

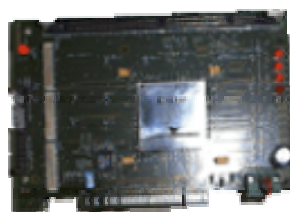


Πολυτεχνείο Κρήτης
Εργαστήριο Μικροεπεξεργαστών και
Υλικού -MHL



Διπλωματική Εργασία: Ζωγόπουλου Σωτήριου

‘ΣΧΕΔΙΑΣΗ & ΥΛΟΠΟΙΗΣΗ ΠΕΙΡΑΜΑΤΙΚΗΣ
ΠΛΑΤΦΟΡΜΑΣ ΓΙΑ ΕΝΕΡΓΑ ΔΙΚΤΥΑ,
ΒΑΣΙΣΜΕΝΗΣ ΣΕ ΑΝΑΔΙΑΤΑΣΣΟΜΕΝΗ ΛΟΓΙΚΗ’



Επιβλέποντες Καθηγητές

Δόλλας Απόστολος
Πνευματικάτος Διονύσιος
Διγαλάκης Βασίλειος

Ευχαριστίες

Ευχαριστώ πάνω από όλα τους γονείς μου για την συμπαράστασή τους σε κάθε βήμα μου κατά τη διάρκεια των σπουδών μου και της διπλωματικής μου εργασίας. Αισθάνομαι την ανάγκη να πω ότι η βοήθεια που μου πρόσφεραν καθημερινά, κατά τη διάρκεια των σπουδών μου, αποτέλεσε πολύ σημαντικό παράγοντα στην επιτυχή ολοκλήρωση των σπουδών μου.

Επίσης, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας, Κ. Δόλλα Απόστολο, που μου έδωσε την ευκαιρία να συμμετέχω στο ερευνητικό του πεδίο, μεταφέροντάς μου τεχνογνωσία και πολύτιμες γνώσεις. Ευχαριστώ τον Κ. Πνευματικάτο Διονύσιο και Κ. Διγαλάκη Βασίλειο, καθώς και τους Κυρίους Κιμιωνή Μάρκο και Λιγεράκη Μιχάλη για τα θέματα υλοποίησης και σχεδίασης αντίστοιχα. Θα ήθελα επίσης να αναφερθώ στην πολύτιμη βοήθεια που είχα από το Κ. Καλοκαιρινό Γεώργιο (ΙΤΕ-Ηρακλείου) και στο Κ. Τσικουδί Νικόλαο, υπάλληλο της Intracom για την άψογη συνεργασία που είχαμε.

Τέλος αισθάνομαι την ανάγκη να εκφράσω την ευγνωμοσύνη μου στους φίλους και συνεργάτες μου, για τις πολύτιμες συμβουλές τους καθώς και για την ψυχολογική υποστήριξη που μου παρείχαν.

Με εκτίμηση

Στους αγαπητούς μου γονείς.

ΠΕΡΙΕΧΟΜΕΝΑ

| | | |
|-------|---|----|
| 1 | Εισαγωγή..... | 5 |
| 2 | Σχετικές Εργασίες & Παρεμφερή Προϊόντα..... | 7 |
| 3 | Αρχιτεκτονική Πλατφόρμας..... | 10 |
| 3.1 | Σύνθεση Συστήματος..... | 10 |
| 3.3 | Κατάτμηση Συστήματος..... | 13 |
| 3.4 | Κατανομή Πόρων..... | 16 |
| 3.4.1 | PCI..... | 17 |
| 3.4.2 | SRAM..... | 18 |
| 3.4.3 | SDRAM..... | 20 |
| 3.4.4 | UTOPIA..... | 21 |
| 3.5 | Χρονισμός της πλατφόρμας..... | 22 |
| 3.6 | Προγραμματισμός της FPGA..... | 26 |
| 4 | Σχεδίαση Πολυεπίπεδου Τυπωμένου Κυκλώματος..... | 27 |
| 4.1 | Γενικά χαρακτηριστικά..... | 28 |
| 4.2 | Επιλογή του πλήθους των επιπέδων..... | 30 |
| 4.3 | Επιλογή και χρήση παθητικών εξαρτημάτων και αγωγών..... | 36 |
| 4.3.1 | Πυκνωτές, αντιστάσεις & Φερριτικοί Πυρήνες..... | 36 |
| 4.3.2 | Επιλογή του Package των Εξαρτημάτων..... | 39 |
| 4.3.3 | Board to Board connectors..... | 41 |
| 4.4 | Δημιουργία netlist..... | 43 |
| 4.5 | Placement, Rules, Route and DRC..... | 46 |
| 4.5.1 | Τοποθέτηση εξαρτημάτων & δημιουργία netlist..... | 46 |
| 4.5.2 | Κανόνες σχεδίασης..... | 49 |
| 4.5.3 | Δρομολόγηση σημάτων (Routing)..... | 53 |
| 4.6 | DRC..... | 59 |
| 4.7 | Output Files..... | 61 |
| 5 | Υλοποίηση και Δοκιμή Συστημάτων..... | 62 |
| 6 | Συμπεράσματα..... | 67 |
| | Παράρτημα Α..... | 68 |
| | ΑΝΑΦΟΡΕΣ..... | 71 |

