με καταχωρητές στις εισόδους και την έξοδο.



Εικόνα 5.1.4 Πλήρης αθροιστής 1bit.

Η ανάπτυξη των designs γίνεται με γλώσσα περιγραφής vhdl. Η αρχιτεκτονική του αθροιστή 32x32 και των πολλαπλασιαστών 16x16, είναι συμπεριφοράς (behavioral), ενώ του full adder είναι δομική (structural). Η σύνθεση με τις νέες βιβλιοθήκες έγινε χωρίς κανένα πρόβλημα, για τα δύο είδη αρχιτεκτονικής, στην τεχνολογία 45 nm. Εν συνεχεία, με δεδομένη την σύνθεση των designs, γίνεται αντικατάσταση των cells τεχνολογίας 45nm με τα αντίστοιχα των άλλων τεχνολογιών. Κατά αυτόν τον τρόπο καταλήγουμε σε κοινά κυκλώματα ώστε να είμαστε σε θέση να συγκρίνουμε τα αποτελέσματα. Διαφορετικά κάνοντας σύνθεση για κάθε βιβλιοθήκη, ο design compiler την υλοποιεί κάθε φορά με διαφορετικό τρόπο, με μόνο κριτήριο να μην υπερβεί τον χρονικό περιορισμό που του ορίζουμε.

Σύμφωνα με στοιχεία που προκύπτουν από τα Verilog αρχεία των designs, αλλά και από το report του SoC Encounter μετά την τοποθέτηση των cells, για την σύνθεση των πιο πάνω έχουμε:

- Για τη σύνθεση του αθροιστή χρησιμοποιήθηκαν 233 cells.
- Για τη σύνθεση του πολλαπλασιαστή mult16x16 χρειάστηκαν 675 cells
- Για τον mult1616 με καταχωρητές, χρειάστηκαν 601 cells
- Για τον full adder χρειάστηκαν 5 cells

## 5.1.2. Αποτελέσματα χρονισμού

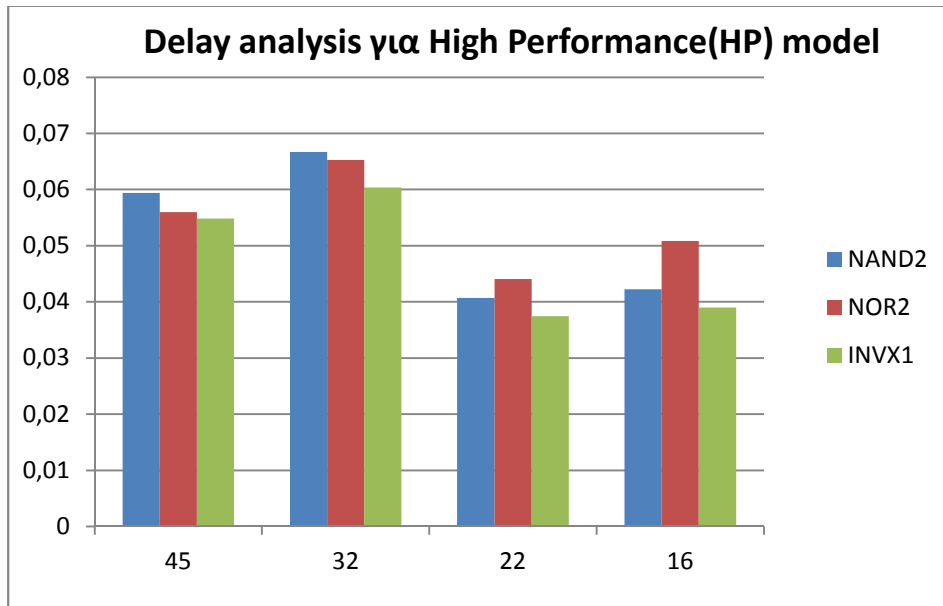
Τα αποτελέσματα για τη μέτρηση της καθυστέρησης, εκφράζονται σε nsec. Ο τρόπος μέτρησης της καθυστέρησης γίνεται για κάθε είσοδο των designs μέχρι τις εξόδους του.

### 5.1.2.1. Αποτελέσματα χρονισμού σε επίπεδο πύλης

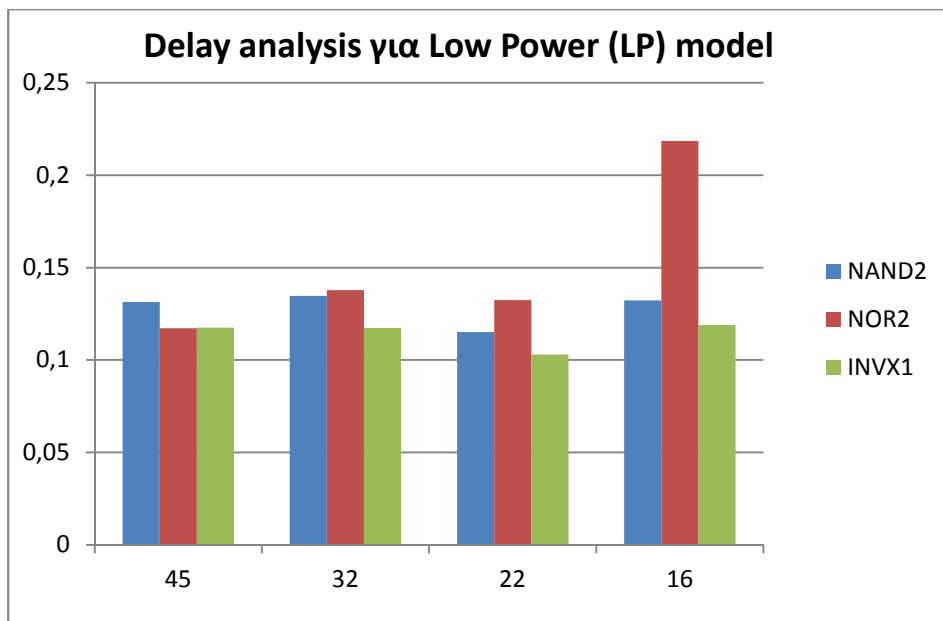
Αρχικά όπως αναφέρθηκε, παρουσιάζονται τα αποτελέσματα τριών βασικών cells, που χρησιμοποιούνται σε μεγάλο βαθμό κατά τη σύνθεση designs. Επιλέχθηκε ένα cell πύλης NAND, μιας πύλης NOR και ενός INVX1. Είναι επίσης σκόπιμο να γίνει αντιπαραβολή των αποτελεσμάτων των ίδιων cells, για όλες τις βιβλιοθήκες που κατασκευάστηκαν. Επειδή τα αποτελέσματα εδώ λαμβάνονται κατευθείαν από το LIB αρχείο, διαβάζοντας τους αντίστοιχους look up πίνακες, επιλέγουμε τις μεσαίες τιμές (median) των πινάκων. Όλα τα αποτελέσματα, σχετικά με τα τρία cells, δίνονται για τις ακόλουθες τιμές των φορτίων εξόδου και slew σήματος εισόδου.

**Πίνακας 5.1.1 Χωρητικά φορτία και slew σήματος εισόδου στα οποία αντιστοιχούν τα αποτελέσματα χρονισμού και κατανάλωσης των τριών βασικών πυλών που επιλέξαμε.**

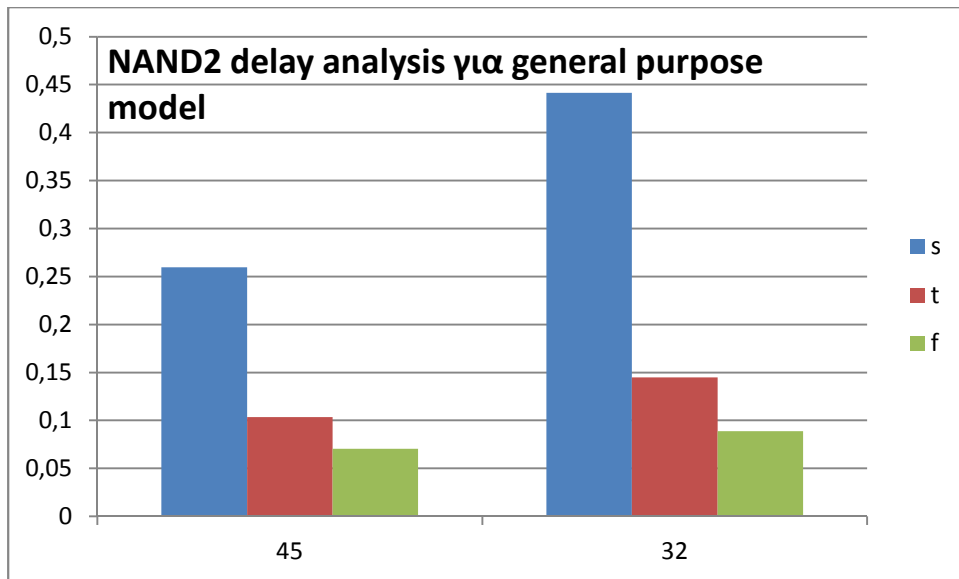
Τεχνολογία	CLOAD	SLEW
45NM	0,01848	0,06
32NM	0,018024	0,0108
22NM	0,009012	0,006
16NM	0,009012	0,012



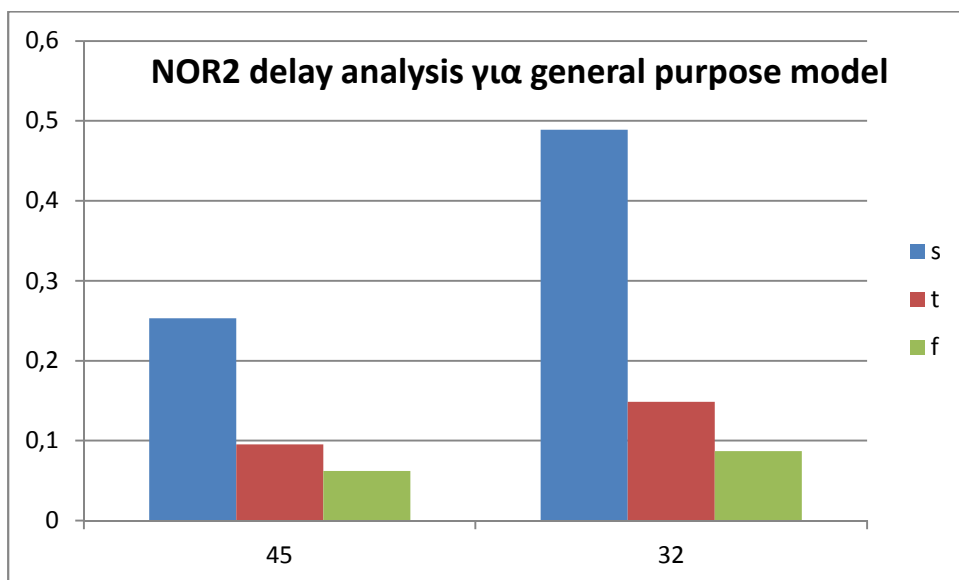
Εικόνα 5.1.5 Σύγκριση αποτελεσμάτων χρονισμού για κάθε αντιπροσωπευτικό cell, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).



Εικόνα 5.1.6 Σύγκριση αποτελεσμάτων χρονισμού για κάθε αντιπροσωπευτικό cell, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).

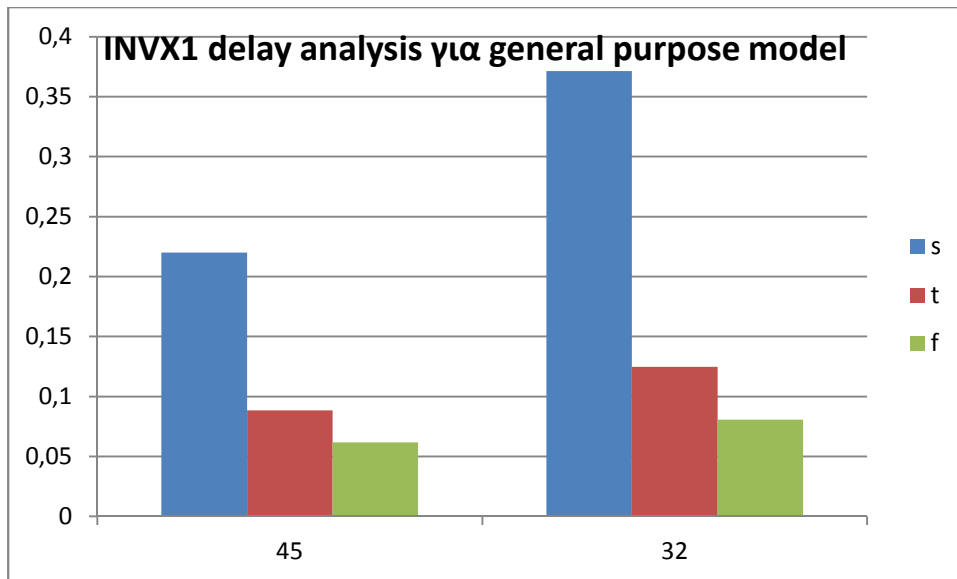


Εικόνα 5.1.7 Σύγκριση αποτελεσμάτων χρονισμού για το cell NANDx2, για general purpose model και process corner tt, ss και ff.



Εικόνα 5.1.8 Σύγκριση αποτελεσμάτων χρονισμού για το cell NORx2, για general purpose model και για process corner tt, ss και ff.





Εικόνα 5.1.9 Σύγκριση αποτελεσμάτων χρονισμού για το cell INVX1, για general purpose model και για process corner tt, ss και ff.

Τα αποτελέσματα παρατηρούμε ότι απέχουν από τα όσα αναφέραμε στη θεωρία. Από τη θεωρία είδαμε πως από γενιά σε γενιά προκύπτει μείωση της καθυστέρησης κατά 30%. Για τις τεχνολογίες που εργαστήκαμε όμως, κινούμαστε σε οριακά σημεία κλιμάκωσης των τάσεων λειτουργίας και κατωφλίου (Ειδικά για τις τάσεις κατωφλίου, παρατηρείται αύξηση της τάσης κατωφλίου σε μερικές περιπτώσεις). Έτσι θα περίμενε κανείς να δει μια σταθεροποιητική τάση για την καθυστέρηση. Αυτό δεν γίνεται όμως εμφανές στα αποτελέσματά μας. Αντίθετα, ιδιαίτερα στη μετάβαση από τα 45nm στα 32nm, παρατηρείται μια αύξηση γύρω στο 50%.

Σύμφωνα με το “νόμο της δύναμης του α” για την καθυστέρηση, έχουμε:

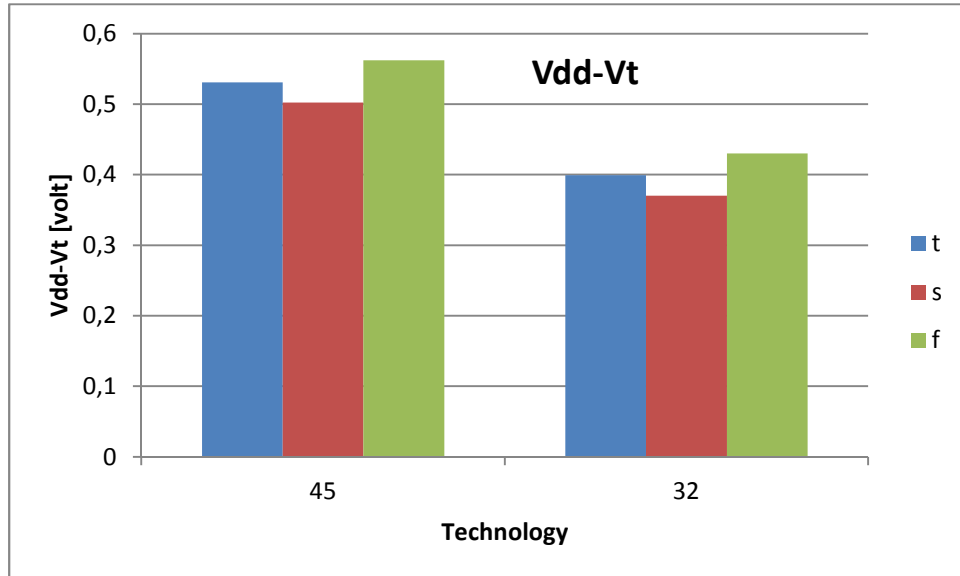
$$d = \frac{C_L \cdot V_{dd}}{(V_{gs} - V_{th})^\alpha}$$

Όπου  $d$  η καθυστέρηση,  $C_L$  το φορτίο,  $V_{gs}$  η τάση πύλης,  $V_{th}$  η τάση κατωφλίου και  $V_{dd}$  η τάση λειτουργίας. Αν λάβουμε υπόψη πως η τάση  $V_{gs}$  φθάνει μέχρι την τιμή  $V_{dd}$ , η πιο πάνω σχέση γίνεται:

$$d = \frac{A}{(V_{dd} - V_{th})^\alpha}$$

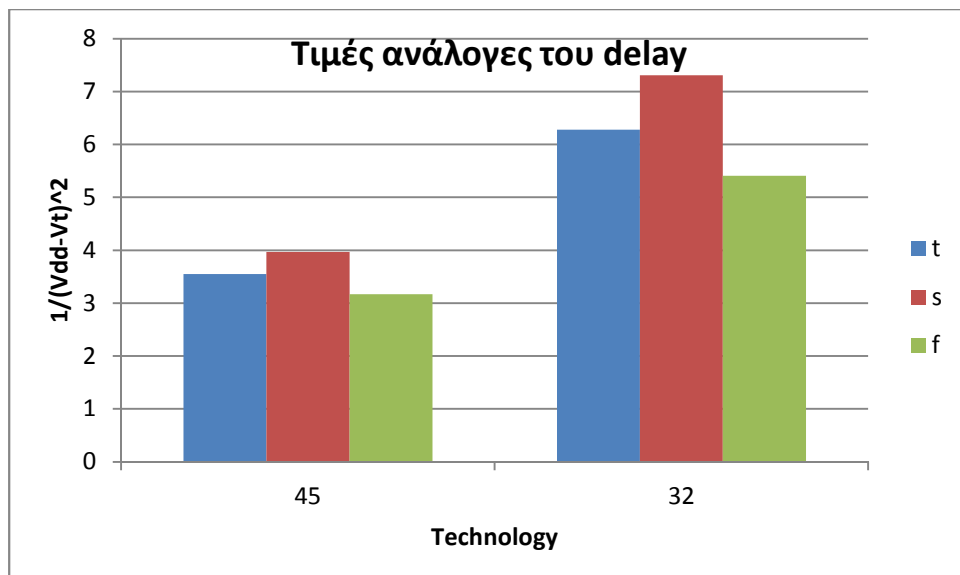
Με  $A = C_L \cdot V_{dd}$ . Δηλαδή η καθυστέρηση  $d$  είναι ανάλογη της σχέσης  $\frac{1}{(V_{dd} - V_{th})^\alpha}$ .

Για τον λόγο αυτό παραθέτουμε στοιχεία σχετικά με την διαφορά  $V_{dd} - V_{th}$ , για τις δυο τεχνολογίες 45nm και 32nm.



Εικόνα 5.1.10 Διαγράμματα της διαφοράς  $V_{dd} - V_{th}$  των τρανζίστορ σε κάθε τεχνολογία.

Στη συνέχεια σχεδιάζουμε τα αντίστοιχα διαγράμματα για τον λόγο  $\frac{1}{(V_{dd} - V_{th})^\alpha}$  θέτοντας  $\alpha=2$ .

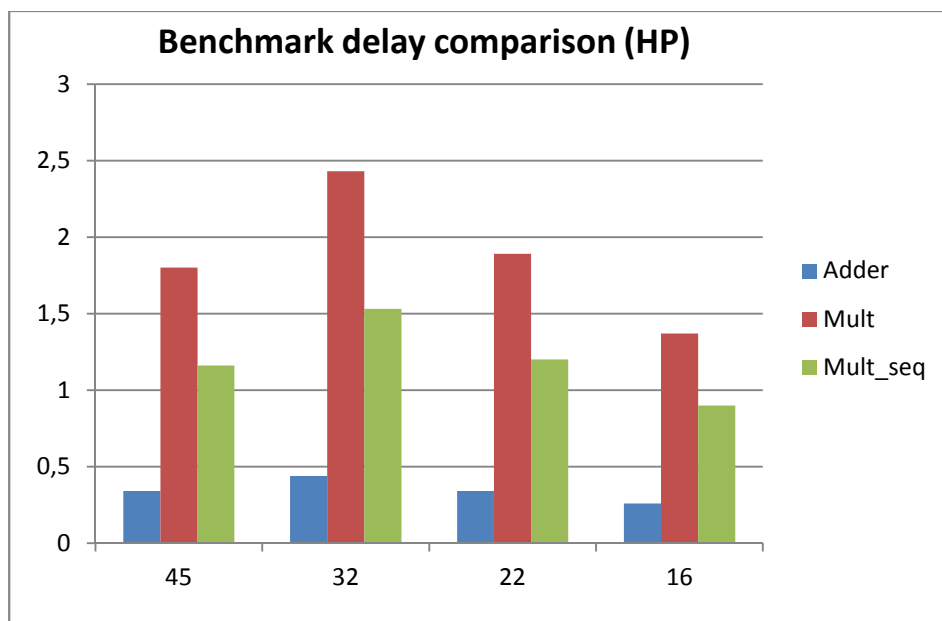


Εικόνα 5.1.11 Παράσταση της ποσότητας  $1/(V_{dd} - V_{th})^\alpha$ .

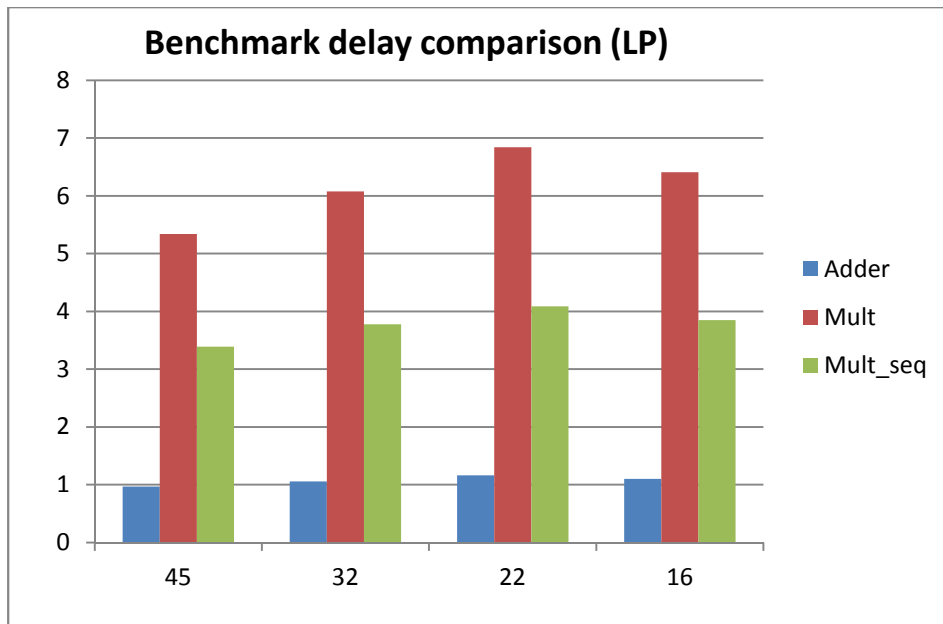
Οι παραπάνω προτάσεις επαληθεύονται, αν συγκρίνουμε με την καθυστέρηση κάποιου cell. Βλέπουμε τότε πως η εικόνα των γραφημάτων είναι η ίδια.

### 5.1.2.2. Αποτελέσματα χρονισμού για designs

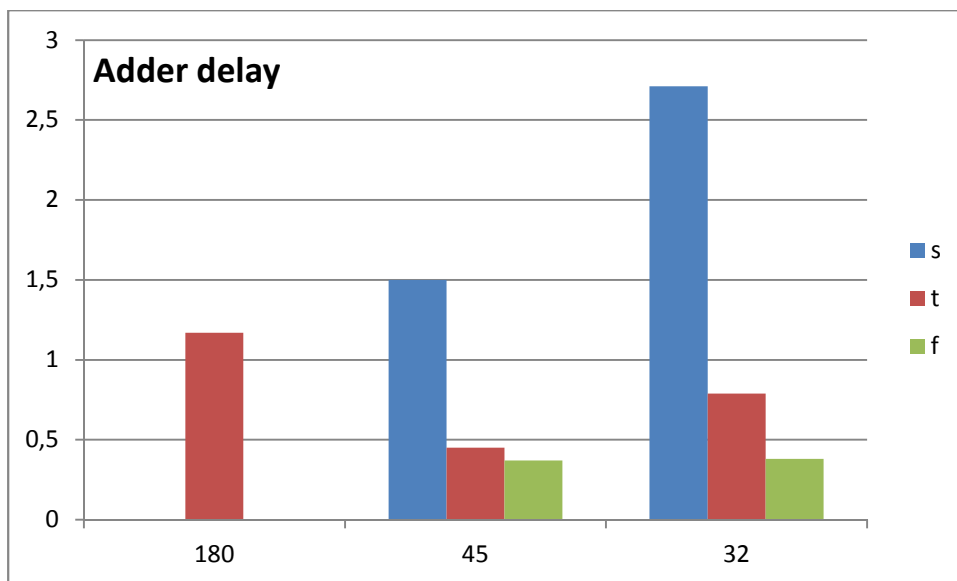
Ακολουθώντας τον ίδιο τρόπο αναπαράστασης των αποτελεσμάτων, όπως έγινε με τα cells, δίνονται οι τιμές για την καθυστέρηση που μετρήθηκε στα τρία designs. Στον αθροιστή 32x32, στον πολλαπλασιαστή 16x16 και στον πολλαπλασιαστή 16x16 με καταχωρητές στις εισόδους και στην έξοδο.



