

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ,
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

**Μεθοδολογία Ανάλυσης Δικτύου Τροφοδοσίας
Ολοκληρωμένων Κυκλωμάτων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Απόστολος Γιαννέτσος

Βόλος, Σεπτέμβριος 2006



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 5050/1

Ημερ. Εισ.: 19-09-2007

Δωρεά: Συγγραφέα

Ταξιθετικός Κωδικός: ΠΤ – ΜΗΥΤΔ

2006

ΓΙΑ

Αφιερώνω την εργασία αυτή
στους γονείς μου

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων και βασικό επιβλέποντα της πτυχιακής αυτής εργασίας κ. Γ. Σταμούλη που μου έδωσε την ευκαιρία να πραγματοποιήσω αυτή την μελέτη. Η υποστήριξή του, η αμέριστη συμπαράστασή του, αλλά και οι διαρκείς και εύστοχες υποδείξεις του βοήθησαν στην έγκαιρη ολοκλήρωση αυτής της μελέτης.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω τον έτερο επιβλέποντα καθηγητή και επισκέπτη καθηγητή στο τμήμα, κ. Ν. Ευμορφόπουλο για τις συμβουλές του στο μαθηματικό υπόβαθρο που χρησιμοποιήθηκε στις μετρήσεις που παρουσιάζονται και τον διδακτορικό φοιτητή του τμήματος Δ. Καραμπατζάκη για την προγενέστερη δουλειά του στα εργαλεία που χρησιμοποιήθηκαν αλλά και για τη γνώση πάνω στα θέματα με τα οποία καταπιάνεται αυτή η διπλωματική εργασία, γνώσεις που απλόχερα μεταλαμπάδευσε. Δεν θα μπορούσα να ξεχάσω τον προπτυχιακό φοιτητή του τμήματος Μ. Τσιαμπά για την βοήθεια που μου προσέφερε.

Τέλος, θα ήθελα να ευχαριστήσω τον αδερφό μου Σάκη για την υποστήριξη που μου επέδειξε. Επίσης ευχαριστώ όλους τους φίλους μου οι οποίοι μου συμπαραστάθηκαν αμέριστα καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής αυτής εργασίας.

Π εριεχόμενα

| | |
|------------------------|---|
| Π ρόλογος | 7 |
|------------------------|---|

| | |
|--|----|
| Ε ισαγωγή | 8 |
| Προσεγγίσεις | 8 |
| Επεξήγηση Προβλημάτων | 9 |
| Τι Σημαίνει Πτώση Τάσης (IR Drop) | 9 |
| Το IR drop επηρεάζει την απόδοση κελιού (cell performance) και του χαρακτηρισμού (characterization) | 9 |
| Αιτίες που προκαλούν IR drop | 10 |
| Υπολογίζοντας τους στόχους για το IR drop | 11 |
| Το IR Drop (Πτώση Τάσης) στο δίκτυο διανομής ισχύος | 14 |
| Συμπτώματα του IR Drop | 17 |
| Ηλεκτρομετανάστευση (Electromigration - EM) | 19 |
| Διορθώνοντας το IR Drop και το Electromigration | 19 |

| | |
|---------------------------------------|----|
| Κ εφάλαιο 1 | 20 |
| Τα Εργαλεία Που Χρησιμοποιήσαμε | 20 |
| BuildGates Synthesis | 20 |
| Virtuoso IC | 24 |
| SKILL | 24 |
| Ultrasim | 25 |
| Spectre Simulator | 26 |
| Ocean | 26 |
| SoC Encounter | 29 |
| IExtreme Engine | 32 |
| DIVA Verification | 34 |

| | |
|--|----|
| Κ εφάλαιο 2 | 37 |
| Μία RTL-to-Grid Σχεδιαστική Ροή Για Έλεγχο Δικτύου Διανομής Ισχύος Βασισμένη σε μία Στατιστική Μηχανή | 37 |
| Περιγραφή του Προβλήματος | 47 |
| Σχεδιαστική Ροή RTL-to-Grid | 48 |
| Πειραματικά Αποτελέσματα | 50 |

| | |
|----------------------------------|----|
| Π αράρτημα A | 52 |
| Αρχεία DEF και PDEF | 52 |
| Π αράρτημα B | 55 |
| read_taps.il..... | 55 |
| sim_proc.il..... | 56 |
| dc_proc.ocn..... | 57 |
| veriloga.va..... | 58 |
| B ιβλιογραφία..... | 60 |
| E-books..... | 60 |
| Δικτυακοί Τόποι..... | 60 |
| Papers..... | 61 |

Πρόλογος

Το 1975 ο Gordon Moore διατύπωσε ότι ανά 2 χρόνια ο αριθμός των τρανζίστορ στα ολοκληρωμένα κυκλώματα θα διπλασιάζεται. Ο λεγόμενος και ως «νόμος του Moore» επαληθεύεται ακόμα και στις μέρες μας. Τα μεγέθη των τρανζίστορ μειώνονται, επιτρέποντας σχεδιασμούς μεγαλύτερης πυκνότητας. Νέα, πιο ισχυρά τρανζίστορ απαιτούν χαμηλής κατανάλωσης πηγές τάσης προς αποφυγή αποτυχιών της συσκευής. Μία χαμηλότερη πηγή τροφοδοσίας σημαίνει λιγότερα επίπεδα θορύβου στο δίκτυο διανομής ισχύος. Από την άλλη μεριά, η ικανότητα για σχεδιασμό ολοένα και αυξανόμενων όσον αφορά την πολυπλοκότητα ολοκληρωμένων οδηγεί σε αυξήσεις στο συνολικό μέγεθος και στις απώλειες ισχύος. Για να σχεδιαστούν μεγαλύτερα ολοκληρωμένα, περισσότερα επίπεδα μετάλλου χρησιμοποιούνται με σκοπό την επίτευξη μεγαλύτερων δρομολογήσεων σημάτων και ισχύος. Τα στενότερα καλώδια έχουν μεγαλύτερη αντίσταση από προηγούμενες τεχνολογίες. Αυτά τα καλώδια με την μεγαλύτερη αντίσταση και τα μεγαλύτερα συνολικά ρεύματα ισχύος (power currents) φυσικά οδηγούν σε αυξήσεις στην πτώση τάσης (IR drop) ή στο θόρυβο του δικτύου διανομής ισχύος. Η συγκρουόμενη σχεδιαστική τάση προς μικρότερα όρια θορύβου σημαίνει ότι πρέπει να αναζητήσουμε μία ισορροπία μεταξύ του εγγενή (inherent) θορύβου στο δίκτυο διανομής ισχύος και των ορίων του θορύβου της πηγής τροφοδοσίας για να επιτύχουμε μία επιτυχημένη σχεδίαση. Όσο οι σχεδιάσεις αυξάνουν σε συχνότητα ή μέγεθος, η περιοχή αυτή για ισορροπία μειώνεται.

Εισαγωγή

Κατά τη μετάβαση σχεδίασης ολοκληρωμένων κυκλωμάτων από τα 0.35μm στα 0.25μm παρουσιάστηκαν προβλήματα όπως αυτό της πτώσης τάσης (IR Drop) και της ηλεκτρομετανάστευσης (electromigration). Προβλήματα όπως αυτά παρουσιάστηκαν λόγω της αυξημένης σχεδιαστικής περιπλοκότητας των chip. Για αυτό πλέον είναι αναγκαία η χρήση εργαλείων ανάλυσης του δικτύου διανομής ισχύος (power grid), ειδικά σε μικροεπεξεργαστές υψηλής ταχύτητας.

Πλέον κάνουμε χρήση εργαλείων για να βρούμε «αδύναμα» σημεία της υλοποίησης του power grid κατά τα στάδια σχεδίασης των blocks αλλά και του full-chip. Τα «αδύναμα» σημεία είναι χαρακτηριστικά της σχεδίασης τα οποία καταλήγουν σε εκτεταμένο IR Drop, electromigration stress ή pin currents κατά την λειτουργία του chip. Τα εργαλεία μας βοηθούν όχι μόνο να εντοπίζουμε «αδύναμα» σημεία αλλά και να κατανοούμε ποιες αποφάσεις υλοποίησης δημιούργησαν το «αδύναμο» σημείο.

Προσεγγίσεις

Η εύρεση «αδύναμων» σημείων μπορεί να χωριστεί σε τρεις προσεγγίσεις :

- Εύρεση αδυναμιών στο power grid οι οποίες είναι πιθανό να έχουν επίπτωση στην ορθή λειτουργία του ολοκληρωμένου, άσχετα από το μέγεθος της επίπτωσης. Αυτή η προσέγγιση είναι και η πιο συνηθισμένη και συναντάται στις στατικές αναλύσεις. Συνίσταται να κάνουμε πρώτα στατική ανάλυση πριν από τη δυναμική, γιατί η στατική ανάλυση μπορεί να βρει προβλήματα γρήγορα. Επιπλέον, η στατική ανάλυση μπορεί να βρει προβλήματα που ακόμα και μία εκτεταμένη δυναμική ανάλυση θα μπορούσε να παραβλέψει λόγω ανεπαρκούς πληρότητας διανυσμάτων ελέγχου (vector coverage).
- Εύρεση αδύναμων σημείων προβλέποντας ένα διάνυσμα ελέγχου χειρίστης περίπτωσης IR drop

- Εύρεση αδύναμων σημείων υπολογίζοντας την ακριβή πτώση τάσης στο πλέγμα για ένα συγκεκριμένο διάλυμα ελέγχου. Αυτή η προσέγγιση είναι συνήθης στη σχεδίαση μνήμης ή όταν το κόστος αλλαγής του σχεδιασμού είναι μεγάλο, οπότε και θέλουμε καθορίσουμε την ακριβή έκταση του IR drop.

Επεξήγηση Προβλημάτων

Τι Σημαίνει Πτώση Τάσης (IR Drop)

Βασιζόμενοι σε θεμελιώδεις πληροφορίες όσον αφορά το φυσικό περιβάλλον ενός ολοκληρωμένου, κατασκευαστικές ανοχές και την βασική εφαρμογή του νόμου του Ohm, ξεκάθαροι υπολογισμοί μπορούν να δώσουν κατάλληλους στόχους για το IR drop στο δίκτυο διανομής ισχύος ενός ολοκληρωμένου. Χωρίς αυτές τις υπολογισμένες επιδιωκόμενες τιμές, οι μηχανικοί σχεδιασμού τυπικά θα στηρίζονταν σε μεθόδους οι οποίες δεν ισχύουν στις μικρότερες γεωμετρίες επεξεργασίας του παρόντος. Δυστυχώς, οι επιπτώσεις του IR drop στον χρονισμό μπορούν να μην φανούν μέχρι και την κατασκευή του πρωτότυπου ολοκληρωμένου, το οποίο βέβαια είναι πολύ αργά.

Το IR drop μπορεί να προκαλέσει πάρα πολύ σημαντικά προβλήματα στον χρονισμό. Υπάρχουν μεγάλης έκτασης πληροφορίες διαθέσιμες για το πως να μειώσουμε αυτή την πτώση με τεχνικές σωστού σχεδιασμού, όπως επίσης και πληροφορίες για εντοπισμό προβλημάτων με εργαλεία ανάλυσης στατικού χρονισμού. Ωστόσο, υπάρχει ανεπάρκεια πληροφοριών στο πως να θέσουμε τον κατάλληλο στόχο για το IR drop εξαρχής.

Το IR drop επηρεάζει την απόδοση κελιού (cell performance) και του χαρακτηρισμού (characterization)

Όπως υποδηλώνει ο όρος, το IR drop είναι μία πτώση τάσης η οποία προκύπτει από το ρεύμα και την αντίσταση που σχετίζονται με το δίκτυο διανομής ισχύος. Αυτή η πτώση τάσης οδηγεί σε δύο τύπους προβλημάτων.

Πρώτον, μία μειωμένη διαφορά τάσης μεταξύ των VDD και VSS pins ενός τυποποιημένου κελιού θα μειώσει την λειτουργική του απόδοση. Αν το κελί βρίσκεται σε κρίσιμο μονοπάτι, αυτή η μείωση θα μπορούσε να μειώσει την συχνότητα λειτουργίας του ολοκληρωμένου. Το IR drop επίσης μειώνει την ανοχή του κελιού στον θόρυβο και σε σπάνιες περιπτώσεις θα μπορούσε να οδηγήσει σε λειτουργικές αποτυχίες. Οι μικρότερες γεωμετρίες διεργασίας έχουν οδηγήσει σε δραματική αύξηση τέτοιων προβλημάτων.

Εξαιτίας της σοβαρότητας τέτοιων θεμάτων, χρειαζόμαστε ΑΣΧ (Ανάλυση Στατικού Χρονισμού). Τα εργαλεία ΑΣΧ παρόλο που έχουν βελτιωθεί σημαντικά σε λειτουργικότητα, ακόμα υποθέτουν ότι όλα τα τυποποιημένα κελιά (standard cells) έχουν την ίδια μέγιστη (VDD συν δέκα τοις εκατό μείον 0V), ονομαστικές (VDD μείον 0V) και ελάχιστες τάσεις (VDD μείον δέκα τοις εκατό μείον 0V). Όπως βλέπουμε από τις παραπάνω τιμές, το VDD μπορεί να έχει μία διακύμανση της τάξης του 10% αλλά το VSS μένει στα 0V.

Ο δεύτερος τύπος προβλημάτων που προκαλεί το IR Drop αφορά τον χαρακτηρισμό των βιβλιοθηκών. Τα standard cells σε μία βιβλιοθήκη χαρακτηρίζονται για να δίνουν ακριβείς προβλέψεις της απόδοσής τους σε πραγματικές συνθήκες μέσα σε ένα καθορισμένο εύρος τιμών τάσης, θερμοκρασίας και συνθήκες επεξεργασίας πυριτίου. Όσο οι γεωμετρίες επεξεργασίας συρρικνώνονται, οι εξισώσεις που απαιτούνται για να διατηρούν την προβλεπόμενη απόδοση να συμβαδίζει με το μετρημένο πυρίτιο (measured silicon) έχουν γίνει πιο περίπλοκες, και σημαντικές παρεκκλίσεις συμβαίνουν όταν ένα standard cell λειτουργεί έξω από τα χαρακτηρισμένο εύρος τιμών του.

Αιτίες που προκαλούν IR drop

Όπως αναφέρθηκε νωρίτερα, το IR drop προκαλείται κυρίως από την αντίσταση του δικτύου κατανομής ισχύος ενός ολοκληρωμένου. Υπάρχουν και

άλλες ενδεχόμενες αιτίες για απώλεια τάσης στα standard cells ενός ολοκληρωμένου, ωστόσο και αυτές οι αιτίες αφορούν ένα σύστημα στο οποίο το ολοκληρωμένο βρίσκεται σε λειτουργία.

Συνήθως, ένα ολοκληρωμένο αλληλεπιδρά με άλλες συσκευές σε μία ΠΤΚ (Πλακέτα Τυπωμένου Κυκλώματος). Η τροφοδοσία ρεύματος του συστήματος είναι επίσης πάνω στην ΠΤΚ (ή κάπου κοντά τουλάχιστον) και παρέχει ισχύ μέσω ενός ρυθμιστή τάσης. Ένας τυπικός ρυθμιστής τάσης παρέχει τάση η οποία είναι σταθερή γύρω από το 1.5 με 2 τοις εκατό της επιδιωκόμενης εξόδου (target output). Οι σύγχρονες ΠΤΚ μπορεί να παρουσιάσουν πτώση τάσης μέχρι και 18mV από τον ρυθμιστή τάσης ως τις πιο απομακρυσμένες συσκευές πάνω στην ΠΤΚ. Συνήθως όμως η τιμή αυτή κυμαίνεται ανάμεσα 11 και 14 mV.

Συνεπώς, το συνολικό IR drop περιλαμβάνει τις πτώσεις τάσης και στην πλακέτα και στο ολοκληρωμένο :

$$IR_{total} = IR_{pcb} + IR_{chip} \quad (1)$$

όπου το IR_{pcb} αποτελείται από IR drop που σχετίζεται με την τροφοδοσία του ολοκληρωμένου με ισχύ, και το IR_{chip} αποτελείται από το IR drop στο ίδιο το ολοκληρωμένο.