ΠΕΡΙΛΗΨΗ

Στα πλαίσια αυτής της διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε μια πειραματική πλατφόρμα για ενεργά δίκτυα, το PLATO. Η ανάγκη της παροχής υπηρεσιών στους κόμβους των μελλοντικών δικτύων, απαιτεί επιπλέον υπολογιστική ισχύ σε αυτούς. Το PLATO προσεγγίζει αυτό το χώρο μέσω Hardware και πιο συγκεκριμένα με τη χρήση αναδιατασσόμενης λογικής (FPGAs), με τη δυνατότητα προσαρμογής σε ATM και Ethernet δίκτυα.

Η υλοποίηση που παρουσιάζεται περιλαμβάνει, 4x4 αμφίδρομες συνδέσεις με ATM δίκτυο, μια μεγάλη FPGA, 256 Mbytes SDRAM για την αποθήκευση των δεδομένων, PCI θύρα για την επικοινωνία με τον γενικής χρήσης επεξεργαστή ενός υπολογιστή, καθώς και διάφορες θύρες επέκτασης για μελλοντική χρήση. Στα πλαίσια αυτής της εργασίας, θα αναφερθεί η διαδικασία σχεδίασης ενός συστήματος αυξημένης πολυπλοκότητας καθώς και διάφορες μεθοδολογίες που αναπτύχθηκαν για την επίλυση σχεδιαστικών προβλημάτων, αλλά και για τη μείωση του χρόνου υλοποίησης.

Στα πλαίσια αυτής της διπλωματικής εργασίας σχεδιάστηκαν δυο τυπωμένα κυκλώματα 8 και 10 επιπέδων αντίστοιχα, με κύριο ολοκληρωμένο κύκλωμα αυτό των FPGAs, με πακέτο επιφανειακής στήριξης BGA 560 και 652 ακροδεκτών αντίστοιχα. Επίσης σχεδιάστηκε μια daughterboard δυο επιπέδων η οποία περιλαμβάνει μια διάταξη από SRAMs. Όλα τα κυκλώματα κατασκευάστηκαν από την INTRACOM και συγκολλήθηκαν στο ITE (Ινστιτούτο Τεχνολογίας και Έρευνας - Ηρακλείου) και δοκιμάστηκαν από το ΠΚ και το ITE και λειτουργούν πλήρως. Τέλος, στην εργασία αυτή θα παρουσιαστούν οι διαδικασίες δοκιμής των συστημάτων.

ΚΕΦΑΛΑΙΟ 1^ο

1 Εισαγωγή.

Τα ενεργά δίκτυα είναι κατά κάποιο τρόπο η σημερινή αντίληψη των έξυπνων δικτύων. Οι τομείς με τους οποίους ασχολούνται έχουν να κάνουν κυρίως με την επεξεργασία των δεδομένων που στέλνονται δια μέσω αυτών, καθώς και με θέματα όπως ‘Ασφάλεια δεδομένων μέσω κρυπτογράφησης’ και ‘Προσαρμοστικότητα σύμφωνα με το φορτίο μετάδοσης μεταξύ των προγραμματιζόμενων κόμβων τους’. Η ευελιξία των ενεργών δικτύων, μέσω κατάλληλου προγραμματισμού, δημιουργεί την αναγκαιότητα της ύπαρξης κάποιας υπολογιστικής ισχύος σε κάθε κόμβο τους. Τυπικά, κάθε κόμβος ενός δικτύου παρέχει την δυνατότητα της δρομολόγησης των πακέτων με βάση τη πληροφορία που περιέχεται στο header του κάθε πακέτου. Πηγαίνοντας ένα βήμα πιο μπροστά, τα ενεργά δίκτυα μπορούν να χρησιμοποιούν το payload του κάθε πακέτου για τη δρομολόγησή του, ‘ισορροπώντας’ κατά κάποιο τρόπο το φορτίο στο δίκτυο ή να επεξεργάζονται το payload κατάλληλα, καθώς το πακέτο μεταφέρεται από κόμβο σε κόμβο.