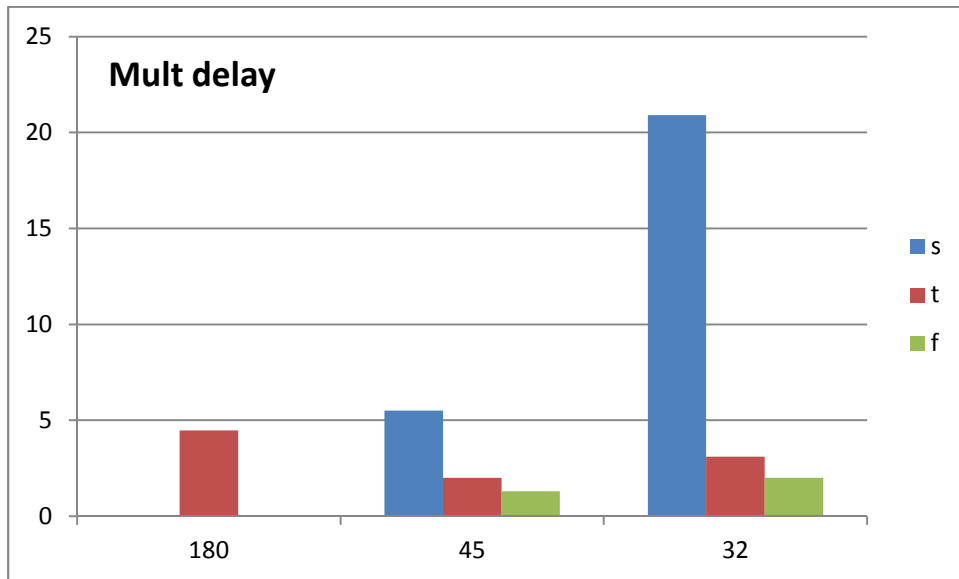
Εικόνα 5.1.12 Σύγκριση αποτελεσμάτων χρονισμού για κάθε design, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance) και process corner typical



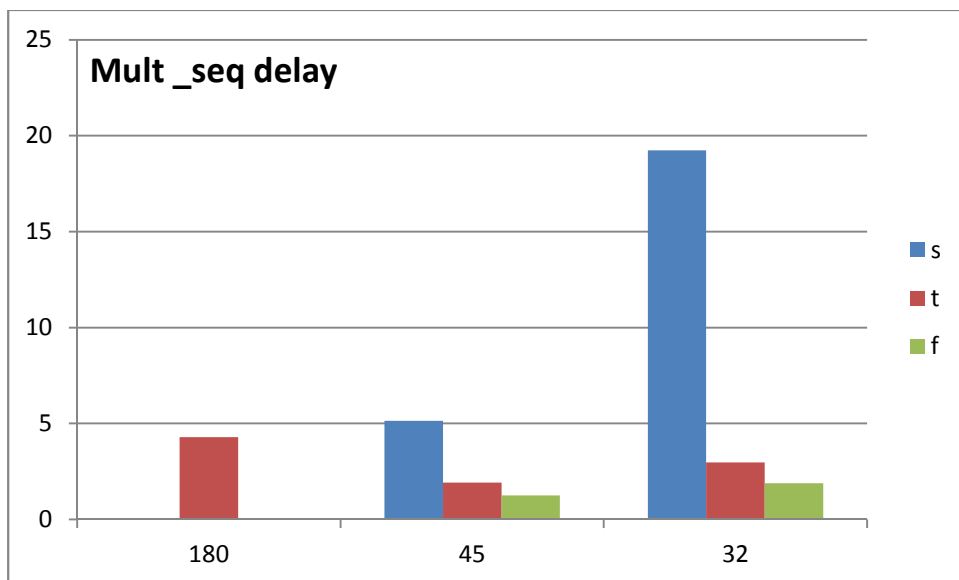
Εικόνα 5.1.13 Σύγκριση αποτελεσμάτων χρονισμού για κάθε design, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power) και process corner typical.



Εικόνα 5.1.14 Σύγκριση αποτελεσμάτων χρονισμού για το design adder 16x16, για process corner typical, slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



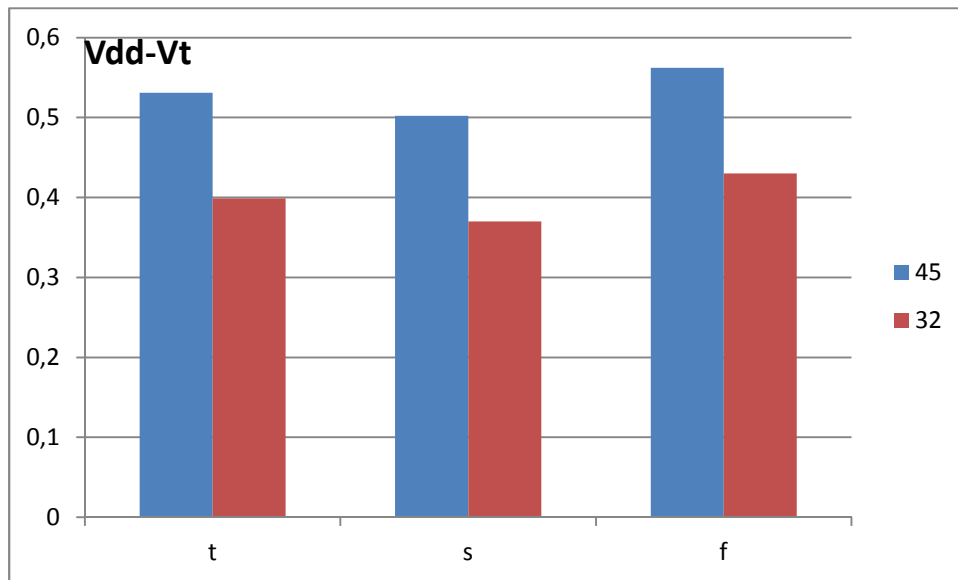
Εικόνα 5.1.15 Σύγκριση αποτελεσμάτων χρονισμού για το design multiplier 32x32, για process corner typical, slow –slow, fast –fast και γενικό μοντέλο τρανζίστορ.



Εικόνα 5.1.16 Σύγκριση αποτελεσμάτων χρονισμού για το design multiplier 32x32 με καταχωρητές στις εισόδους και την έξοδο, για process corner typical, slow –slow, fast –fast και γενικό μοντέλο τρανζίστορ.

Όπως με τα μεμονωμένα cells, έτσι κι εδώ παρατηρείται η ίδια εικόνα. Ισχύει δηλαδή ο νόμος της δύναμης του “α” και τα διαγράμματα της καθυστέρησης, είναι σε

αναλογία με τα διαγράμματα της έκφρασης  $\frac{1}{(V_{dd}-V_{th})^2}$ . Ειδικά για το process corner slow – slow στην τεχνολογία 32nm, παρατηρείται σοβαρή αύξηση της καθυστέρησης και για τα τρία designs. Αντίστοιχα παρατηρώντας το διάγραμμα του μεγέθους ( $V_{dd}-V_t$ ), βλέπουμε πως η χαμηλότερη τιμή του λαμβάνεται για process corner slow – slow και τεχνολογία 32nm.



Εικόνα 5.1.17 Σύγκριση τιμών του μεγέθους  $V_{dd}-V_t$  για process corner ss, ff και typical και τεχνολογίες 45 και 32nm.

Άρα επειδή η έκφραση  $\frac{1}{(V_{dd}-V_{th})^2}$  μεγιστοποιείται για process corner ss στην τεχνολογία 32nm, η καθυστέρηση θα έχει αντίστοιχη πορεία.

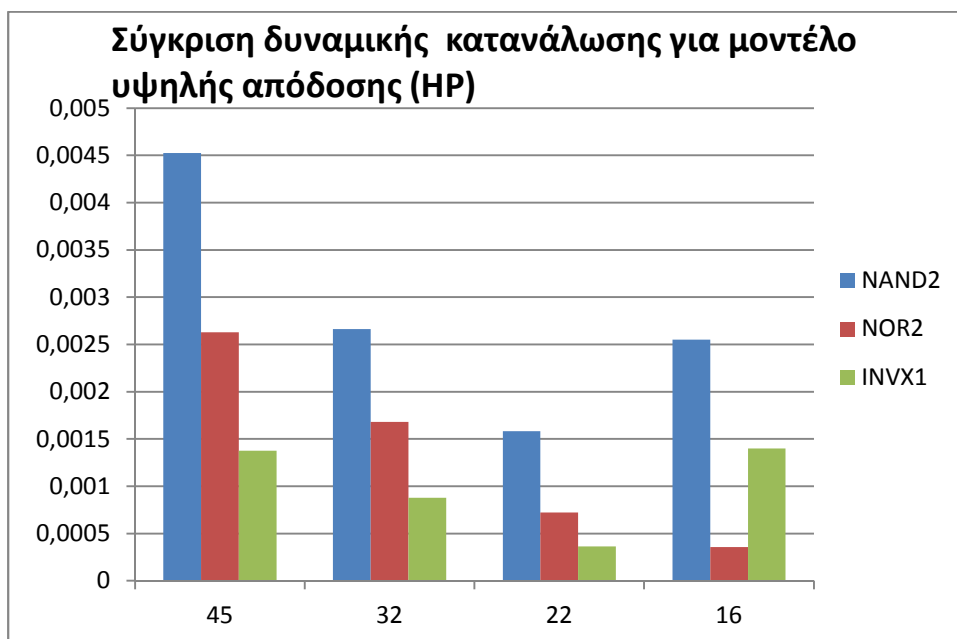
### 5.1.3. Αποτελέσματα κατανάλωσης

Εδώ καταγράφονται και παρουσιάζονται τα αποτελέσματα της δυναμικής και στατικής κατανάλωσης, τόσο σε επίπεδο πύλης, όσο και σε επίπεδο design. Ειδικότερα για την περίπτωση των αποτελεσμάτων σε επίπεδο πύλης, υπενθυμίζεται ότι η ανάγνωση των τιμών κατανάλωσης, γίνεται όπως και στην καθυστέρηση, κατευθείαν από τους look up πίνακες. Οι τιμές που παριστάνονται δηλαδή είναι οι μεσαίες τιμές των πινάκων αυτών και αντιστοιχούν στις τιμές φορτίου και Slew σύμφωνα με τα στοιχεία που δίνει ο Πίνακας 5.1.1.

### 5.1.3.1. Αποτελέσματα δυναμικής κατανάλωσης

Η δυναμική κατανάλωση που δίνεται από τη σχέση  $P_{dynamic} = V^2 * C * f * a$ , (όπου V η τάση τροφοδοσίας, C η χωρητικότητα του φορτίου, f η συχνότητα και a δραστηριότητα του cell) (3). όπως είναι αναμενόμενο αυξάνεται στις τεχνολογίες μεγαλύτερων διαστάσεων εξαιτίας της υψηλότερης τάσης τροφοδοσίας και της μεγαλύτερης χωρητικότητας που παρουσιάζουν τα pin εισόδου των cells. Ας δούμε όμως τι αποτελέσματα παίρνουμε στην πραγματικότητα, τόσο για κάθε ένα από τα τρία cells, όσο και για κάθε design. Τα αποτελέσματα για τη μέτρηση της στατικής και δυναμικής κατανάλωσης είναι σε  $\mu W$ .

#### 5.1.3.1.1. Αποτελέσματα δυναμικής κατανάλωσης σε επίπεδο πύλης



Εικόνα 5.1.3.1 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).

Όσον αφορά τα αποτελέσματα για process corner typical και μοντέλο υψηλής απόδοσης, παρατηρούμε πως υπάρχει πτώση της δυναμικής κατανάλωσης κατά τη μετάβαση από την τεχνολογία 45nm προς τις επόμενες. Εκτός από τα cells NAND2 και INVX1, που παρουσιάζουν μια αύξηση της δυναμικής κατανάλωσης για την τεχνολογία 16nm σε σχέση με την 22nm (INVX1 και NAND2) και 32nm (INVX1).

Για τις τιμές fall power της τεχνολογίας 16nm έχουμε 0.000112 μw για τον αναστροφέα έναντι 0.00024 μw για την αντίστοιχη fall power τιμή στην τεχνολογία 22nm και 0.000431 μw για την τεχνολογία 32nm. Επίσης για την πύλη NAND2 έχουμε fall power 0.000256 μw για την τεχνολογία 16nm έναντι 0.000515 μw για την αντίστοιχη κατανάλωση της τεχνολογίας 22nm. Η διαφορά μεταξύ rise και fall οφείλεται στα διαφορετικά χαρακτηριστικά των τρανζίστορ, που είναι υπεύθυνα για τις rise και fall τιμές. Τις rise τιμές τις καθορίζουν τα PMOS, που παρουσιάζουν δύο με τρεις φορές μειωμένη κινητικότητα φορέων, σε σχέση με τα αντίστοιχα NMOS. Διαφορές των μοντέλων PMOS που χρησιμοποιούνται στην τεχνολογία 16nm, σε σχέση με τα αντίστοιχα των τεχνολογιών 22 και 32nm, είναι η αιτία που προκαλεί μεγαλύτερο rise power για την τεχνολογία 16nm. Αντίθετα στην fall power παρατηρείται φυσιολογική εξέλιξη.

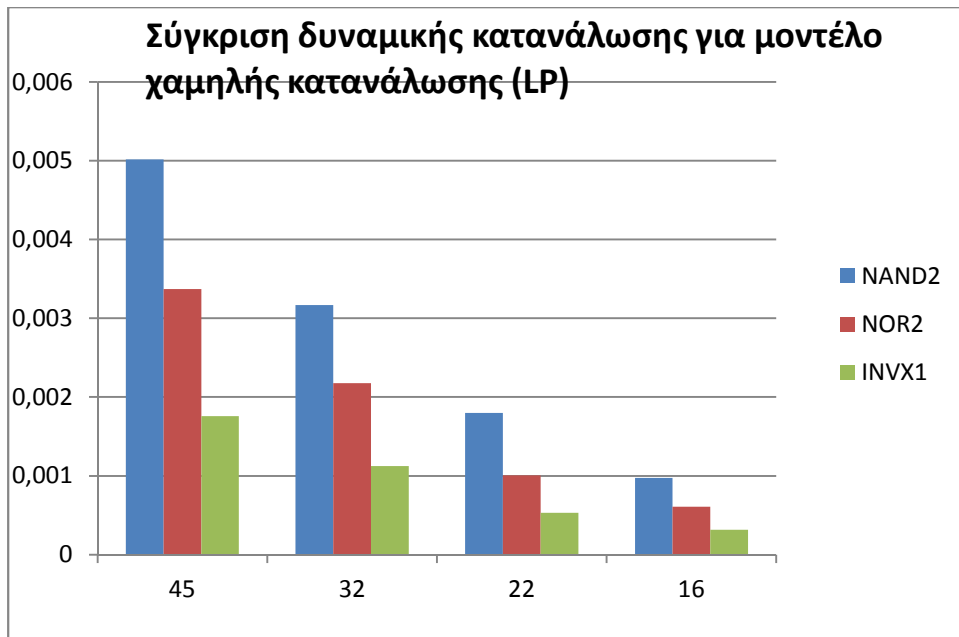
Επίσης στην δυναμική ισχύ περιλαμβάνεται και η ισχύς βραχυκυκλώματος, η οποία είναι ισχύς που καταναλώνεται κατά την μετάβαση 0->1 ή 1->0, όπου για πολύ μικρό χρονικό διάστημα άγουν και τα δύο τρανζίστορ ενός CMOS κυκλώματος. Η ισχύς αυτή δίνεται από τον τύπο:

$$P_{sc} = \frac{k}{12} (V_{DD} - 2V_t)^{\frac{3t_r}{t_p}}$$

Όπου  $k=k_p=k_n$ ,  $t_r$  ο χρόνος rise time και  $t_p$  η περίοδος του παλμού εισόδου. Έτσι με αντιστοιχία μεγαλύτερου rise time στην τιμή που επιλέξαμε, έχουμε εκθετική αύξηση της ισχύος βραχυκύκλωσης.

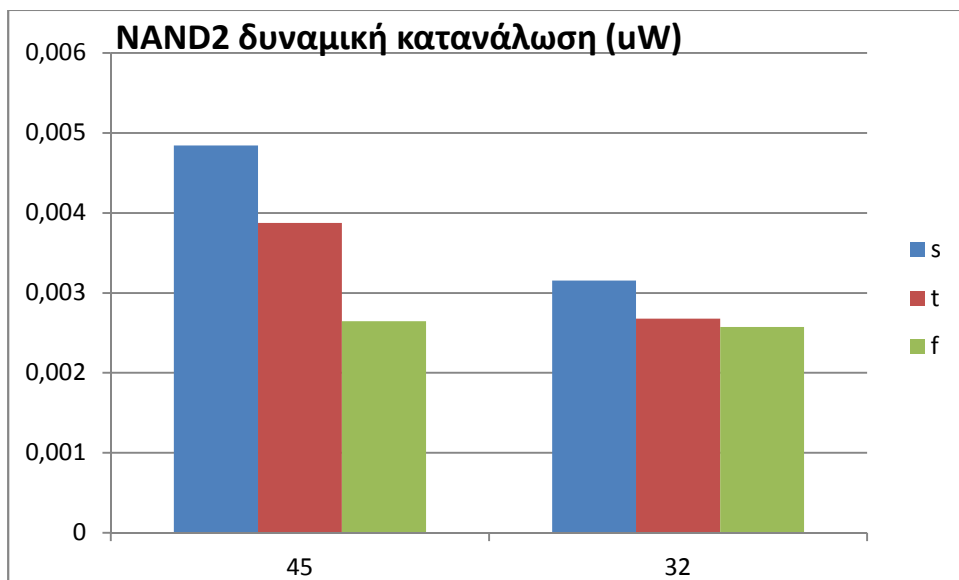
Γενικότερα συμπεραίνουμε, πως η τάση της δυναμικής ισχύος είναι πτωτική, καθώς περνάμε σε επόμενη γενιά τεχνολογίας.



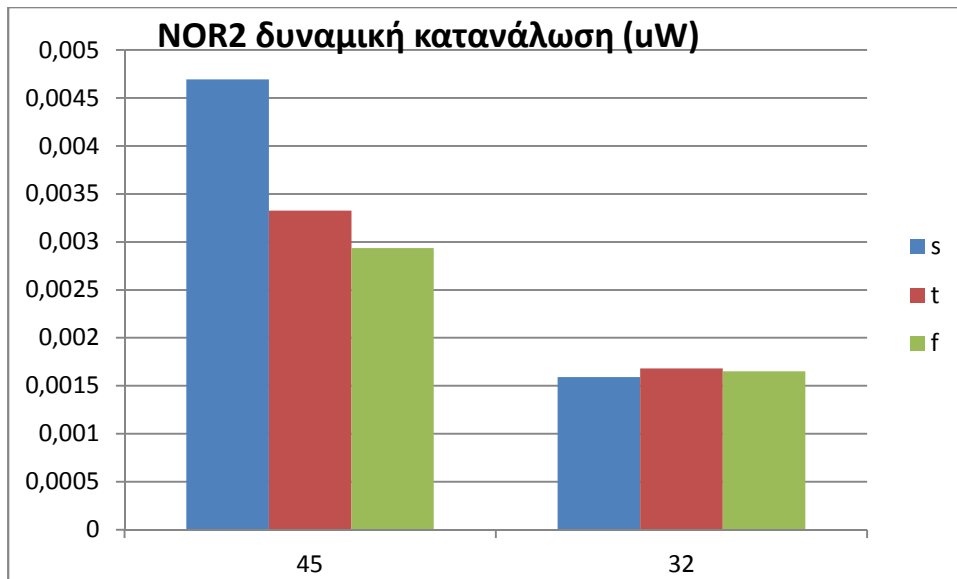


Εικόνα 5.1.3.2 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).

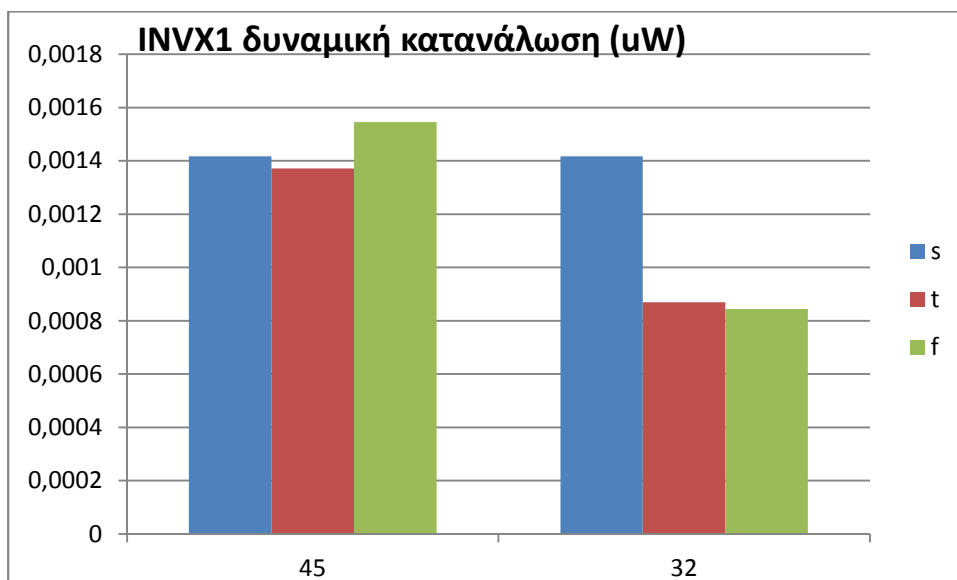
Η διαπίστωση ότι η δυναμική κατανάλωση μειώνεται καθώς μειώνονται οι διαστάσεις της τεχνολογίας, για το μοντέλο χαμηλής κατανάλωσης (LP) είναι ξεκάθαρη. Βλέπουμε την πτωτική πορεία που ακολουθεί η δυναμική κατανάλωση, καθώς περνάμε σε επόμενη γενιά τεχνολογίας.



Εικόνα 5.1.3.3 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για το cell NANDx2, για process corners tt, ss και ff.



Εικόνα 5.1.3.4 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για το cell NORx2, για process corners tt, ss και ff.



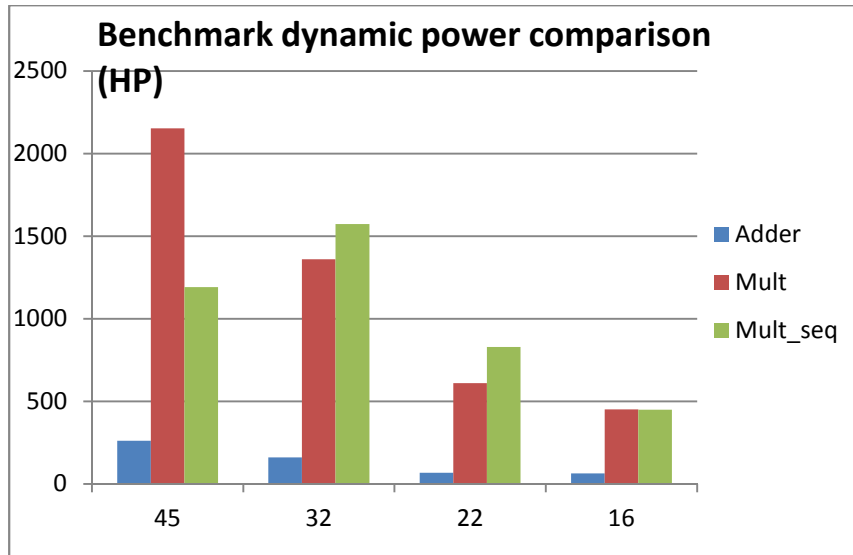
Εικόνα 5.1.3.5 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για το cell INVX1, για process corners tt, ss και ff.

Η πτωτική τάση της δυναμικής ισχύος, στα προηγούμενα γραφήματα, διακρίνεται καλύτερα για κάθε process corner.

### 5.1.3.1.2. Αποτελέσματα δυναμικής κατανάλωσης σε επίπεδο κυκλώματος

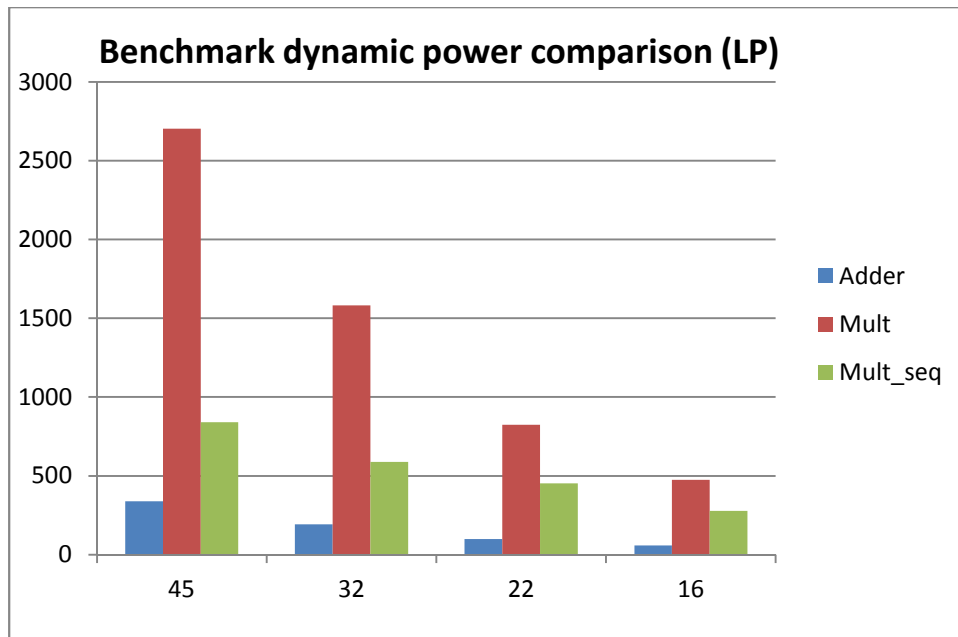
Με τον ίδιο τρόπο αναπαράστασης των αποτελεσμάτων, δίνονται οι τιμές για την δυναμική κατανάλωση που μετρήθηκε στα τρία designs. Στον αθροιστή 32x32, στον

πολλαπλασιαστή 16x16 και στον πολλαπλασιαστή 16x16 με καταχωρητές στις εισόδους και στην έξοδο. Αναμένεται κι εδώ όπως και στις μεμονωμένες πύλες, πτωτική τάση της δυναμικής ισχύος, όσο πηγαίνουμε σε μικρότερες διαστάσεις.



Εικόνα 5.1.3.6 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance) και process corner typical.

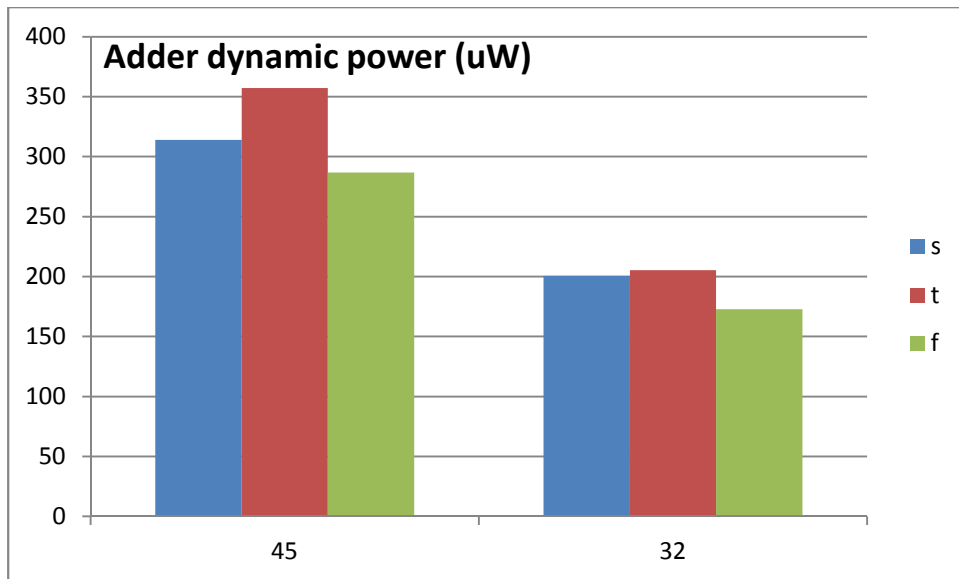
Για το μοντέλο υψηλής απόδοσης, η φθίνουσα πορεία της κατανάλωσης είναι εμφανής, αν και για το ακολουθιακό κύκλωμα του πολλαπλασιαστή, υπάρχει μια παραβίαση του κανόνα, κατά την μετάβαση από την τεχνολογία 45nm στην τεχνολογία 32nm. Εδώ βέβαια η διαφορά δεν οφείλεται στους λόγους που αναπτύχθηκαν στα μεμονωμένα cells, αλλά σε παράγοντες που αφορούν την ανάλυση που έκανε ο design compiler και σχετίζονται με τη δραστηριότητα των επιμέρους δομικών μονάδων του design, μέσα στο χρονικό παράθυρο της εξομοίωσης.