Υπολογίζοντας τους στόχους για το IR drop

Για να επιδείξουμε μία πρακτική μέθοδο καθορισμού των στόχων για το IR drop, ο επόμενος υπολογισμός χρησιμοποιεί το παράδειγμα ενός τυπικού ολοκληρωμένου στα 130nm με 1.2V τάση τροφοδοσίας. Οι αρχικές παραδοχές είναι έτσι οι εξής :

- Ένα standard cell χαρακτηρισμένο στα 1.2V με απόκλιση +/- 10 τοις εκατό
- Μία πηγή τροφοδοσίας στην ΠΤΚ στα 1.2V με απόκλιση +/- 2 τοις εκατό
- IR drop των 18mV από την πηγή τροφοδοσίας στο ολοκληρωμένο

Βασισμένοι στα δεδομένα χαρακτηρισμού ενός standard cell, το δίκτυο διανομής ισχύος του ολοκληρωμένου πρέπει τουλάχιστον να διατηρήσει 1.08V (1.2V μείον 10 τοις εκατό) σε όλα τα standard cells. Μπορεί επιπλέον να προκύψει από τις αρχικές παραδοχές ότι η τάση στην χειρότερη περίπτωση που θα δώσει η πηγή τροφοδοσίας είναι 1.176V (1.2V μείον 2 τοις εκατό). Με διδόμενες αυτές τις τιμές, ένας στόχος για το IR drop υπολογίζεται εύκολα αφαιρώντας την απαιτούμενη τάση από αυτήν που παρέχεται :

$$IR_{target} = V_{supply} - V_{require}$$

$$= 1.176V - 1.08V$$

$$= 0.096V (96 mV)$$

Θέτοντας το IR_{target} ίσο με το IR_{total} στην εξίσωση (1) και χρησιμοποιώντας το IR drop των 18mV που υποθέσαμε στην ΠΤΚ, παίρνουμε ένα νέο στόχο για το IR drop στο ολοκληρωμένο :

$$IR_{chip} = IR_{target} - IR_{pcb} (2)$$

$$= 96 mV - 18.5 mV$$

$$= 77.5 mV$$

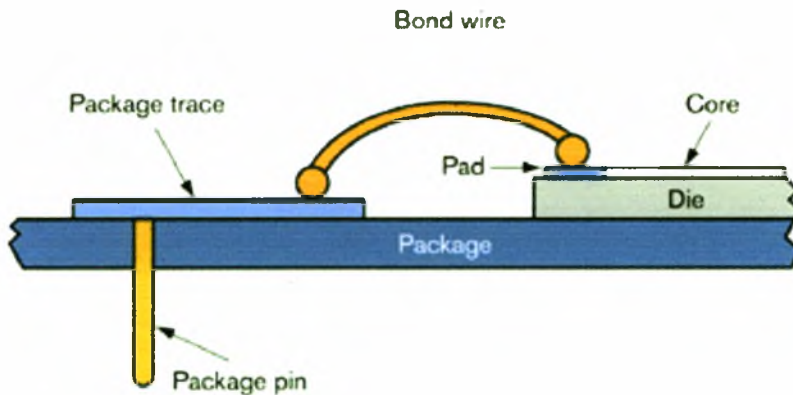
Υποθέτοντας ένα ισόποσο μοίρασμα μεταξύ της πτώσης στο VDD και της ανόδου στο VSS, ένας αρχικός στόχος για το σχεδιασμό του δικτύου διανομής ισχύος μπορεί να είναι τα 38mV σε κάθε ένα από τα VDD και VSS δίκτυα από τα pins του ολοκληρωμένου στα standard cells. Ωστόσο, ο στόχος εξακολουθεί να παρουσιάζει μία κάπως ασαφή και πιθανώς παραπλανητική προϋπόθεση στον φυσικό σχεδιαστή. Ας πάρουμε υπόψη μία ακόμα βελτίωση του στόχου :

$$IR_{chip} = IR_{package} + IR_{die} (3)$$

Όπου το $IR_{package}$ αποτελείται από το IR drop μέσα στο πακέτο, και το IR_{die} αποτελείται από το IR drop στο die. Το δεύτερο συστατικό μπορεί να σπάσει περαιτέρω ως εξής :

$$IR_{die} = IR_{pads} + IR_{core} (4)$$

όπου το IR_{pads} αποτελείται το IR drop κατά πλάτος των bond pads και του I/O cell, και το IR_{core} αποτελείται από το IR drop από το I/O cell στο μακρινότερο standard cell. Πρέπει να σημειώσουμε ότι η τοποθεσία του «μακρινότερου standard cell» εξαρτάται και από τις επιλογές μας στο floorplan αλλά και από την λειτουργία του κυκλώματος. Το παρακάτω σχήμα δείχνει τα ξεχωριστά μέρη του IR_{chip} να οδηγεί σε IR_{core} για ένα wire-bond ολοκληρωμένο.



Σχήμα E-1 : Πτώση Τάσης Κατά Μήκος Του Μονοπατιού Εισόδου στο Δίκτυο Ισχύος ενός SoC

Συνδυάζοντας τις σχέσεις (3), (4), και την τιμή για το IR_{chip} που βρήκαμε από την σχέση (2) μας παρέχει μία φόρμουλα για να καθορίσουμε ένα ενακτέο (actionable) IR drop για την διεργασία του φυσικού σχεδιασμού :

$$IR_{core} = 77.5 \text{ mV} - IR_{package} - IR_{pads} \quad (5)$$

Για να καθορίσουμε τις τιμές για τα $IR_{package}$ και IR_{pads} πρέπει να λάβουμε υπ' όψιν ότι το $IR_{package}$ αποτελείται από δύο μέρη :

$IR_{package} = IR_{trace} + IR_{bondwire}$ όπου το IR_{trace} είναι το IR drop κατά πλάτος των traces μέσα στο package και το $IR_{bondwire}$ είναι το IR drop κατά πλάτος του bond wire. Οι τιμές για αυτές τις παραμέτρους διαφέρουν σημαντικά ανάλογα με τον τύπο του package, τον αριθμό των package pins αποκλειστικά για το VDD και το VSS, και την συνολική σκέδαση ισχύος στην συσκευή.

Το IR pads μπορεί να υπολογιστεί βασιζόμενοι στην αντίσταση των καλωδίων στο I/O cell. Πολλαπλασιάζουμε την αντίσταση με το ρεύμα που αναμένεται

να περάσει από το cell. Υπάρχει η επιλογή δύο εναλλακτικών λύσεων για αυτό το ρεύμα :

- Απόλυτα χειρίστη περίπτωση βασιζόμενοι στον σχεδιασμό του κελιού και στα όρια ρευμάτων που μπορεί να περάσουν, τα οποία εξαρτώνται από την τεχνολογία.
- Αναμενόμενη χειρίστη περίπτωση βασιζόμενοι στην προβλεπόμενη σκέδαση ρεύματος του ολοκληρωμένου και τον αριθμό των VDD (ή VSS) κελιών.

Η επιλογή εξαρτάται από τα επιθυμητά όρια και την ακρίβεια της προβλεπόμενης σκέδασης ρεύματος. Εκ νέου, οι ακόλουθοι υπολογισμοί υποθέτουν 10mV για το IR_{pads} .

Αντικαθιστώντας τις υπολογισμένες και τις τιμές που υποθέσαμε στην σχέση (5) παίρνουμε :

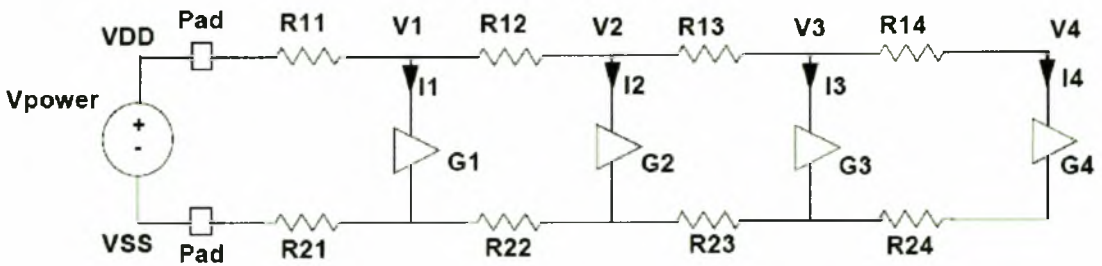
$$\begin{aligned} IR_{core} &= 77.5 \text{ mV} - 10 \text{ mV} - 10 \text{ mV} \\ &= 57.5 \text{ mV} \end{aligned}$$

Τελικά, μοιράζοντας ισόποσα αυτήν την τιμή μεταξύ της επιτρεπτής VDD πτώσης και VSS ανόδου, παίρνουμε ένα στόχο για το IR drop των 28.75mV για κάθε ένα δίκτυο από τα I/O cell στο πιο απομακρυσμένο standard cell.

Το IR Drop (Πτώση Τάσης) στο δίκτυο διανομής ισχύος

Το IR Drop είναι μία μείωση στην τάση που προκύπτει και στα δίκτυα τροφοδοσίας και στα δίκτυα γείωσης στα ολοκληρωμένα κυκλώματα. Ο σχεδιασμός ολοκληρωμένων κυκλωμάτων υποθέτει την ύπαρξη μίας ιδανικής πηγής τροφοδοσίας η οποία μπορεί συνεχώς να προσφέρει οποιαδήποτε ποσότητα ρεύματος ώστε να διατηρήσει την απαιτούμενη τάση στο ολοκληρωμένο. Στην πραγματικότητα ωστόσο, οι όλο και πιο στενές γραμμές μετάλλου προκαλούν αύξηση της αντίστασης και επομένως αύξηση και της πτώσης τάσης στο ολοκληρωμένο. Η ποσότητα της πτώσης τάσης εξαρτάται

από το effective resistance από τα pads στις πύλες. Το σχήμα E-2 απεικονίζει την έννοια του IR Drop



Σχήμα E-2 : Τυπική Δομή Δικτύου Διανομής Ισχύος

Το παραπάνω σχήμα δείχνει μία πηγή τροφοδοσίας συνδεδεμένη με τα pads του ολοκληρωμένου. Το δίκτυο διανομής ισχύος απεικονίζεται από τις αντιστάσεις R11-R14 για το VDD και τις αντιστάσεις R21-R24 για το VSS. Οι G1-G4 είναι οι συνδέσεις μεταξύ των λογικών πυλών στο δίκτυο διανομής ισχύος. Χαρακτηριστικά, όταν πραγματοποιείται προσομοίωση σε επίπεδο transistor, αυτές οι τάσεις (V1 – V4) υποτίθεται ότι είναι ίσες. Με άλλα λόγια, όλες οι αντιστάσεις R11-R14 και R21-R24 θα ήταν 0.0 ohms και έτσι όλες οι πύλες G1-G4 θα είχαν ιδανικές τάσεις τροφοδοσίας, VDD και VSS. Στην πραγματικότητα, οι αντιστάσεις στο δίκτυο διανομής ισχύος ενός ολοκληρωμένου είναι μη μηδενικές. Για παράδειγμα, η πύλη G4 ποτέ δεν έχει ιδανική τάση VDD στο pin τροφοδοσίας του όταν είναι ενεργή. Στην πραγματικότητα έχει μικρότερη τάση. Το ρεύμα που περνάει από την τροφοδοσία στην πύλη G4 πρέπει να περάσει από το δίκτυο διανομής ισχύος. Ένα ρεύμα, I , που περνάει από μία effective resistance, R , εισάγει μία πτώση τάσης, $V=IR$. Το IR drop στο δίκτυο διανομής τροφοδοσίας VSS αποτελεί μία αύξηση της τάσης VSS στις πύλες G1-G4.

Το σχήμα E-2 επίσης δείχνει την πολυπλοκότητα των δικτύων διανομής ισχύος και του IR drop. Ας υποθέσουμε ότι η πύλη G4 έχει ένα ρεύμα λόγω VDD I_4 αμπέρ. Καμία άλλη πύλη δεν έχει ρεύμα. Το ρεύμα I_4 περνάει από την τροφοδοσία μέσω του δικτύου διανομής ισχύος στην πύλη G4. Το IR drop στην πύλη G4 είναι σε αυτή την περίπτωση $I_4(R_{11}+R_{12}+R_{13}+R_{14})$. Επιπρόσθετα, λόγω του ρεύματος στην πύλη G4, η πύλη G2 δεν έχει ιδανική

τροφοδοσία. Έχει ένα IR drop $I_4(R_{11}+R_{12})$. Επομένως, το ρεύμα κάθε πύλης στη σχεδίαση προκαλεί κάποιου είδους IR drop σε όλες τις πύλες του σχεδίου. Υποθέτοντας στιγμιαία ρεύματα I_1 - I_4 στις πύλες G_1 - G_4 αντίστοιχα, στο σχέδιο E-2, το IR drop στην πύλη G_4 θα ήταν το εξής :

$$I_1(R_{11}) + I_2(R_{11} + R_{12}) + I_3(R_{11} + R_{12} + R_{13}) + I_4(R_{11} + R_{12} + R_{13} + R_{14})$$

Το IR drop μπορεί να είναι είτε τοπικό είτε ολικό φαινόμενο.

- Το IR drop είναι τοπικό φαινόμενο όταν ένας αριθμός πυλών πολύ κοντά μεταξύ τους αλλάζουν κατάσταση ταυτοχρόνως, προκαλώντας έτσι IR drop. Το τοπικό IR drop μπορεί επίσης να προκληθεί από μία μεγαλύτερη αντίσταση σε ένα συγκεκριμένο κομμάτι του δικτύου διανομής ισχύος, όπως π.χ. η αντίσταση R_{14} να είναι πολύ μεγαλύτερη από το αναμενόμενο
- Το IR drop είναι επίσης ολικό φαινόμενο όταν δραστηριότητα σε μία περιοχή του ολοκληρωμένου προκαλεί IR drop σε άλλες περιοχές. Σε ένα καλά ισορροπημένο δίκτυο διανομής ισχύος με ισόποσα κατανεμημένο ρεύμα, το δίκτυο διανομής ισχύος τυπικά έχει ένα σύνολο από ισοδυναμικών επιφάνειες IR drop οι οποίες σχηματίζουν ομόκεντρους κύκλους με κέντρο το μέσο του ολοκληρωμένου, το οποίο έχει το μεγαλύτερο IR drop. Τα ρεύματα των μεγάλων πυλών μέσα στο ολοκληρωμένο αυξάνουν το μέγεθος τέτοιων ισοδυναμικών επιφανειών IR drop.

Οι διατυπώσεις για το IR drop μας δείχνουν πόσο σημαντικό είναι για τις πύλες σε ένα σχεδιασμό να αλλάζουν καταστάσεις σε διαφορετικές χρονικές στιγμές. Αν αλλάζουν κατάσταση όλες ταυτοχρόνως, το τοπικό αλλά και το ολικό IR drop σε ένα ολοκληρωμένο θα είναι πολύ μεγάλο. Σε μερικά κομμάτια του σχεδίου ωστόσο, είναι σημαντικό οι πύλες να αλλάζουν κατάσταση ταυτοχρόνως. Το ρολόι και τα latches που αυτό οδηγεί πρέπει να αλλάζουν κατάσταση ταυτοχρόνως σε ένα σύγχρονο ψηφιακό σύστημα, οπότε κάποιο ποσό IR drop είναι αναπόφευκτο σε ένα ολοκληρωμένο.

Το σχήμα E-2 επίσης δείχνει την διαφορά μεταξύ του μέσου και μέγιστου IR drop. Αν όλες οι πύλες άλλαζαν κατάσταση ταυτοχρόνως, το IR drop στην

πύλη G4 υπολογίζεται όπως δείξαμε προηγουμένως. Αν αυτό το μέγιστο ρεύμα παρουσιαζόταν σε παραπάνω από ένα τέταρτο του κύκλου του ρολογιού, το μέσο IR drop στην πύλη G4 καθ' όλη τη διάρκεια του κύκλου του ρολογιού θα ήταν το ένα τέταρτο αυτής της τιμής. Το μέγιστο IR drop σε οποιοδήποτε σημείο του ολοκληρωμένου μπορεί επομένως να είναι πολύ μεγαλύτερο από το μέσο IR drop σε ένα κύκλο ρολογιού. Όσο το μέγεθος ενός μπλοκ αυξάνει, ο λόγος του μέγιστου IR drop προς το μέσο IR drop μειώνεται κοντά στο 1.0, γιατί η πιθανότητα πολλών πυλών να αλλάζουν κατάσταση ταυτοχρόνως μειώνεται αισθητά. Οι πρωτεύουσες αιτίες της στιγμιαίας εμφάνισης IR drop (θόρυβος) είναι η εναλλαγή κατάστασης του ρολογιού, ενός δίαυλου ή των signal pads. Αυτά τα προβλήματα οφείλονται σε μεγάλο αριθμό μεγάλων πυλών που αλλάζουν κατάσταση συγχρόνως σε μία περιοχή.