Με ελάχιστες εξαιρέσεις, σχεδόν όλα τα ερευνητικά προγράμματα σε ενεργά δίκτυα έχουν μελετηθεί και αναπτυχθεί με κατάλληλο λογισμικό, έχοντας αντιληφθεί σωστά το γεγονός ότι τα δίκτυα του μέλλοντος θα περιέχουν τεράστια υπολογιστική ισχύ στους κόμβους τους, αντίστοιχη και κυρίως συμβατή με αυτή των σημερινών επεξεργαστών υψηλής τεχνολογίας. Και ενώ μια τέτοια προσέγγιση είναι ιδανική για τη υλοποίηση καινούριων ιδεών, αφήνει ενδιαφέροντες τομείς όπως την επεξεργασία του payload των πακέτων σε πραγματικό χρόνο. Κάτι τέτοιο θα μπορούσε να είναι εξαιρετικά σημαντικό για εφαρμογές όπως:

1. ‘Κατανομή του φορτίου σε πραγματικό χρόνο για e-commerce servers’,
2. ‘Αναγνώριση φωνής σε πραγματικό χρόνο για v-commerce servers’,
3. ‘Βελτίωση πρωτοκόλλου’.

Για να έχουμε τη δυνατότητα να πειραματιστούμε με τέτοιου είδους εφαρμογές αναπτύξαμε μια νέα επαναπρογραμματιζόμενη πλατφόρμα για ATM δίκτυα, το PLATO.

Περιγραφή των ενοτήτων.

Στο 2^ο Κεφάλαιο θα αναφερθούμε σε σχετικές εργασίες που έχουν αναπτυχθεί στο χώρο των Ενεργών Δικτύων, καλύπτοντας τις διαφορετικές προσεγγίσεις καθώς και διάφορα παρεμφερή προϊόντα παρόμοιας αρχιτεκτονικής με του PLATO.

Στο 3^ο Κεφάλαιο θα παρουσιαστεί η αρχιτεκτονική του συστήματος και η διαδικασία κατάτμησής του σε υποσυστήματα και θα γίνει αναλυτική αναφορά σε κάθε ένα από αυτά.

Στο 4^ο Κεφάλαιο θα περιγραφεί η διαδικασία σχεδίασης ενός πολυεπίπεδου τυπωμένου κυκλώματος καθώς και διάφορες μεθοδολογίες επίλυσης σχεδιαστικών προβλημάτων όπως επιλογή του κατάλληλου πλήθους επιπέδων και μεθοδολογίας κατασκευής του netlist.

Στο 5^ο Κεφάλαιο θα γίνει αναφορά στη υλοποίηση και στον ποιοτικό έλεγχο του συστήματος, με τη χρήση πειραματικών εφαρμογών.

Στο 6^ο Κεφάλαιο θα αναφερθούμε στα συμπεράσματα που προκύπτουν από την ανάλυση και λειτουργικότητα του PLATO.

ΚΕΦΑΛΑΙΟ 2^ο

2 Σχετικές Εργασίες & Παρεμφερή Προϊόντα

Στο χώρο της έρευνας των ενεργών δικτύων και πιο συγκεκριμένα, στη σχεδίαση της αρχιτεκτονικής των κόμβων τους, υπάρχουν δυο διαφορετικές προσεγγίσεις. Η ‘In-band’, και η ‘Out-of-band’ node programming. Στην In-band ή αλλιώς προσέγγιση μέσω ολοκληρωμένων κυκλωμάτων, κάθε πακέτο περιλαμβάνει ένα μικρό πρόγραμμα και την κατάλληλη πληροφορία του. Όταν αυτό φτάσει στον κατάλληλα σχεδιασμένο κόμβο, εκτελείται το πρόγραμμα, χρησιμοποιώντας τα απαραίτητα δεδομένα που περιέχονται στο ίδιο το πακέτο. Στην Out-of-band προσέγγιση ή αλλιώς διακριτή προσέγγιση, σε κάθε κόμβο του δικτύου ‘φορτώνεται’ και εκτελείται κατάλληλος κώδικας, υλοποιώντας την επιθυμητή εφαρμογή. Σε αντίθεση με την In-band, στην Out-of-band προσέγγιση απαιτείται η αρχικοποίηση, αλλά και η αναβάθμιση των δυνατοτήτων του κάθε κόμβου. Λόγω του μεγάλου εύρους των εφαρμογών των ενεργών δικτύων οι δυο αυτές μέθοδοι τείνουν στην ενοποίηση τους.