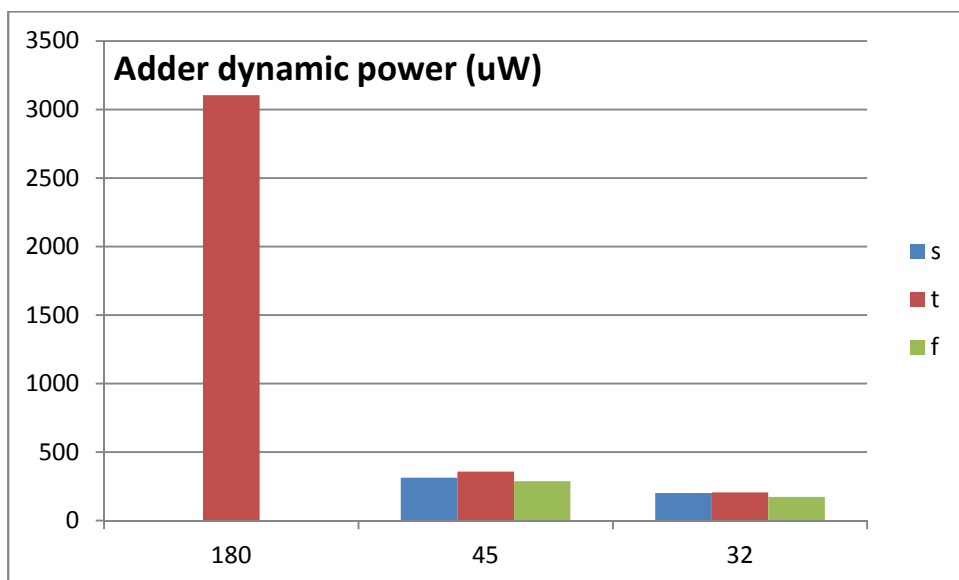
Εικόνα 5.1.3.7 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power) και process corner typical.

Τα πράγματα είναι σαφώς καλύτερα για το μοντέλο χαμηλής κατανάλωσης, όπου επαληθεύεται η θεωρητική σχέση υπολογισμού, της δυναμικής κατανάλωσης.

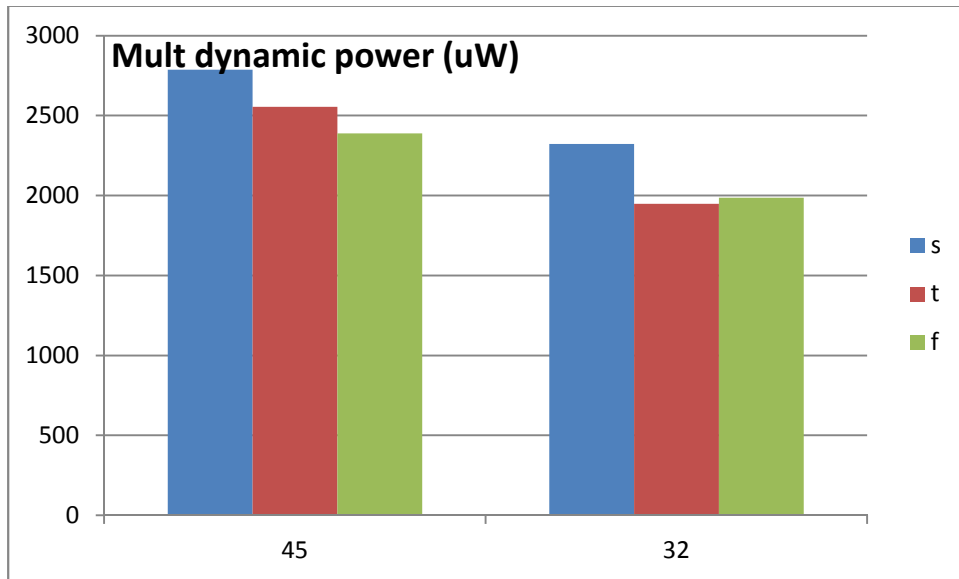
Επειδή, για τη δυναμική κατανάλωση η διαφορά μεταξύ αποτελεσμάτων στην τεχνολογία των 180nm, σε σχέση με των τεχνολογιών 45, 32, 22, και 16nm είναι μεγάλη, δίνονται τα ίδια διαγράμματα μια φορά με την 180nm και μια φορά χωρίς, με σκοπό την ακριβέστερη παρουσίασή τους.



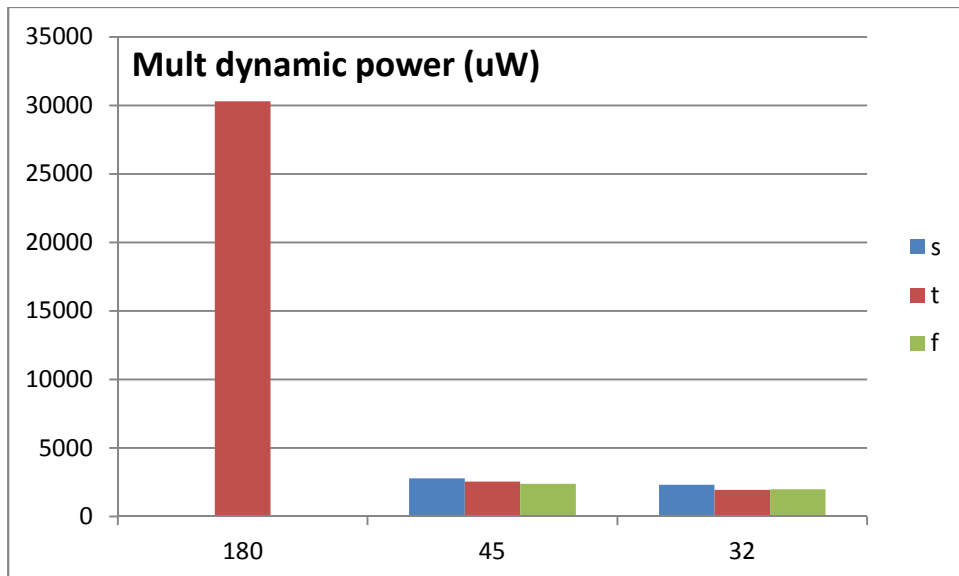
Εικόνα 5.1.3.8 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης, για το design adder 32x32 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



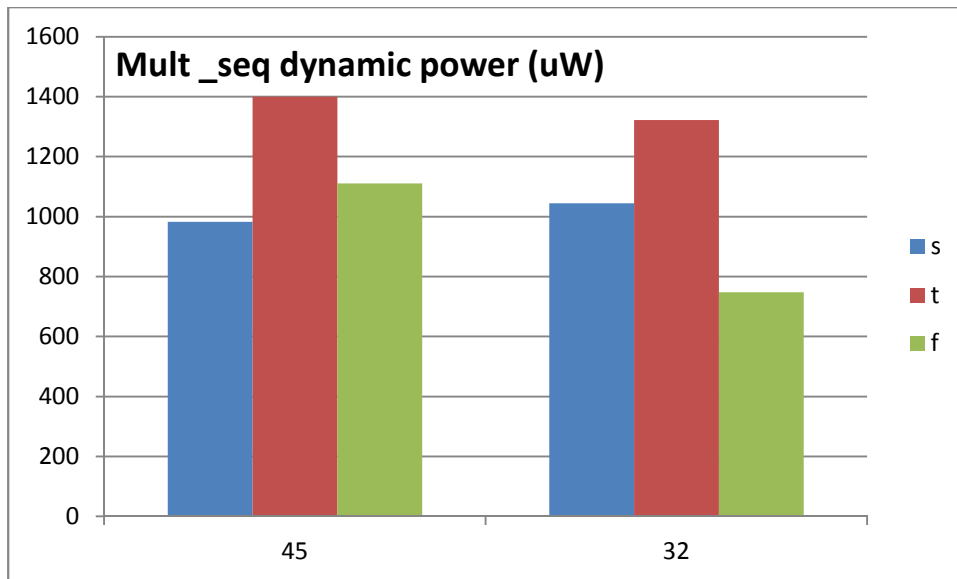
Εικόνα 5.1.3.9 Ίδιο με την προηγούμενη εικόνα, με αποτελέσματα και για τα 180nm.



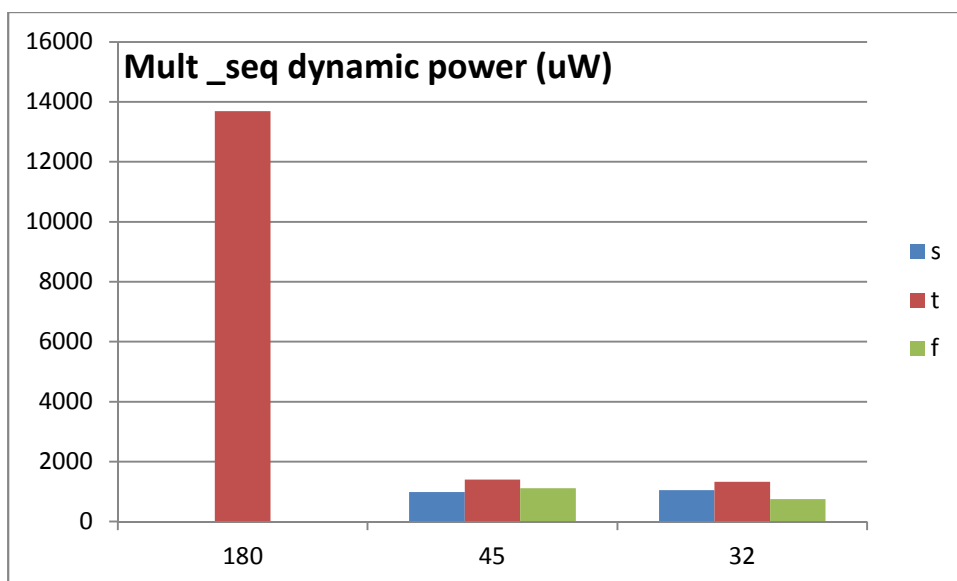
Εικόνα 5.1.3.10 Σύγκριση δυναμικής κατανάλωσης, για το design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



Εικόνα 5.1.3.11 Ίδιο με την προηγούμενη εικόνα, με αποτελέσματα και για τα 180nm.



Εικόνα 5.1.3.12 Σύγκριση αποτελεσμάτων δυναμικής κατανάλωσης, για το design multiplier 16x16 με καταχωρητές στις εισόδους και στην έξοδο, για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



Εικόνα 5.1.3.13 Ίδιο με την προηγούμενη εικόνα, με αποτελέσματα και για τα 180nm.

Σαν γενική παρατήρηση σχετικά με τα αποτελέσματα, είναι η σημαντική μείωση της δυναμικής κατανάλωσης ανάμεσα στα δύο άκρα των τεχνολογιών που ερευνούμε. Αυτό σημαίνει πως υπάρχει μείωση της χωρητικότητας εισόδου των pins, πέραν της μείωσης της τάσης τροφοδοσίας που ήδη γνωρίζουμε πως συμβαίνει. Έτσι η ισχύς  $P = C * V^2 * f$  μειώνεται όπως επιβεβαιώνουν τα αποτελέσματα στην πλειοψηφία τους.

### 5.1.3.2. Αποτελέσματα στατικής κατανάλωσης

Στις παλαιότερες τεχνολογίες, η στατική κατανάλωση ήταν ένα πολύ μικρό ποσοστό της συνολικής ισχύος ενός cell. Η ύπαρξη στατικής ισχύος, παρά το γεγονός ότι στην CMOS τεχνολογία καταναλώνεται ισχύς όταν τα τρανζίστορ δραστηριοποιούνται, οφείλεται στα ανάστροφα ρεύματα των επαφών.

Τα ρεύματα αυτά είναι:

- Sub – threshold leakage το οποίο εμφανίζεται όταν το τρανζίστορ είναι σε κατάσταση αποκοπής.
- Gate leakage εξαιτίας φαινομένου tunneling, μεταξύ του ημιαγωγού κάτω από το οξειδίο της πύλης και της πύλης του τρανζίστορ. Το ρεύμα αυτό αυξάνεται σημαντικά για πάχος οξειδίου μικρότερο των 3nm και για μεγάλες τάσεις πύλης.
- Band to Band Tunneling Leakage. Το ρεύμα αυτό δημιουργείται εξαιτίας της ανάστροφης πόλωσης των επαφών substrate – drain και substrate – source και εξαρτάται από τη νόθευση του ημιαγωγού του substrate.

Όσο πηγαίνουμε σε μικρότερων διαστάσεων τεχνολογία, παρατηρείται αύξηση των (ανάστροφων) ρευμάτων διαρροής (leakage) μεταξύ substrate και διαχύσεων. Η τιμή της στατικής ισχύος δίνεται από τη σχέση

$$P_s = \sum_{i=1}^n (i_{0i}) V_{DD}$$

Όπου  $i_0$  το ρεύμα της ανάστροφα πολωμένης επαφής που δίνεται από τη γνωστή σχέση:

$$i_0 = i_s \left( e^{qV/kT} - 1 \right)$$

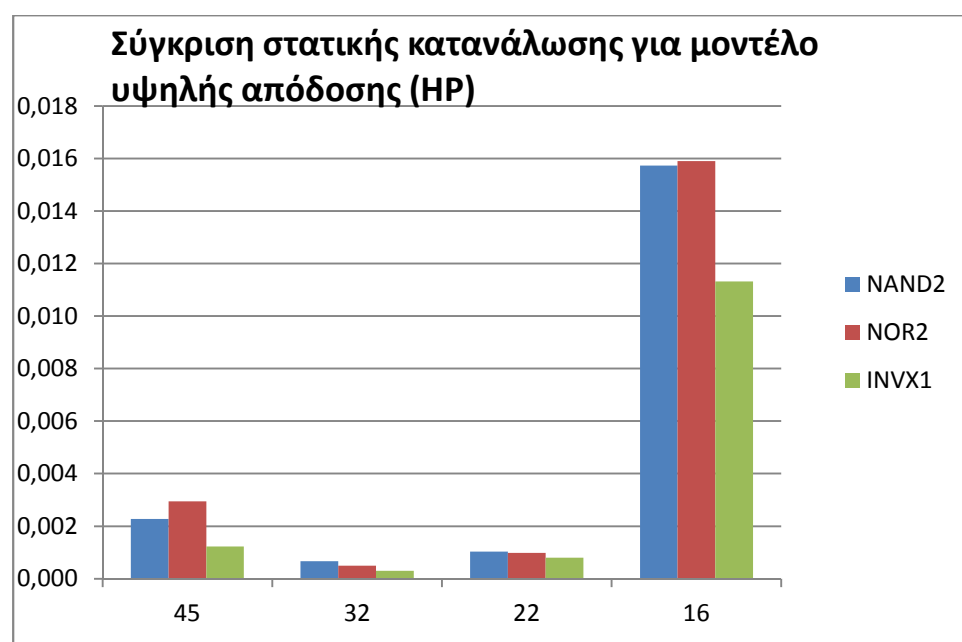
Το ρεύμα  $i_0$  εξαρτάται κυρίως από το  $V_t$  και το πάχος του οξειδίου στην πύλη  $t_{ox}$ . Όσο μικραίνουν τα  $V_t$  και  $t_{ox}$ , καθώς το εύρος της περιοχής απογύμνωσης μικραίνει, το ρεύμα διαρροής αυξάνεται εκθετικά.



Στη συνέχεια ακολουθούν τα αποτελέσματα για τη στατική κατανάλωση, όπου αναμένεται η αύξησή της, όσο κατεβαίνουμε σε πολύ μικρές διαστάσεις των τρανζίστορ (πλησιάζοντας τα 16nm).

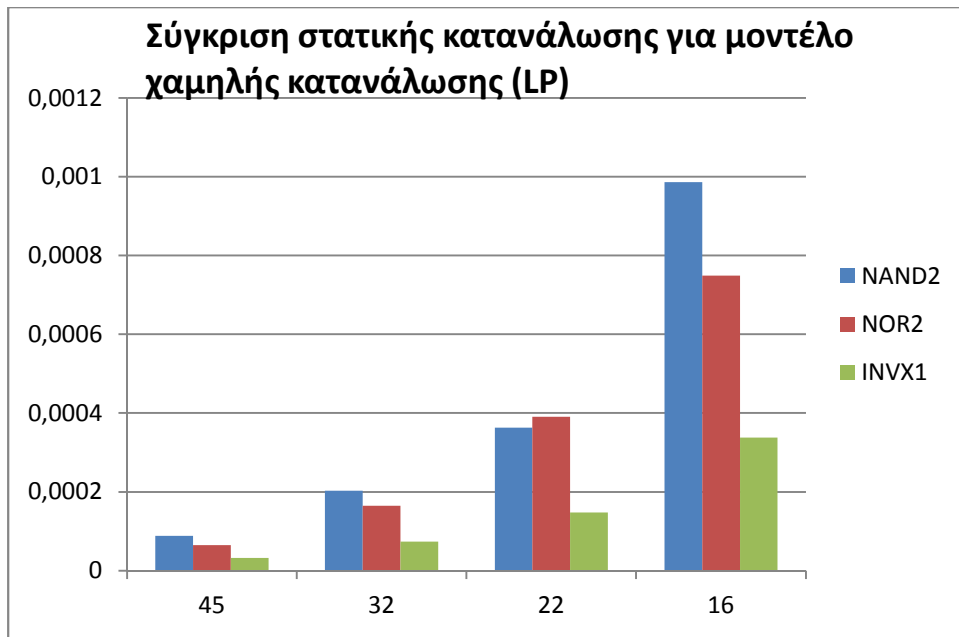
### 5.1.3.2.1. Αποτελέσματα στατικής κατανάλωσης σε επίπεδο πύλης

Με τον τρόπο παρουσίασης των αποτελεσμάτων που έχουμε δει ως τώρα, ακολουθεί η στατική κατανάλωση σε επίπεδο πύλης για κάθε μοντέλο τρανζίστορ.

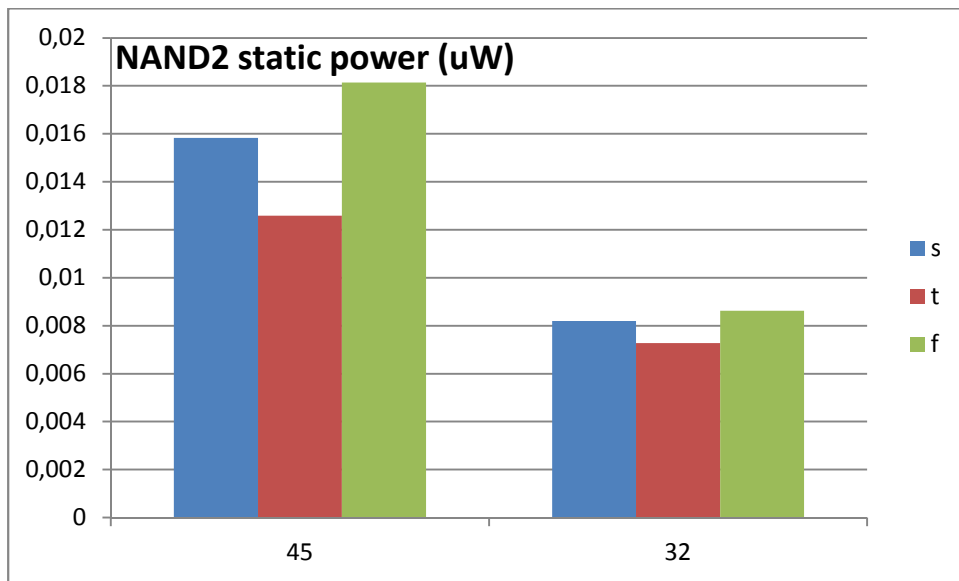


Εικόνα 5.1.3.14 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).

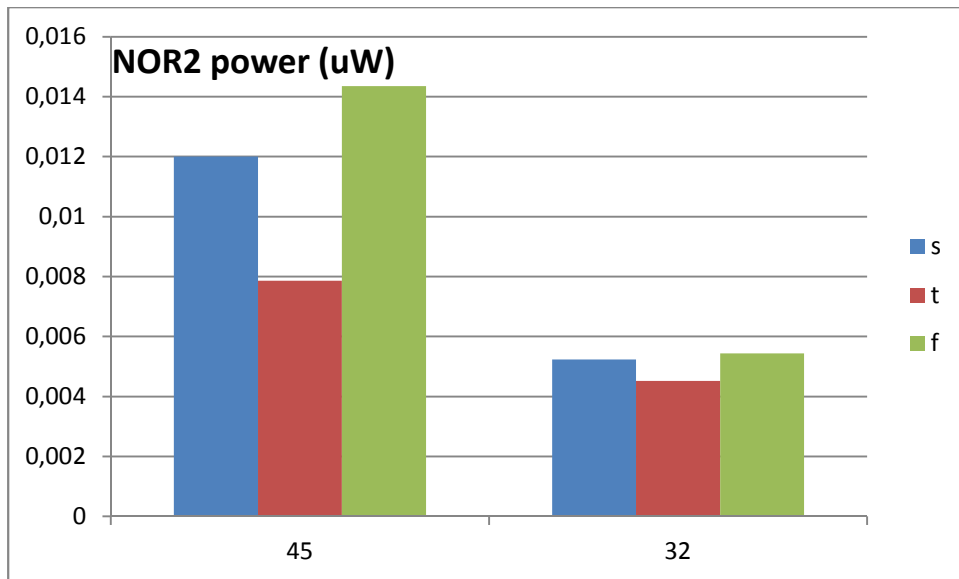
Σε επίπεδο πύλης παρατηρούμε πως σε απόλυτη τιμή, υπάρχει σημαντική αύξηση στατικής κατανάλωσης στην τεχνολογία των 16nm (όπως αναμενόταν). Προφανώς για τόσο μικρές διαστάσεις τρανζίστορ, τα ρεύματα διαρροής των ανάστροφα πολωμένων επαφών (substrate – διαχύσεων), λαμβάνουν σημαντικές τιμές. Ειδικότερα στο μοντέλο χαμηλής κατανάλωσης (επόμενο γράφημα), έχουμε σταδιακή αύξηση της στατικής κατανάλωσης (σε απόλυτες τιμές αγνοώντας προς το παρόν τη σχέση της με τη συνολική κατανάλωση)



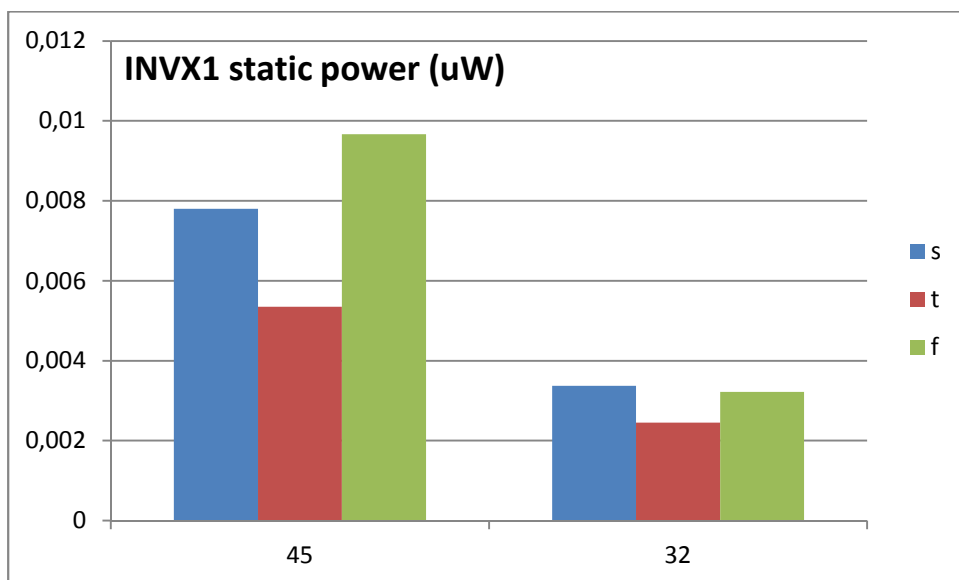
Εικόνα 5.1.3.15 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε αντιπροσωπευτική πύλη, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).



Εικόνα 5.1.3.16 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για το cell NANDx2, για process corners tt, ss και ff.



Εικόνα 5.1.3.17 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για το cell NORx2, για process corners tt, ss και ff.



Εικόνα 5.1.3.18 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για το cell INVx1, για process corners tt, ss και ff.

Παρατηρώντας τα παραπάνω γραφήματα, βγάζουμε διάφορα συμπεράσματα. Καταρχήν στο μοντέλο της χαμηλής κατανάλωσης (low power), βλέπουμε μια σταδιακή αύξηση της στατικής κατανάλωσης, που γίνεται πολύ μεγαλύτερη στην τεχνολογία 16nm. Στο μοντέλο general purpose, περνώντας από την τεχνολογία 45 στην τεχνολογία 32nm, παρατηρείται μείωση της στατικής κατανάλωσης, παρά το γεγονός ότι οι διαστάσεις των τρανζίστορ μειώθηκαν. Η εξήγηση μπορεί να αναζητηθεί στη σχέση που δίνει το ρεύμα διαρροής Sub –threshold.