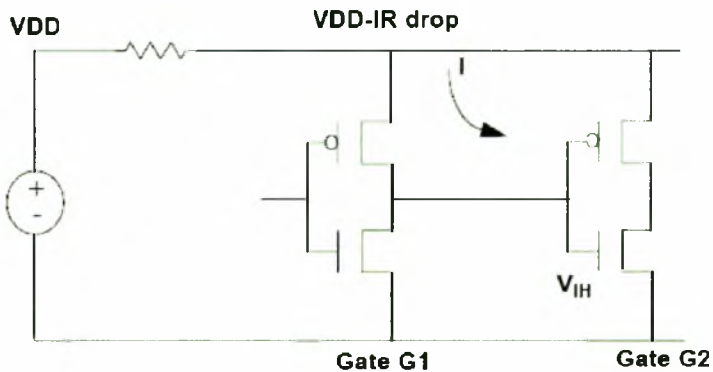
Συμπτώματα του IR Drop

Λόγω του ότι ο αριθμός αποτυχιών που προκύπτουν από το IR drop αυξήθηκε σημαντικά μόλις τελευταία, πολλοί σχεδιαστές δεν υπολογίζουν την κατανομή ισχύος σαν πιθανή πηγή αποτυχίας στο ολοκληρωμένο. Τα συμπτώματα των προβλημάτων που προκύπτουν από το IR drop αρκετά συχνά μοιάζουν με αυτές προβλημάτων χρονισμού ή ακεραιότητας σήματος. Τα συμπτώματα προβλημάτων IR drop περιλαμβάνουν τα εξής :

- Μη λειτουργικά ολοκληρωμένα. Αν το ολικό IR drop είναι πολύ μεγάλο όταν ένα ολοκληρωμένο λειτουργεί, οι λογικές πύλες δυσλειτουργούν. Η αποτυχία μοιάζει με αυτήν της λογικής λειτουργικής αποτυχίας ή κατασκευαστικού προβλήματος, παρόλο που η λογική προσομοίωση δείχνει ότι η σχεδίαση είναι σωστή. Ένας τρόπος διάγνωσης αυτού του συμπτώματος είναι να αυξήσουμε την τάση της πηγής τροφοδοσίας και να δούμε αν το κύκλωμα λειτουργεί.
- Λειτουργικές αποτυχίες κατά τη διάρκεια λειτουργίας ή λειτουργικές αποτυχίες εξαρτώμενες από δεδομένα. Τα τοπικά IR drop προβλήματα ευαισθητοποιούνται (sensitized) ή αναγκάζονται να συμβούν από συγκεκριμένες λειτουργίες στενής εγγύτητας, όπως όλα τα bit ενός διαύλου να αλλάζουν κατάσταση ταυτοχρόνως. Σε κανονική λειτουργία, η συγκεκριμένη ευαισθητοποίηση μπορεί να μην συμβεί.

Ωστόσο, μία συγκεκριμένη εισαγωγή δεδομένων μπορεί να ενεργοποιήσει το πρόβλημα. Αυτό το σύμπτωμα εμφανίζεται σαν λογική λειτουργική αποτυχία εκείνου του κομματιού του ολοκληρωμένου.

- Αποτυχία χρονισμού. Όταν το ολικό IR drop είναι υψηλό, αλλά όχι τόσο υψηλό που να προκαλέσει λογική αποτυχία, το σύμπτωμα είναι μία αποτυχία χρονισμού στο ολοκληρωμένο. Σε αυτή την περίπτωση, το IR drop επιβραδύνει την ταχύτητα της λειτουργίας της πύλης. Πειράματα έχουν δείξει ότι ένα πέντε τοις εκατό (5%) σε μία πύλη μπορεί να επιβραδύνει την ταχύτητα της μέχρι και δεκαπέντε τοις εκατό (15%). Η πτώση στην ταχύτητα οφείλεται σε δύο μηχανισμούς, όπως φαίνεται και στο σχήμα E-3. Κατά πρώτον, το IR drop στην είσοδο τροφοδοσίας της πύλης G1 επιβραδύνει την ταχύτητα φόρτισης (charging speed) της εξόδου της πύλης. Με λίγα λόγια, τάση εξόδου της πύλης G1 ανεβαίνει μόνο μέχρι VDD μείον το IR drop. Ακολούθως, αυτή η πτώση μειώνει την λογική υψηλή τάση εισόδου (logical high input voltage) στην πύλη G2, μειώνοντας έτσι την ταχύτητα της να αλλάζει καταστάσεις.



Σχήμα E-3 Αντίκτυπος του IR Drop στην ταχύτητα της πύλης

- Αποτυχίες χρονισμού κατά τη διάρκεια λειτουργίας ή αποτυχίες χρονισμού εξαρτώμενες από δεδομένα. Όπως και στις λογικές αποτυχίες κατά τη διάρκεια της λειτουργίας, συγκεκριμένες εισαγωγές δεδομένων μπορούν να προκαλέσουν IR drop το οποίο φαίνεται σαν

μία αποτυχία χρονισμού. Μία αποτυχία χρονισμού κατά τη διάρκεια της λειτουργίας τέτοιου τύπου αποτελεί σύμπτωμα του προβλήματος της χωρητικής σύζευξης μεταξύ γραμμών σήματος. Ένας τρόπος διάγνωσης αυτού του συμπτώματος είναι να μειώσουμε την συχνότητα του ρολογιού και να δούμε αν το ολοκληρωμένο δουλεύει.

Ηλεκτρομετανάστευση (Electromigration - EM)

Η ηλεκτρομετανάστευση χρησιμοποιείται σαν ένας γενικός όρος για να περιγράψουμε μηχανισμούς αποτυχίας στα μεταλλικά καλώδια (metal wires) ενός ολοκληρωμένου που προκαλούνται από την κίνηση των μεταλλικών ατόμων (metal atoms) σε ένα καλώδιο λόγω του μεγάλου φόρτου ρεύματος (current stress)

Διορθώνοντας το IR Drop και το Electromigration

Η καλύτερη λύση σε ένα διδόμενο IR drop πρόβλημα εξαρτάται από τον τύπο του IR drop, την αρχιτεκτονική του ολοκληρωμένου, την διάταξη του ολοκληρωμένου και την λειτουργικότητα. Οι ακόλουθες προσεγγίσεις έχουν χρησιμοποιηθεί επιτυχώς :

- διαπλάτυνση των γραμμών των μετάλλων
- προσθήκη ή αφαίρεση καναλιών τροφοδοσίας για ανακατεύθυνση του ρεύματος
- κλιμάκωση της εναλλαγής καταστάσεως μεγάλων συσκευών
- μείωση των σημείων όπου τοποθετούνται buffers
- προσθήκη πυκνωτών αποσύζευξης
- χρήση της τεχνολογίας solder-bump
- προσθήκη περισσότερων πηγών VDD

Κ εφάλαιο 1

Τα Εργαλεία Που Χρησιμοποιήσαμε

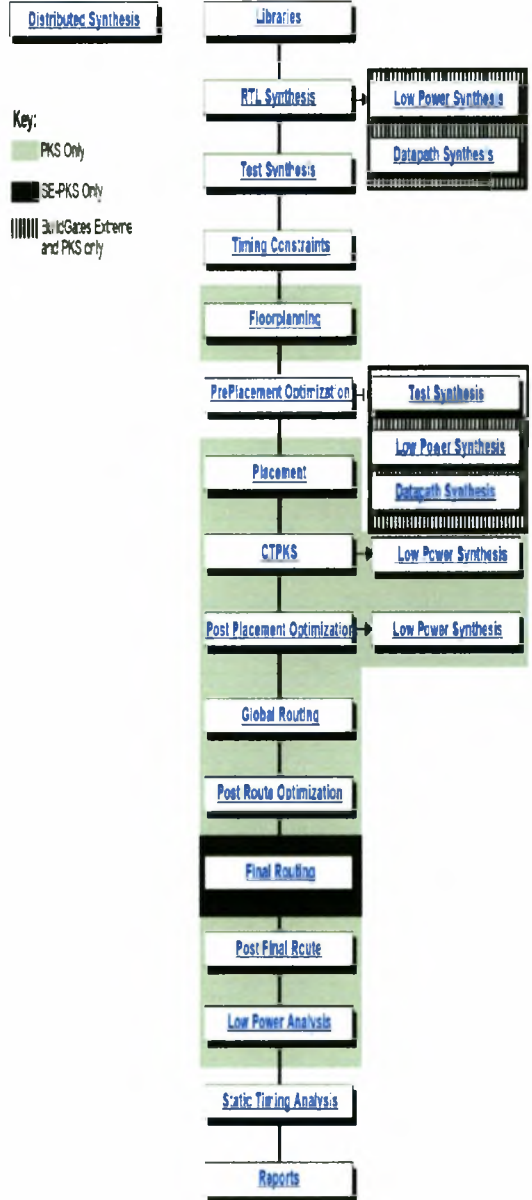
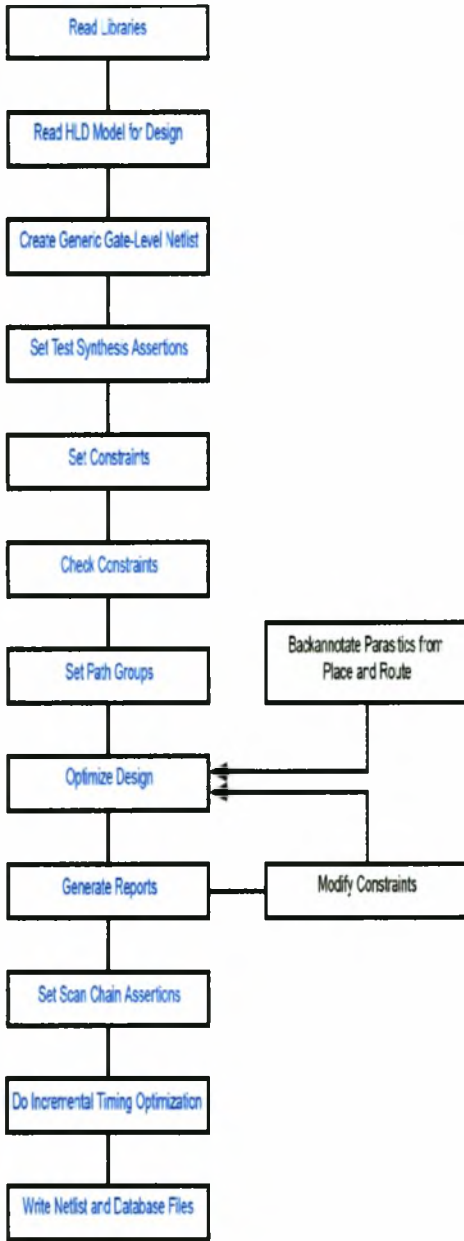
Το λογισμικό που χρησιμοποιήσαμε, προέρχεται αποκλειστικά από την εταιρεία Cadence. Η Cadence αποτελεί μία από τις μεγαλύτερες εταιρείες που αναπτύσσουν λογισμικό για προσομοίωση, σύνθεση και έλεγχο ψηφιακών κυκλωμάτων. Ακολουθούν τα εργαλεία που χρησιμοποιήσαμε στην παρούσα διπλωματική.

BuildGates Synthesis

Το BuildGates Synthesis είναι ένα EDA (Electronic Design Automation) εργαλείο της Cadence το οποίο επιτελεί με μεγάλη ταχύτητα και ακρίβεια το synthesis του κυκλώματος μας σε chip-level. Στα παρακάτω σχήματα μπορούμε να δούμε πως το BuildGates Synthesis συμβάλλει στην συνολική ροή σχεδίασης :

Basic BuildGates Flow

BuildGates Synthesis

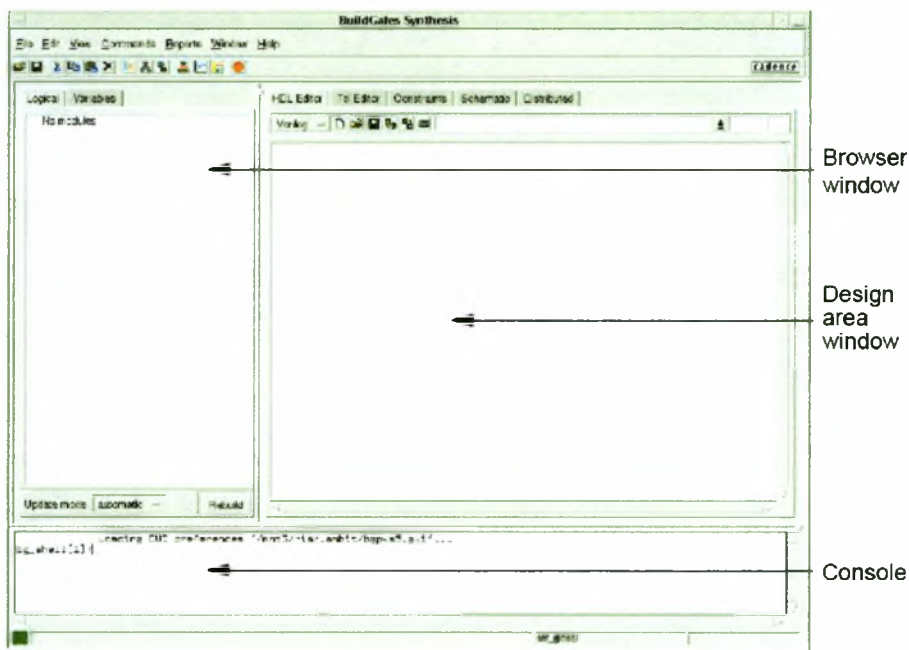


Στο σημείο αυτό θα δώσουμε και τον πρώτο «επίσημο» ορισμό της σύνθεσης ενός ψηφιακού κυκλώματος :

«Σύνθεση ονομάζεται η διαδικασία κατά την οποία μετατρέπουμε μια σχεδίαση, η οποία μας δίνεται σε HDL κώδικα, σε ένα βέλτιστο netlist σε επίπεδο πυλών το οποίο προσδιορίζεται πλήρως από μια τεχνολογική βιβλιοθήκη»

Ακολουθεί μία τυπική διαδικασία synthesis ενός κυκλώματος μέσα από το περιβάλλον του BuildGates Synthesis :

Εκκινούμε το πρόγραμμα με την εντολή *bgx_shell*. Είναι όμως πιο εύκολο να εκκινήσουμε και να χρησιμοποιήσουμε το πρόγραμμα με το γραφικό του περιβάλλον χρησιμοποιώντας την εντολή *bgx_shell -gui* αφού έτσι μπορούμε να βλέπουμε γραφικά ανά πάσα στιγμή την όλη διαδικασία του synthesis. Το γραφικό περιβάλλον του BuildGates Synthesis είναι το παρακάτω :



Έπειτα εκτελούμε μία σειρά από scripts γραμμένα σε TCL τα οποία με την σειρά τους επιτελούν τα εξής:

1. Ορισμός των διάφορων φακέλων που χρειάζεται το πρόγραμμα, όπως οι φάκελοι όπου θα τοποθετηθούν τα διάφορα reports, οι φάκελοι μέσα

στους οποίους βρίσκεται το κύκλωμά μας σε RTL μορφή αλλά και ορισμός ίσως κάποιων τιμών όπως καθυστερήσεις για διάφορα κελιά.