Η ανάπτυξη των δυναμικών δικτύων ‘ενθαρρύνει’ την εξέλιξη εφαρμογών, βασισμένων σε ολοκληρωμένα κυκλώματα, χωρίς την υποστήριξη εκτεταμένου λειτουργικού συστήματος και γλώσσας υποστήριξης των ενεργών κόμβων. Στην αντίθετη κατεύθυνση υπάρχει πολύ μεγαλύτερη ευελιξία με τη χρήση εφαρμογών βασισμένων σε κατάλληλο λογισμικό, έχοντας όμως το κόστος της επιπρόσθετης καθυστέρησης.

Το ANTS είναι ερευνητικό πρόγραμμα που αναπτύχθηκε στο Τεχνολογικό Ίδρυμα της Μασαχουσέτης. Κάθε πακέτο περιέχει τα απαραίτητα δεδομένα, καθώς και μια αναφορά σε μια Java εφαρμογή που πρέπει να τοποθετηθεί στο κόμβο, αν ήδη δεν είναι παρούσα πριν αρχίσει η επεξεργασία του. Είναι απαραίτητο να εγκατασταθεί το ANTS στον ενεργό κόμβο για να εκτελέσει τη διαδικασία που θα του υποδείξει το περιεχόμενο του πακέτου, καθώς επίσης και να αναλάβει θέματα ασφάλειας μέσα από τεχνικές όπως ‘sandboxing’ και ‘Java bytecode verification’. Αντίστοιχα ερευνητικά προγράμματα είναι το BBN και το ‘SwitchWare’ στο Upenn και Bellcore.

Στο CANES ερευνητικό πρόγραμμα του πανεπιστημίου Georgia Tech και του Πανεπιστημίου του Κεντάκι, η υλοποίηση των ενεργών κόμβων υλοποιείται με την χρήση του λειτουργικού συστήματος του κόμβου (Node Operating System-NodeOS) και των Execution Environments (EE) που λαμβάνουν θέση σε υψηλότερο επίπεδο από ότι το NodeOS.

Από όσο μας είναι γνωστό, μόνο δυο ερευνητικά προγράμματα ασχολήθηκαν με την υλοποίηση ειδικών πλατφόρμων ανάπτυξης για ενεργά δίκτυα. Το ANN του Πανεπιστημίου της Ουάσινγκτον σχεδίασε μια hardware εφαρμογή. Η πλατφόρμα τους χρησιμοποιεί έναν γενικής χρήσης επεξεργαστή, μια μεγάλη FPGA (άνω των 100,000 πυλών), 64 Mb μνήμη και ένα ATM Port Interconnect Controller (APIC) ολοκληρωμένο, συνδεδεμένα στο PCI ενός υπολογιστή. Το Programmable Protocol Processing Pipeline (P4) ερευνητικό πρόγραμμα υλοποίησε μια πλατφόρμα αποτελούμενη από μια ομάδα από FPGAs της εταιρίας Altera και της οικογένειας FLEX8000, οι οποίες χρησιμεύουν σαν υπολογιστικές μονάδες και ενός 'switching array', που επιλέγει ποια από τις FPGAs θα συμμετέχουν στη διαδικασία.

Οι εφαρμογές των ενεργών δικτύων περιλαμβάνουν: Active reliable multicast, Self-organizing network, caching, Multicast video distribution, Online auctions, Active bridging, Protocol boosters, Active congestion control, Distributed firewalls και Cell filtering. Προσεγγίσεις μέσω λογισμικού, όπως τα ANTS, Smart Packets, PLAN and CANES, έχουν ήδη υλοποιηθεί. Η δική μας πλευρά προσέγγισης στο θέμα των ενεργών δικτύων είναι τέτοια, που απαιτεί την σχεδίαση και την υλοποίηση μιας πλατφόρμας ανάπτυξης, η οποία να χρησιμοποιεί και να ανακαλύπτει τις δυνατότητες ενός ενεργού, 155 Mbps, 4x4 ATM διακόπτη (switch). Χωρίς το τμήμα του διακόπτη, είναι γεγονός ότι υπάρχουν πολλές, παρόμοιας αρχιτεκτονικής, πλατφόρμες στο εμπόριο. Η υπολογιστική τους ισχύ είναι εξαιρετικά μεγάλη, πολλές φορές με τη χρήση παραπάνω από μιας FPGA. Βρίσκονται σε αρκετά μεγάλη ποικιλία και ως προς τις οικογένειες των ολοκληρωμένων και τη μνήμη που χρησιμοποιούν. Σχεδόν όλες έχουν κάποιες θύρες επέκτασης, δίνοντας σε κάποιους connectors αρκετά σήματα προς χρήση. Η χρήση κάποιας από αυτές τις πλατφόρμες αναμφισβήτητα θα διευκόλυνε την ολοκλήρωση του ερευνητικού. Σε μια τέτοια περίπτωση θα χρειαζόταν να σχεδιάσουμε μια πλατφόρμα, η οποία θα υλοποιούσε το φυσικό επίπεδο του ATM δικτύου και θα διασυνδεόταν με την κεντρική κάρτα από τους connectors που προαναφέραμε. Δυστυχώς, κάτι τέτοιο δεν ήταν εφικτό, εξαιτίας της κατανομής των πόρων των FPGAs. Το πλήθος των I/O, η τοπολογία των