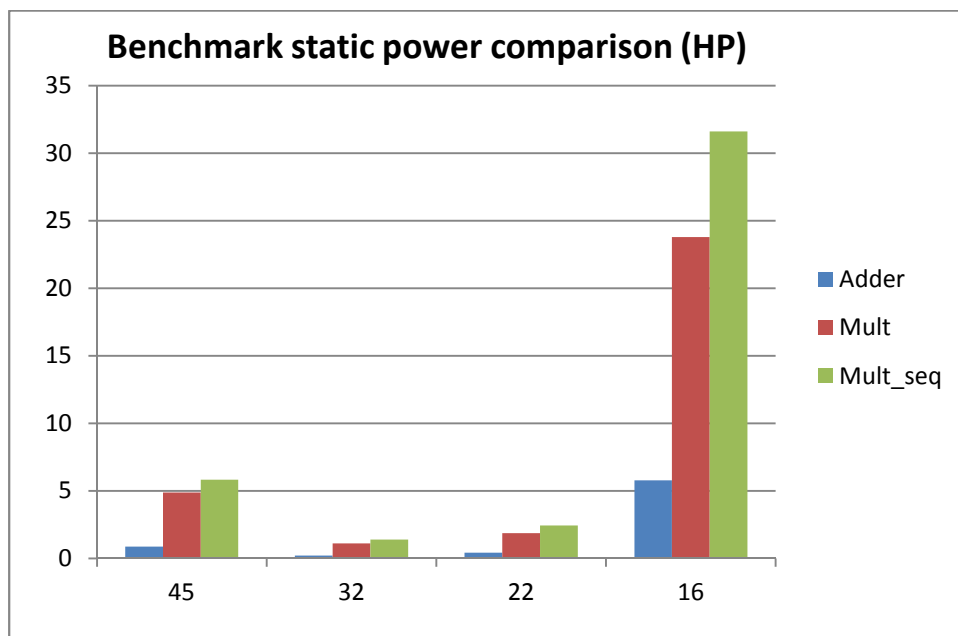
$$I_{sub} = I_0 \cdot \left[ 1 - \exp\left(\frac{V_{ds}}{v_{th}}\right) \right] \cdot \exp\left[\frac{V_{gs} - V_t - V_{off}}{n \cdot v_{th}}\right]$$

Με  $I_0 = \mu \cdot \frac{W}{L} \cdot \sqrt{\frac{q \cdot \epsilon_{si} \cdot NDEP}{2\phi_s}} \cdot v_t^2$ ,  $v_{th}$  : Thermal voltage,  $V_{off}$  : Offset voltage,  $V_t$ : Threshold voltage,  $n$ : παράμετρος ταλάντωσης sub-threshold. (4)

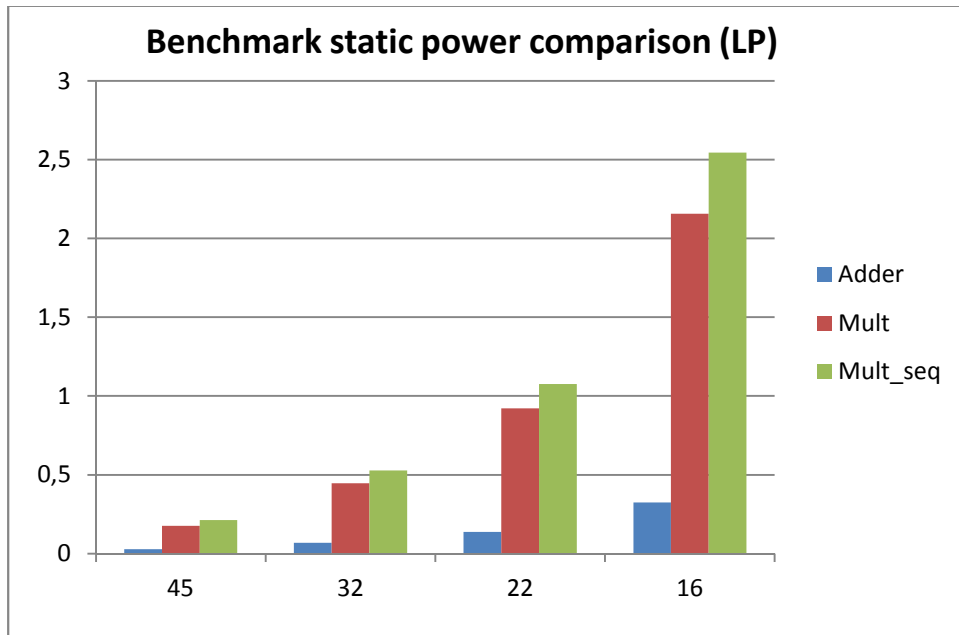
Εξετάζοντας τώρα τις τιμές του  $V_t$  που έχουν τα αντίστοιχα τρανζίστορ της κάθε τεχνολογίας, διακρίνουμε μικρότερο  $V_t$  στην τεχνολογία 45nm σε σχέση με αυτό της τεχνολογίας 32nm. Αυτό σημαίνει πως για την τεχνολογία 32nm, περιμένουμε μικρότερο ρεύμα διαρροής αφού ο εκθετικός όρος της προηγούμενης σχέσης μικραίνει. Έτσι η εικόνα του γραφήματος δεν είναι τίποτα περισσότερο από την επαλήθευση αυτής της σχέσης.

### 5.1.3.2.2. Αποτελέσματα στατικής κατανάλωσης σε επίπεδο design

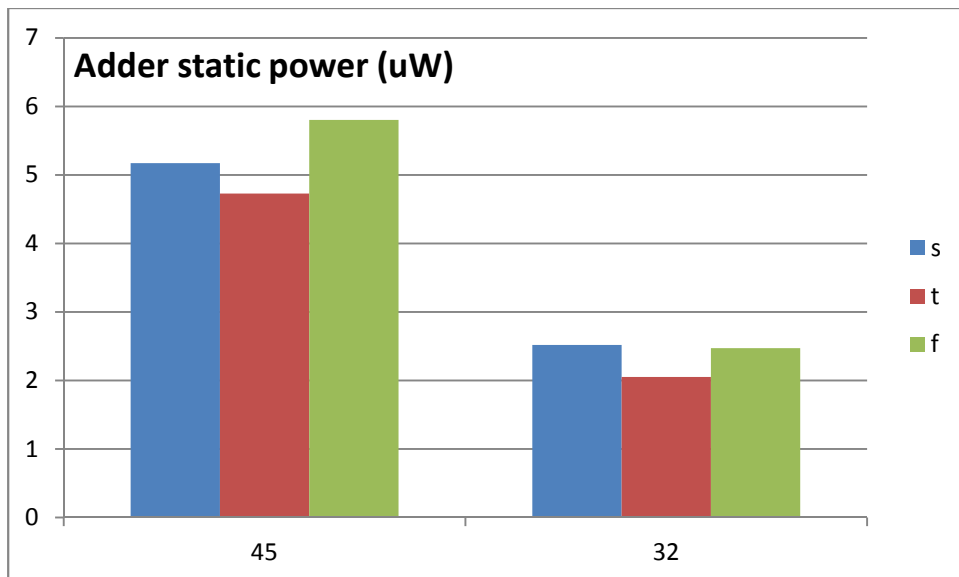
Συνεχίζουν αντίστοιχα αποτελέσματα στατικής κατανάλωσης, για τα τρία designs. Για τον αθροιστή 32x32, για τον πολλαπλασιαστή 16x16 και για το ακολουθιακό design του πολλαπλασιαστή 16x16.



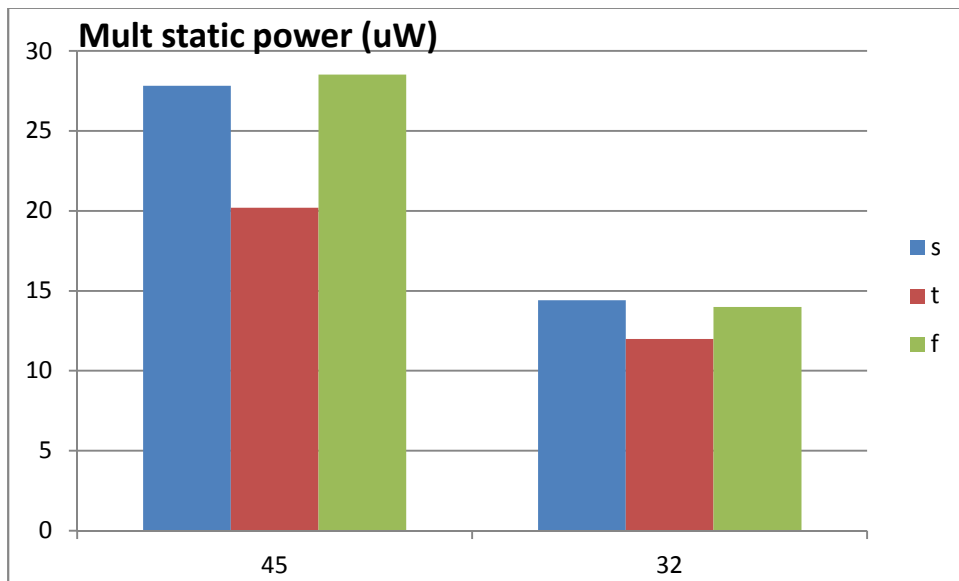
Εικόνα 5.1.3.19 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).



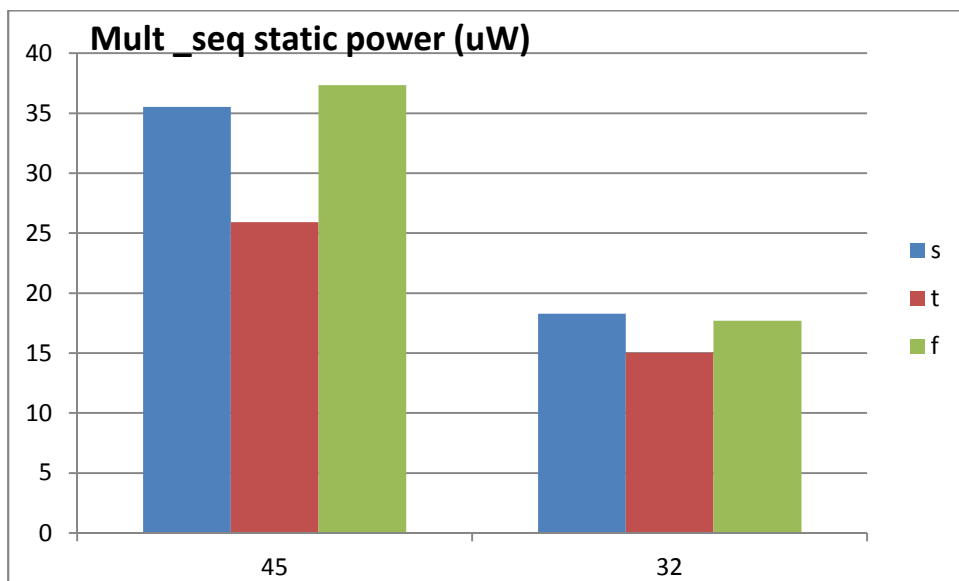
Εικόνα 5.1.3.20 Σύγκριση αποτελεσμάτων στατικής κατανάλωσης για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).



Εικόνα 5.1.3.21 Σύγκριση στατικής κατανάλωσης, για το design adder 32x32 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



Εικόνα 5.1.3.22 Σύγκριση στατικής κατανάλωσης, για το design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



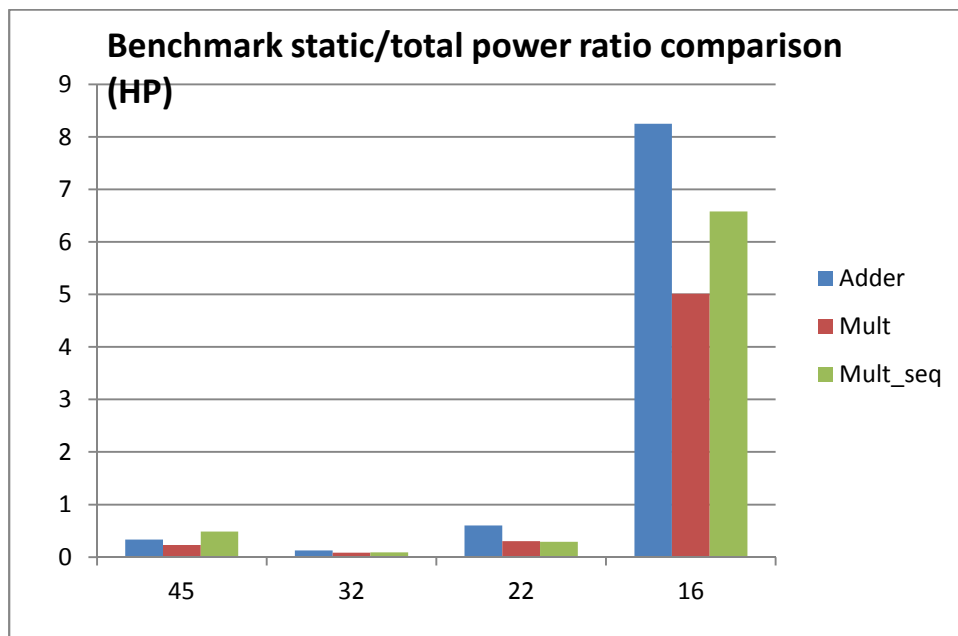
Εικόνα 5.1.3.23 Σύγκριση στατικής κατανάλωσης, για το design της ακολουθιακής εκδοχής του multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.

Για τα designs παρατηρούνται ανάλογα αποτελέσματα με αυτά των μεμονωμένων cells. Η αιτιολογία εδώ βέβαια, για την ποιότητα των αποτελεσμάτων, αναζητείται στις ιδιότητες της εξομοίωσης. Για να έχουμε ρεαλιστική απεικόνιση της στατικής κατανάλωσης, θα πρέπει κάποια μέρη του κυκλώματος να είναι σε κατάσταση “ηρεμίας”. Δηλαδή όσο τα μέρη που απαρτίζουν το design, εκτελούν κάποιες λογικές

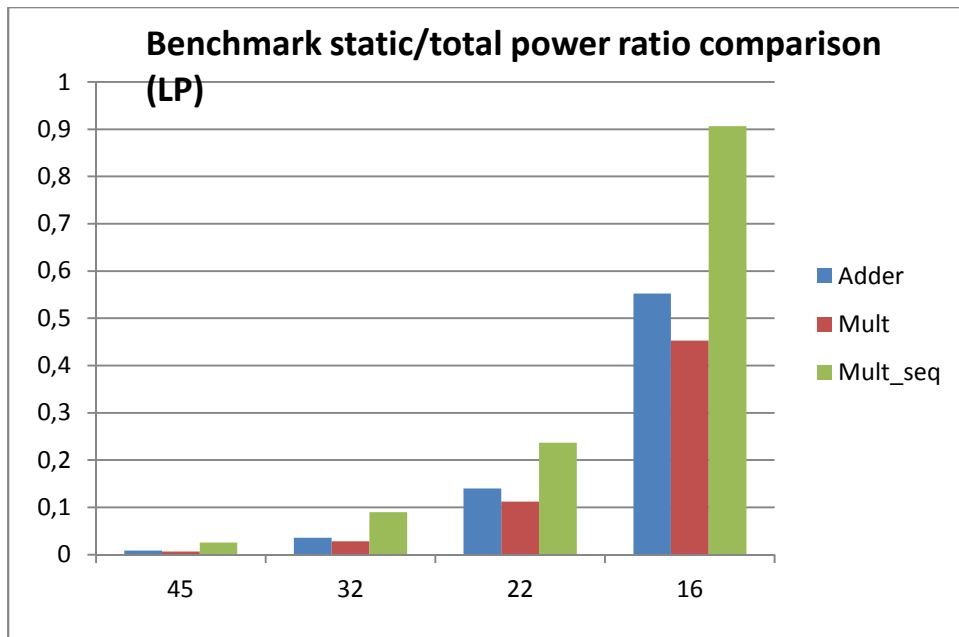
ή αριθμητικές πράξεις, καταναλώνουν δυναμική ισχύ. Είναι δύσκολο να υπολογίσουμε στατική ισχύ σ' αυτήν την περίπτωση, μέσα στον περιορισμένο χρόνο της εξομοίωσης, όπου κανένα δομικό στοιχείο του design, δεν μπορεί να συμπεριφέρεται στατικά. Σε επίπεδο συστήματος και με κατάλληλη επιλογή διανυσμάτων εξομοίωσης, θα μπορούσαμε να εξαγάγουμε ρεαλιστικότερα αποτελέσματα στατικής ισχύος.

### 5.1.3.3. Ποσοστιαία συνεισφορά στατικής κατανάλωσης στην ολική κατανάλωση

Εκτός από την απόλυτη τιμή της στατικής κατανάλωσης, μας ενδιαφέρει πρωτίστως η έκφρασή της ως το ποσοστό επί τοις εκατό της ολικής κατανάλωσης. Το ποσοστό αυτό επηρεάζεται από τις συνθήκες εξομοίωσης εξαιτίας της δραστηριότητας των μονάδων του κυκλώματος. Οι γραφικές παραστάσεις που ακολουθούν αναφέρονται σ' αυτές τις τιμές και οι τιμές είναι επί τοις εκατό.

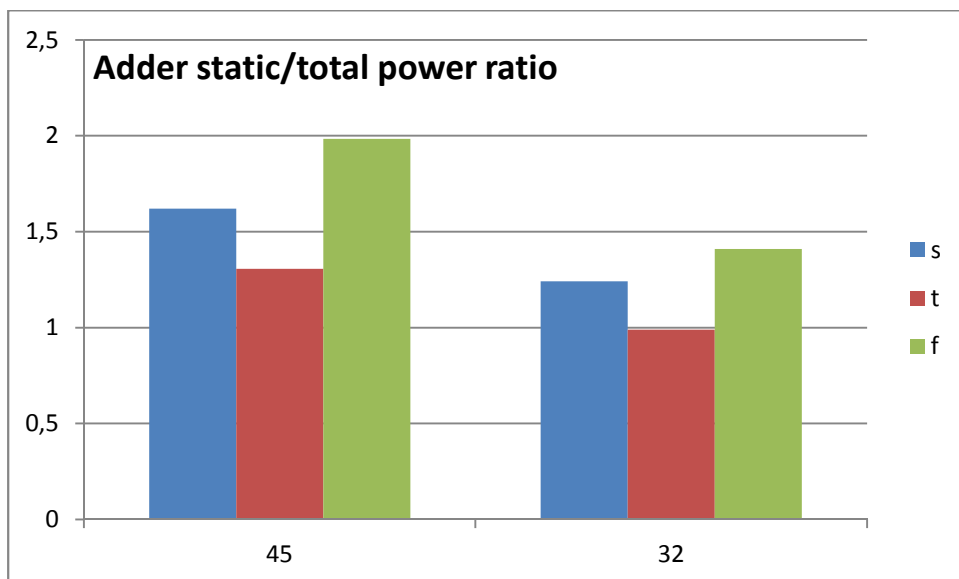


Εικόνα 5.1.3.24 Σύγκριση λόγου στατικής κατανάλωσης/συνολικής για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ υψηλής απόδοσης (high performance).



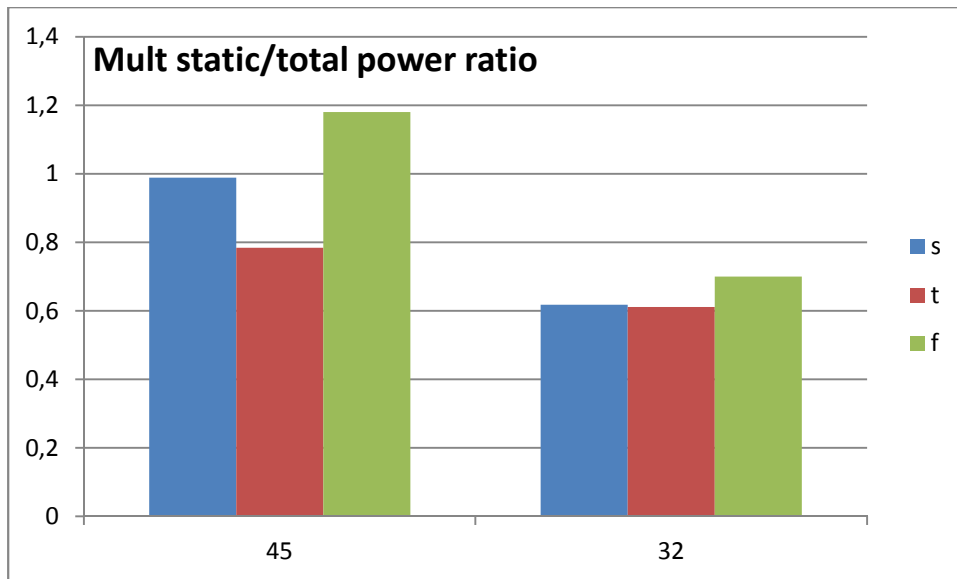
Εικόνα 5.1.3.25 Σύγκριση λόγου στατικής κατανάλωσης/συνολικής για κάθε ένα από τα τρία designs, με μοντέλο τρανζίστορ χαμηλής κατανάλωσης (low power).

Με τα πρώτα στοιχεία που προκύπτουν εμφανίζεται μια καλύτερη εικόνα από τεχνολογία σε τεχνολογία, ωστόσο τα ποσοστά που καταγράφονται είναι σε χαμηλά επίπεδα και εδώ ισχυροποιείται η αιτιολογία των συνθηκών εξομοίωσης. Εάν δηλαδή τα ίδια designs, μέσα στο χρονικό παράθυρο της εξομοίωσης, έμεναν για μεγάλα διαστήματα αδρανής, χωρίς να εκτελούν κάποιες λειτουργίες, τότε η στατική ισχύς θα καταλάμβανε μεγαλύτερο ποσοστό της συνολικής κατανάλωσης.

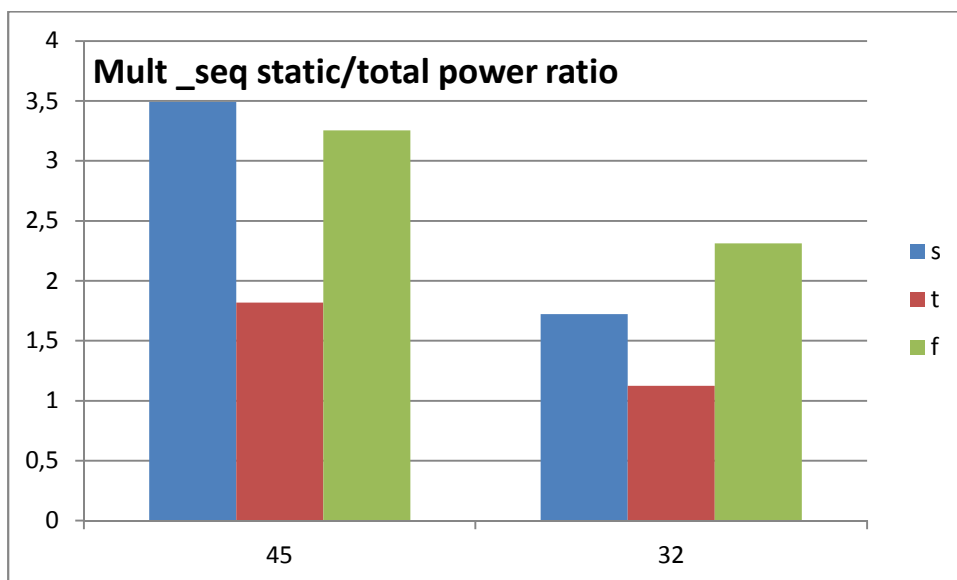


Εικόνα 5.1.3.26 Σύγκριση λόγου στατικής κατανάλωσης/ολικής, για το design adder 32x32 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.





Εικόνα 5.1.3.27 Σύγκριση λόγου στατικής κατανάλωσης/ολικής, για το design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.



Εικόνα 5.1.3.28 Σύγκριση λόγου στατικής κατανάλωσης/ολικής, για την ακολουθιακή εκδοχή του design multiplier 16x16 για process corner typical (t), slow – slow, fast – fast και γενικό μοντέλο τρανζίστορ.

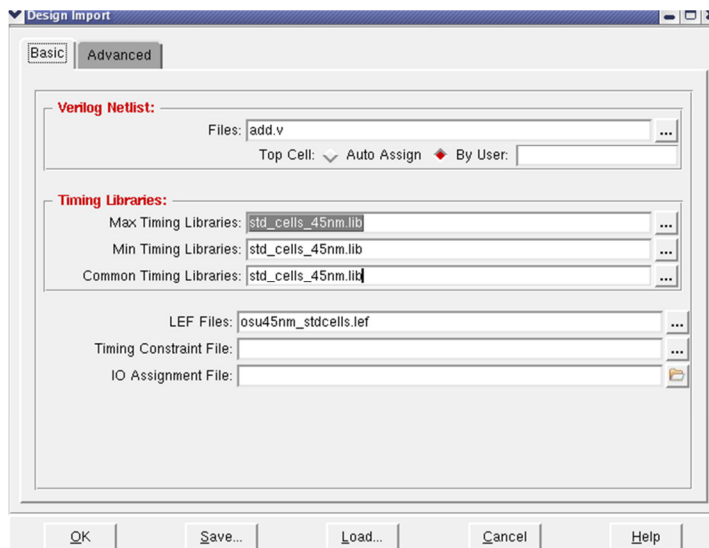
Συγκρίνοντας τα στοιχεία για τα τρία process corners στις τεχνολογίες 45 και 32nm, βλέπουμε πως κινούμαστε σε πολύ χαμηλά ποσοστά. Επιπλέον η αρχική παρατήρηση που έγινε σε επίπεδο πύλης, σχετικά με την σύγκριση για την στατική κατανάλωση στις τεχνολογίες 45 και 32nm, αντανακλάται ως τα designs. Το ποσοστό δηλαδή της στατικής κατανάλωσης έναντι της ολικής, παραμένει χαμηλότερο για την τεχνολογία 32nm σε σχέση με την τεχνολογία 45nm.

## 5.2. Αποτελέσματα Placement και routing

Προκειμένου να πάρουμε σωστά αποτελέσματα στην παρούσα φάση, μετά την επιτυχή κλιμάκωση των αρχείων lef, δε θα πρέπει να προκύπτουν σφάλματα με τη γεωμετρία και τους κανόνες που τη διέπουν, όπως αυτά καταγράφονται στο σχετικό log file (menu tools-> log view ).

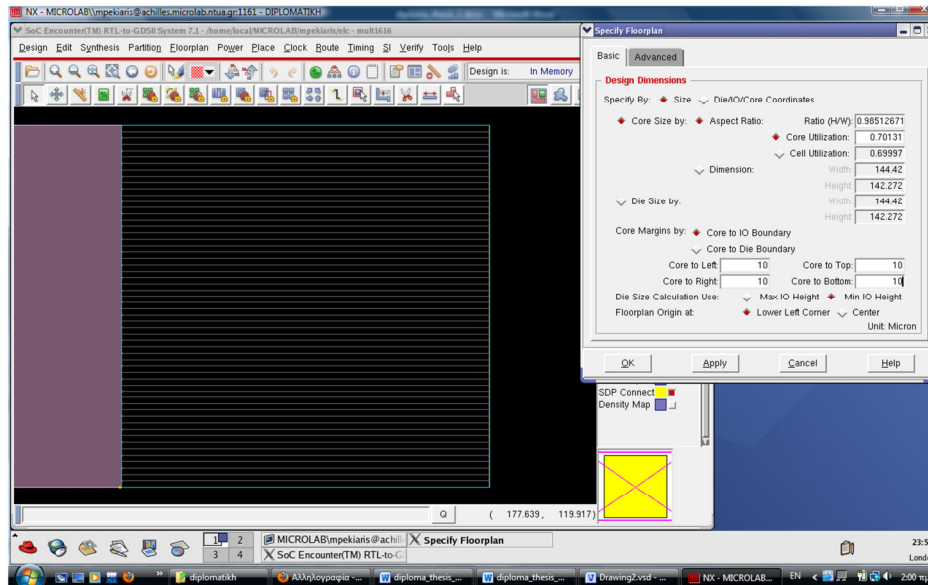
Η εφαρμογή παρέχει τη δυνατότητα χειρισμού, μέσω γραφικού περιβάλλοντος ( GUI), κάνοντας πιο εύκολη τη διαδικασία η οποία μπορεί να περιγραφεί από τα εξής βήματα:

- Ορισμός design.v, βιβλιοθήκης για στοιχεία χροτισμού, μέσω του αρχείου LIB και δήλωση αρχείου LEF.



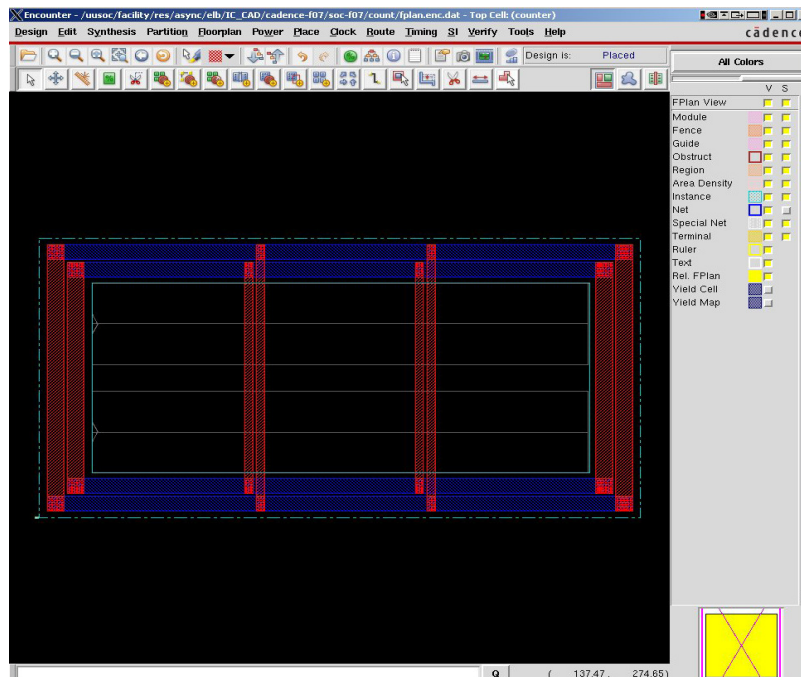
Εικόνα 5.2.1 Περιβάλλον του SoC Encounter για την δήλωση του design και των βιβλιοθηκών.