2. Διαβάζουμε τις βιβλιοθήκες που θα χρησιμοποιήσουμε στο κύκλωμά μας. Πιο συγκεκριμένα διαβάζουμε τα *.tlf* (timing library files) των βιβλιοθηκών μας, τα οποία περιέχουν όλες τις πληροφορίες για τους χρονισμούς αυτών.
3. Ακολουθεί το RTL synthesis του κυκλώματός μας. Με την εντολή *read_verilog* γίνεται το parsing του RTL κώδικα και με την εντολή *do_build_generic* φτιάχνεται η εσωτερική αναπαράσταση.
4. Εφαρμογή όλων των design constraints που θέλουμε να εφαρμόσουμε στο κύκλωμά μας. Με εντολές όπως π.χ. οι *set_internal_delay*, *set_external_delay*, *set_drive_cell* μπορούμε να θέσουμε κάθε είδους περιορισμό στο κύκλωμά μας.
5. Σε αυτό το σημείο καλείται ίσως η πιο βαρύνουσα σημασίας εντολή σε όλο το synthesis, η *do_optimize*. Επιτελεί το mapping στα κελιά των βιβλιοθηκών αλλά και λογική βελτιστοποίηση μέχρι όλα τα constraints που θέσαμε να ικανοποιηθούν. Μόλις τελειώσει το synthesis, θα υπάρξει ένα timing report από την Common Timing Engine (CTE) αφού το BuildGates περιλαμβάνει ένα εργαλείο στατικής ανάλυσης χρονισμού (STA tool). Το report αυτό περιλαμβάνει το ελάχιστο slack του σχεδιασμού μας, το οποίο ουσιαστικά είναι ο χρόνος στον οποίο το κρίσιμο μονοπάτι είναι γρηγορότερο από το clock signal του κρίσιμου μονοπατιού.
6. Γράψιμο αποτελεσμάτων της διαδικασίας του synthesis. Τυπικές εντολές που χρησιμοποιούμε είναι οι *report_timing*, *report_analysis_coverage*, *report_area*, *write_verilog*, *write_adb*.

Virtuoso IC

Από το Virtuoso της Cadence χρησιμοποιήσαμε κυρίως το Virtuoso Schematic Editor και το Virtuoso Layout Editor.

Το Virtuoso Schematic Editor αποτελεί το περιβάλλον σύνθεσης σχεδιασμού της πλατφόρμας Virtuoso Custom Design. Μπορούμε να το χρησιμοποιήσουμε από το προσδιορισμό ακόμα της αρχιτεκτονικής μας μέχρι και σε υλοποιήσεις σε επίπεδο τρανζίστορ. Χρησιμοποιώντας τον hierarchy editor, η οπτική αναπαράσταση μεγάλων και σύνθετων σχεδιασμών απλοποιείται αρκετά. Η καλά οργανωμένη βιβλιοθήκη μας επιτρέπει να δουλεύουμε γρήγορα τόσο σε επίπεδο πυλών όσο και σε επίπεδο τρανζίστορ, ενώ οι δυνατότητες του wire routing μας βοηθά να ενώσουμε εύκολα τα devices μας. Κατά την σχεδιαστική μας ροή κάνουμε εκτεταμένη χρήση των δυνατοτήτων ελέγχου του σχεδιασμού μας που μας προσφέρει το Virtuoso Schematics Editor. Μπορούμε να κάνουμε έλεγχο του σχεδιασμού μας για drawing και electrical rule παραβάσεις, όπως επικαλυπτόμενα στοιχεία, ανοιχτές οι βραχυκυκλωμένες συνδέσεις, μη συνδεδεμένες εισόδους και εξόδους, συνέπεια αντικειμένων και μη έγκυρα ονόματα έτσι ώστε να διασφαλίσουμε την ακρίβεια του σχεδιασμού μας. Επιπλέον μπορούμε να ελέγξουμε τις συνδέσεις απ' άκρο σε άκρο ολόκληρης της ιεραρχίας του σχεδιασμού μας έτσι ώστε να επαληθεύσουμε ότι όλα τα ονόματα των pin ταιριάζουν και ότι έχουν γίνει όλες οι ενώσεις με τα καλώδια.

SKILL

Η γλώσσα SKILL αναπτύχθηκε από την Cadence για να χρησιμοποιείται από τις διάφορες σουίτες εργαλείων της. Επιτρέπει στον χρήστη να γράψει ένα “script” το οποίο εκτελεί οποιαδήποτε εντολή της Cadence. Η SKILL είναι μία interpretive γλώσσα όπως η LISP και η Perl. Ωστόσο, υποστηρίζει μία πιο

συμβατική σύνταξη, παρόμοια με αυτή της C. Σχεδιάστηκε για να δουλεύει σε επαναληπτικές διεργασίες και μερικές από τις λειτουργίες της βασίζονται σε λίστες. Παρέχει ένα ασφαλές, υψηλού επιπέδου περιβάλλον προγραμματισμού το οποίο χειρίζεται αυτόματα πολλές παραδοσιακές προγραμματιστικές λειτουργίες, όπως η διαχείριση μνήμης. Είναι ιδανική για γρήγορο prototyping. Μπορεί κάποιος να τεκμηριώσει σταδιακά τα βήματα του αλγορίθμου του πριν να τα ενσωματώσει σε ένα μεγαλύτερο πρόγραμμα. Τα λάθη διαχείρισης αποθήκευσης είναι από τα πιο συνηθισμένα στην ανάπτυξη λογισμικού. Η SKILL παρέχει αυτόματη διαχείριση αποθήκευσης και έτσι γλιτώνει τον προγραμματιστή από αυτόν τον φόρτο. Επίσης παρέχει έτοιμες ρουτίνες για διαχείριση λιστών και περίπλοκων εξαιρέσεων, καθιστώντας έτσι τον κώδικά μας πιο συνοπτικό, και άρα πιο εύκολα «συντηρήσιμο». Όπως είπαμε, στην SKILL τα προγράμματα αναπαριστώνται σαν λίστες (όπως και στην LISP), μπορούν να χειριστούν σαν δεδομένα. Μπορεί κάποιος δυναμικά να δημιουργήσει, τροποποιήσει ή επιλεκτικά να αποτιμήσει ορισμούς συναρτήσεων και εκφράσεων. Επιπλέον, η SKILL υποστηρίζει επιλεκτική αποτίμηση εκφράσεων μέσα σε εισαγωγικά. Η επιλεκτική αποτίμηση εξαλείφει τις μακριές προτάσεις κλήσεων στις εντολές list και append.

Μία από τις πιο συνήθεις χρήσεις της SKILL είναι για την κατασκευή περιβαλλόντων διεπαφής. Παρέχει εργαλεία για tracing, αποσφαλμάτωσης και profiling, ικανά για την δημιουργία μεγαλεπίβολων project. Μας επιτρέπει να προσαρμόσουμε υπάρχουσες CAD εφαρμογές και μας βοηθά στην ανάπτυξη νέων.

Ultrasim

Ο Virtuoso Ultrasim Full-chip Simulator είναι ο FastSPICE προσομοιωτής της Cadence. Ο Ultrasim είναι ένας γρήγορος και πολυχρηστικός ιεραρχικός προσομοιωτής, ο οποίος χρησιμοποιείται για την επαλήθευση αναλογικών, mixed-signal και ψηφιακών κυκλωμάτων. Μπορεί να χρησιμοποιηθεί για την

λειτουργική επαλήθευση κυκλωμάτων μνήμης με δισεκατομμύρια τρανζίστορ αλλά και για προσομοίωση υψηλής ακρίβειας σύνθετων αναλογικών κυκλωμάτων. Έχει δυνατότητες για deep-submicron (DSM) αναλύσεις συμπεριλαμβανομένων χρονισμού, ενέργειας, θορύβου και αξιοπιστίας. Παρόλο που είναι πολύ πιο γρήγορος από το SPICE (10.000 φορές γρηγορότερος) χάνει μόνο συν πλην ένα τοις εκατό (+/- 1%) σε ακρίβεια, λειτουργώντας στην πιο επακριβή κατάσταση. Μας επιτρέπει έτσι με αυτό τον τρόπο να κάνουμε γρήγορες προσομοιώσεις των κυκλωμάτων μας, χωρίς να χάνουμε σε ακρίβεια. Πολύ χρήσιμοι είναι επίσης και οι αλγόριθμοι σύμπτυξης και μείωσης κατά την pre-layout και post-layout προσομοίωση για τα εξαγόμενα postlayout netlists.

Spectre Simulator

Ο Spectre είναι ένας προσομοιωτής κυκλωμάτων ο οποίος προσομοιώνει αναλογικά και ψηφιακά κυκλώματα. Οι βασικές δυνατότητες του SPECTRE είναι ίδιες σε λειτουργία και εφαρμογή με το SPICE. Και οι δύο προσομοιωτές χρησιμοποιούν τους ίδιους βασικούς αλγορίθμους όπως πεπλεγμένες μέθοδοι ενσωμάτωσης, Newton-Raphson και επιλύσεις πινάκων, με τη διαφορά ότι κάθε αλγόριθμος είναι υλοποιημένος εκ νέου. Έτσι ο Spectre Simulator κερδίζει σε ταχύτητα, ακρίβεια, αξιοπιστία και ευελιξία σε σύγκριση με τους υπάρχοντες SPICE-like προσομοιωτές.

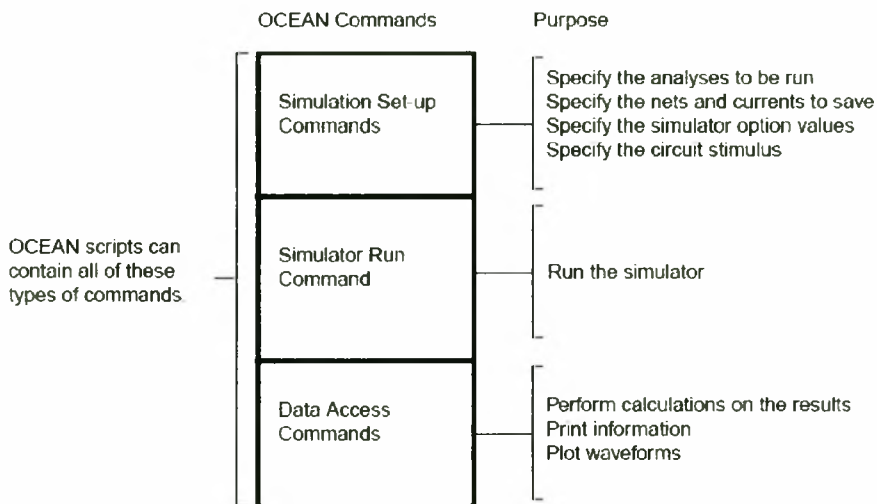
Ocean

Η OCEAN αποτελεί μία scripting language. Τα αρχικά της προκύπτουν από το Open Command Environment for Analysis. Είναι βασισμένη στην SKILL (και

άρα κληρονομεί όλες τις ιδιότητες που αναφέραμε παραπάνω). Χρησιμοποιείται για να «σετάρει», να προσομοιώνει και να αναλύει κυκλωματικά δεδομένα. Η OCEAN τρέχει αποκλειστικά σε κονσόλα και δεν έχει κάποιο γραφικό περιβάλλον. Επιτρέπει στον χρήστη να εκτελεί τις όποιες εντολές μέσα από ένα διαδραστικό περιβάλλον ή να δημιουργεί script τα οποία περιέχουν τις απαιτούμενες εντολές, τα οποία script μετέπειτα φορτώνονται μέσα στο περιβάλλον της OCEAN. Ουσιαστικά αυτό που κάνουμε με την OCEAN είναι να ελέγχουμε τα κυκλώματα που σχεδιάζουμε υπό διάφορες συνθήκες. Μετά τις οποιεσδήποτε αλλαγές στα κυκλώματά μας, αρκεί μόνο να ξανατρέξουμε τα script που έχουμε δημιουργήσει για να τα ελέγξουμε. Με την OCEAN μπορούμε :

- να δημιουργήσουμε script τα οποία μπορούμε να τρέχουμε επαναλαμβανόμενα για να επιβεβαιώνουμε την απόδοση των κυκλωμάτων μας.
- να τρέχουμε παρατεταμένες αναλύσεις όπως παραμετρικές και στατιστικές πιο αποτελεσματικά
- να τρέχουμε προσομοιώσεις από ένα μη γραφικό, απομακρυσμένο τερματικό

Μπορούμε να φτιάξουμε OCEAN scripts έτσι ώστε να μπορέσουμε να φέρουμε εις πέρας μία πλήρους εύρους προσομοίωση και όλες τις εργασίες πρόσβασης στα δεδομένα αυτής που θα έπρεπε διαφορετικά να εκτελέσουμε «χειροκίνητα» μέσα από το περιβάλλον του Virtuoso Analog Design . Ένα script γραμμένο σε OCEAN μπορεί να περιέχει τρεις τύπους εντολών, όπως φαίνεται και από το παρακάτω σχεδιάγραμμα



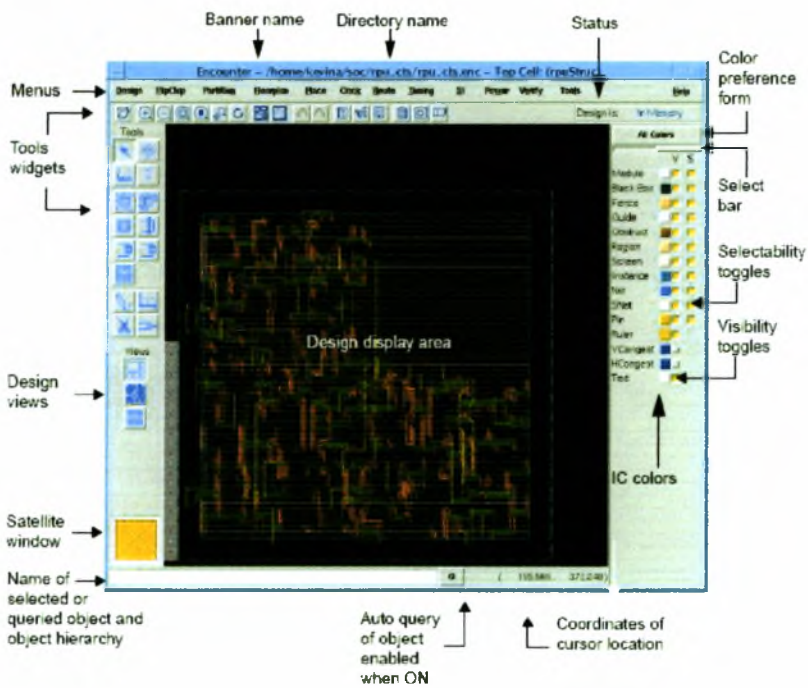
Άλλο ένα πολύ ενδιαφέρον χαρακτηριστικό της OCEAN είναι η δυνατότητα της να μπορεί να τρέχει σε ένα σύνολο υπολογιστικών συστημάτων κάνοντας χρήση των εντολών της για κατανεμημένη επεξεργασία. Οι εντολές αυτές επιτρέπουν στον χρήστη να ορίσει πού και πότε αυτές θα εκτελεστούν και του δίνουν τη δυνατότητα να παρακολουθεί και να ελέγχει τις δουλειές με διάφορους τρόπους. Χρησιμοποιώντας τις εντολές κατανεμημένης επεξεργασίας μπορούμε :

- να υποβάλλουμε μία ή και περισσότερες δουλειές σε μία ουρά κατανεμημένης επεξεργασίας
- να καθορίσουμε ένα διακομιστή ή ένα σύνολο διακομιστών στους οποίους θα κατανείμουμε τις δουλειές
- να δούμε την κατάσταση των δουλειών
- να καθορίσουμε πότε μία δουλειά θα τρέξει ή σε ποια σειρά ένα σύνολο από δουλειές θα τρέξει
- να αναστείλουμε και να συνεχίσουμε δουλειές
- να ακυρώσουμε δουλειές

SoC Encounter

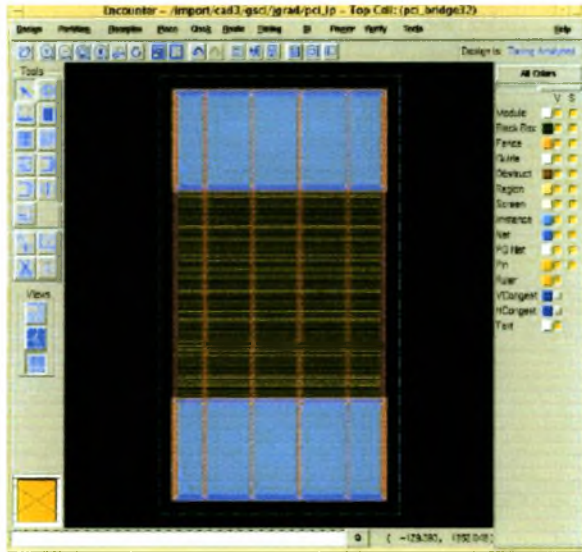
Ο SoC Encounter αποτελεί ένα εργαλείο για ιεραρχική RTL-to-GDSII φυσική υλοποίηση. Μεταξύ άλλων, κάποια σημαντικά χαρακτηριστικά του SoC Encounter είναι τα εξής : RTL σύνθεση, βελτιστοποίηση χρονοισμού, βελτιστοποίηση φυσικής σύνθεσης, παραγωγή GDSII αρχείων.