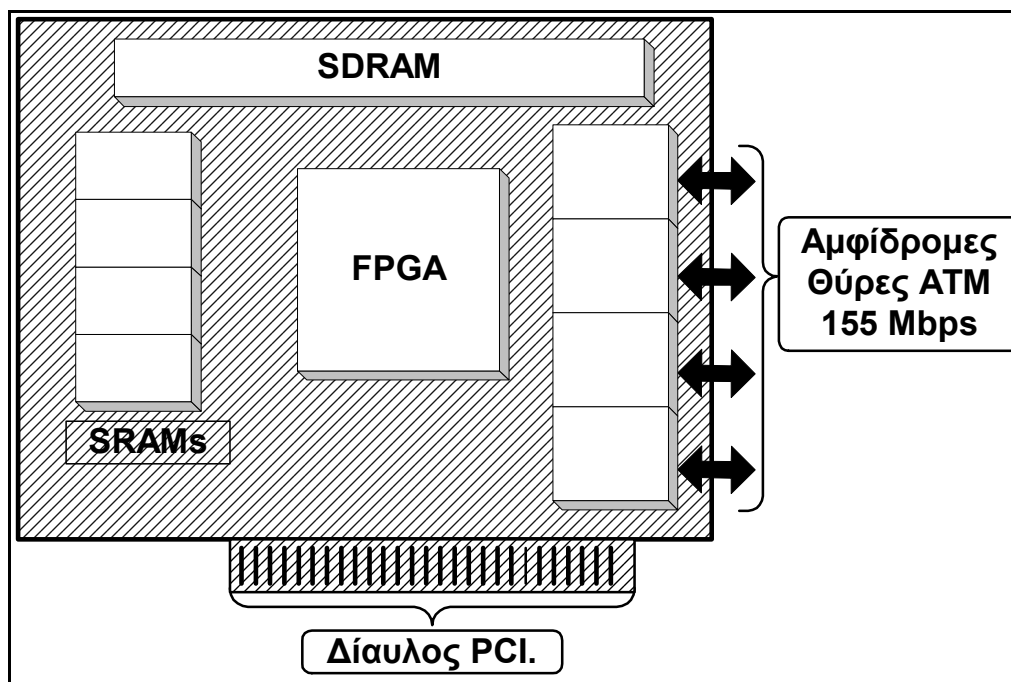
εξαρτημάτων και η χρήση ακατάλληλων μνημών στις πλατφόρμες του εμπορίου μας ανάγκασαν να εγκαταλείψουμε αυτήν την εκδοχή και να κατασκευάσουμε τη δική μας πλατφόρμα, βελτιστοποιώντας τους απαραίτητους πόρους.

ΚΕΦΑΛΑΙΟ 3^ο

3 Αρχιτεκτονική Πλατφόρμας.

3.1 Σύνθεση Συστήματος.

Η πλατφόρμα ανάπτυξης PLATO προορίζεται για ανάπτυξη εφαρμογών στη ερευνητική περιοχή των Ενεργών Δικτύων και της Επεξεργασίας Λόγου σε Πραγματικό Χρόνο. Κατά τη διάρκεια αυτής της διπλωματικής εργασίας κατασκευάστηκαν δυο εκδόσεις, PLATO X1.0 και PLATO A1.0. Η βασική αρχιτεκτονική είναι ίδια και στις δυο εκδόσεις και περιγράφεται γραφικά στο Σχήμα 3.1.



Σχήμα 3.1 Βασική αρχιτεκτονική του συστήματος.