- Καθορισμός του floor plan. Εδώ καθορίζονται τα όρια, μέσα στα οποία θα γίνει η τελική τοποθέτηση των κελιών και η διασύνδεση μεταξύ τους.



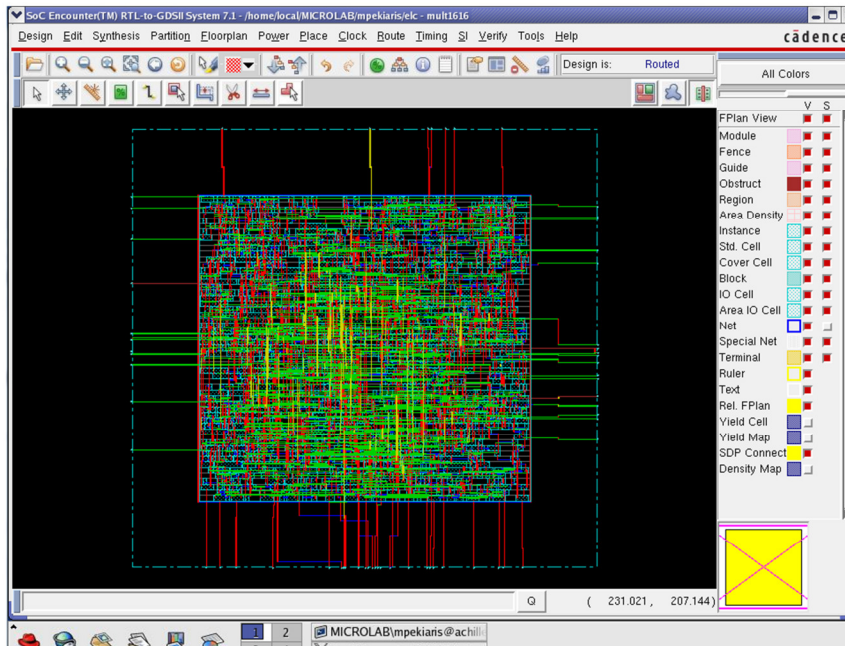
Εικόνα 5.2.2 Ορισμός floor plan.

- Καθορισμός αγωγών τροφοδοσίας. Μπορεί να κατασκευαστούν περιμετρικά του die, δακτύλιοι για το vdd και gnd, ώστε να είναι ευκολότερη η σύνδεση του υπόλοιπου κυκλώματος με την τροφοδοσία. Για τον ίδιο λόγο μπορούν να δημιουργηθούν ειδικές λωρίδες αγωγών. Η επιλογή του επιπέδου μετάλλου, καθώς και χαρακτηριστικά τους (πλάτος αγωγών, απόσταση), γίνεται μέσω του αντίστοιχου user interface tab.



Εικόνα 5.2.3 Τρόπος τοποθέτησης power rings.

- Τοποθέτηση και διασύνδεση. Η τοποθέτηση γίνεται με την επιλογή place std cells, ενώ το routing με την επιλογή nanoroute.

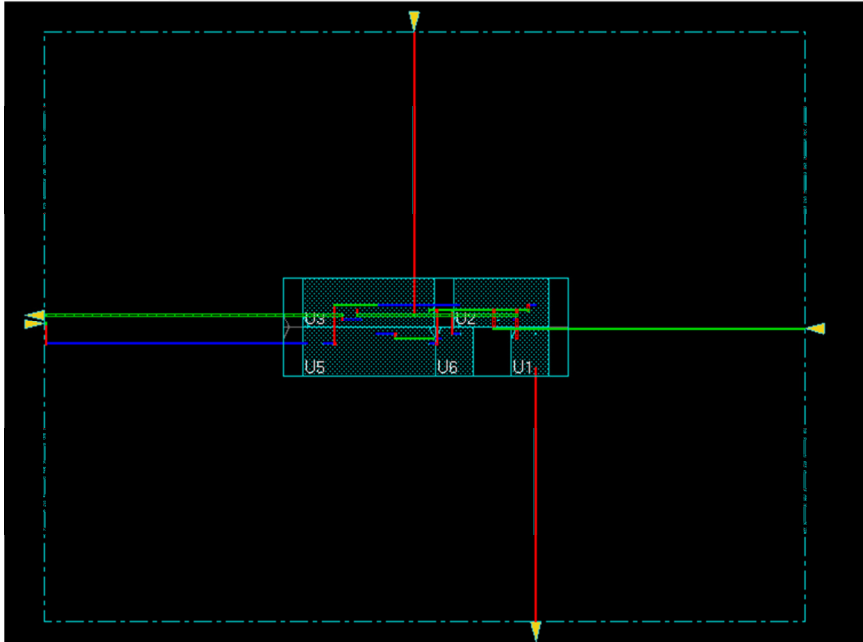


Εικόνα 5.2.4 Τοποθέτηση και διασύνδεση των cells.

- Εξαγωγή παρασιτικών χωρητικότητας καλωδιώσεων και pins. Οι παρασιτικές χωρητικότητες καταγράφονται τέλος με την επιλογή Timing -> ExtractRC, στο αρχείο design.cap.

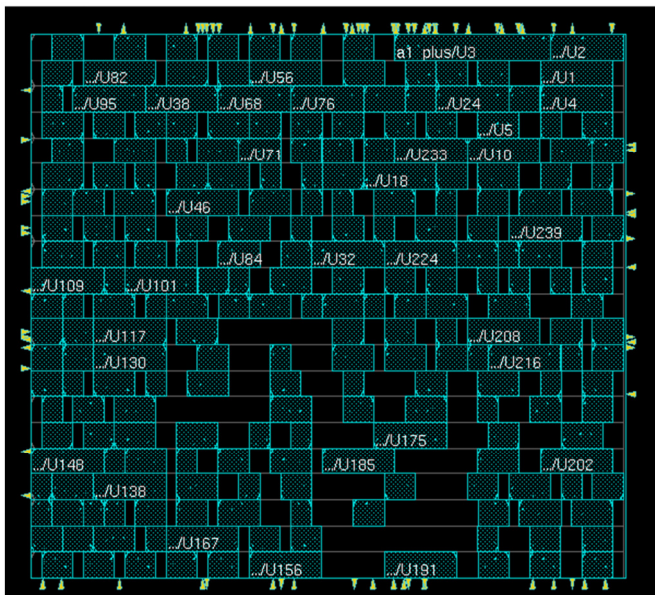
### 5.2.1. Τοποθέτηση και διασύνδεση σε διάφορα designs

Για την επαλήθευση του επιτυχούς χαρακτηρισμού, γίνεται προσπάθεια τοποθέτησης και διασύνδεσης (placement and routing), σε διάφορα designs. Για λόγους απλούστευσης ξεκινάμε τη διαδικασία δοκιμών με έναν απλό πλήρη αθροιστή του ενός bit.



Εικόνα 5.2.5 τοποθέτηση και διασύνδεση των cells που συνθέτουν έναν πλήρη αθροιστή.

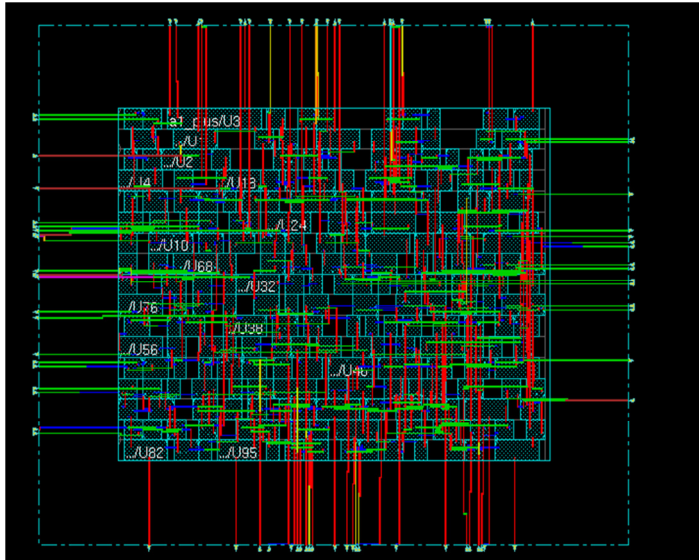
Ακολουθούν τα αποτελέσματα της τοποθέτησης των cells και της διασύνδεσής τους, για τα τρία designs. Του αθροιστή 32x32, του πολλαπλασιαστή 16x16 και της ακολουθιακής εκδοχής του πολλαπλασιαστή 16x16.



Εικόνα 5.2.6 Διάταξη cells για τον αθροιστή 32x32 bit στα 45nm.

Στο σχήμα αυτό διακρίνεται καθαρά η διάταξη, ο προσανατολισμός καθώς και η πυκνότητα με την οποία είναι τοποθετημένα τα cells. Η τοποθέτηση των cells έγινε

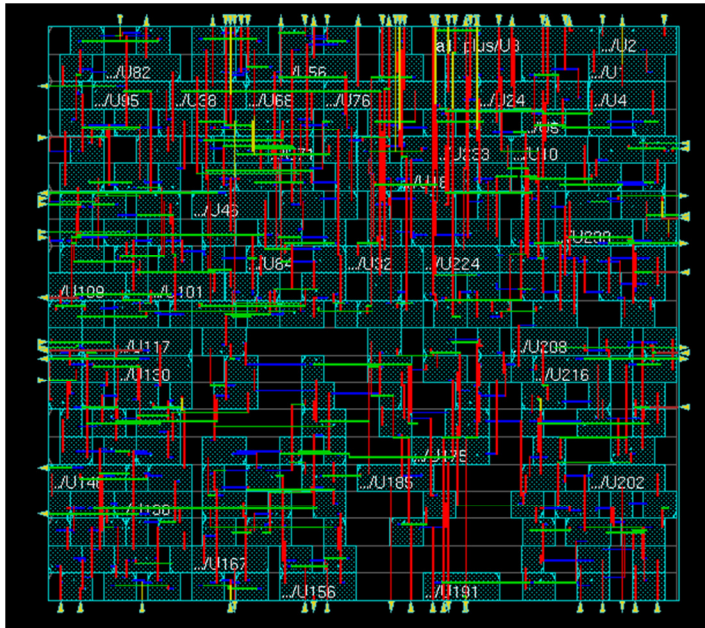
στην τεχνολογία των 45nm. Στο επόμενο σχήμα δίνεται άλλη μια περίπτωση του ίδιου design, όπου έχει γίνει τοποθέτηση και διασύνδεση για την ίδια τεχνολογία, με ορισμό ταυτόχρονα των περιθωρίων (floor planning) του χώρου τοποθέτησης.



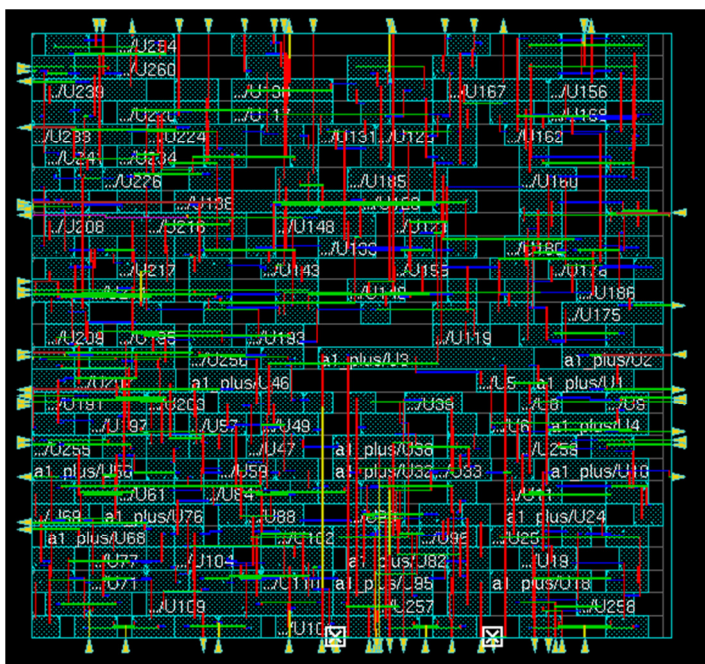
Εικόνα 5.2.7 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 45nm.

Με επιτυχία ολοκληρώθηκε τοποθέτηση και διασύνδεση του αθροιστή, για τις υπόλοιπες τεχνολογίες.

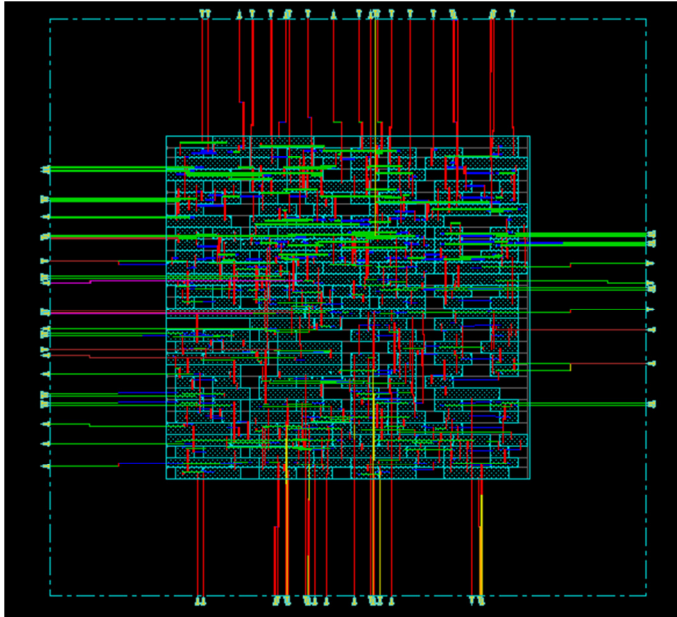




Εικόνα 5.2.8 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 32nm.



Εικόνα 5.2.9 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 22nm.



Εικόνα 5.2.10 Τοποθέτηση και διασύνδεση των cells για τον αθροιστή 32x32 bit στα 16nm.

Ακολουθεί ο πίνακας με τις παρασιτικές χωρητικότητες του αθροιστή για κάθε τεχνολογία, που έδωσε το εργαλείο encounter μετά την εξαγωγή τους με την κατάλληλη εντολή. Συγκεκριμένα έχει καταγραφεί η **μεσαία** τιμή (median) από όλες τις μετρούμενες τιμές, για την ολική χωρητικότητα (χωρητικότητα καλωδίου + χωρητικότητα ακίδας) και τη χωρητικότητα καλωδίου ανά μονάδα μήκους για κάθε τεχνολογία, συμπεριλαμβανομένης και της τεχνολογίας 180nm από την οποία προέκυψαν με κλιμάκωση (scaling) οι βιβλιοθήκες των υπόλοιπων τεχνολογιών.

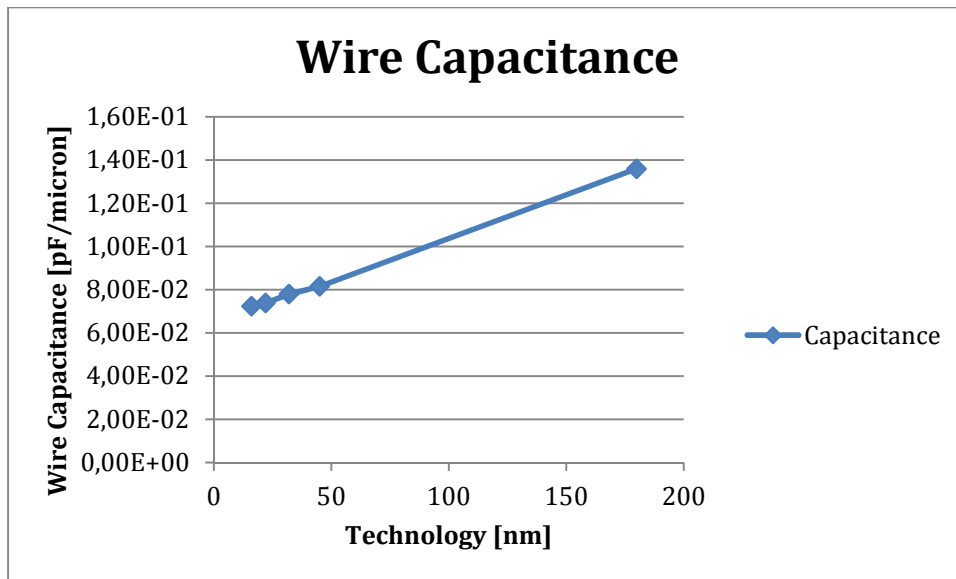
Πίνακας 5.2.1 Μεσαίες τιμές συνολικής χωρητικότητας καλωδίων και χωρητικότητας καλωδίου/micron που έδωσε το εργαλείο SoC Encounter.

technology [nm]	Total Capacitance [pF]	Capacitance [pF/micron]
180	3,10E-02	1,36E-01
45	5,28E-03	8,15E-02
32	3,73E-03	7,79E-02
22	1,92E-03	7,38E-02
16	1,38E-03	7,23E-02

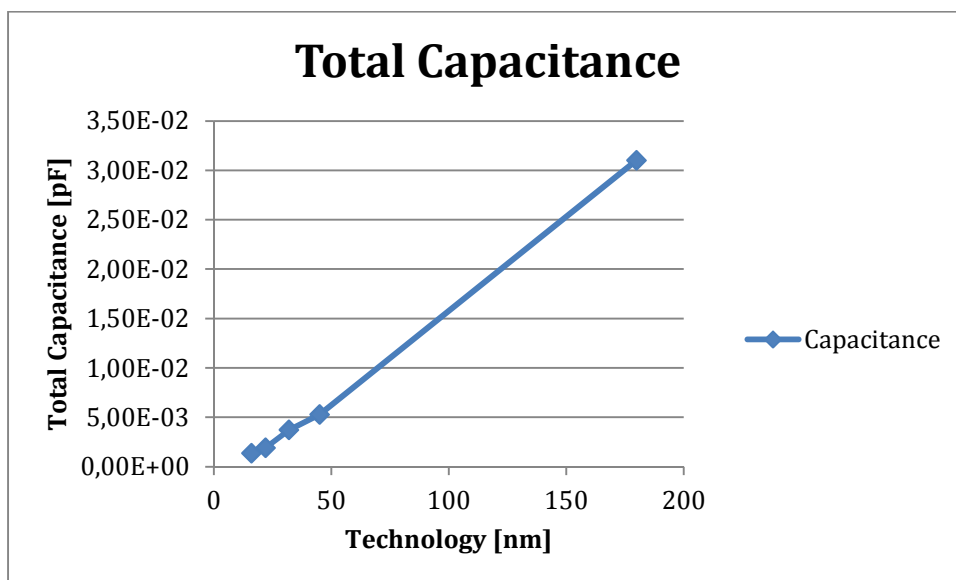
Βλέπουμε λοιπόν πως τα αποτελέσματα των ηλεκτρικών χαρακτηριστικών των γραμμών διασύνδεσης ακολουθούν τη γραμμική κλιμάκωση που έχουμε κάνει στις παραμέτρους του LEF αρχείου.

Αντίστοιχα προκύπτει το γραφικό αποτέλεσμα των προηγούμενων.



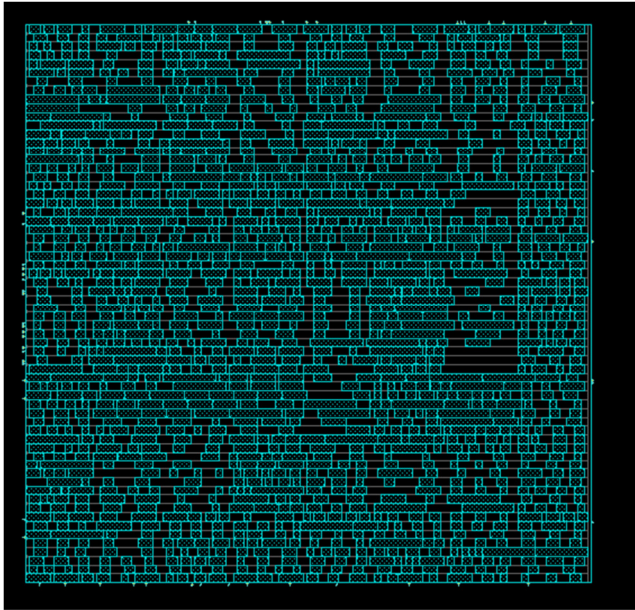


Εικόνα 5.2.11 χωρητικότητα καλωδίου/μονάδα μήκους συναρτήσει της τεχνολογίας για τον αθροιστή 32x32, που έδωσε το εργαλείο SoC Encounter.

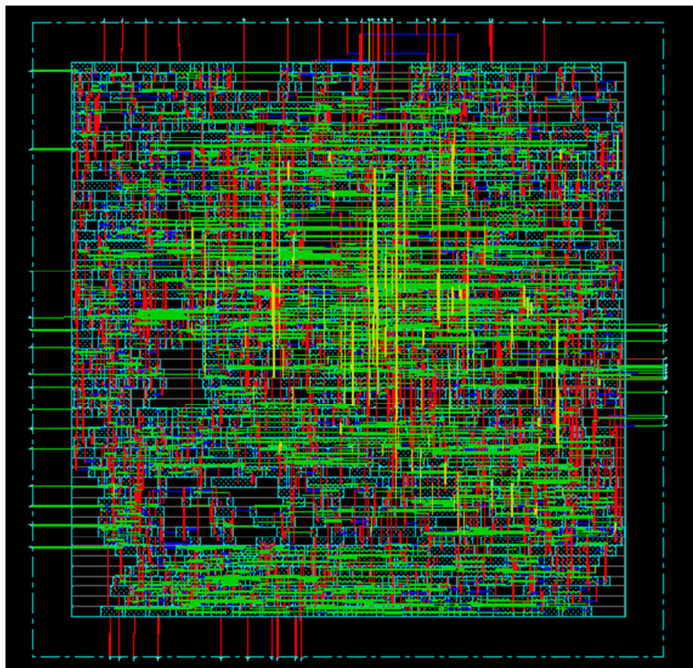


Εικόνα 5.2.12 Ολική χωρητικότητα (καλωδίου + ακίδας) συναρτήσει της τεχνολογίας για τον αθροιστή 32x32 που έδωσε το εργαλείο SoC Encounter.

Προχωρώντας με τον πολλαπλασιαστή 16x16, πραγματοποιούμε αρχικά τοποθέτηση των cells στην τεχνολογία των 45nm. Στη συνέχεια γίνεται τοποθέτηση και διασύνδεση για τις υπόλοιπες τεχνολογίες.



Εικόνα 5.2.13 Τοποθέτηση cells για την υλοποίηση του πολλαπλασιαστή 16x16 στα 45nm.

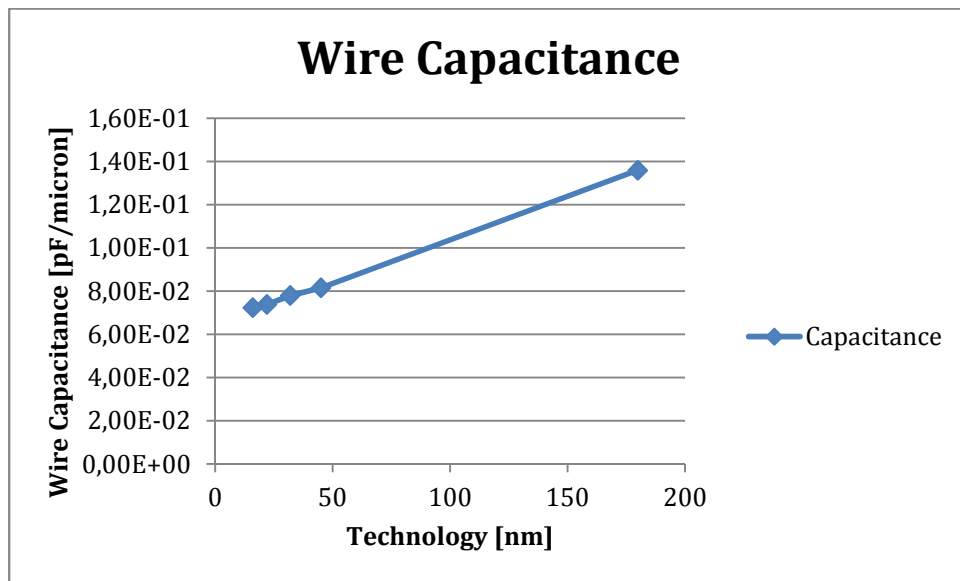


Εικόνα 5.2.14 Layout για τον πολλαπλασιαστή 16x16 στην τεχνολογία 45nm

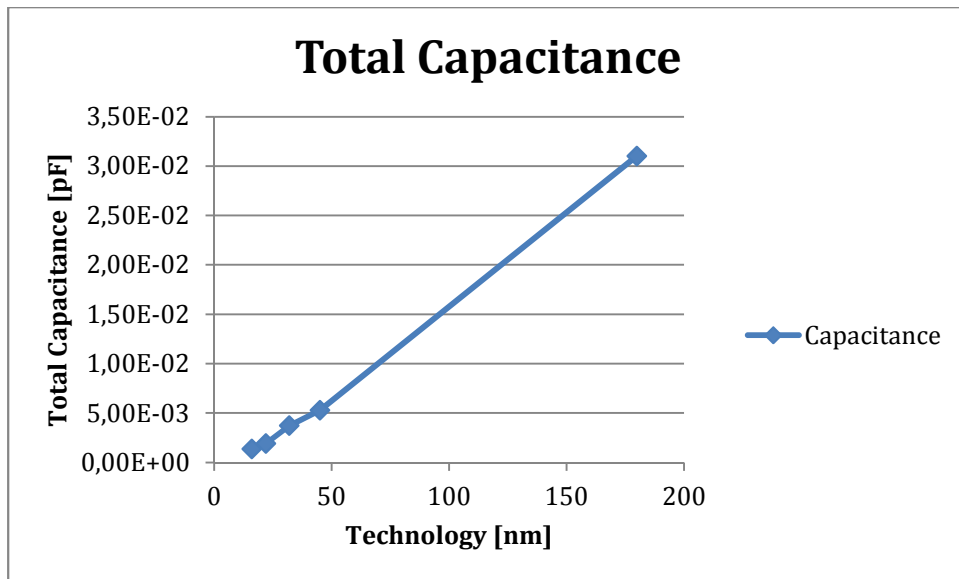
Όπως στον αθροιστή, έτσι κι εδώ παίρνοντας τα αποτελέσματα από τα αρχεία καταγραφής των μεσαίων τιμών των παρασιτικών χωρητικοτήτων, σχεδιάζουμε τις καμπύλες της ολικής χωρητικότητας και της χωρητικότητας του καλωδίου ανά μονάδα μήκους.