Αφού έχουμε ολοκληρώσει την διαδικασία της σύνθεσης όπως περιγράψαμε παραπάνω (βλ. BuildGates Synthesis), χρησιμοποιούμε το SoC Encounter για την φυσική υλοποίηση. Εκκινούμε το πρόγραμμα με την εντολή *encounter* και μας εμφανίζεται το κεντρικό παράθυρο του προγράμματος :

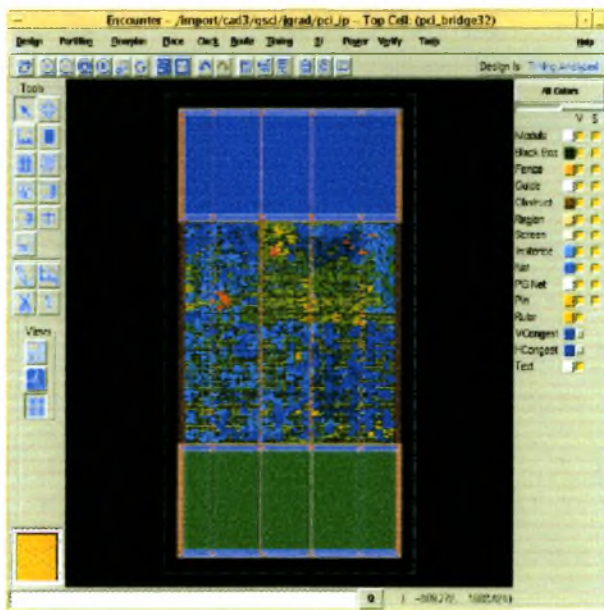


Όπως και στην περίπτωση του BuildGates Synthesis, έτσι και εδώ κάνουμε χρήση *scripts* γραμμένα σε TCL για να αυτοματοποιήσουμε λίγο την διαδικασία. Η μη χρήση *scripts* θα απαιτούσε από τον σχεδιαστή να πληκτρολογεί μεμονωμένα τις όποιες εντολές του είναι απαραίτητες για να επιτελέσει την δουλειά του.

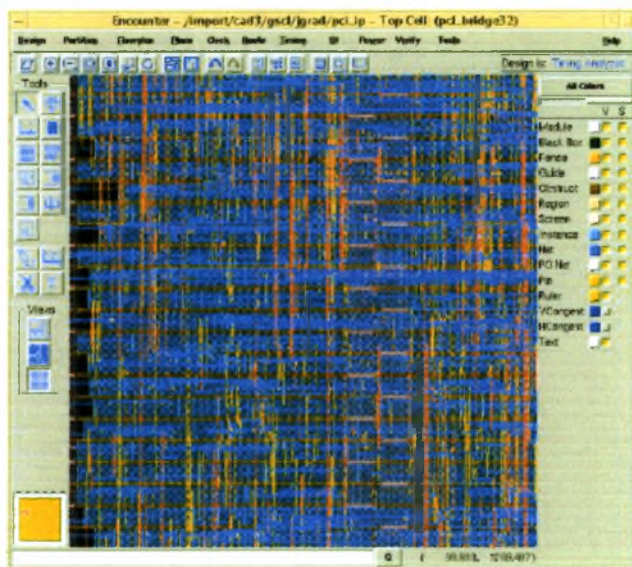
Στην συνέχεια θα περιγράψουμε ένα τυπικό flow στο SoC Encounter (σαν παράδειγμα χρησιμοποιήσαμε την φυσική υλοποίηση μίας PCI-Wishbone γέφυρας). Αρχικά, με τη χρήση ενός tcl script ορίζουμε μερικές μεταβλητές



Σε αυτό το σημείο εκτελείται ένα αρχικό placement των block (δεν υπάρχουν filler cells ακόμα), ακολουθούμενο από το Clock Tree Synthesis. Το Clock Tree Synthesis εισάγει clock buffers στο σχεδιασμό μας για να ελαχιστοποιήσουμε το clock skew αλλά και για να βελτιστοποιήσουμε τους χρόνους ανόδου και καθόδου των σημάτων ρολογιού. Μετά το πέρας της παραπάνω διαδικασίας, παίρνουμε το εξής :



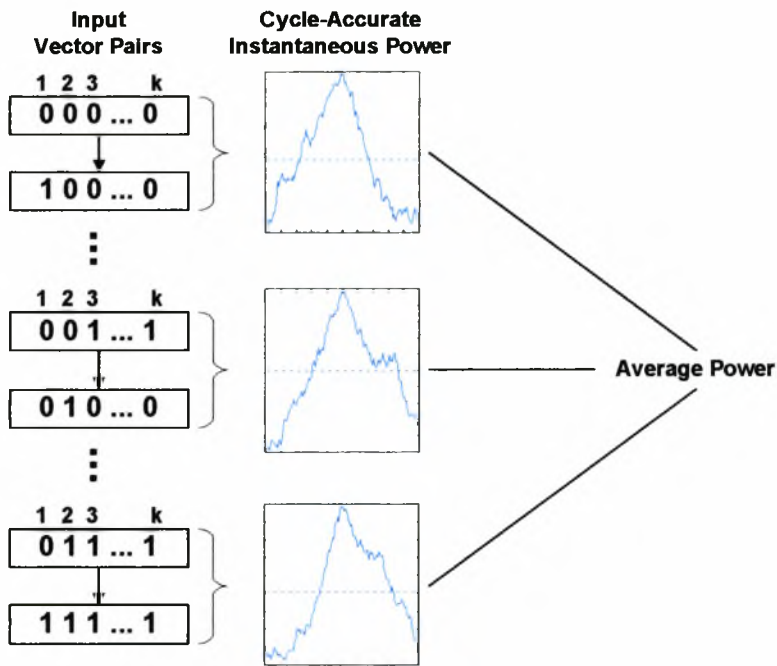
Αυτό που πλέον μένει είναι να γίνει το power routing, να τοποθετήσουμε filler cells για να μπορέσουμε να προχωρήσουμε στο Global και Detailed Routing. Το τελικό αποτέλεσμα θα μοιάζει κάπως έτσι :



Αφού επαληθεύσουμε το κύκλωμά μας, εξάγουμε τα τελικά αποτελέσματα και τις τελικές μας μετρήσεις.

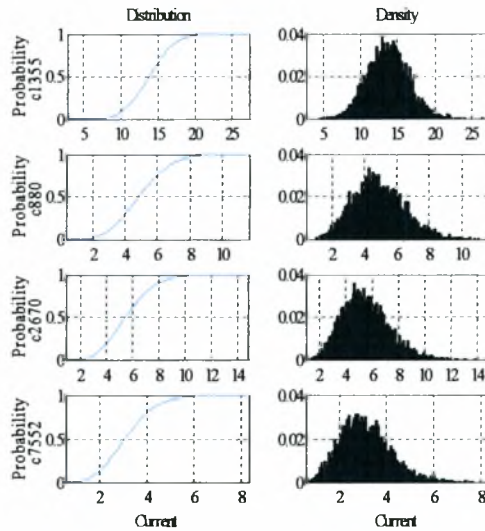
IExtreme Engine

Ο IExtreme είναι μία μηχανή γραμμένη σε C. Στο flow που θα περιγράψουμε παρακάτω (βλ. Κεφάλαιο 2), χρησιμοποιούμε τον IExtreme για να εκτιμήσουμε τα μέγιστα ρεύματα χειρότερης περίπτωσης σε κάθε καταβόθρα. Ο IExtreme εκμεταλλεύεται την πιο σύγχρονη έρευνα στην στατιστική εκτίμηση μέγιστου ρεύματος από την αρχή της Θεωρίας Μεγίστων Αριθμών (*Extreme Value Theory – EVT*), η οποία είναι το παρεμφερές πεδίο της στατιστικής για την εκτίμηση του άγνωστου μέγιστου ενός σχετικού πληθυσμού από ένα (ή και περισσότερα) δείγματά του.



Γραφική απεικόνιση του εναλλακτικού (υπό τη μορφή πληθυσμιακού μέσου) ορισμού της μέσης ισχύος στα κυκλώματα CMOS VLSI.

Μερικές προηγούμενες τεχνικές στηριζόμενες στην στατιστική αξιοποίηση μερικών δεδομένων προσομοίωσης εμφανίστηκαν, αλλά είτε δεν στηρίζονταν στην ΘΜΑ είτε δεν έκαναν επαρκή χρήση της θεωρίας. Η εγγενής δυσκολία στο να υπολογίσουμε τα μέγιστα ρεύματα $i_{mx,k}$ βρίσκεται στο ότι η κυματομορφή των ρευμάτων $i_k(t)$ που «αντλείται» από κάθε block k εξαρτάται από το ζευγάρι διανυσμάτων (v_1, v_2) , τα οποία εφαρμόζονται στο κύκλωμα στα όρια της ακμής το ρολογιού (για να επιβάλλουν μία μετάβαση λογικής κατάστασης). Αν η μέγιστη τιμή (peak value) $i_{p,k}$ του $i_k(t)$ χρειάστηκε παραπάνω από ένα κύκλο ρολογιού ως συνάρτηση των (v_1, v_2) , τότε το $i_{mx,k}$ ισούται με την μέγιστη τιμή αυτού του μέγιστου ρεύματος σε ένα κύκλο ρολογιού μεταξύ όλων των πιθανών ζευγαριών (v_1, v_2) .



Εμπειρική συνάρτηση κατανομής και συνάρτηση πυκνότητας ενός cycle-accurate ακρώρειου ρεύματος

Από μία στατιστική άποψη, όλο το σύνολο των ζευγών των διανυσμάτων μπορεί να εδραιώσει έναν αρχικό πληθυσμό μεγέθους 4^r (όπου r είναι ο αριθμός των αρχικών εισόδων – ή ο αριθμός των bits σε κάθε διάνυσμα εισόδου) στο οποίο η ποσότητα του cycle-accurate ακρώρειου ρεύματος θεωρείται σαν μία τυχαία μεταβλητή X . Υποθέτοντας ότι η X χαρακτηρίζεται από cumulative distribution (cdf) συνάρτηση $F(x)$ (και συσχετιζόμενη συνάρτηση πυκνότητας $f(x) = dF(x)/dx$) η οποία θεωρείται συνεχής και παραγωγίσιμη (βλ. Σχήμα 2), τότε το πρόβλημα του καθορισμού του συνολικού μέγιστου ρεύματος μπορεί να αναχθεί σε πρόβλημα υπολογισμού του άγνωστου μέγιστου ενός διδόμενου στατιστικού πληθυσμού με cdf $F(x)$.

DIVA Verification

Το DIVA είναι ένα σύνολο από εργαλεία φυσικής επαλήθευσης το οποίο μας επιτρέπει να βρίσκουμε και να διορθώνουμε σχεδιαστικά λάθη. Με το DIVA ελέγχουμε τη φυσική σχεδίαση και την ηλεκτρική λειτουργικότητα και μπορούμε να εκτελέσουμε συγκρίσεις μεταξύ layout και schematic. Πολύ

σημαντικό στοιχείο είναι ότι με το DIVA μπορούμε να εκτελούμε βηματικούς (incremental) ελέγχους σε περιοχές που αλλάζουμε.

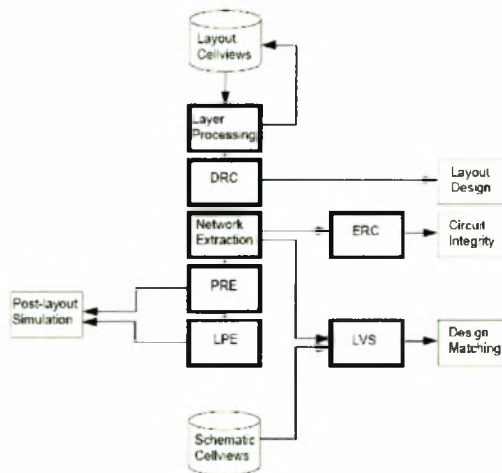
Τα εργαλεία που αποτελούν το DIVA Verification είναι τα εξής :

- Design Rule Checker (iDRC)
- Layout Parasitic Extractor (iLPE)
- Parasitic Resistance Extractor (iPRE)
- Electrical Rules Checker (iERC)
- Layout Versus Schematic (iLVS)

Στο διάγραμμα που ακολουθεί μπορούμε να δούμε ποιες λειτουργίες μπορούμε να εκτελέσουμε και με ποιο εργαλείο :

| Program | Process | Product |
|-------------|------------------------------------|---------|
| DRC/Extract | Layer Processing | iDRC |
| | Design Rule Checking | iDRC |
| | Connectivity Extraction | iDRC |
| | Device Extraction | iLPE |
| | Parameter Extraction | iLPE |
| | Parasitic Extraction | iLPE |
| | Resistance Extraction | iPRE |
| ERC | Electrical Rule Checking | iERC |
| LVS | Layout Versus Schematic Comparison | iLVS |
| | Correspondence Point Generation | iLVS |

ενώ το flow που χρησιμοποιούμε για verification χρησιμοποιώντας το DIVA είναι το ακόλουθο :



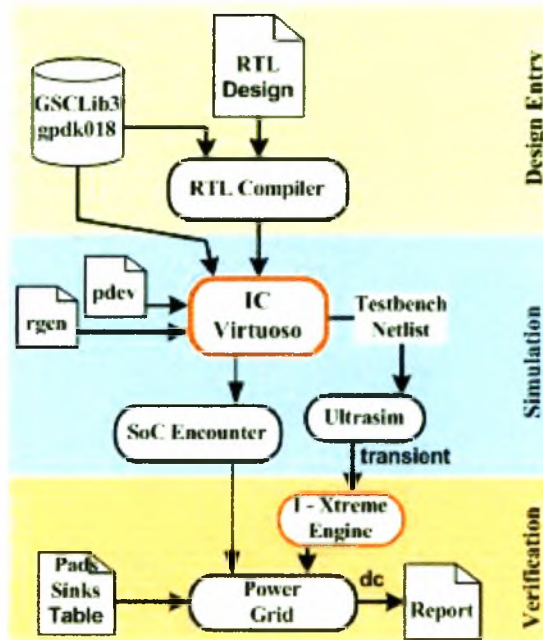
Στο δικό μας flow που ακολουθεί, χρησιμοποιήσαμε μόνο τον iPRE (μέσα από το περιβάλλον του Virtuoso XL), ο οποίος εξάγει τις παρασιτικές αντιστάσεις και χωρητικότητες και μετατρέπει τις διασυνδέσεις του κυκλώματος σε ένα πλήρες δίκτυο αντίστασης-χωρητικότητας (RC network).

Κεφάλαιο 2

Μία RTL-to-Grid Σχεδιαστική Ροή Για Έλεγχο Δικτύου Διανομής Ισχύος Βασισμένη σε μία Στατιστική Μηχανή

Στο παρόν κεφάλαιο θα δείξουμε πως με την χρήση των κατάλληλων εργαλείων μπορούμε να κάνουμε verify το power grid του κυκλώματος μας. Θα χρησιμοποιήσουμε σαν παράδειγμα μία 8 bit Αριθμητική Λογική Μονάδα (ALU).

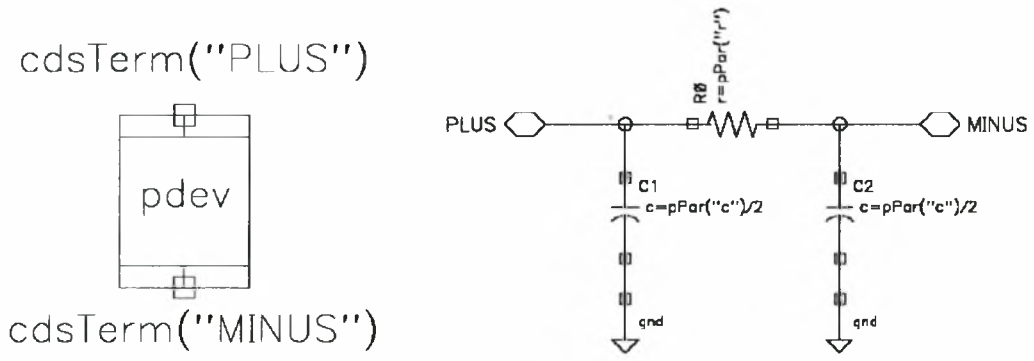
Η πρώτη προσέγγιση που θα χρησιμοποιήσουμε περιγράφεται από το flow που φαίνεται παρακάτω :



Μέσα από το περιβάλλον Cadence Virtuoso IC και κάνοντας χρήση της τεχνολογικής βιβλιοθήκης GSCLib3.0 (gpdk018) ο σχεδιαστής πρέπει να σχεδιάσει το κύκλωμα. Όπως έχουμε αναφέρει, χρησιμοποιούμε το symbolPT view για τα GSCLib3.0 τυποποιημένα κελιά (standard cells).