Όπως αναφέρθηκε στο 2^ο κεφάλαιο, η πλατφόρμα ανάπτυξης σχεδιάστηκε ώστε να έχει τη δυνατότητα να χρησιμοποιείται για την εύρεση των δυνατοτήτων ενός ενεργού, 155 Mbps, 4x4 ATM διακόπτη (switch). Με βάση αυτόν τον άξονα, οι ελάχιστες προδιαγραφές του συστήματος είναι :

- ο Δυνατότητα διασύνδεσης με ATM δίκτυο, είτε αυτό είναι οπτικής ίνας, είτε χαλκού, καθώς και δυνατότητα διασύνδεσης με Ethernet (10/10 ή Gigabit) με κάποια μελλοντική επέκταση.
- ο Ελάχιστη καθυστέρηση στην επεξεργασία των πακέτων και δυνατότητα να υλοποιεί πρωτόκολλα χωρίς τη χρήση επεξεργαστή.
- ο Δυνατότητα αποθήκευσης συρμών από πακέτα και επεξεργασία του payload σε κάθε ένα από αυτά.
- ο Δυνατότητα επικοινωνίας με έναν επεξεργαστή γενικής χρήσης με σκοπό την περαιτέρω επεξεργασία των δεδομένων και την αποθήκευση του περιεχομένου των πακέτων ή την στατιστική μελέτη κάποιων δεδομένων για την παραγωγή πολύτιμων συμπερασμάτων όσον αφορά την αποτελεσματικότητα των πρωτοκόλλων που υλοποιούμε.
- ο Επιπλέον θύρες επέκτασης για μελλοντικές χρήσεις.

Το σύστημα τελικά υλοποιήθηκε με την χρήση μιας μεγάλης FPGA, η οποία θα είναι και η μόνη υπολογιστική μονάδα της πλατφόρμας. Στην έκδοση PLATO X1.0 χρησιμοποιήθηκε η FPGA Virtex XCV1000 της εταιρίας Xilinx, ενώ στην έκδοση PLATO A1.0 η Apex 20K400E της Altera. Αποφασίστηκε, αντί να χρησιμοποιηθεί ένας επεξεργαστής γενικής χρήσης να χρησιμοποιηθεί το PCI Bus ενός υπολογιστή, σχεδιάζοντας το σύστημα πάνω σε μια PCI κάρτα, για την επικοινωνία με τον επεξεργαστή του υπολογιστή και την περαιτέρω επεξεργασία των δεδομένων. Η FPGA θα αναλάβει την διαχείριση του PCI της SDRAM της SRAM και του Utopia. Με πολύ συντηρητικούς υπολογισμούς υπολογίζεται ότι η διαχείριση αυτών των πόρων θα καταλαμβάνει το πολύ το 30% της συνολικής χωρητικότητας της Virtex και ακόμα λιγότερο της Apex, εφόσον η δεύτερη είναι μεγαλύτερη, αφήνοντας ικανοποιητική χωρητικότητα για την ανάπτυξη και υλοποίηση διάφορων πρωτοκόλλων.

Εκτός της FPGA η πλατφόρμα αποτελείται και από κατάλληλο κύκλωμα χρονισμού, θύρα προγραμματισμού της FPGA, καθώς όπως προαναφέραμε περιληπτικά και από τα εξής τέσσερα βασικά υποσυστήματα:

- ο Τη διασύνδεση μεταξύ της πλατφόρμας ανάπτυξης και ATM δικτύου με τη χρήση κατάλληλου ολοκληρωμένου κυκλώματος, το οποίο υλοποιεί το

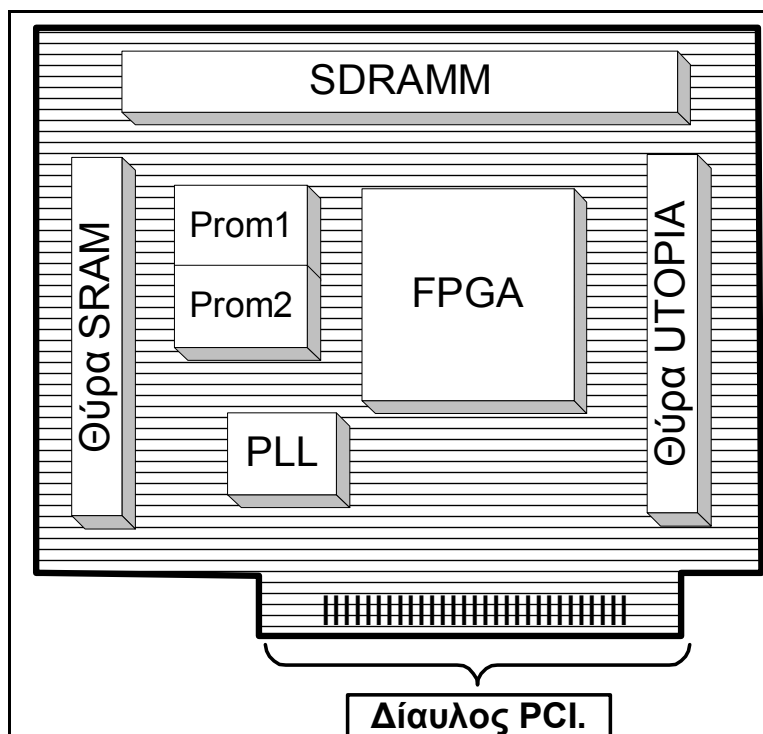
πρωτόκολλο UTOPIA 2-ATM [χρησιμοποιείται για να υλοποιήσει το φυσικό επίπεδο του ATM].