Πίνακας 5.2.2 Πίνακας 5.2.2 Ολική χωρητικότητα (καλωδίου + ακίδας) συναρτήσει της τεχνολογίας για τον πολλαπλασιαστή 16x16 που έδωσε το εργαλείο SoC Encounter.

technology [nm]	Total Capacitance [pF]	Capacitance [pF/micron]
180	3,06E-02	1,36E-01
45	5,10E-03	8,14E-02
32	3,73E-03	7,81E-02
22	1,89E-03	7,38E-02
16	1,31E-03	7,22E-02



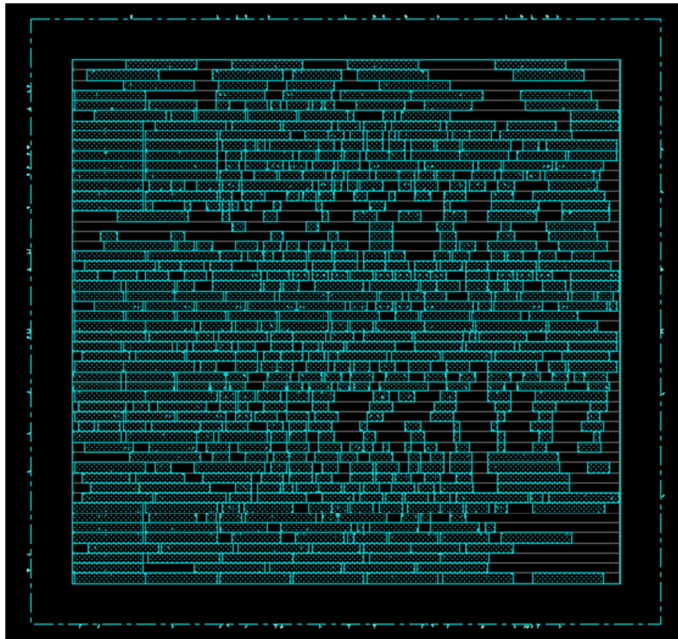
Εικόνα 5.2.15 χωρητικότητα καλωδίου/μονάδα μήκους συναρτήσει της τεχνολογίας για τον πολλαπλασιαστή 16x16 που έδωσε το εργαλείο SoC Encounter.



Εικόνα 5.2.16 Ολική χωρητικότητα (καλωδίου + ακίδας) συναρτήσει της τεχνολογίας για τον πολλαπλασιαστή 16x16 που έδωσε το εργαλείο SoC Encounter.

Το συμπέρασμα κι εδώ είναι το ίδιο. Δηλαδή τα ηλεκτρικά χαρακτηριστικά των designs, ακολουθούν γραμμική κλιμάκωση (scaling), όπως άλλωστε έχει γίνει στα αντίστοιχα στο περιεχόμενο των αρχείων LEF.

Χωρίς κανένα πρόβλημα ακολούθησε τοποθέτηση και διασύνδεση της ακολουθιακής εκδοχής του πολλαπλασιαστή. Η τοποθέτηση των cells του κυκλώματος αυτού για τεχνολογία 16nm φαίνεται στο επόμενο σχήμα.

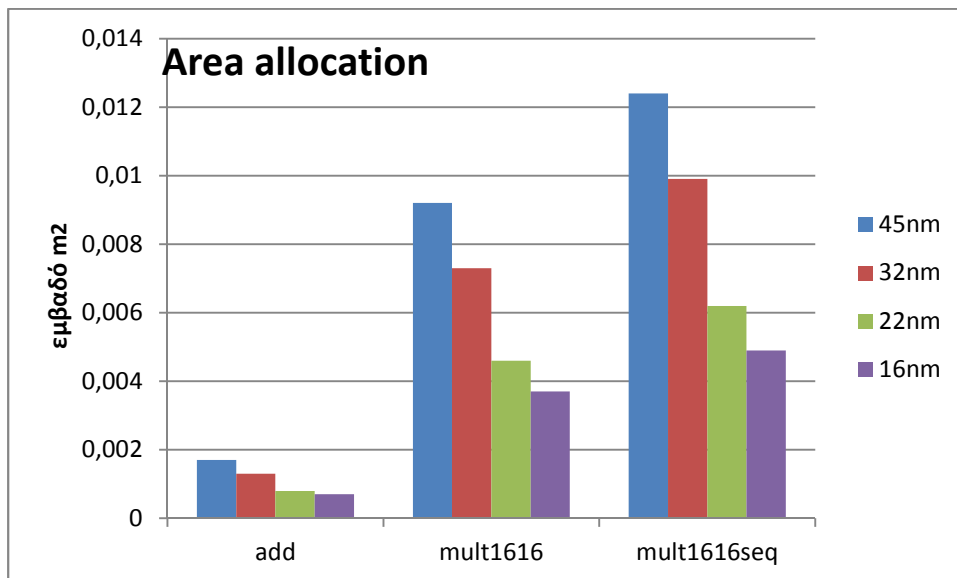


Εικόνα 5.2.17 Τοποθέτηση των cells που απαρτίζουν την ακολουθιακή εκδοχή του πολλαπλασιαστή 16x16 στην τεχνολογία 16nm.

Ένα στοιχείο που ενδιαφέρει, είναι το εμβαδό (area) που καταλαμβάνει κάθε design για κάθε τεχνολογία. Επιπλέον η πυκνότητα με την οποία είναι τοποθετημένα τα cells, δίνει ένα ποιοτικό χαρακτηριστικό της σωστής εκμετάλλευσης του διαθέσιμου χώρου από το εργαλείο τοποθέτησης των cells.

Πίνακας 5.2.3 Αποτελέσματα για το εμβαδό που καταλαμβάνει κάθε design.

Τεχνολογία/λειτουργία		area (mm <sup>2</sup> )
45nm	add	0.0017
	mult1616	0.0092
	mult1616seq	0.0124
32nm	add	0.0013
	mult1616	0.0073
	mult1616seq	0.0099
22nm	add	0,0008
	mult1616	0,0046
	mult1616seq	0,0062
16nm	add	0,0007
	mult1616	0,0037
	mult1616seq	0,0049



Εικόνα 5.2.18 Γραφική παράσταση της περιοχής που καταλαμβάνει κάθε design, σε σχέση με τη χρησιμοποιούμενη τεχνολογία.

Όπως παρατηρούμε όσον αφορά το εμβαδό τοποθέτησης (area), ανάλογα με τον αριθμό των cells που συνθέτουν το κύκλωμα και ανάλογα με την τεχνολογία που χρησιμοποιούμε έχουμε το αναμενόμενο αποτέλεσμα. Δηλαδή όσο κατεβαίνουμε σε τεχνολογία, απαιτούμε μικρότερη επιφάνεια για να τοποθετήσουμε το ίδιο design. Όπως παρατηρήθηκε στο εμβαδό που καταλαμβάνει κάθε cell, έτσι κι εδώ το scaling δεν γίνεται κατά αναλογία με το τετράγωνο της τιμής του συντελεστή κλιμάκωσης, όπως θα περιμέναμε, αλλά (προσεγγιστικά) σε αναλογία με την τιμή του. Αυτό οφείλεται κι εδώ, στο γεγονός ότι η κλιμάκωση (scaling), γίνεται μόνο κατά τη διεύθυνση Y και σε τιμές που πλησιάζουν το εγγύτερο ακέραιο πολλαπλάσιο του manufacturing grid.

### 5.3. Αποτελέσματα σε επίπεδο συστήματος

Για μια πληρέστερη αποτίμηση των βιβλιοθηκών που παρήχθησαν μέσω χαρακτηρισμού, επιλέχθηκε η λύση της υλοποίησης ενός μεγαλύτερου design, σε σχέση με αθροιστές και πολλαπλασιαστές, δηλαδή με περισσότερα timing paths και standard-cells. Στόχος ήταν να χρησιμοποιηθούν ενδεικτικά κάποιες από τις

βιβλιοθήκες που δημιουργήθηκαν στα πλαίσια της εργασίας αυτής μέσω του χαρακτηρισμού στο RTL-to-GDSII physical implementation flow, δηλαδή στη σύνθεση και στην τοποθέτηση-και-διασύνδεση του design, ώστε να παραχθεί το layout του.

Στα πλαίσια αυτά, επιλέξαμε την τεχνολογία των 32nm και συγκεκριμένα τη βιβλιοθήκη που αντιστοιχεί στο typical corner (0.9V, 25 C). Το σύστημα που υλοποιήθηκε περιελάμβανε δύο πυρήνες LEON3 SPARC, συνδεδεμένους πάνω στο διάδρομο AMBA ως masters (AHB bus), καθώς και περιφερειακές μονάδες παράλληλης εισόδου/εξόδου (I/Os), μονάδες σειριακής εισόδου/εξόδου (UARTs) και χρονιστές (timers), ως slaves στον APB.

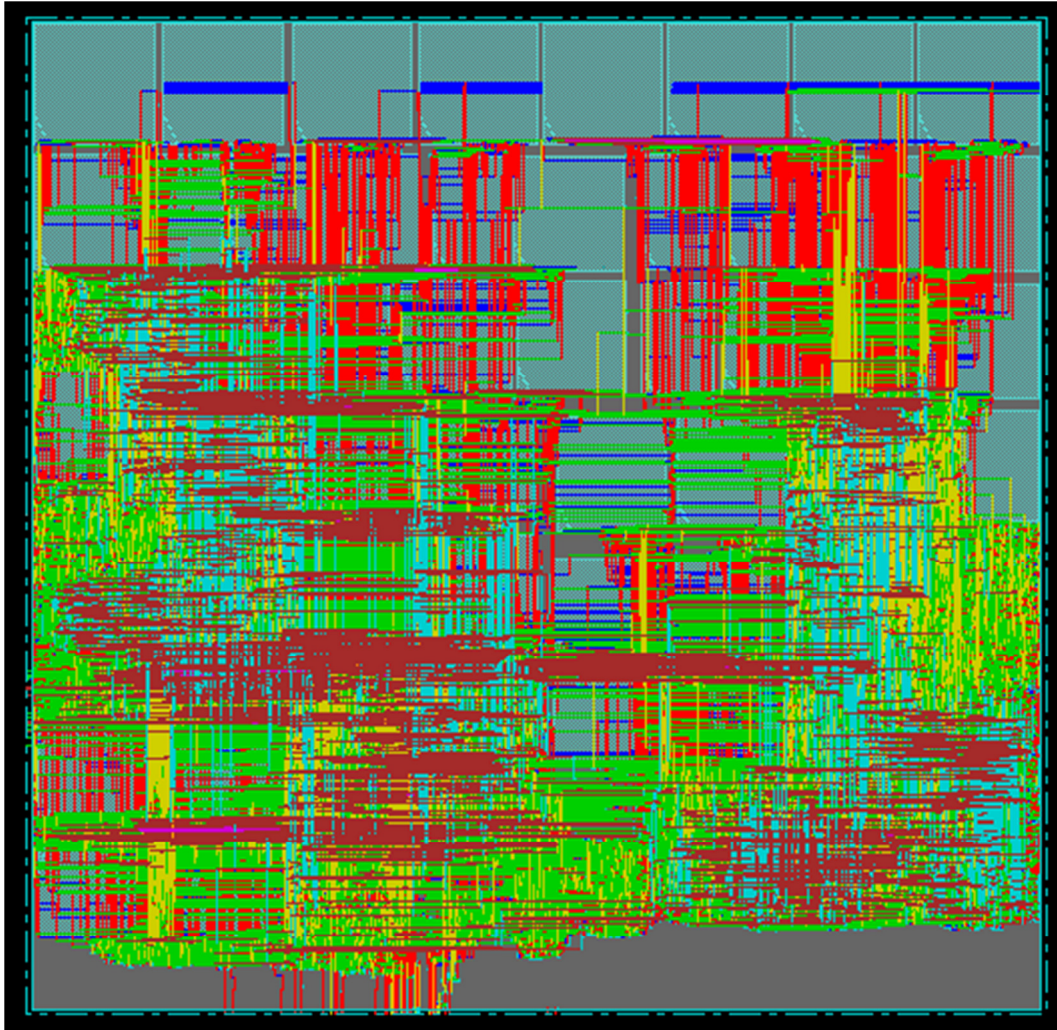
Η σύνθεση της generic VHDL περιγραφής, που παρέχεται από την Gaisler Research (5), πραγματοποιήθηκε στον Synopsys Design Compiler, με timing constraint 1.6ns. Η post-synthesis gate-level netlist οδηγήθηκε στο Cadence SoC Encounter, όπου και παρήχθη το τελικό φυσικό σχέδιο, από το οποίο παρατίθενται τα αποτελέσματα χρονισμού, επιφάνειας, δυναμικής και στατικής κατανάλωσης, στον πίνακα που ακολουθεί.

**Πίνακας 5.3.1 Αποτελέσματα για σύστημα με δύο πυρήνες LEON3 SPARC, σε τεχνολογία 32nm (General Purpose model, typical corner).**

<i>Design</i>	<i>Leon3 MP-SoC (2 cores)</i>
<i>Technology node</i>	<i>32nm @ 0.9V, 25C (GP PTM model-typical)</i>
<i>Critical path delay(ns)</i>	<i>4.45ns</i>
<i>Area (core size)</i>	<i>1940x1896 <math>\mu\text{m}^2</math></i>
<i>Dynamic Power (@ 200MHZ)</i>	<i>72mW</i>
<i>Static (Leakage) Power (@ 200MHZ)</i>	<i>9mW</i>
<i>Total Power (@ 200MHZ)</i>	<i>81mW</i>

Τα αποτελέσματα όσον αφορά στη συνολική κατανάλωση ισχύος, ήταν στα αναμενόμενα πλαίσια, δηλαδή όχι πάνω από 100 mW, δεδομένης και της εφαρμογής, που εκτελέστηκε σε συχνότητα 200 MHz στο ModelSim και παρήγαγε ένα .vcd αρχείο της τάξης των 2 GB. Ακόμα, η αναλογία στατικής και δυναμικής κατανάλωσης φαίνεται υπέρ της δυναμικής, αλλά η στατική είναι στην ίδια τάξη μεγέθους, με μια αναλογία 1:8, δηλαδή όχι έξω από τα αναμενόμενα, με βάση τις προβλέψεις για το leakage στα 32nm.





Εικόνα 5.3.1 1. Το 1940x1896  $\mu\text{m}^2$  layout του LEON3-based MP-SoC με δύο πυρήνες στο SoC Encounter.

Επίσης, τα αποτελέσματα χρονισμού εμφανίζονται ελαφρώς καλύτερα σε σχέση με αντίστοιχες υλοποιήσεις του ίδιου συστήματος στα 45nm (TSMC process, typical corner). Ωστόσο, η επιφάνεια φαίνεται σχετικά μεγάλη, σε σχέση με τα 45nm, δεδομένου ότι το scaling στα blocks των cells, που δηλώνονται στο αρχείο .LEF, έγινε μόνο κατά την κατακόρυφη διεύθυνση στις δικές μας βιβλιοθήκες.

Με αφετηρία αυτά τα αποτελέσματα, στόχος πλέον θα μπορούσε να είναι η υλοποίηση του συγκεκριμένου design ή άλλων αντίστοιχου μεγέθους (της τάξης των 20K – 30 K σε standard-cells) στα 22 και στα 16 nm, ώστε να είναι δυνατή η εξαγωγή συγκριτικών αποτελεσμάτων για τις βιβλιοθήκες, με βάση πολύπλοκα ψηφιακά συστήματα.



# Κεφάλαιο 6

## Συμπεράσματα – Προτάσεις

### 6.1. Συμπεράσματα

Με την ολοκλήρωση του παρόντος έργου προκύπτουν ορισμένα χρήσιμα αποτελέσματα που αφορούν τις τεχνολογίες κάτω των 45nm. Ολοκληρώθηκε επιτυχώς όλο το work flow που αναλύθηκε απ' την αρχή, επαληθεύοντας με τα αποτελέσματα τη λειτουργία των νέων βιβλιοθηκών. Επιπλέον καταφέραμε να αναδείξουμε έναν αλγόριθμο, για τα βήματα σχεδίασης νέων βιβλιοθηκών, τόσο για τα εργαλεία σύνθεσης, όσο και για τα εργαλεία τοποθέτησης και διασύνδεσης, για τεχνολογίες που βρίσκονται ακόμα σε πρώιμο στάδιο ανάπτυξης. Για το πρώτο σκέλος βλέπουμε πως στις τεχνολογίες κάτω των 45nm παρατηρείται μια τάση σταθεροποίησης των περιθωρίων βελτίωσης των αποτελεσμάτων που αφορούν την καθυστέρηση, η οποία ακολουθεί το νόμο της δύναμης του "α". Δηλαδή είναι αντιστρόφως ανάλογη της δύναμης του "α" του μεγέθους ( $V_{dd}-V_t$ ). Επιπλέον παρατηρείται ελάττωση του ρυθμού μείωσης της δυναμικής κατανάλωσης, όσο πηγαίνουμε σε μικρότερες διαστάσεις.

Οι συνέπειες αυτές μπορούν να δικαιολογηθούν, αν αναλογιστούμε ότι η κλιμάκωση πέραν αυτών των τεχνολογιών, ακολουθεί το μοντέλο της γενικής κλιμάκωσης, όπου οι διαστάσεις κλιμακώνονται με διαφορετικό συντελεστή, σε σχέση με τις τάσεις τροφοδοσίας απ' ενός και απ' εταίρου επειδή μικραίνοντας οριακά τις φυσικές διαστάσεις, οδηγούμαστε σε μεγαλύτερες τιμές αντιστάσεων  $R$  των στοιχείων των κυττάρων, αντισταθμίζοντας εν γένει το όφελος της ελάττωσης των χωρητικότητων που συμβαίνει σε μικρότερες διαστάσεις. Συνέπειες αναπόφευκτες και στενά συνδεδεμένες με την σημερινή τεχνική ανάπτυξης των ημιαγωγών, αναλογιζόμενοι επιπλέον το όριο που τίθεται για τις τάσεις (λειτουργίας και  $V_t$ ), από τις φυσικές ιδιότητες των ίδιων των ημιαγωγών.

Για την στατική κατανάλωση, η οποία εξαρτάται κι αυτή από το  $V_t$ , παρατηρούμε πως ανάλογα με το μοντέλο (τάσεις κατωφλίου  $V_t$ ), λαμβάνονται μικρότερες ή μεγαλύτερες τιμές. Κοινή διαπίστωση είναι ότι στην τεχνολογία κάτω των 22nm, αυξάνεται απότομα, εξαιτίας της αύξησης των ρευμάτων διαρροής στις ανάστροφα

πολωμένες επαφές των τρανζίστορ των cells. Βέβαια οι τιμές των ποσοστών της στατικής ισχύος έναντι της ολικής, απέχουν από τις αναμενόμενες, αλλά όπως εξηγήθηκε και στην αποτύπωση των σχετικών γραφημάτων, η αναζήτηση της αιτίας πρέπει να γίνει στα ποιοτικά χαρακτηριστικά της εξομοίωσης των designs. Σε ποιο ποσοστό δηλαδή συμμετέχουν τα διάφορα δομικά στοιχεία του στις λειτουργίες που εκτελούνται και για πόσο χρόνο μένουν ανενεργά, ώστε να είναι δυνατή η εξαγωγή αξιόπιστων τιμών στατικής κατανάλωσης. Απόδειξη της πρότασης αυτής είναι τα αποτελέσματα σε επίπεδο συστήματος, όπου το ποσοστό της στατικής κατανάλωσης είναι μέσα στα αναμενόμενα (για τη συγκεκριμένη τεχνολογία 32nm) περιθώρια.

Αναλυτικότερα εξετάζοντας τα αποτελέσματα που έδωσε η σύνθεση των κυκλωμάτων, διαπιστώνουμε σαφή ελάττωση στην καθυστέρηση και στη δυναμική κατανάλωση κατά τη μετάβαση από την τεχνολογία των 180nm στην τεχνολογία των 45nm, με μικρή αύξηση της κατανάλωσης διαρροών (συγκρίνοντας πάντα για όμοιες συνθήκες επεξεργασίας και χαρακτηρισμού). Στην τεχνολογία 32nm παρατηρείται και πάλι ελάττωση στη δυναμική κατανάλωση και αύξηση για την τιμή της καθυστέρησης, σε σχέση μ' αυτές της τεχνολογίας 45nm, η οποία όπως αναφέραμε εξαρτάται από το αντίστροφο της δύναμης του “α” της διαφοράς  $V_{dd}-V_t$ .

Η ανοδική τάση της στατικής κατανάλωσης σε σχέση με την ολική κατανάλωση είναι εμφανής, για το μοντέλο χαμηλής κατανάλωσης.

Για τις τεχνολογίες 22 και 16nm παρατηρείται σταθεροποιητική τάση στην τιμή της καθυστέρησης και τάση ελάττωσης της δυναμικής κατανάλωσης. Για την στατική κατανάλωση παρατηρείται απότομη αύξηση στην τεχνολογία 16nm.

Για τα βήματα που ακολουθήθηκαν, εμπεδώθηκε ένας αλγόριθμος που μπορεί να εφαρμοσθεί αποτελεσματικά σε κάθε περίπτωση που απαιτείται η κατασκευή μιας νέας βιβλιοθήκης. Η ανάγκη αυτή μπορεί να αφορά την εμφάνιση νέων spice model cards των τρανζίστορ για παράδειγμα ή την κατασκευή βιβλιοθήκης σε νέα τεχνολογία. Για την διαδικασία κλιμάκωσης των αρχείων .lef προέκυψε αντίστοιχος αλγόριθμος που μπορεί να εφαρμοσθεί μελλοντικά σε παρόμοια προβλήματα.

## 6.2. Προτάσεις και μελλοντικές προεκτάσεις

Περιθώρια βελτίωσης της διαδικασίας αφορούν το μέρος του χαρακτηρισμού της βιβλιοθήκης.

- Κυρίως η εξαγωγή παρασιτικών χωρητικοτήτων της εξόδου κάθε κυττάρου (cell), θα έδινε ρεαλιστικότερα αποτελέσματα χρονισμού.
- Επίσης θα μπορούσε να επιχειρηθεί αντί της γραμμικής κλιμάκωσης κάποιο άλλο είδος εισάγοντας μη γραμμικότητα ανάλογα με κάποιο χαρακτηριστικό (π.χ. ανάλογα με το είδος του cell).
- Η επέκταση των βιβλιοθηκών για κατασκευαστικές ανοχές και για καλώδια στις εκάστοτε τεχνολογίες. Δηλαδή εύρεση κατάλληλων παραμέτρων και τιμών για τις κατασκευαστικές ανοχές, όπως το μήκος και νόθευση του καναλιού και ο χαρακτηρισμός των .lib, με στατιστικό τρόπο (με χρήση τεχνικών Monte Carlo) για τις ανοχές.
- Σχεδιασμός βελτιστοποιημένων κελιών για μείωση κατανάλωσης στατικής ισχύος, όπως εισαγωγή επιπλέον τρανζίστορ για power gating, ή/και χρήση τρανζίστορ με διαφορετικές τάσεις κατωφλίου.



# ΠΑΡΑΡΤΗΜΑ

## ΠΑΡΑΡΤΗΜΑ 1

Σ' αυτό το παράρτημα δίνονται οι java κώδικες των εφαρμογών που αναπτύχθηκαν για το scaling τόσο της netlist τεχνολογίας 180 nm όσο και του lef αρχείου της ίδιας τεχνολογίας.

### 7.1. ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ ΚΛΙΜΑΚΩΣΗΣ (SCALING) ΤΟΥ SPICE NETLIST

Ο πρώτος κώδικας αφορά την εφαρμογή που αναλαμβάνει το scaling της SPICE netlist. Αυτή η εφαρμογή δέχεται ως είσοδο το αρχικό αρχείο και το κενό αρχείο στο οποίο θα καταγραφεί η κλιμακωμένη netlist. Το μέρος του κώδικα που ακολουθεί περιέχει μόνο την κλάση που αναπτύχθηκε για την κλιμάκωση και έχει παραληφθεί όλος ο υπόλοιπος κώδικας που αφορά το user interface.

```
public class str_replacement {
    public void str_replacement(){
        /*ορισμός της κλάσης που θα περιέχει μέθοδο για το scaling των
        παραμέτρων που περιέχει το αρχείο osu018_stdcells.sp*/
    }
    public void replaceAllWords1(FileReader fr) {
        /*ορισμός της μεθόδου replaceAllWords. Η μέθοδος αυτή κάνει το scaling
        των παραμέτρων*/
        if (jTextField1.getText().equals("")){ /*σε περίπτωση που το
        πεδίο που δέχεται το όνομα του αρχείου εισόδου, είναι κενό, τύπωσε
        το αντίστοιχο μήνυμα*/

        ShowDialogBox db = new ShowDialogBox();

        return;
    }

    int
    scaling_factor=Integer.parseInt(jTextField1.getText());
```

```

double w=0, ad=0, pd=0, as=0, ps=0; //ορισμός μεταβλητών
BufferedReader br = null;
br = new BufferedReader(fr);/* δημιουργία αντικειμένου
BufferedReader για άνοιγμα και διάβασμα του αρχείου
osu018_stdcells.sp μέσω ειδικού stream*/

String str;

PrintWriter outputStream = null;

try {
    outputStream=new PrintWriter(new
FileOutputStream(file1.getPath())); /*δημιουργία stream σύνδεσης
με το αρχείο εισόδου */
    } catch (FileNotFoundException e) {
System.out.println("Error opening the output file. " +
e.getMessage());
    }

    try {
        while ((str = br.readLine()) != null) {/*όσο η
είσοδος από το stream που συνδέει το αρχείο
osu018_stdcells.sp δεν είναι κενό, διάβασε την
είσοδο*/

            if (str.contains("w=2u")){w=2; /*εάν το string που
έλαβες περιέχει τον χαρακτήρα("w=2u", τότε
αντικατέστησέ το με "w=2u/scaling_factor"
και "l=0.2"με "l="+((double)0.2/scaling_factor */

str=str.replaceAll("w=2u", "w="+((double)2/scaling_factor+"u");
str=str.replaceAll("l=0.2u",
"l="+((double)0.2/scaling_factor+"u");

        }
/*παρομοίως κάνε τις αντικαταστάσεις για όλες τις δυνατές περιπτώσεις w */
        else if (str.contains("w=1u")){
            w=1;
str=str.replaceAll("w=1u", "w="+w/scaling_factor+"u");
str=str.replaceAll("l=0.2u", "l="+((double)0.2/scaling_factor+"u");

        }

        else if (str.contains("w=4u")){
            w=4;
str=str.replaceAll("w=4u", "w="+w/scaling_factor+"u");
str=str.replaceAll("l=0.2u", "l="+((double)0.2/scaling_factor+"u");

        }

        else if (str.contains("w=3u")){
            w=3;
str=str.replaceAll("w=3u", "w="+w/scaling_factor+"u");
str=str.replaceAll("l=0.2u", "l="+((double)0.2/scaling_factor+"u");

```

```

    }
    else if (str.contains("w=1.5u")){
        w=1.5;
str=str.replaceAll("w=1.5u", "w="+w/scaling_factor+"u");
str=str.replaceAll("l=0.2u", "l="+((double)0.2/scaling_factor+"u");

    }

    else if (str.contains("w=3.6u")){
        w=3.6;
str=str.replaceAll("w=3.6u", "w="+w/scaling_factor+"u");
str=str.replaceAll("l=0.2u", "l="+((double)0.2/scaling_factor+"u");

    }

        else if (str.contains("w=4.8u")){
            w=4.8;
str=str.replaceAll("w=4.8u", "w="+w/scaling_factor+"u");
str=str.replaceAll("l=0.2u", "l="+((double)0.2/scaling_factor+"u");

        }

ad=as*w/scaling_factor*w/scaling_factor; pd=ps=4*w/scaling_factor;
str=str.replaceAll("ad=0p", "ad="+ad+"p"); str=str.replaceAll("as=0p",
"as="+as+"p"); str=str.replaceAll("pd=0u", "pd="+pd+"u");
str=str.replaceAll("ps=0u", "ps="+ps+"u");

        outputStream.println(str);

        /*υπολόγισε και τις τιμές των ad, ad, pd,
ps)*/ } catch (IOException ex) {

            Logger.getLogger(Library_scalingView.class.getName()).log
(L level.SEVERE, null, ex);

        }
        outputStream.close(); //μετά την ανάγνωση της τελευταίας
γραμμής του αρχείου, κλείσε το αρχείο*/

    }

}

```

Επίσης δίνεται και ο κώδικας που ενσωματώνεται πίσω απ' τα δύο κουμπιά διάδρασης.