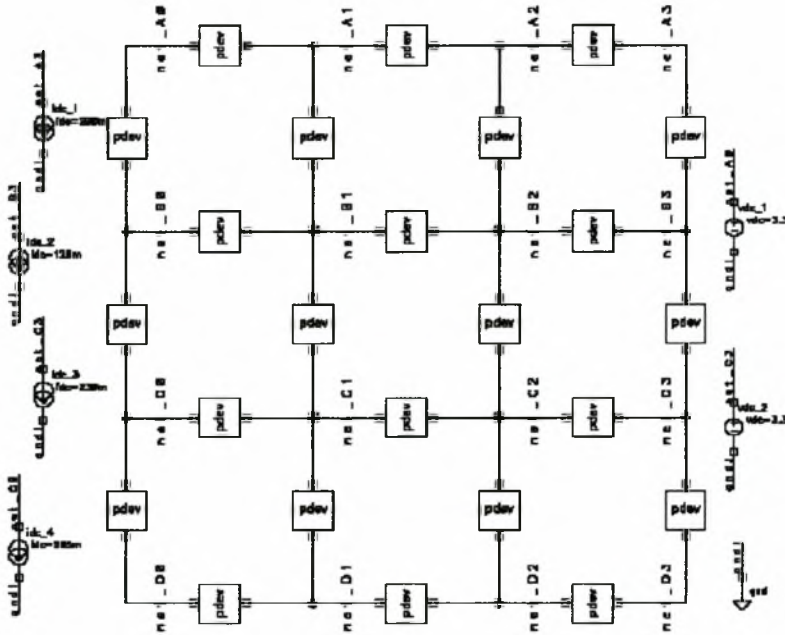
Για την μοντελοποίηση του δικτύου διανομής χρησιμοποιούμε μία παραμετρική συσκευή ("pdevice" ή "pdev") η οποία περιέχει όλες τις cdf

ιδιότητες ενός “presistor” κελιού από την analogLib. Ένα pdevice περιέχει δεδομένα για την παρασιτική αντίσταση μίας διακλάδωσης του δικτύου διανομής. Μπορεί επίσης να περιέχει δεδομένα για την παρασιτική χωρητικότητα και είναι ευέλικτο να προσθέσει πληροφορία για επαγωγικές επιπτώσεις. Χρησιμοποιούμε το “pdevice” για να καταγράψουμε την τρέχουσα κυματομορφή μετά την ανάλυση ενός testbench netlist και να παράγουμε το δίκτυο διανομής ισχύος.



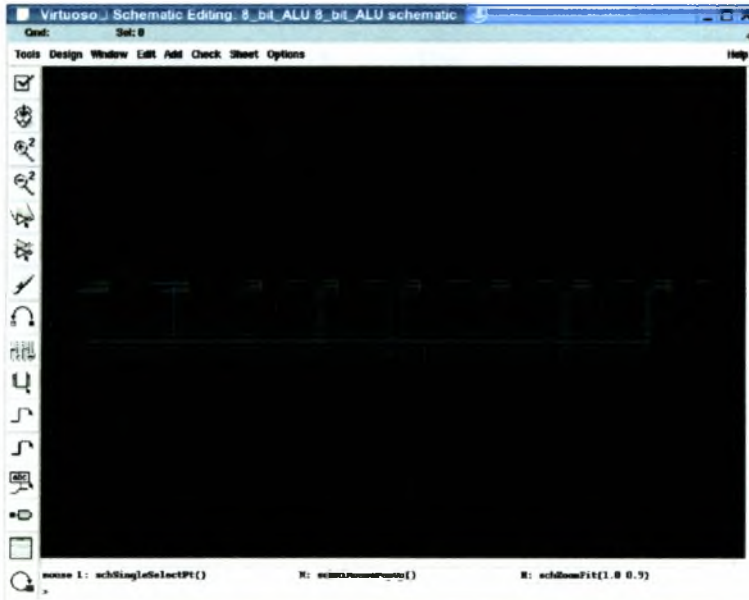
Το symbol και schematic cell view της παραμετρικής συσκευής (pdevice)

Το μοντέλο που θα υιοθετήσουμε για το δίκτυο διανομής ισχύος όσον αφορά τη δική μας σχεδιαστική ροή είναι το γραμμικό μοντέλο αντίστασης (`pdev`) με dc πηγές ρεύματος από την analogLib (`idc`) σε κάθε καταβόθρα και dc πηγές τάσης (`vdc`) για τα pads, αφού ψάχνουμε για στατικά διανύσματα ρεύματος για DC ανάλυση

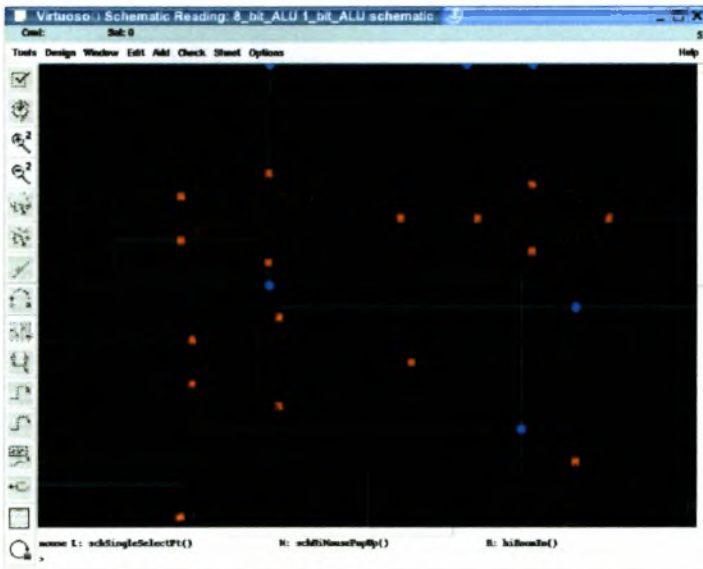


Παράδειγμα σχηματικού δικτύου διανομής ισχύος (4x4 κόμβοι) με τέσσερις πηγές ρευμάτων (sinks) και δύο πηγές τάσης (pads)

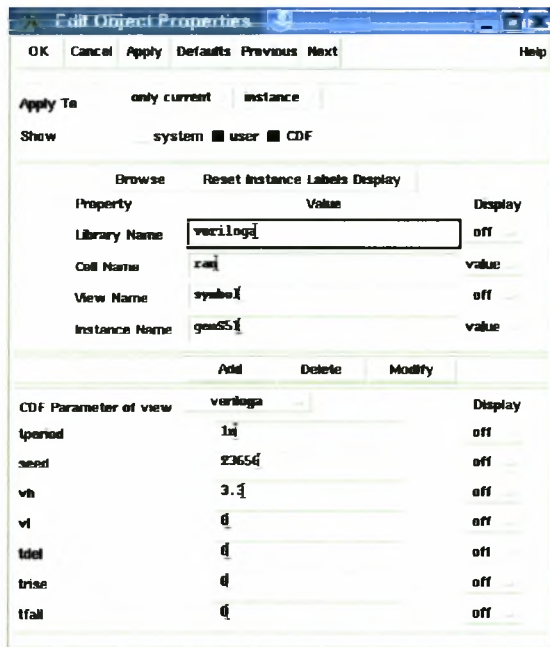
Τελικά το κύκλωμα που προκύπτει είναι το παρακάτω (τα blocks που βλέπουμε είναι το symbol view των 1 bit ALU τις οποίες χρησιμοποιήσαμε) :



Αφού κατεβούμε ιεραρχικά και εστιάζοντας σε κάποιο κομμάτι του κυκλώματος, μπορούμε να διακρίνουμε το symbolPT view που χρησιμοποιούμε για τα GSCLib3.0 standard cells :

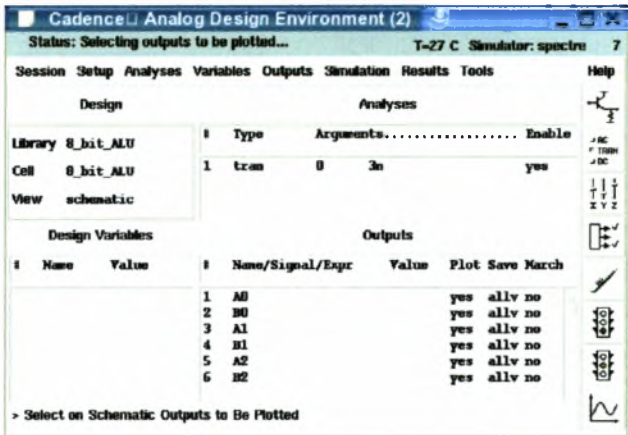


Αφού τελειώσει η σχεδίαση, πρέπει σε κάθε block να αναθέσουμε ένα pdevice. Αφού ολοκληρωθεί και αυτή η διαδικασία, ο σχεδιαστής πρέπει να τοποθετήσει σε κάθε είσοδο του κυκλώματος random generators (ορίζοντας την περίοδο, το random seed και τα χαρακτηριστικά του παλμού)

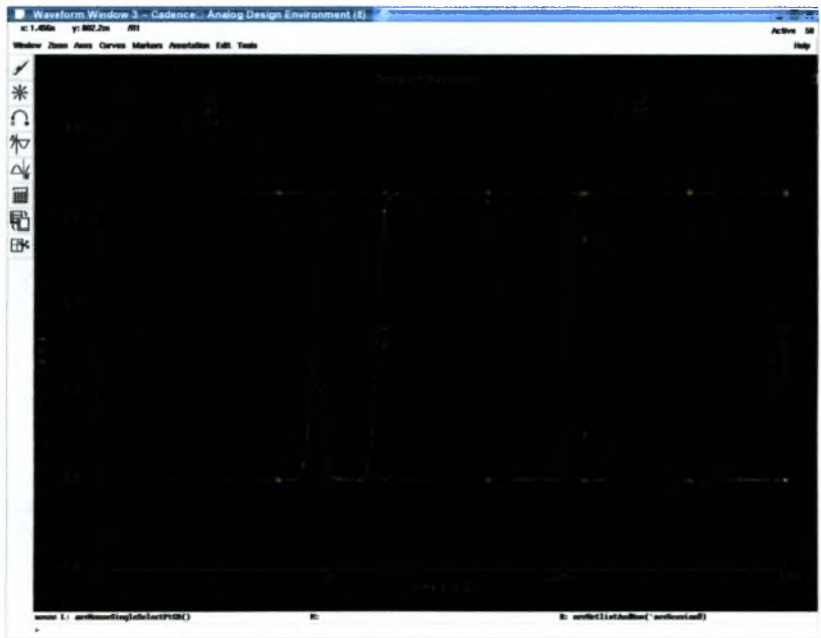


αλλά και να τοποθετήσει στις εξόδους του κυκλώματος πυκνωτές για τα φορτία εξόδου.

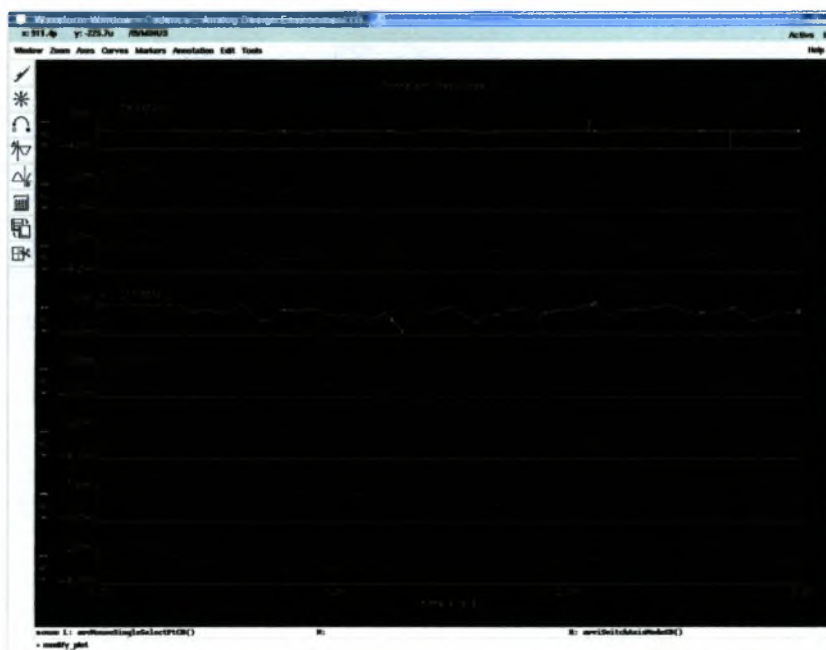
Αφού τα κάνουμε όλα αυτά, έρχεται η σειρά της προσομοίωσης. Για το σκοπό αυτό, θα χρησιμοποιήσουμε τον Spectre Simulator. Αφού ορίσουμε τον χρόνο που θα διαρκέσει η προσομοίωση, το είδος της ανάλυσης (στην περίπτωση μας θα χρησιμοποιήσουμε transient analysis) και ποιες εισόδους και εξόδους θέλουμε να φανούν στο τελικό γράφημα



τελικά παίρνουμε τα αποτελέσματα της προσομοίωσης :

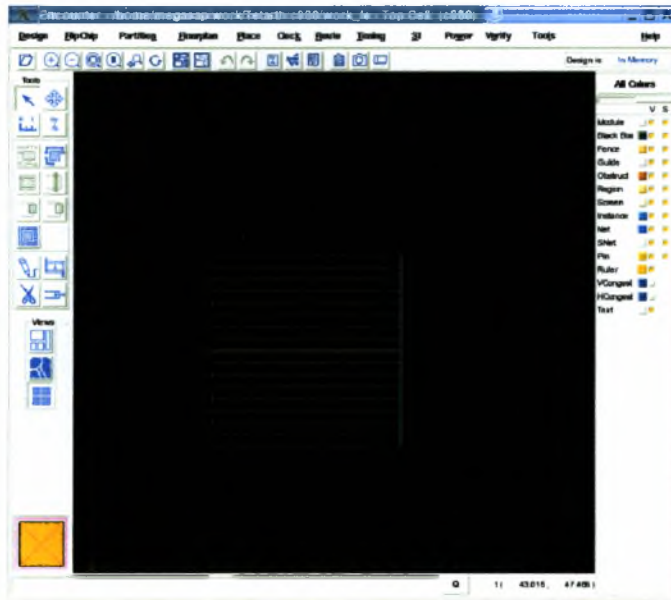


για να διαπιστώσουμε και γραφικά ότι το κύκλωμά μας λειτουργεί κανονικά ενώ μπορούμε να δούμε και τα την κυματομορφή των ρευμάτων στα pdevices

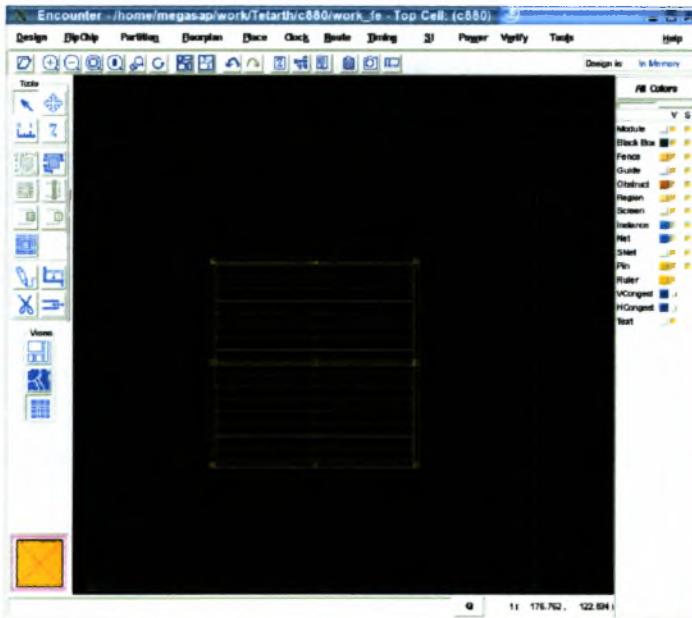


Από τις παραπάνω κυματομορφές των ρευμάτων, μπορούμε να βρούμε τα peak currents, τα οποία στην συνέχεια θα εισάγουμε στο IXtreme Engine για να υπολογίσει τα μέγιστα ρεύματα χειρότερης περίπτωσης σε κάθε καταβόθρα.