- Μια μνήμη 256 MB 133 MHz SDRAM για αποθηκευτικό χώρο.
- Μια διάταξη από ασύγχρονες SRAMs μνήμες για την χρήση τους στο ερευνητικό πρόγραμμα 'Επεξεργασία Λόγου σε Πραγματικό Χρόνο', οι οποίες θα χρησιμεύουν για Look Up Tables.
- PCI θύρα για την επικοινωνία της πλατφόρμας με τον επεξεργαστή.

3.3 Κατάτμηση Συστήματος.

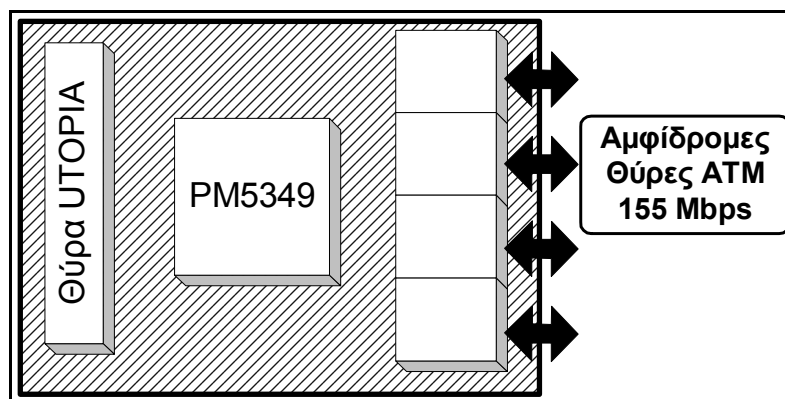
Για την υλοποίηση του PLATO προτιμήθηκε να χωριστεί η γενική αρχιτεκτονική του Σχήματος 3.1.1 σε τρία υποσυστήματα. Το κάθε ένα από αυτά περιλαμβάνει:

- Η κεντρική μονάδα που ονομάζουμε mainboard και περιέχει:
 - ο την FPGA,
 - ο τη θύρα για το PCI,
 - ο τη θύρα προγραμματισμού της FPGA,
 - ο την SDRAM,
 - ο ένα PLL για το χρονισμό της πλατφόρμας και,
 - ο στην έκδοση PLATO X1.0 μια διάταξη από μνήμες για το ‘power up’ προγραμματισμό της Virtex.



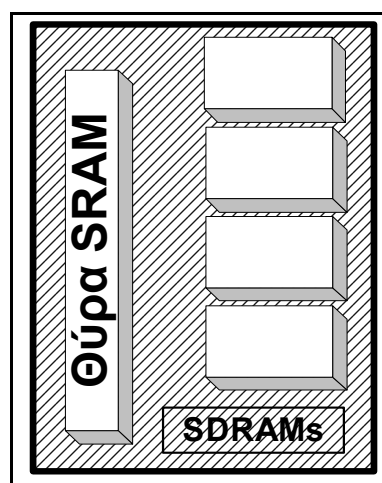
Σχήμα 3.2 Αρχιτεκτονική του mainboard.

- Μια μικρότερη πλακέτα, που ονομάζουμε daughterboard και υλοποιεί το physical επίπεδο για το ATM δίκτυο, η οποία περιέχει:
 - ο το ολοκληρωμένο PM5349 της εταιρείας PMC Sierra, το οποίο αναλαμβάνει την ψηφιοποίησης της μεταδιδόμενης πληροφορίας από το δίκτυο προς τη κάρτα και την αντίστροφη διαδικασία για τη μετάδοση πληροφορίας από τη κάρτα προς το δίκτυο, με δυνατότητα χειρισμού τεσσάρων I/O ports για ATM δίκτυα 155 Mbps,
 - ο τέσσερα RJ-45 slots για την διασύνδεση του daughterboard με καλώδιο ή οπτική ίνα του ATM δικτύου.



Σχήμα 3.3 Αρχιτεκτονική του UTOPIA daughterboard.