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt)
{
    str_replacement replace =new str_replacement();
    //Δημιούργησε νέο object str_replacement()με όνομα replace
    try
    {
        FileReader fr = new FileReader(file); //Δημιούργησε νέο
object fileReader με όνομα fr

        replace.replaceAllWords1(fr); /*εφάρμοσε τη
μέθοδο replaceAllWords1 της κλάσης str_replacement στο
string του fr*/
        }catch(FileNotFoundException e)
        {
        }catch(IOException e)
        {
        }
    }

}

private void jTextField3MouseClicked(java.awt.event.MouseEvent
evt) {
    JFileChooser fc1 = new JFileChooser();
    int returnVal = fc1.showOpenDialog(fc1);
    if (returnVal ==
JFileChooser.APPROVE_OPTION) { file1=
fc1.getSelectedFile();
    jTextField3.setText( file1.getName());
    }
}
}

```

## 7.2. ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ ΚΛΙΜΑΚΩΣΗΣ ΤΟΥ ΑΡΧΕΙΟΥ LEF

Ο επόμενος κώδικας αφορά την εφαρμογή που αναλαμβάνει το scaling του LEF αρχείου και όπως και προηγουμένως είναι μόνο το μέρος του κώδικα που αφορά την κλάση που αναπτύχθηκε για αυτό τον σκοπό.

```

public class str_replacement {
    public void str_replacement() {
    }/*ορισμός της κλάσης που θα περιέχει μέθοδο για το scaling των
παραμέτρων που περιέχει το αρχείο osu018_stdcells.lef*/

    public void replaceAllWords1(FileReader fr) {

```



```

        /*ορισμός της μεθόδου replaceAllWords1. Η μέθοδος αυτή
        κάνει το scaling των παραμέτρων που περιέχει το αρχείο
        osu018_stdcells.lef*/
if
    (jTextField1.getText().equals(""))
    ){ ShowDialogBox db = new
    ShowDialogBox();
    /*σε περίπτωση που το πεδίο που δέχεται το όνομα του αρχείου
    εισόδου, σταμάτησε*/

    return;
}
//ορισμός μεταβλητών
    double
    scaling_factor=Double.parseDouble(jTextField1.getText());
boolean macro=false, odd=false; /*ορισμός flags για έλεγχο εάν
βρισκόμαστε μέσα σε ένα macro cell και επεξεργαζόμαστε τις διαστάσεις
του. Επίσης εάν βρισκόμαστε σε άρτιο επίπεδο διασύνδεσης. */
    NumberFormat formator = new DecimalFormat("0.000"); /*ορισμός
    format για την ακρίβεια των δεκαδικών ψηφίων του manufacturing
    grid*/
    double number ;
    double MAN_GRID=0.05/scaling_factor;
    BigDecimal bd = new BigDecimal("0");
    String temp=formator.format(MAN_GRID);
    temp=temp.replaceAll(",",".");
    System.out.print(temp);
    MAN_GRID=Double.parseDouble(temp);

    String target;
    BufferedReader br=null;
    br = new BufferedReader(fr); /*δημιουργία ειδικού αντικειμένου
    για την αποθήκευση του character stream που λαμβάνεται από
    το αντικείμενο σύνδεσης με το αρχείο εισόδου*/

    String str;

    PrintWriter outputStream = null;

    try {
        outputStream = new PrintWriter(new
        FileOutputStream(file1.getPath()));
        //δημιουργία αντικειμένου σύνδεσης με το αρχείο εισόδου
    } catch (FileNotFoundException e) {
        System.out.println("Error opening the output file. " +

```

```

e.getMessage());
    }
    try {
        while ((str = br.readLine()) != null) {
            /*διάβαζε συνεχώς το αρχείο γραμμή γραμμή, όσο
            αρχείο δεν έχει φθάσει τέλος*
            */
            if (str.contains("MANUFACTURINGGRID 0.05 ;")){
                /*αντικατάσταση manufacturing grid με τη νέα τιμή
                manufacturing grid/scaling factor, κρατώντας πάντα
                την ακρίβεια στα τρία δεκαδικά ψηφία */
                str=str.replaceAll("MANUFACTURINGGRID
                0.05 ;", "MANUFACTURINGGRID "+formatter.format(MAN_GRID)+"
                ;");
            }
            if ((str.contains("SIZE "))) {
                /*αν υπάρχει η λέξη κλειδί SIZE πρέπει να γίνει scaling στις τιμές
                του SIZE*/
                NumberFormat num = new DecimalFormat("0.000");
                //επιλογή ακρίβειας αποτελέσματος
                target
                =str.substring(str.indexOf("SIZE")+1,
                str.indexOf(";"));

                StringTokenizer st = new StringTokenizer(target);
                /*δημιουργία αντικειμένου για χειρισμό του string κάθε γραμμής
                διαβάσματος*/
                str=" SIZE ";
                while (st.hasMoreTokens()) {
                    //επέλεξε το πρώτο και τρίτο string μετά τη λέξη κλειδί, που
                    περιέχουν τις τιμές και κάνει scaling, ελέγχοντας ταυτόχρονα και την
                    ακέραια πολλαπλότητα ως προς το manufacturing grid*/
                    if (st.countTokens()==1){
                        number=Double.parseDouble(
                        st.nextToken())/ scaling_factor;
                        if ((number%MAN_GRID) != 0.0){
                            number--=(number%MAN_GRID);
                        }
                        str=str+num.format(number)+" ";
                    }
                    else if (st.countTokens()==3){
                        number=Double.parseDouble(
                        st.nextToken());
                        if ((number%MAN_GRID) !=
                        0.0){ number
                        =(number%MAN_GRID);
                        }
                    }

                    str=str+num.format(number)+" ";
                }
            }
        }
    }
}

```

```

    }
    else
    str=str+st.nextToken()
    +" ";
    str=str.replaceAll(", "
    , ".");
    }
    str=str+";" ;
}
if (str.contains("MACRO ")) /*Αν περιέχεται η λέξη
κλειδί
MACRO σήκωσε την σχετική σημαία */
    macro
    =true; if
    (macro){
        /*Εάν βρισκόμαστε μέσα στο MACRO Cell και
        συναντήσουμε τις λέξεις κλειδιά metal1, metal3, metal5, metal7,
        metal9, ανεβάζουμε τη σημαία περιτιού layer, διαφορετικά την αφήνουμε
        false*/
        if (str.contains("metal1") ||
        str.contains("metal3")|| str.contains("metal5")||
        str.contains("metal7")|| str.contains("metal9"))
            odd=true;
        else if (str.contains("metal2") ||
        str.contains("metal4")|| str.contains("metal6")||
        str.contains("metal8"))
            odd=false;
        }
        if (!(macro)){
        //Για οτιδήποτε είναι εκτός MACRO, κάνουμε scaling ανάλογο του
        συντελεστή κλιμάκωσης και ελέγχουμε ως προς τη διατήρηση του ακέραιου
        πολλαπλάσιου του manufacturing grid
            if (str.contains("WIDTH 0.3 TO 60 ;")){
                if (((0.3/scaling_factor)%MAN_GRID) !=
                0.0){ str="WIDTH
"+formator.format((0.3/scaling_factor)-
                ((0.3/scaling_factor)%MAN_GRID)); if
                (((60.0/scaling_factor)%MAN_GRID) != 0.0)
                    str=str+" TO
"+formator.format(60.0/scaling_factor-
                (60.0/scaling_factor)%MAN_GRID)+" ";
                    else
                        str=str+" TO "+60.0/scaling_factor+" ";
                }
                else
                    str="WIDTH
"+(0.3)/scaling_factor+" TO "+60.0/scaling_factor+"
";
            }

```

```

    }
    else if (str.contains("WIDTH 0.5 TO 60 ;")){
        if (((0.5/scaling_factor)%MAN_GRID) !=
            0.0){ str="WIDTH
"+formator.format(0.5/scaling_factor-
                    (0.5/scaling_factor)%MAN_GRID); if
                    ((60.0/scaling_factor)%MAN_GRID) !=
                    0.0)
                        str=str+" TO
"+formator.format(60.0/scaling_factor-
(60.0/scaling_factor)%MAN_GRID)+" ";
                else
                    str=str+" TO "+60.0/scaling_factor+" ";
            }
            else
                str="WIDTH
"+0.5/scaling_factor+" TO "+60.0/scaling_factor+"
";
        }
        else if (str.contains("0.45 ")){
            if (((0.45/scaling_factor)%MAN_GRID)
                != 0.0) str=str.replaceAll("0.45 ",
formator.format((0.45/scaling_factor)-
(0.45/scaling_factor)%MAN_GRID)+" "); else
                str=str.replaceAll("0.45 ", 0.45/scaling_factor+"
");
        }
        else if (str.contains("1 ;")){
            if (((1.0/scaling_factor)%MAN_GRID)
                != 0.0) str=str.replaceAll("1 ;",
formator.format((1.0/scaling_factor)-
((1.0/scaling_factor)%MAN_GRID)+" ");
            else
                str=str.replaceAll("1 ;", 1.0/scaling_factor+" ");
        }
        else if (str.contains("0.1 ")){
            if (((0.1/scaling_factor)%MAN_GRID)
                != 0.0) str=str.replaceAll("0.1 ",
formator.format(0.1/scaling_factor-
((0.1/scaling_factor)%MAN_GRID)+" "); else
                str=str.replaceAll("0.1 ", 0.1/scaling_factor+" ");
        }
        }
        else if (str.contains("0.3 ")){
            BigDecimal divided=new

```

```

        BigDecimal("0.3"); if
        (((divided.divide(bd.valueOf( scaling_factor))).remainder(bd.valueOf(M
        AN_GRID))).doubleValue() != 0.0){
            System.out.print("0.3/scaling_factor%MAN
        _GRID= "+0.3/scaling_factor % 0.005+"\n");
            str=str.replaceAll("0.3 ",
        formator.format(((divided.divide(bd.valueOf( scaling_factor))).double
        Value( )-
        ((divided.divide(bd.valueOf( scaling_factor))).remainder(bd.valueOf(MA
        N_GRID))).doubleValue()))+" ");
        }
        else
        str=str.replaceAll("0.3 ", 0.3/scaling_factor+" ");
    }
    else if (str.contains("0.4 ")){
        if (((0.4/scaling_factor)%MAN_GRID) !=
        0.0){ str=str.replaceAll("0.4 ",
        formator.format(0.4/scaling_factor-
        ((0.4/scaling_factor)%MAN_GRID))+ " ");
        System.out.print("0.4/scaling_factor%MAN_GRID=
        "+0.4/scaling_factor % 0.005+"\n");
        }
        else
        str=str.replaceAll("0.4 ", 0.4/scaling_factor+" ");
    }
    else if (str.contains("0.5 ")){
        if (((0.5/scaling_factor)%MAN_GRID)
        != 0.0) str=str.replaceAll("0.5 ",
        formator.format(0.5/scaling_factor-
        ((0.5/scaling_factor)%MAN_GRID))+ " "); else
        str=str.replaceAll("0.5 ", 0.5/scaling_factor+" ");
    }
    else if (str.contains("0.6 ")){
        if (((0.6/scaling_factor)%MAN_GRID)
        != 0.0) str=str.replaceAll("0.6 ",
        formator.format(0.6/scaling_factor-
        ((0.6/scaling_factor)%MAN_GRID))+ " "); else
        str=str.replaceAll("0.6 ", 0.6/scaling_factor+" ");
    }
    else if (str.contains("0.7 ")){
        if (((0.7/scaling_factor)%MAN_GRID)
        != 0.0) str=str.replaceAll("0.7 ",
        formator.format(0.7/scaling_factor-((0.7/scaling_factor)%MAN_GRID))+
        " ");

```

```

        else
            str=str.replaceAll("0.7 ", 0.7/scaling_factor+" ");
        }
        else if (str.contains("0.8 ")){
            if (((0.8/scaling_factor)%MAN_GRID) !=
                0.0) str=str.replaceAll("0.8 ",
formator.format(0.8/scaling_factor-((0.8/scaling_factor)%MAN_GRID))+
                "); else
            str=str.replaceAll("0.8 ", 0.8/scaling_factor+" ");
        }
        else if (str.contains("1.6 ")){
            if (((1.6/scaling_factor)%MAN_GRID) !=
                0.0) str=str.replaceAll("1.6 ",
formator.format(1.6/scaling_factor-((1.6/scaling_factor)%MAN_GRID))+
                "); else
            str=str.replaceAll("1.6 ", 1.6/scaling_factor+" ");
        }
        else if (str.contains("60 ")){
            if (((60/scaling_factor)%MAN_GRID) !=
                0.0) str=str.replaceAll("60 ",
formator.format(60/scaling_factor-((60/scaling_factor)%MAN_GRID))+
                "); else
            str=str.replaceAll("60 ", 60.0/scaling_factor+" ");
        }
        else if (str.contains("RECT
        ")){ Double num=0.0 ;
target
        =str.substring(str.indexOf("RECT")+ "RECT".length()+1,
        str.indexOf(";"));

            StringTokenizer st = new
            StringTokenizer(target);
            System.out.println(target);
            str="          RECT ";
            while (st.hasMoreTokens()) {
                number=Double.parseDouble(
st.nextToken())/scaling_factor;
                if (( number%MAN_GRID) != 0.0)
                    number= number-
                    number%MAN_GRID;

                str=str+formator.format(number)+" ";
            }

            str=str+";" ;

```

```

    }
        else if (str.contains("RESISTANCE
                                RPERSQ
                                Q ")){
            NumberFormat num = new DecimalFormat("0.000");
            target =str.substring(str.indexOf("RESISTANCE
RPERSQ")+ "RESISTANCE    RPERSQ".length()+1, str.indexOf(";"));
            StringTokenizer st = new
StringTokenizer(target);
            str=" RESISTANCE    RPERSQ ";
            while (st.hasMoreTokens()) {
                number=Double.parseDouble(
st.nextToken()*scaling_factor;
                str=str+num.format(number)+" ";
            }

            str=str+";" ;
        }

        else if (str.contains("THICKNESS
                                ")){
            NumberFormat num = new DecimalFormat("0.000");
            target =str.substring(str.indexOf("THICKNESS
                                ")+"THICKNESS
                                ".length(),
str.indexOf(";"));

            str=" THICKNESS
                                ";
            str=str+num.format(number)+" ";
            str=str+";" ;
        }

        else if (str.contains("CAPACITANCE    CPERSQDIST
                                ")){
            NumberFormat num = new DecimalFormat("0.000");

            target =str.substring(str.indexOf("CAPACITANCE
CPERSQDIST")+ "CAPACITANCE
CPERSQDIST".length()+1, str.indexOf(";"));
            StringTokenizer st = new StringTokenizer(target);
            str=" CAPACITANCE    CPERSQDIST ";
            while (st.hasMoreTokens()) {
                number=Double.parseDouble(
st.nextToken().toUpperCase()*scaling_factor;
                str=str+number+" ";
                str=str.replaceAll(",",".");
            }

            str=str+";" ;

```

```

    }

    else if (str.contains("EDGECAPACITANCE ")){
        NumberFormat num = new DecimalFormat("0.000");

target
=str.substring(str.indexOf("EDGECAPACITANCE")+ "EDGECAPACITANCE".length()+1, str.indexOf(";"));
        StringTokenizer st = new
            StringTokenizer(target
                ); str="
EDGECAPACITANCE ";
        while
            (st.hasMoreTokens()) {
number=Double.parseDouble(st.nextToken().toUpperCase())*scaling_factor; str=str+number+" ";
            str=str.replaceAll(",",".");
        }

        str=str+";" ;
    }

}

// Αν βρισκόμαστε μέσα στο MACRO, τότε κάνουμε scaling κατά τη
// διεύθυνση X ή Y του παραλληλόγραμμου σχήματος των λωρίδων μετάλλου
// ανάλογα με το αν έχουμε περιττό ή άρτιο επίπεδο
    else if ((macro&&str.contains("RECT
"))) { Double num =0.0;

        target
=str.substring(str.indexOf("RECT")+ "RECT".length()+1,
str.indexOf(";"));

        System.out.println(target);
        StringTokenizer st = new
            StringTokenizer(target);

str="                RECT                ";
        while (st.hasMoreTokens()) {
// Εάν βρισκόμαστε σε περιττό επίπεδο τότε γίνεται scaling κατά τον
// άξονα Y γιατί ο προσανατολισμός είναι οριζόντιος, κρατώντας πάντα το
// αποτέλεσμα ακέραιο πολλαπλάσιο του manufacturing grid
            if

                (st.countTokens()==3||st.countTokens()==1){
                    number=Double.parseDouble(

```



```

st.nextToken())/scaling_factor;

        if (( number%MAN_GRID) !=
            0.0) number= number-
            number%MAN_GRID;
        }
        else{
            number=Double.parseDouble(
                st.nextToken()); if ((
                number%MAN_GRID) != 0.0)
                number= number- number%MAN_GRID;
            }

            str=str+formator.format(number)+" ";
        }

// Εάν βρισκόμαστε σε άρτιο επίπεδο τότε γίνεται scaling κατά τον
// άξονα X γιατί ο προσανατολισμός είναι κατακόρυφος, κρατώντας πάντα το
// αποτέλεσμα ακέραιο πολλαπλάσιο του manufacturing grid
        else if (!odd)
            {
                if (st.countTokens()==4||st.countTokens()==2){
                    number=Double.parseDouble(st.nextToken())/scaling_factor; if (( number%MAN_GRID) != 0.0)
                    number= number- number%MAN_GRID;
                }
            }

        else{
            number=Double.parseDouble(
                st.nextToken()); if ((
                number%MAN_GRID) != 0.0)
                number= number-
                number%MAN_GRID;
            }
            str=str+formator.format(number)+" ";
        }
    }

// Μορφοποίησε κατάλληλα το string ώστε να υπάρχει συμβατότητα με τη
// σύνταξη του LEF αρχείου
        str=str+";" ;
    }

        str=str.replaceAll(", ",
        , ".");
        outputStream.println(s
        tr);

    }

} catch (IOException ex) {

```

```

        Logger.getLogger(Library_scalingView.class.getName())
            .log(L
level.SEVERE, null, ex);
    }

    outputStream.close();
}
}

```

Ακολουθεί τέλος ο κώδικας που ενσωματώθηκε στα κουμπιά διάδρασης με την εφαρμογή.

```

        JFileChooser fc1 = new JFileChooser();
        int returnVal = fc1.showOpenDialog(fc1);
        if (returnVal ==
        JFileChooser.APPROVE_OPTION) { file1=
        fc1.getSelectedFile();
// Πέρασμα του ονόματος του επιλεγμένου αρχείου στο κατάλληλο
TextField
jTextField3.setText( file1.getName());
        }
    }

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {

        JFileChooser fc = new
        JFileChooser(); int returnVal =
        fc.showOpenDialog(fc);

        if (returnVal ==
        JFileChooser.APPROVE_OPTION) { file=
        fc.getSelectedFile();
        jTextField2.setText( file.getPath());
// Πέρασμα του path του επιλεγμένου αρχείου στο κατάλληλο TextField
        }

    }

private void jButton2MouseClicked(java.awt.event.MouseEvent
evt)
{

        str_replacement replace =new str_replacement();

        //Με το πάτημα του button, δημιουργία αντικειμένου της
κλάσης str_replacement που δημιουργήσαμε
try
    {
        replace.replaceAll(fr);
    }catch(FileNotFoundException e)
    {

```

```
    }catch (IOException e)  
  
}
```



## ΠΑΡΑΡΤΗΜΑ 2

### 8.1. SIMULATION SETUP FILE

Αυτό το αρχείο είναι ένα ASCII αρχείο που χρησιμοποιείται ως είσοδος για τον χαρακτηρισμό της βιβλιοθήκης και περιέχει διάφορες πληροφορίες όπως για τα (process corner), τιμές θερμοκρασίας και τάσης (Vdd). Άλλα στοιχεία χρήσιμα για την εξομοίωση είναι οι στάθμες τάσεως για τα διάφορα κατώφλια (Vtp, Vtn), καθώς και τις high/low λογικές στάθμες. Τα χωρητικά φορτία που παρουσιάζουν οι ακροδέκτες σύνδεσης εξόδων, οι συνθήκες slew rate των ακροδεκτών εισόδου, τα περιθώρια των αποτελεσμάτων χαρακτηρισμού, οι παράγοντες διόρθωσης για τον υπολογισμό των αποτελεσμάτων του χαρακτηρισμού και περιορισμοί για χρόνους καθυστέρησης, συγκράτησης και αποκατάστασης είναι απαραίτητες πληροφορίες που μεταφέρει το setup αρχείο. Ακολουθεί η δομή του setup αρχείου που θα χρησιμοποιηθεί στον χαρακτηρισμό. Το αρχείο setup χωρίζεται σε δύο επιμέρους κομμάτια. Το πρώτο είναι το define section και το δεύτερο είναι το control section. Στο define section γίνεται ο καθορισμός των παραμέτρων, των groups των μεταβλητών και των διαφόρων περιθωρίων που θα χρησιμοποιηθούν από το control section.

```
set process (best, typical, worst) {
simulation = STD_CELLS_45NM_SIM ;
index = X1 ;
```

Οι συντακτικοί κανόνες του setup αρχείου επιτρέπουν τη χρήση δύο wildcards. Ο χαρακτήρας '\*' δηλώνει την ύπαρξη άγνωστου αριθμού χαρακτήρων μετά ενός γνωστού string. Για παράδειγμα το string "Mem\*" δηλώνει οτιδήποτε ταιριάζει με το Mem (Memory, Mem1, Mem32K κλπ.). Ο χαρακτήρας '?' δηλώνει την ύπαρξη ενός χαρακτήρα, πέραν του string του οποίου προηγείται ή έπεται. Η group είναι μια παράμετρος, η οποία προδιαγράφει διάφορες ομάδες από κύτταρα ή pins. Η δήλωση index προδιαγράφει τις τιμές των slew rates εισόδου και των χωρητικότητων εξόδου, που θα χρησιμοποιηθούν κατά τον χαρακτηρισμό της βιβλιοθήκης. Τα slews που προσδιορίζονται από το RSLEW ή το FSLEW υπερσχύουν αυτών που καθορίζονται από την παράμετρο SLEW. Ο ελάχιστος αριθμός των τιμών του διάνυσματος είναι 2 και ο μέγιστος 32. Όπως βλέπουμε και στο προηγούμενο παράδειγμα, κάθε index περιέχει ένα διάνυσμα με τιμές slew και ένα διάνυσμα με χωρητικά φορτία. Μέσω

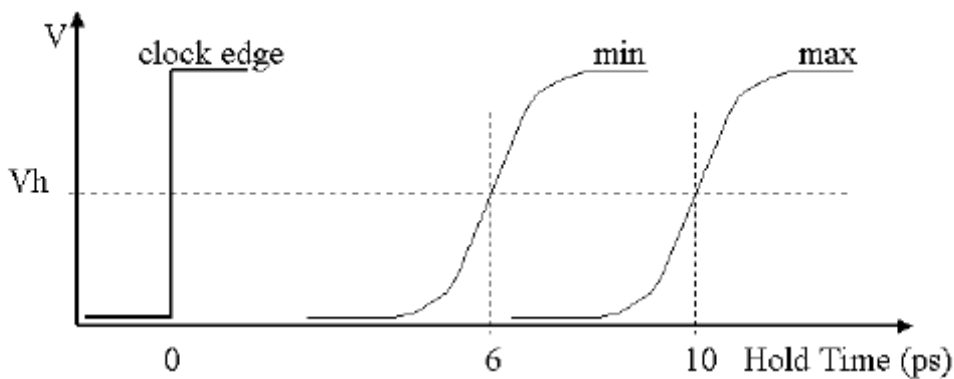
αυτών των διανυσμάτων, κατά το χρόνο των εξομοιώσεων που γίνονται στον χαρακτηρισμό, δημιουργούνται look up πίνακες που περιέχουν πληροφορία χρόνων καθυστέρησης, μεταφοράς και δυναμικής κατανάλωσης. Επιπλέον σε κάθε Index υπάρχει και ένα διάνυσμα BSLEW. Αυτό έχει τιμές για δυαδική (binary) αναζήτηση<sup>12</sup> (6). Με τη δήλωση Margin ορίζονται οι παράγοντες των περιθωρίων για τον χαρακτηρισμό της βιβλιοθήκης. Οι παράγοντες αυτοί περιλαμβάνουν δυο τιμές. Η μία είναι η σχετική τιμή (scale) και η άλλη είναι η απόλυτη τιμή (offset). Οι τιμές αυτές εφαρμόζονται στα αποτελέσματα με τον τύπο  $results = number * scale + offset$ . Οι δηλώσεις των παραγόντων περιθωρίων είναι ταξινομημένες στον πίνακα που ακολουθεί:

Πίνακας 8.1.1 Κυριότερες παράμετροι του setup αρχείου.

Descriptor	Keyword	Type	Default
Input capacitance Factors	Cap	<i>scale, offset</i>	1.0, 0.0
Wire capacitance factors	Wcap	<i>scale, offset</i>	1.0, 0.0
Wire resistance factors	Wresist	<i>scale, offset</i>	1.0, 0.0
Cell delay factors	Delay	<i>scale, offset</i>	1.0, 0.0
Effective capacitance Factors	Ecap	<i>scale, offset</i>	1.0, 0.0
Power consumption Factors	Power	<i>scale, offset</i>	1.0, 0.0
Slew rate factors	Slew	<i>scale, offset</i>	1.0, 0.0
I/O path delay factors	Iopath	<i>scale, offset</i>	1.0, 0.0
Interconnect delay Factors	Interconnect	<i>scale, offset</i>	1.0, 0.0
Setup constraint factors	Setup	<i>scale, offset</i>	1.0, 0.0
Hold constraint factors	Hold	<i>scale, offset</i>	1.0, 0.0
Release constraint Factors	Release	<i>scale, offset</i>	1.0, 0.0
Removal constraint Factors	Removal	<i>scale, offset</i>	1.0, 0.0
Recovery constraint Factors	Recovery	<i>scale, offset</i>	1.0, 0.0
Pulse width constraint Factors	Width	<i>scale, offset</i>	1.0, 0.0

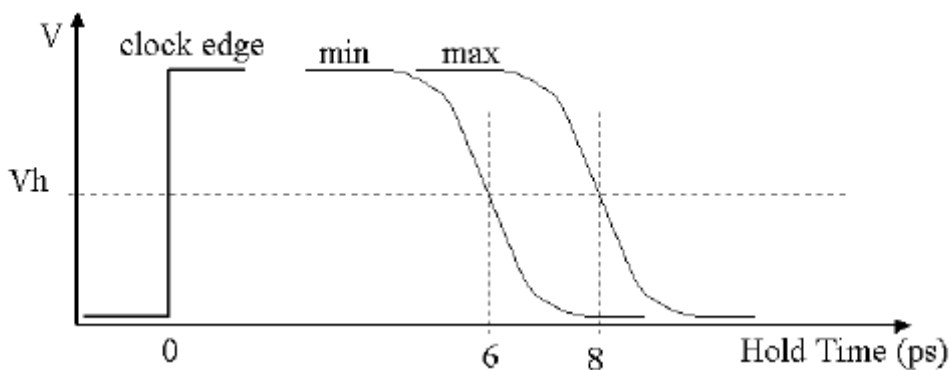
<sup>12</sup> Πρόκειται για επαναληπτική μέθοδο αναζήτησης, όπου κάθε φορά το διάστημα αναζήτησης διχοτομείται και ανάλογα με το αν η τιμή στη διχοτόμο είναι μεγαλύτερη ή μικρότερη η διαδικασία συνεχίζεται αντίστοιχα στο πριν ή μετά τη διχοτόμο διάστημα. Αυτό επαναλαμβάνεται μέχρι η τιμή που αναζητάμε να απέχει από την διχοτόμο  $\pm$  ένα αποδεκτό περιθώριο.

Αντίστοιχα με τη δήλωση *nominal*, προσδιορίζονται οι παράγοντες μέσω των οποίων υπολογίζονται οι μέσες τιμές για τον χαρακτηρισμό της βιβλιοθήκης. Εδώ μπορούν να καθοριστούν δύο τιμές *scale*. Μια για τους ανερχόμενους και μια για τους κατερχόμενους παλμούς. Ο υπολογισμός των μέσων τιμών ενός μεγέθους, γίνεται από την μέγιστη και την ελάχιστη τιμή του σύμφωνα με τη σχέση  $average\_value = (max - min) * factor + min$ . Ένα παράδειγμα για τον υπολογισμό του χρόνου συγκράτησης για ανερχόμενο παλμό φαίνεται στην Εικόνα 8.1.1



Εικόνα 8.1.1 Τρόπος υπολογισμού της μέσης τιμής του Hold time για ανερχόμενο παλμό.

Εάν στο *setup* αρχείο είχαμε *Nominal check 0.5 0.4*, τότε η μέση τιμή του χρόνου συγκράτησης είναι  $Average\ rise\ hold\ time = \{ (10 - 6) * 0.5 \} + 6 = 8ns$ . Αντίστοιχα για κατερχόμενο παλμό όπως στην Εικόνα 8.1.2 θα έχουμε  $Average\ fall\ hold\ time = \{ (10 - 6) * 0.4 \} + 6 = 8ns$ .



Εικόνα 8.1.2 Τρόπος υπολογισμού της μέσης τιμής του Hold time για κατερχόμενο παλμό.

Οι δηλώσεις για τις nominal τιμές καταγράφονται στον ακόλουθο πίνακα:

**Πίνακας 8.1.2 Δηλώσεις για τις Nominal τιμές διαφόρων μεγεθών.**

Descriptor	Keyword	Type	Default
Gate capacitance Factor	Cap	<i>value, value</i>	0.5, 0.5
Constraint factor	Check	<i>value, value</i>	1.0,1.0
Current source Factor	Current	<i>value, value</i>	0.5, 0.5
Power factor	Power	<i>value, value</i>	0.5, 0.5
Slew rate factor	Slew	<i>value, value</i>	0.5, 0.5
Delay factor	Delay	<i>value, value</i>	0.5, 0.5

Η δήλωση PROCESS προσδιορίζει τις συνθήκες εξομοίωσης. Αυτές είναι οι παράμετροι των MOS ( $V_t$ , Process Corner) και οι συνθήκες κάτω από τις οποίες γίνεται η εξομοίωση, όπως θερμοκρασία και τάση λειτουργίας. Πρέπει πριν οποιαδήποτε άλλη δήλωση παραμέτρων, να έχουν ήδη δηλωθεί οι παράμετροι της PROCESS. Οι παράμετροι που αφορούν τη δήλωση PROCESS καταγράφονται στον ακόλουθο πίνακα:

**Πίνακας 8.1.3 Παράμετροι που αφορούν την ενότητα Process του Setup αρχείου.**

Descriptor	Keyword	Type	Default
Voltage	Voltage	<i>Value</i>	None
Temperature	Temperature	<i>Value</i>	None
Process corners for loading cell models from library database	Gtcorner	<i>strings</i>	<i>process_name</i>
Library section names in the library model file for SPICE simulation	section	<i>String</i>	None
Library Model file Name	Lib	<i>String</i>	None
Model file name	Model	<i>String</i>	None
Voltage of NMOS Gate	Vtn	<i>Value</i>	None
Voltage of PMOS Gate	Vtp	<i>Value</i>	None
RC process corner	RCcorner	<i>String</i>	None