Στη συνέχεια, πρέπει ο σχεδιαστής μέσα από το περιβάλλον του SoC Encounter να σχεδιάσει το power grid για το κύκλωμά του, με τις προδιαγραφές που αυτός θέλει. Ο σχεδιαστής θα πρέπει να έχει το netlist του κυκλώματός του σε μορφή verilog καθώς επίσης και τα αρχεία βιβλιοθηκών της τεχνολογίας GSCLib που αυτός χρησιμοποιεί. Μετά την αρχικοποίηση των διάφορων απαραίτητων καταλόγων για το SoC Encounter (όπως αυτοί της αναφοράς όπου θα σώζονται τα όποια αποτελέσματα, της εξόδου όπου θα σώζονται τα όποια αρχεία .lef, .tlf, .def) και το φόρτωμα του αρχείου .conf όπου ουσιαστικά περιέχονται όλες οι πληροφορίες για το που βρίσκονται τα απαιτούμενα αρχεία που αναφέραμε παραπάνω παίρνουμε το εξής :



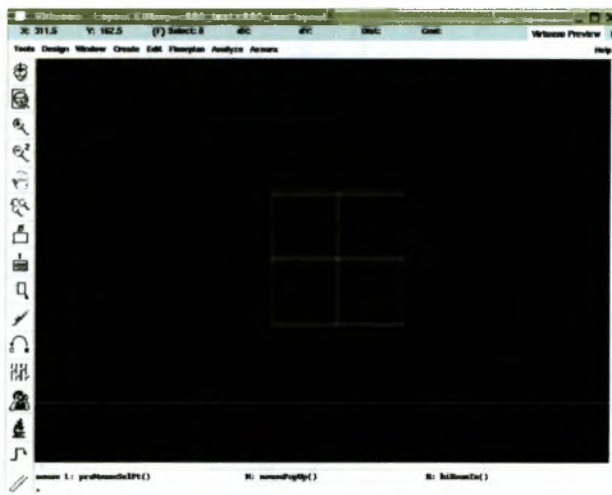
το οποίο ουσιαστικά δεν είναι τίποτα άλλο παρά το block του κυκλώματος μας με τα pins της τροφοδοσίας και της γείωσης. Έπειτα με τις εντολές *addRing* και *addStripe* σχεδιάζουμε το power grid που επιθυμούμε για το κύκλωμά μας. Οι παραπάνω εντολές δίνουν την ευχέρεια στον σχεδιαστή να κάνει το δίκτυο διανομής ισχύος όσο πυκνό ή όσο αραιό αυτός θέλει. Στο παράδειγμά μας, φτιάξαμε ένα power grid 2x2 :



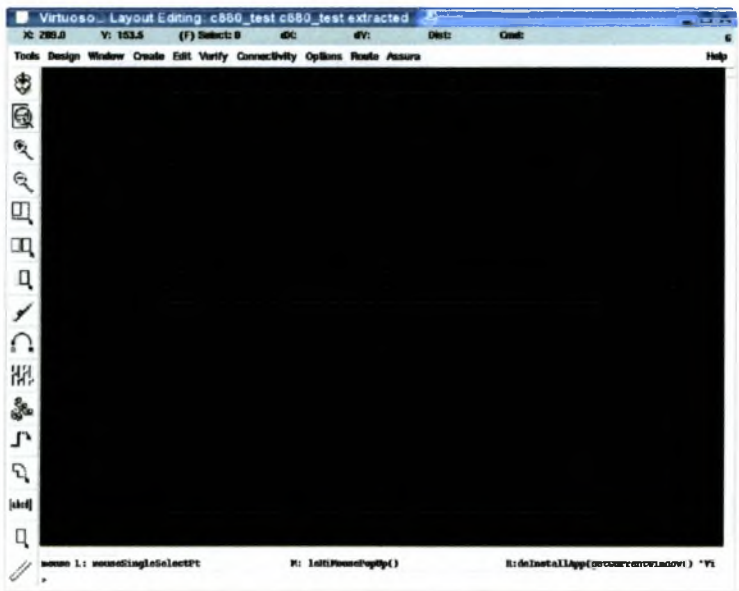
Στο εικονιζόμενο δίκτυο διανομής ισχύος, τα κάθετα layers είναι φτιαγμένα με Metal5 και τα οριζόντια με Metal6, ενώ όπως παρατηρούμε δεν βάλουμε νίαις μεταξύ των μετάλλων. Στο τέλος με την εντολή *defOut* σώζουμε το *.def* (Design Exchange Format) του κυκλώματός μας.

Με τον παραπάνω τρόπο, φτιάχνουμε δίκτυα διανομής τροφοδοσίας τα οποία δεν μπορούμε να τροποποιήσουμε εκ των υστέρων. Έστω ότι ο σχεδιαστής αποφάσισε ότι το πλάτος των γραμμών των μετάλλων θα έπρεπε να είναι μεγαλύτερο ή μικρότερο. Για να κάνει οποιαδήποτε αλλαγή, θα έπρεπε να επαναλάβει όλα τα βήματα που αναφέραμε παραπάνω με τις κατάλληλες αλλαγές στις διάφορες εντολές. Όπως εύκολα προκύπτει, κάτι τέτοιο δεν είναι εφικτό σε μεγάλου εύρους κυκλώματα.

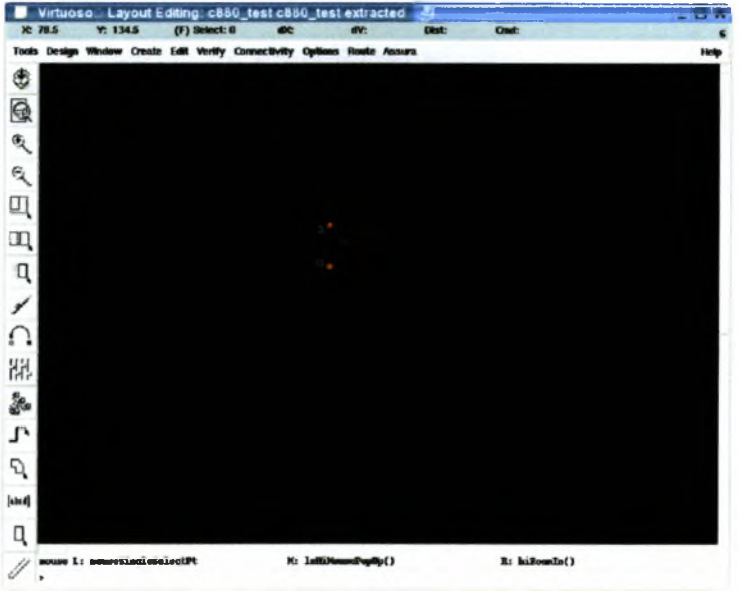
Μπορούμε πολύ εύκολα να ξεπεράσουμε τέτοιου είδους προβλήματα, απλά μετασχηματίζοντας τα κυκλώματα σε μία μορφή, στην οποία μπορούμε εύκολα να εφαρμόσουμε τις όποιες αλλαγές θέλουμε να κάνουμε. Αυτό που θα κάνουμε είναι να εισάγουμε το *.def* αρχείο που πήραμε από το SoC Encounter στο IC. Φτιάχνουμε μία νέα βιβλιοθήκη την οποία κάνουμε attach στην τεχνολογική βιβλιοθήκη *gpdk* (αυτό γίνεται γιατί για το κύκλωμα του παραδείγματος χρησιμοποιήσαμε την *GSCLib*). Έπειτα εισάγουμε το *.def* αρχείο (File->Import->DEF...), όπου ουσιαστικά ορίζουμε που βρίσκεται το *.def* αρχείο και το view name (χρησιμοποιήσαμε το layout στο παράδειγμά μας). Πλέον στην βιβλιοθήκη που φτιάξαμε, υπάρχει το layout view του κυκλώματός μας :



Το κύκλωμα στην παραπάνω του μορφή, μπορεί πλέον εύκολα να τροποποιηθεί με τη χρήση του Virtuoso Layout (XL). Το επόμενο βήμα είναι να πάρουμε το extracted view για το power grid μας με τη χρήση του DIVA, το οποίο ουσιαστικά μας δείχνει τις πραγματικές αντιστάσεις σε κάθε branch :



Αν εστιάσουμε σε κάποιο από τα branches, θα δούμε την αντίσταση που έχει υπολογίσει το DIVA :



Στο flow που περιγράψαμε παραπάνω, όπως παρατηρούμε, ο σχεδιαστής είναι «υποχρεωμένος» να εκτελέσει κάθε βήμα μεμονωμένα. Αν θέλει να ξαναεκτελέσει το flow, μετά από τυχόν αλλαγές που έκανε στο κύκλωμά του, θα πρέπει να επαναλάβει κάθε βήμα μεμονωμένα. Γίνεται αμέσως αντιληπτό ότι για κυκλώματα ευρείας κλίμακας κάτι τέτοιο είναι σχεδόν ανέφικτο.

Σε αυτό το σημείο θα παρουσιάσουμε ένα διαφορετικό flow, το οποίο στηρίζεται στην χρήση αυτοματοποιημένων δεσμών ενεργειών (scripts) για να επιτελέσουμε την ίδια ακριβώς εργασία. Περιληπτικά, τα βήματα του flow είναι τα παρακάτω :

1. Έχουμε το RTL design του ολοκληρωμένου μας (technology-independent) σε μορφή vhdl ή verilog το οποίο και κάνουμε synthesis χρησιμοποιώντας την βιβλιοθήκη GSCLib. Έπειτα σώζουμε το πλέον technology-dependent netlist το οποίο προκύπτει (σε verilog μορφή).
2. Εισάγουμε το παραγόμενο netlist από το προηγούμενο βήμα στο Virtuoso IC και ελέγχουμε το κύκλωμα για τυχόν σφάλματα.
3. Γίνεται χρήση της SKILL για να φτιάξουμε αυτόματα το testbench για το netlist και με διάφορες διεργασίες να ετοιμάσουμε το κύκλωμα μας για μία πρώιμη προσομοίωση
4. Γίνεται χρήση του Ultrasim ως προσομοιωτή. Ο Ultrasim είναι ο FastSPICE προσομοιωτής της Cadence για να κάνουμε verify το κύκλωμά μας
5. Γίνεται χρήση των peak currents από την προσομοίωση που έγινε στο προηγούμενο βήμα, τα οποία εισάγονται στην IXtreme Engine για να υπολογιστούν τα ρεύματα χειρότερης περίπτωσης
6. Χρήση εκ νέου της SKILL για την δημιουργία του power grid του κυκλώματος μας
7. Χρήση του Spectre simulator για την εκτέλεση dc ανάλυσης στο κύκλωμα μας. Μετά το πέρας της προσομοίωσης και επαλήθευσης, δημιουργείται η αναφορά για όλες τις τυχόν παραβιάσεις που αναφέρθηκαν. Ο Spectre Simulator της Cadence χρησιμοποιείται για

λεπτομερή ανάλυση σε επίπεδο τρανζίστορ (κάνοντας προσομοιώσεις σε επίπεδο SPICE).

Περιγραφή του Προβλήματος

Υποθέτουμε ότι το δίκτυο διανομής ισχύος (ή τουλάχιστον το κομμάτι του το οποίο χρειάζεται επαλήθευση) έχει εξαχθεί σαν ένα γραμμικό δίκτυο με διακλαδώσεις αντιστάσεων το οποίο έχει $q + p$ κόμβους, εκ των οποίων οι p κόμβοι είναι συνδεδεμένοι στην εξωτερική πηγή τροφοδοσίας μέσω power pads (τα οποία είναι είτε τοποθετημένα στην περιφέρεια του πλέγματος στην περίπτωση ενός wire-bond package, είτε είναι διασκορπισμένα μέσα σε όλο το πλέγμα στην περίπτωση της μεθοδολογίας C4), και οι εναπομείναντες q κόμβοι είναι χωρισμένοι σε n κόμβους καταβόθρες (sink nodes) (με πηγές ρευμάτων σε ένα εξωτερικό κόμβο γείωση – ground node) και $q - n$ εσωτερικούς κόμβους (για τους οποίους συνήθως $q \gg n$).

Σύμφωνα με την Τροποποιημένη Κομβική Ανάλυση – Modified Nodal Analysis (MNA), το $q \times 1$ διάνυσμα των στιγμιαίων τάσεων $\underline{U}(t)$ (συγκριτικά με τη γείωση) σε όλους τους εσωτερικούς και κόμβους καταβόθρες καθορίζεται από τα στιγμιαία ρεύματα που «αντλούνται» από τις καταβόθρες και τη συγκεκριμένη δομή του δικτύου, και δίδεται από την παρακάτω εξίσωση πινάκων :

$$\underline{G} \cdot \underline{U}(t) = -\underline{I}(t) + \underline{G} \cdot \underline{V}_{vdd} \quad (1)$$

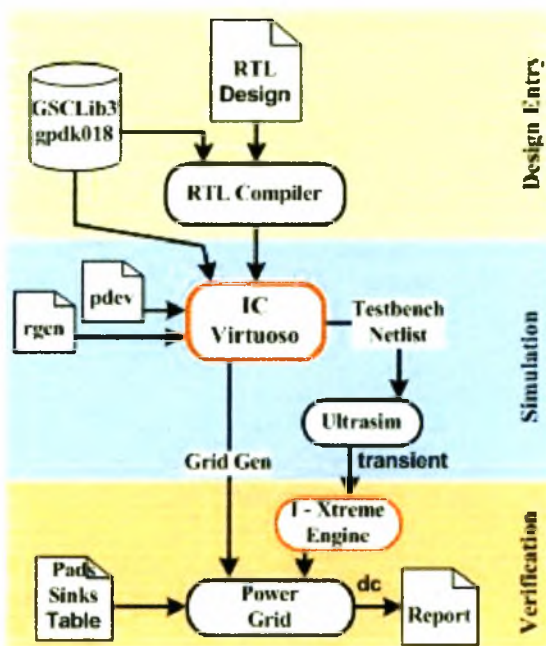
όπου το \underline{G} είναι ο $q \times q$ πίνακας διαγωγιμότητας (conductance) του δικτύου (σχηματιζόμενο από τις διαγωγιμότητες των διακλαδώσεων του δικτύου), $\underline{I}(t)$ είναι ένα $q \times 1$ διάνυσμα των διεγέρσεων των ρευμάτων στους κόμβους (με θετικά ρεύματα στη γείωση στους κόμβους καταβόθρες και μηδενικές καταχωρήσεις σε όλους τους εσωτερικούς κόμβους), και \underline{V}_{vdd} είναι άλλο ένα $q \times 1$ διάνυσμα με όλες τις καταχωρήσεις ίσες με την τάση τροφοδοσίας V_{dd} . Ορίζοντας σαν $\underline{V}(t) = \underline{V}_d - \underline{U}(t)$ την πτώση τάσης σε όλους τους κόμβους

μπορούμε να ξαναγράψουμε την παραπάνω εξίσωση του δικτύου σε μία μορφή που μπορεί να λυθεί κατευθείαν για τις τιμές των πτώσεων τάσεων :

$$\underline{G} \cdot \underline{V}(t) = \underline{I}(t) \tag{2}$$

Σχεδιαστική Ροή RTL-to-Grid

Η σχεδιαστική ροή RTL-to-Grid περιγράφεται από το παρακάτω διάγραμμα :



Το πρώτο βήμα είναι η σύνθεση του RTL Design (vhdl ή verilog) χρησιμοποιώντας την βιβλιοθήκη GSCLib3.0 (τεχνολογία gpdk018). Εκτελούμε μια δέσμη ενεργειών (script) για γρήγορη σύνθεση αφού είναι τυπικό για την δημιουργία testbench netlists. Μετά την σύνθεση αποθηκεύουμε το flat technology-dependent netlist της σχεδίασης (verilog).

Το δεύτερο βήμα είναι να εισάγουμε το flat netlist στο Cadence Virtuoso IC χρησιμοποιώντας την symbolPT (ακίδες ρεύματος και γείωσης) προβολή για τα standard cells της GSCLib3.0. Τα κελιά για το pdevice και την random

generator (γραμμένα σε verilog) υπάρχουν επίσης στην τρέχουσα βιβλιοθήκη.