```

Process typical {
voltage = 1 ; // as voltage
temp = 25 ; /* as temperature */
Vtn = 0.469 ; // nmos Vt
Vtp = 0.418 ; // pmos Vt
} ;

//ff
Process best {
voltage = 1 ; // as voltage
temp = 0 ; /* as temperature */
Vtn = 0.424 ;
Vtp = 0.372 ;
};

//ss
Process worst {
voltage = 1 ; // as voltage
temp = 125 ; /* as temperature */
Vtn = 0.484 ;
Vtp = 0.432 ;
} ;

Signal STD_CELLS_45NM_GROUP {
unit = REL ; // relative value
Vh = 1.0 1.0 ; // 100% rise/fall
Vl = 0.0 0.0 ;
Vth = 0.5 0.5 ; // 50% rise/fall
Vsh = 0.8 0.8 ;
Vsl = 0.2 0.2 ;
tsmax = 1.0n ; // maximum output slew rate
} ;

Index X1 {
Load = 0.00082P 0.00330P 0.00842P 0.01848P 0.03861P 0.07870P
0.18975P ;
BSlew = 0.0385N 0.2N 1.0360N ; // optional for binary search
Slew = 0.0385N 0.047N 0.08N 0.1N 0.2N 0.8N 1.0360N ;
} ;

Index X2 {
load = 0.00165P 0.00660P 0.01683P 0.03696P 0.07722P 0.15741P 0.37950P
;
} ;

Index X4 {
load = 0.00330P 0.01320P 0.03366P 0.07392P 0.15444P 0.31482P 0.75900P
;
} ;

Index X8 {
load = 0.00660P 0.02640P 0.06732P 0.14784P 0.30888P 0.62964P 1.51800P
;
} ;

Index CLKBUF1 {

```

```
Load = 0.00082P 0.00330P 0.00842P 0.01848P 0.03861P 0.07870P
0.18975P ;
} ;

Index CLKBUF2 {
load = 0.00165P 0.00660P 0.01683P 0.03696P 0.07722P 0.15741P 0.37950P
;
} ;

Index CLKBUF3 {
load = 0.00248P 0.00990P 0.02524P 0.05544P 0.11583P 0.23612P 0.56925P
;
} ;

Index DFFSR {
Load = 0.00082P 0.00330P 0.00842P 0.01848P 0.03861P 0.07870P
0.18975P ;
BSlew = 0.006N 0.018N 0.024N ;
} ;

Index LATCH {
Load = 0.00082P 0.00330P 0.00842P 0.01848P 0.03861P 0.07870P
0.18975P ;
} ;

Index DFF {

BSlew = 0.1385N 0.3N 0.9N ;
} ;

Group X1 {
CELL = *X1 ;
} ;

Group X2 {
CELL = *X2 ;
} ;

Group X4 {
CELL = *X4 ;
} ;

Group X8 {
CELL = *X8 ;
} ;

Group CLKBUF1 {
CELL = *CLKBUF1 ;
} ;

Group CLKBUF2 {
CELL = *CLKBUF2 ;
} ;

Group CLKBUF3 {
CELL = *CLKBUF3 ;
```

```

} ;

Group DFFSR {
CELL = *DFFSR DFFPOSX1 DFFNEGX1 ;
} ;

Group LATCH {
CELL = *LATCH ;
} ;

Group DFF {
CLK_PIN = DFF*.CLK;
CELL = DFF*;
};

Margin m0 {
setup = 1.0 0.0 ;
hold = 1.0 0.0 ;
release = 1.0 0.0 ;
removal = 1.0 0.0 ;
recovery = 1.0 0.0 ;
width = 1.0 0.0 ;
delay = 1.0 0.0 ;
power = 1.0 0.0 ;
cap = 1.0 0.0 ;
} ;

Nominal n0 {
delay = 0.5 0.5 ; // as rise fall
power = 0.5 0.5 ;
cap = 0.5 0.5 ;
} ;

Simulation STD_CELLS_45NM_SIM {
transient = 1.0n 100n 10p ;
biseq = 10.0n 10.0n 10ps ; // binary search
resistance = 10MEG;
} ;

```

Με τη δήλωση **SIGNAL** θέτουμε τις στάθμες των σημάτων εισόδου ή εξόδου, για την παραγωγή των σημάτων εισόδου, ή την αρχικοποίηση των σταθμών των τάσεων εξόδου, για την μέτρηση της καθυστέρησης και της κατανάλωσης.

Οι επιμέρους παράμετροι που δηλώνονται εδώ είναι:

Πίνακας 8.1.4 Παράμετροι που καθορίζουν τη μορφή και τις στάθμες του σήματος εισόδου ή εξόδου.

Descriptor	Keyword	Type	Default
Voltage level unit	Unit	REL   ABS REL is a percentage of the voltage level unit; ABS is the fixed value.	REL
High-level voltage	Vh	Value	1.0
Low-level voltage	Vl	Value	0.0
Threshold-level Voltage	Vth	Value	0.5
High-slew voltage Level	Vsh	Value	0.8
Low-slew voltage Level	Vsl	Value	0.2
Maximum output slew, which is used for max_cap calculation	Tsmax	Value	3.0 ns
Input circuit for generating nonlinear input slew	Incir	"string"	""
Scaling factor for input slew	slew_derate	Value	0.0

Οι πιο πάνω παράμετροι μπορούν να πάρουν τιμές με τρεις τρόπους. Σύμφωνα με τον πρώτο τρόπο μπορούν να δοθούν τέσσερις τιμές για κάθε μία από τις παραπάνω παραμέτρους όπου η κάθε μία αφορά χωριστά κάθε ένα από τα κατώφλια δυναμικού για input-rise, input-fall, output-rise, και output-fall. Έτσι γράφοντας  $v_{th} = 0.1 \ 0.2 \ 0.15 \ 0.25$  ; σημαίνει

input rise  $V_{th} = 0.1 \text{ V}$   
input fall  $V_{th} = 0.2 \text{ V}$   
output rise  $V_{th} = 0.15 \text{ V}$   
output fall  $V_{th} = 0.25 \text{ V}$

Ο δεύτερος τρόπος επιτρέπει την απόδοση με ζεύγος τιμών όπως στο ακόλουθο παράδειγμα  $v_{th} = 0.1 \ 0.2$  ; όπου σημαίνει

input-rise Vth = input-fall Vth = 0.1 V  
 output-rise Vth = output-fall Vth = 0.2 V

Σύμφωνα με τον τρίτο τρόπο αποδίδεται μια τιμή η οποία καταχωρείται και στις τέσσερις παραμέτρους. Η πρόταση  $v_{th} = 0.1$  ; σημαίνει

input-rise Vth = input-fall Vth = 0.1 V  
 output-rise Vth = output-fall Vth = 0.1 V

Η δήλωση simulation καθορίζει τις μεταβλητές ελέγχου της εξομοίωσης όπως αναφέρονται στο σχετικό πίνακα.

**Πίνακας 8.1.5** Δηλώσεις που αφορούν το είδος της εξομοίωσης, την δυαδική αναζήτηση, την τιμή της αντίστασης Pulling και το όνομα του προκαθορισμένου κυκλώματος, που παράγει μη γραμμικό Slew στην είσοδο.

Descriptor	Keyword	Type	Default
Transient simulation	Transient	<i>value, value, value</i>	1.0 ns, 20 ns, 10 ps
Binary search	Bisec	<i>value, value, value</i>	3.0 ns, 3.0 ns, 100 Ps
Pulling resistance	Resistance	<i>value, value, value</i>	1000000
Default input circuit for generating nonlinear input slew	Incir	<i>"string"</i>	<i>""</i>

Στη δομή του setup αρχείου που έχει γραφτεί στην αρχή έχουμε

```
Simulation STD_CELLS_45NM_SIM {
transient = 1.0n 100n 10p ;
bisec = 10.0n 10.0n 10ps ; // binary search
resistance = 10MEG;
} ;
```

Που σημαίνει πως η εξομοίωση για το χαρακτηρισμό θα τρέξει για χρόνο από 1nsec έως 100nsec με βήμα 10 psec. Η δυαδική αναζήτηση η οποία χρησιμοποιείται για τον ακριβή υπολογισμό του χρόνου συγκράτησης και αποκατάστασης, θα ψάξει για το πολύ 10nsec με βήμα των 10psec. Η pullup και pulldown αντίσταση που χρησιμοποιείται στις tristate μετρήσεις, είναι 10 Megohm.

Το δεύτερο κομμάτι του αρχείου είναι το control section. Το μέρος αυτό του setup αρχείου αναλαμβάνει την ανάθεση των τιμών που δηλώθηκαν στο Define section, στις διάφορες διεργασίες. Η εντολή db\_prepare που είδαμε στην προηγούμενη

ενότητα του command file, διαβάζει το control section του αρχείου setup και εγκαθιστά όλες τις διεργασίες που αναφέρονται εκεί καθώς και τις παραμέτρους που ορίζονται στις δηλώσεις της βάσης δεδομένων. Τέτοιες δηλώσεις ελέγχου ακολουθούν στη συνέχεια.

```

set process (best,typical,worst) {
simulation = STD_CELLS_45NM_SIM ;
index = X1 ;
signal =STD_CELLS_45NM_GROUP ;
margin = m0 ;
nominal = n0 ;
} ;

set index (best,typical,worst) {
Group(X1) = X1 ;
} ;

set index (best,typical,worst) {
Group(X2) = X2 ;
} ;
set index (best,typical,worst) {
Group(X4) = X4 ;
} ;

set index (best,typical,worst) {
Group(X8) = X8 ;
} ;

set index (best,typical,worst) {
Group(CLKBUF1) = CLKBUF1 ;
} ;

set index (best,typical,worst) {
Group(CLKBUF2) = CLKBUF2;
} ;

set index (best,typical,worst) {
Group(CLKBUF3) = CLKBUF3 ;
} ;

set index (best,typical,worst) {
Group(DFFSR) = DFFSR ;
} ;

set index (best,typical,worst) {
Group(LATCH) = LATCH ;
} ;

set index (best,typical,worst) {
Group(DFF) = DFF ;
} ;

```

Η εντολή SET\_CELL αναθέτει τιμές σύμφωνα με τις δηλώσεις που περιέχει σε συγκεκριμένα κελιά. Οι δηλώσεις της εντολής αυτής υπερκαλύπτουν τις αναθέσεις

που έγιναν από τις εντολές SET\_PROCESS και SET\_GROUP αλλά όχι τις αναθέσεις που έγιναν από την εντολή SET\_PIN.

Με την εντολή SET\_DEFINES μπορούν να ανατεθούν νέες δηλώσεις σε pins, cells ή groups και υπερκαλύπτονται οι τιμές που έχουν ανατεθεί από την εντολή SET\_PROCESS. Η μορφή της είναι ίδια με αυτήν των εντολών SET\_SIGNAL, SET\_SIMULATION, SET\_INDEX, SET\_MARGIN, and SET\_NOMINAL. Προσοχή εδώ πρέπει να δοθεί όταν αναθέτονται τιμές με τις εντολές SET\_PINS, SET\_GROUPS και SET\_CELL. Όταν χρησιμοποιείται μια από τις προηγούμενες, δεν πρέπει να καλείται η SET\_DEFINES γιατί μπορεί να προκύψει σύγκρουση (conflict).

Παρόμοια εντολή είναι η SET\_PIN η οποία αναθέτει δηλώσεις σε ένα pin και υπερκαλύπτει τις εντολές SET\_PROCESS, SET\_CELL, και SET\_GROUP.

Τέλος με την εντολή SET\_PROCESS γίνεται ορισμός των παραμέτρων για τον χαρακτηρισμό της βιβλιοθήκης.

Σε κάθε μία απ' τις τεχνολογίες που θα εφαρμοσθεί ο χαρακτηρισμός, θα πρέπει να τροποποιηθεί κατάλληλα το setup αρχείο. Η τροποποίηση αυτή αφορά τις τάσεις κατωφλίου και λειτουργίας τη χωρητικότητα εξόδου και τα slew rates των εισόδων.

## 8.2. Configuration File (elccfg)

Το αρχείο αυτό περιέχει μεταβλητές του περιβάλλοντος ή setup directives που εκτελούνται κατά τη διάρκεια του χαρακτηρισμού. Για τον ορισμό των μεταβλητών περιβάλλοντος δημιουργούμε προτάσεις της μορφής *variable\_name=value*;

Ενώ για την δημιουργία directives δημιουργούμε προτάσεις της μορφής *DIRECTIVE\_NAME="value"*; . Οι είσοδοι που απαιτεί η διαδικασία χαρακτηρισμού δηλώνονται υπό μορφή directives.

Το ακόλουθο αρχείο elccfg είναι αυτό που θα χρησιμοποιηθεί ως βάση για τον χαρακτηρισμό των βιβλιοθηκών για τις τεχνολογίες 45nm, 32nm, 22nm, 16nm.

```
#Specify the environment variable settings.
EC_SIM_USE_LSF=1;
EC_SIM_LSF_CMD=" ";
EC_SIM_LSF_PARALLEL=1;
EC_SIM_TYPE="hspice";
EC_SIM_NAME="hspice";
EC_SPICE_SIMPLIFY=1;
EC_CHAR=\u201cECSM-TIMING ECSM-POWER\u201d;
```

```
#Specify the characterization inputs.
SUBCKT="std_cells_45nm.sp";
MODEL="berkeley_45nm.m";
DESIGNS="INVX1 AND2X1 AND2X2 AOI21X1 AOI22X1 BUFX2 BUFX4 CLKBUF1
CLKBUF2 CLKBUF3 DFFNEGX1 DFFPOSX1 DFFSR FAX1 HAX1 INVX1 INVX2 INVX4
INVX8 LATCH MUX2X1 NAND2X1 NAND3X1 NOR2X1 NOR3X1 OAI21X1 OAI22X1
OR2X1 OR2X2 TBUF1X1 TBUF2X1 XNOR2X1 XOR2X1";
SETUP="setup_45nm";
PROCESS=typical;
XDESIGNS="";
CORNER="tt";
```

### 8.3. Command file (cmd\_file)

Το αρχείο αυτό έχει καταγεγραμμένη την σειρά εκτέλεσης των εντολών του characterizer. Οι εντολές που χρειάστηκε να εκτελεστούν είναι οι επόμενες:

```
db_open std_cells_45nm
db_prepare -f
db_spice -s hspice -p typical -keep_log
db_output -p typical -alf std_cells_45nm.alf
alf2lib -lib std_cells_45nm.lib -alf std_cells_45nm.alf -def
property_file
alf2veri -alf std_cells_45nm.alf -verilog std_cells_45nm.v
db_close
exit
```

Αρχικά λοιπόν ανοίγει ένα αρχείο βάσης δεδομένων με όνομα `std_cells_45nm`. Η επόμενη εντολή προετοιμάζει τις συνθήκες χαρακτηρισμού. Κατά τη προετοιμασία φορτώνεται το μοντέλο `spice`, εκτελείται η αναγνώριση του κυκλώματος, και δημιουργούνται τα διανύσματα εξομοίωσης. Η εντολή `db_spice -s hspice -p typical -keep_log` εκτελεί την εξομοίωση και το αποτέλεσμα του χαρακτηρισμού καταγράφεται στο αρχείο `std_cells_45nm.alf` με την εκτέλεση της εντολής `db_output -p typical -alf std_cells_45nm.alf`. Στη συνέχεια με την εντολή `alf2lib -lib std_cells_45nm.lib -alf std_cells_45nm.alf -def property_file`

παράγουμε το αρχείο LIB το οποίο εδώ θα περιλαμβάνει και πληροφορία για το area (αρχείο `property_file`).

Με την εντολή `alf2veri -alf std_cells_45nm.alf -verilog std_cells_45nm.v` παράγουμε το αρχείο Verilog.

Μετά την ολοκλήρωση των πιο πάνω πρέπει να κλείσει η βάση δεδομένων, γι αυτό καλείται η εντολή `db_close`.



## 8.4. Property file (property\_file)

Το αρχείο αυτό περιέχει πληροφορία για το είδος των I/O κάθε cell, και τον χώρο που το κάθε cell καταλαμβάνει. Κατά αυτόν τον τρόπο γίνεται μια εκτίμηση από το design compiler του χώρου που απαιτεί η σύνθεση ενός design. Ένα μέρος του property file που χρησιμοποιούμε στην τεχνολογία των 45nm δίνεται, για να δούμε τη σχετική δομή του.

```
cell AND2X1 {
  pin A {
    direction input;
  };
  pin B {
    direction input;
  };
  pin Y {
    direction output;
  };
  footprint AND2;
  area 20.0;
};
cell INVX1 {
  pin A {
    direction input;
  };
  pin Y {
    direction output;
  };
  footprint INV;
  area 10.0;
};
```

Το συγκεκριμένο μέρος του αρχείου, περιγράφει δύο cells. Μια πύλη AND2X1 και έναν αναστροφέα. Βλέπουμε λοιπόν πως σε κάθε cell προσδιορίζεται η κατεύθυνση του I/O pin και ορίζεται κάποια επιφάνεια (area) που καταλαμβάνει το συγκεκριμένο cell.

## 8.5. Εκκίνηση της εφαρμογής Encounter Library Characterizer

Ο characterizer ξεκινά από το command prompt του λειτουργικού Unix δίνοντας την εντολή:

```
elc
[-version]
[-cdb makecdb_cmd_file]
```

```

[-dpm makedpm_cmd_file]
[-slc slc_cmd_file]
[-r rechar_cmd_file]
[-S cmd_file]
[-Q]
[-L log_file]
[-lic { elcgx1 | elcx1 | etsgx1 | etsx1 | ets1 }]
[-C log_file]
[-al].

```

Η σύνταξη της εντολής δέχεται προαιρετικά ένα από τα ορίσματα που καταγράφονται ανάλογα με το είδος του χαρακτηρισμού που θέλουμε να κάνουμε και τις αναφορές σχετικά με τη διαδικασία χαρακτηρισμού που θέλουμε να καταγράψουμε. Συνήθως αρκεί, ο καθορισμός του *cmd\_file* για να γίνει χαρακτηρισμός για την κατανάλωση και τον χρονισμό. Κατά την εκκίνηση του characterizer γίνεται αρχικοποίηση του περιβάλλοντος φλοιού (>elc) επεξεργάζοντας τις μεταβλητές του ή διαβάζοντάς τις από το αρχείο configuration file (elccfg). Στο command prompt του elc μπορούν να εκτελεστούν πέραν των εντολών του characterizer, όλες οι εντολές του λειτουργικού unix. Με την εκκίνηση της εφαρμογής του characterizer διαβάζονται οι εντολές που περιγράφονται στο αρχείο *cmd\_file* και εκτελούνται ενώ τα αποτελέσματα καταγράφονται σε log file με default όνομα *elc.log*.

## 8.6. Μεταγλώττιση του αρχείου LIB σε .db

Για τη μεταγλώττιση αρκεί να γραφούν οι απαραίτητες εντολές του library compiler με τη σειρά κλήσης τους.

```

lc_shell
read_lib std_cells_45nm.lib
write_lib      std_cells_45nm      -format      db      -output
std_cells_45nm.db
exit

```

```

dc_shell-xg-t
read_db std_cells_45nm.db
report_lib std_cells_45nm
exit

```

Για το σκοπό αυτό θα πρέπει να δημιουργήσουμε ένα Tcl script το οποίο θα αυτοματοποιήσει το βήμα αυτό.

Για περισσότερες λεπτομέρειες σχετικά με το χειρισμό του εργαλείου Encounter Library Characterizer, ανατρέξτε στο σχετικό user guide απ' όπου έχουν προέλθει οι τα περιεχόμενα του παρόντος παραρτήματος.

## 8.7. Σύνθεση με τη βοήθεια του Design Compiler

Για τη σύνθεση των designs, πρέπει να εκτελεστούν κάποιες εντολές στον design compiler. Προκειμένου να γίνει ευκολότερα αυτή η διαδικασία, γράφτηκαν αντίστοιχα scripts. Ένα τέτοιο script είναι αυτό που πραγματοποιεί τη σύνθεση του αθροιστή.

```
dc_shell-xg-t
read_vhdl add.vhd
current_design add
elaborate add
link
compile
report_area, report_timing, report_power
write -h -format verilog -output add.sv
write -h -format ddc -output add.ddc
exit
```

Για τη σύνθεση ακριβώς του ίδιου κυκλώματος, με διαφορετικής τεχνολογίας όμως cells, πρέπει να κρατήσουμε την ίδια σύνθεση με την προηγούμενη. Το script που ακολουθεί, κάνει αυτό ακριβώς.

```
read_verilog mult1616_new.sv
current_design mult1616
set auto_wire_load_selection true
create_clock "clk" -name clk -period 2.0
Link
```

Οπότε μπορούμε στο τέλος να προσαρτήσουμε τις κατάλληλες εντολές για timing και power analysis.

```
report_area, report_timing, report_power
write -h -format verilog -output add.sv
write -h -format ddc -output add.ddc
```



## Αναφορές-βιβλιογραφία

1. **MEYER, GRAY HURST LEWIS.** *ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΑΝΑΛΟΓΙΚΩΝ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ.* 2007.
2. **SMITH, SEDRA.** *ΜΙΚΡΟΗΛΕΚΤΡΟΝΙΚΑ ΚΥΚΛΩΜΑΤΑ.* 1994.
3. **WESTE, N.H.** *ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ.* 1996.
4. **Lang Lin, Wayne Burleson.** *Leakage-Based Differential Power Analysis (LDPA).* Massachusetts : Department of Electrical and Computer Engineering, University of Massachusetts at Amherst.
5. **gaisler.** [Ηλεκτρονικό] [www.gaisler.com](http://www.gaisler.com).
6. **Cadence.** *ELC User Guide.* September 2008.
7. **Sohnius, Richard.** *Prof. Dr. U.Bruning . s.l. : University of Mannheim. Standard Cell Characterization.*
8. **Predictive Technology Models. Berkeley University.** [Ηλεκτρονικό] <http://ptm.asu.edu/>.
9. *Design Challenges of Technology Scaling . Borkar, Shekhar.*
10. **Cadence.** *LEF/DEF Language Reference Product Version 5.6.* [Ηλεκτρονικό] September 2004.
11. **University, Digital ASIC Group Lund.** *Digital ASIC Design Flow.* [Ηλεκτρονικό] October 2005.
12. **Wichern, Don.** *Characterizing Cells and Writing a Technology Library File.* February 2005.
13. **Synopsys.** *Liberty User Guide Voll.*
14. **Borkar, Shekhar.** *DESIGN CHALLENGES OF TECHNOLOGY SCALING.* s.l. : Intel.