Το επόμενο βήμα είναι να εκτελέσουμε την transient προσομοίωση της σχεδίασης. Στη σχεδιαστική μας ροή χρησιμοποιούμε την SKILL για την αυτοματοποιημένη παραγωγή του testbench netlist. Κατά τη διάρκεια της πρώτης διεργασίας χωρίζουμε τυχαία τη σχεδίαση σε blocks επιλεγμένου αριθμού πυλών (π.χ. 50). Σε κάθε block αναθέτουμε ένα pdevice. Η δεύτερη διεργασία τοποθετεί random generators σε κάθε είσοδο (ορίζοντας την περίοδο, το random seed και τα χαρακτηριστικά του παλμού), τα φορτία εξόδου και την αναφορά τάσης (V_{dd}) του testbench. Η τελευταία διεργασία δημιουργεί το ocean script για την transient ανάλυση (καθορισμός της διάρκειας της προσομοίωσης, του προσομοιωτή και ρύθμιση του προσομοιωτή). Ελέγχουμε τη σχεδίαση για συνεκτικότητα.

Το ocean script ελέγχει την προσομοίωση και αποθηκεύει τις κυματομορφές των ρευμάτων σε κάθε pdevice και τελικά τα μέγιστα ρεύματα σε κάθε κύκλο ρολογιού κάθε περιόδου ελέγχονται από άλλο script. Ο σχεδιαστής μπορεί επίσης να εισάγει την κυματομορφή στο Matlab για να ελέγξει τις συναρτήσεις κατανομής και πυκνότητας των ρευμάτων. Ο επιλεγμένος fast spice προσομοιωτής είναι ο Ultrasim της Cadence, ρυθμισμένο να δουλεύει σε digital fast mode για να μειώσουμε το χρόνο προσομοίωσης.

Αφού τα μέγιστα ρεύματα έχουν εξαχθεί από την προσομοίωση, μία εσωτερική διεργασία καλεί την IXtreme Engine για να υπολογίσει τα μέγιστα ρεύματα χειρότερης περίπτωσης σε κάθε καταβόθρα. Η μηχανή είναι μία διεργασία σε C η οποία εκτελεί όλους τους στατιστικούς υπολογισμούς. Υποθέτουμε ότι τα μέγιστα ρεύματα εμφανίζονται ταυτόχρονα σε κάθε καταβόθρα.

Επειδή η μεθοδολογία δεν υποστηρίζει πληροφορία από τα layout views, το δίκτυο διανομής ισχύος παράγεται από μία διεργασία SKILL. Το cell view του δικτύου διανομής είναι «σχηματικό», το μέγεθος είναι παραμετροποιήσιμο και χρησιμοποιούμε pdevice για να αναπαραστήσουμε τις αντιστάσεις σε

κάθε διακλάδωση. Σαν είσοδο στο δίκτυο διανομής χρησιμοποιούμε ένα sinks-and-pads πίνακα κατανομής για να δηλώσουμε την τοποθέτηση των πηγών ρεύματος και τάσεων σε αυτό.

Το τελικό βήμα είναι η dc ανάλυση του δικτύου διανομής (προσομοιωτής Spectre). Μία διεργασία σε ocean εκκινεί την ανάλυση και επίσης επιτελεί την επαλήθευση σε κάθε καταβόθρα. Στο τέλος δημιουργείται μία αναφορά και όλες οι παραβιάσεις καταγράφονται.

Πειραματικά Αποτελέσματα

Τα κυκλώματα που επιλέχθηκαν για την πειραματική επαλήθευση της μεθόδου είναι από την σουίτα δοκιμής επιδόσεων ISCAS85. Το κύκλωμα υλοποιήθηκε με την τεχνολογία GPDK 0.18 um με 6 γραμμές μετάλλου, αντίσταση φύλλου (sheet resistance) 0.1 ohm/square και πηγή τροφοδοσίας $V_{dd}=3.3V-2.7V$. Τα κυκλώματα διασπάστηκαν σε ένα τυχαίο αριθμό λειτουργικών block περιέχοντας 50 standard cells το καθένα, και τοποθετήθηκαν χρησιμοποιώντας διαφορετικά προφίλ floorplanning. Η εκτίμηση των μέγιστων ρευμάτων σε κάθε block εκτελείται πρώτα και μετά τα εξαγόμενα αποτελέσματα καταγράφονται στον Πίνακα I. Η κύρια πληροφορία περιλαμβάνει το όνομα του κυκλώματος στην στήλη 1, τον αριθμό των blocks στην στήλη 2, τον αριθμό των διανυσματικών ζευγών που χρησιμοποιήσαμε για την εκτίμηση στην στήλη 3, το αναγνωριστικό του ρεύματος του block, το μέγιστο ρεύμα στο δείγμα που χρησιμοποιήθηκε για την εκτίμηση, την εκτίμηση του πραγματικού μέγιστου ρεύματος στην στήλη 6. Μετά την dc ανάλυση τα αποτελέσματα για την πτώση τάσης σε κάθε καταβόθρα (A:net_A3, B:net_B2, C:net_C2, D:net_C1 και E:net_Do, βλ. Figure 4) του κυκλώματος c1355 καταγράφονται στον Πίνακα II.

| Circuit | # blocks | # vectors | I_{max} | Sample maximum (mA) | Estimated maximum (mA) |
|----------------------|----------|-----------|-------------|---------------------|------------------------|
| <i>c1355</i> 3.3V | 5 | 5000 | I_{max_A} | 27.24 | 34.73 |
| | | | I_{max_B} | 24.85 | 32.69 |
| | | | I_{max_C} | 28.58 | 36.81 |
| | | | I_{max_D} | 12.31 | 16.15 |
| | | | I_{max_E} | 20.37 | 28.06 |
| <i>c1355</i> 2.7V | 5 | 5000 | I_{max_A} | 18.51 | 23.42 |
| | | | I_{max_B} | 15.55 | 21.61 |
| | | | I_{max_C} | 17.27 | 22.39 |
| | | | I_{max_D} | 8.34 | 10.99 |
| | | | I_{max_E} | 11.46 | 16.85 |
| <i>c880</i> 3.3V | 8 | 4000 | I_{max} | 11.66 | 18.08 |
| <i>c2670</i> 3.3V | 13 | 4000 | I_{max} | 14.82 | 21.4 |
| <i>c7552</i> 3.3V | 42 | 4000 | I_{max} | 8.32 | 13.02 |

Πίνακας I Αποτελέσματα για τις εκτιμώμενες τιμές των μέγιστων ρευμάτων

| Circuit | | Maximum IR-drop | | | | | |
|----------|----------|-----------------|-------|-------|-------|-------|-------|
| c1355 | | V_{dd} | A | B | C | D | E |
| Topology | WireBond | 3.3V | 27.8m | 22.9m | 20.6m | 20.1m | 24.2m |
| | | 2.7V | 17.7m | 14.3m | 12.5m | 12.2m | 14.4m |
| | C4 | 3.3V | 18.8m | 11.3m | 11.9m | 8.4m | 14.8m |
| | | 2.7V | 11.9m | 6.8m | 6.9m | 4.7m | 8.5m |

Πίνακας II Αποτελέσματα για το IR-Drop σε κάθε κόμβο καταβόθρα

Παράρτημα Α

Αρχεία DEF και PDEF

Στο προηγούμενο κεφάλαιο είδαμε πως χρησιμοποιώντας το DEF (Design Exchange Format) αρχείο το οποίο πήραμε από το SoC Encounter και εισάγοντας το στο IC καταφέραμε και πάρουμε το extracted view του power grid μας. Έτσι κατέστη δυνατό να υπολογίσουμε τις παρασιτικές αντιστάσεις σε κάθε branch του power grid μας.

Με βάση μία πρώτη έρευνα που έγινε για το πόσο χρήσιμα είναι τα DEF αρχεία καταλήξαμε στα εξής:

| SYNOPTISYS | DEF SUPPORT |
|--------------------|--------------------|
| IC COMPILER | X |
| FLOORPLAN COMPILER | X |
| PHYSICAL COMPILER | X |
| MILKYWAY | X |
| CADENCE | |
| SOC ENCOUNTER | X |
| ICFB | X |
| | |

Περισσότερες πληροφορίες για την σύνταξη των αρχείων DEF είναι διαθέσιμη στο `lefdefref.pdf` στον φάκελο docs του SOC4.2 ή του IC.

Ακολουθεί ένα παράδειγμα DEF αρχείου:

Χειρισμός δικτύων ισχύος στα DEF και PDEF αρχείων

Στα DEF αρχεία μας πρέπει να περιγράψουμε επακριβώς τις γραμμές τροφοδοσίας.

Στο παράδειγμα του DEF αρχείου που ακολουθεί ισχύουν τα εξής :

1. Υπάρχουν δύο ειδικά δίκτυα (special nets).
2. Το δίκτυο της γείωσης GND συνδέεται με όλα τα κελιά δια μέσου των GND ακίδων τους
3. Τα ROUTED και NEW περιγράφουν κομμάτια μετάλλου που σχηματίζουν τη δομή του δικτύου ισχύος. Αυτό το δίκτυο βρίσκεται στο m1 μήκους 100 μονάδων.
4. Το δίκτυο της τροφοδοσίας VDD συνδέεται με όλα τα κελιά δια μέσου των VDD ακίδων τους.
5. Τα ROUTED και NEW περιγράφουν κομμάτια μετάλλου που σχηματίζουν τη δομή του δικτύου ισχύος. Αυτό το δίκτυο βρίσκεται στο m2 μήκους 100 μονάδων.
6. Τέλος της περιγραφής των ειδικών δικτύων.

```
#  
# Παράδειγμα τμήματος από DEF αρχείο δείχνοντας την περιγραφή των  
# γραμμών τροφοδοσίας  
#
```

```
SPECIALNETS 2 ;  
- GND ( * GND )  
+ ROUTED m1 100 ( -200 -200 ) ( 200 * )  
NEW m1 100 ( -200 200 ) ( 200 * ) + USE GROUND ;  
- VDD ( * VDD )  
+ ROUTED m2 100 ( -200 -200 ) ( 200 * )  
NEW m2 100 ( -200 200 ) ( 200 * ) + USE POWER ;  
END SPECIALNETS
```

Στην περιγραφή στο DEF αρχείο, πρέπει να προσδιορίσουμε τα δίκτυα ισχύος ως εξής :

- Το δίκτυο της γείωσης GND πρέπει να έχει τη δήλωση + *USE GROUND*
- Το δίκτυο της τροφοδοσίας VDD net πρέπει να έχει τη δήλωση + *USE POWER*

Τα αρχεία PDEF έχουν παρόμοιες PNET απαιτήσεις με αυτό. Η δομή (TYPE) Απαιτείται για να προσδιορίσουμε τα δίκτυα ισχύος σαν pwr ή gnd.

```
(PNET GND
(TYPE gnd)
(ROUTE
(LAYER_WIDTH 3 100)
(3 ( -200 -200 ) ( 200 * ) )
)
(ROUTE
(LAYER_WIDTH 3 100)
( 3 ( -200 200 ) ( 200 * ) )
)
)
(PNET VDD
(TYPE pwr)
(ROUTE
(LAYER_WIDTH 5 100)
(5 ( -200 -200 ) ( * 200 ) )
)
(ROUTE
(LAYER_WIDTH 5 100)
( 5 ( 200 -200 ) ( * 200 ) )
)
)
```

Παράρτημα Β

Στο παρόν παράρτημα, θα δούμε τις αυτοματοποιημένες δέσμες ενεργειών (scripts) που χρησιμοποιήσαμε στο κεφάλαιο 3.

read_taps.il

Η παραπάνω αυτοματοποιημένη ενέργεια γραμμένη σε SKILL, φορτώνει τις τιμές για τα ρεύματα και τις τάσεις στα καθορισμένα nets του κυκλώματος.

```
; load current sources locations
```

```
i_taps=  
list(  
'("net_A3" "200m")  
'("net_B3" "120m")  
'("net_D3" "230m")  
'("net_C0" "205m")  
)
```

```
; load voltage sources locations
```

```
v_taps=  
list(  
'("net_A0" 3.3)  
'("net_C3" 3.3)  
)
```


sim_proc.il

Η παραπάνω αυτοματοποιημένη ενέργεια γραμμένη σε SKILL, φτιάχνει το OCEAN script και εκτελεί την προσομοίωση με τις επιθυμητές παραμέτρους.

```
procedure( SimGen( lib cell stime sstep sper)

let( (libi celli)

libi =lib
celli = cell
newlib = strcat( libi "_sim")
newcell1 = strcat( celli "_try1")
newcell2 = strcat( celli "_try2")

cv = dbOpenCellViewByType(newlib newcell2 "schematic" ""
'a)
```

Ένα παράδειγμα χρήσης της παραπάνω διεργασίας :

```
SimGen( "iscas89" "c880" "10n" "0.01" "2.5n")
```

dc_proc.ocn

Η παραπάνω αυτοματοποιημένη ενέργεια γραμμένη σε OCEAN, εκτελεί την dc ανάλυση με τις επιθυμητές παραμέτρους

```
procedure( dcIRdrop( lib cell voltRef IRDper )
```

```
let( (libi celli)
```

```
libi =lib
```

```
celli = cell
```

```
newlib = strcat( libi "_sim")
```

Ένα παράδειγμα χρήσης της παραπάνω διεργασίας :

```
dcIRdrop( "iscas89" "test2" 3.3 0.1)
```

veriloga.va

Το παραπάνω αρχείο γραμμένο σε veriloga, αποτελεί ουσιαστικά τους random generators που χρησιμοποιήσαμε

```
//      FUNCTION: Random Bitstream
//      VERSION: $Revision: 2.6 $
//      AUTHORS: Modelwriter Standard Library
//              Cadence Design Systems, Inc.
//
// GENERATED BY: Cadence Modelwriter 2.25
//              ON: Mon Oct 11 10:00:02 CEST 2004
//
// Description:      Random Single Bit Genetator
//
// PARAMETERS:
//      seed = Initial seed for random number generator
//      tdel = Delay into Period which transistion begins
[Sec]
//      tfall = Falling transition time of data stream
[Sec]
//      tperiod = Period of data stream [Sec]
//      trise = Rising transition time of data stream
[Sec]
//      vh = voltage to output for a Logic 1 [V]
//      vl = voltage to output for a Logic 0 [V]
//

`include "discipline.h"
`include "constants.h"

module rand_bit_stream (vout1);
    output vout1;
    electrical vout1;
    parameter real tperiod = 0.0n from (0:inf);
    parameter integer seed = 23133;
    parameter real      vh = 3.3;
    parameter real      vl = 0.0;
```

```

parameter real    tdel = 0.0n from [0:tperiod);
parameter real    trise = 0.0n from [0:tperiod);
parameter real    tfall = 0.0n from [0:tperiod);

real next, vout_val;
integer bit, rand;
integer seed_int;
analog begin
    @ ( initial_step( "ac","tran","dc","xf" )) begin
        next = $abstime + tperiod;
        seed_int=seed;
    end

    $bound_step(tperiod);
    rand = $random(seed_int);
    bit = abs(rand %2);

    @ ( timer(next)) begin
        vout_val = (bit == 1 ? vh : vl);
        next = next + tperiod;
    end

    v(vout1) <+ transition(vout_val, tdel, trise,
tfall);
end
endmodule

```

Bιβλιογραφία

E-books

- [1] **VoltageStorm Transistor-Level PGS User Guide**
Product Version 4.0.2 March 2004
- [2] **Synopsys Insight : A practical Methodology Calculates IR Drop Targets for SoCs**
- [3] **Generic Standard Cell Library IP Block Implementation Tutorial**
- [4] **SKILL Language User Guide**
Product Version 06.30 September 2003
- [5] **SKILL Language Reference**
Product Version 06.30 September 2003
- [6] **OCEAN Reference**
Product Version 5.0 October 2003
- [7] **Encounter User Guide**
Product Version 3.3 January 2004
- [8] **DIVA Reference**
Product Version 5.0 July 2003
- [9] **UltraSim User Guide**
Product Version 4.1 March 2004
- [10] **Spectre Circuit Simulator Reference**
Product Version 5.0 September 2003

Δικτυακοί Τόποι

- [1] <http://www.cadence.com>

Papers

[1] An RTL-to-Grid Design Flow for Power Grid Verification Based On a Statistical Estimation Engine

D. P. Karampatzakis, N.E. Evmorfopoulos, M. K. Tsiampas and G. I. Stamoulis Dept. of Computer and Communication Engineering
University of Thessaly



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ



004000085802