- Μια δεύτερη daughterboard η οποία περιέχει:
 - ο τέσσερις ασύγχρονες SRAMs των 12ns, σε παράλληλη σύνδεση αυτών ανά δυο.



Σχήμα 3.4 Αρχιτεκτονική του SRAM daughterboard.

Η κατασκευή όλου του συστήματος σε μια πλακέτα δεν προτιμήθηκε, επειδή υστερούσε σε δυο σημεία. Το βασικότερο από αυτά είναι ότι η πλατφόρμα ανάπτυξης θα έπρεπε να έχει τη δυνατότητα διασύνδεσης με ATM χαλκού ή οπτικής ίνας. Στη περίπτωση που όλο το σύστημα υλοποιούνταν σε μια πλατφόρμα ανάπτυξης, θα έπρεπε να έχουμε δυο τέτοιες πλατφόρμες, μια για κάθε είδος physical δικτύου. Αντίθετα, χωρίζοντας το σύστημα σε τρία μικρότερα και υλοποιώντας το κάθε ένα σε ξεχωριστή πλακέτα, μια εκ των οποίων είναι η βασική πλακέτα με την FPGA, και αντικαθιστώντας ανάλογα με το είδος του δικτύου το daughterboard του UTOPIA μπορούμε να πετύχουμε διασύνδεση και με τους δυο τύπους ATM δικτύου ή ακόμα και με Ethernet, έχοντας μόνο μια mainboard. Με αυτό το διαχωρισμό της αρχιτεκτονικής και με την υλοποίηση τριών πλακετών αντί μιας, μας δίνεται η δυνατότητα να χρησιμοποιήσουμε τα δυο daughterboards και στις δυο εκδόσεις του PLATO και πιθανότατα στις μελλοντικές.

Επίσης, η διάσπαση του συστήματος σε τρία μέρη, προτιμήθηκε γιατί δημιουργούνται απλούστερα υποσυστήματα, τα οποία έχουν μεγαλύτερες πιθανότητες λειτουργίας σε σχέση με τη πιθανότητα λειτουργίας όλου του συστήματος σε μια πλατφόρμα ανάπτυξης. Μικρότερα συστήματα είναι πιο πιθανό να σχεδιαστούν δίχως ελαττώματα, ενώ παράλληλα το στάδιο της υλοποίησης είναι σαφώς πιο εύκολο. Για παράδειγμα η έκδοση PLATO X1.0 είναι υλοποιημένη σε 8 επίπεδα και η PLATO A1.0 είναι σε 10 επίπεδα, σε αντίθεση με τα daughterboards SRAM και UTOPIA, που είναι υλοποιημένα σε 2 και 6 επίπεδα αντίστοιχα.

Επίσης, η πιθανότητα δυσλειτουργίας του ενός εκ των τριών πλακετών δεν μας υποχρεώνει να επανασχεδιάσουμε και τα τρία υποσυστήματα. Για παράδειγμα, αν είχαμε μια πλακέτα για όλα τα υποσυστήματα και κατά την κόλληση της FPGA προέκυπτε ανεπανόρθωτη βλάβη, όλο το σύστημα θα ήταν άχρηστο, εφόσον είναι πολύ απίθανο και να συγκολληθεί επιτυχώς ξανά η FPGA.. Αντίθετα, αν το ίδιο συνέβαινε στη περίπτωση των τριών πλακετών, τα δυο daughterboards θα ήταν λειτουργήσιμα και το μόνο που θα ήθελε επανασχεδίαση ή επανάληψη της υλοποίησής του θα ήταν η mainboard.

Οι πλακέτες μεταξύ τους ενώνονται με SMD connectors των 51 και 69 pins. Για την διασύνδεση του daughterboard της SRAM χρησιμοποιούνται τρεις connectors των 51 pins, ενώ για την διασύνδεση του daughterboard του UTOPIA χρησιμοποιούνται δυο connectors των 69 pins.

3.4 Κατανομή Πόρων.

Οι πόροι του συστήματος περιγράφονται με βάση την FPGA που χρησιμοποιεί αυτό, εφόσον αυτή είναι η κεντρική μονάδα επεξεργασίας και διαχείρισης των πόρων του συστήματος. Στον Πίνακα 3.1 φαίνονται αναλυτικά το πλήθος και το ποσοστό των I/O της εκάστοτε FPGA σε σχέση με τα περιφερειακά της.

| | | PLATO X1.0 | | PLATO A1.0 | |
|------------------|---------------------|-------------------|--------|-------------------|--------|
| | | # | % | # | % |
| PCI | <i>Data/Address</i> | 32 | 14.10% | 32 | 11.67% |
| | <i>Control</i> | 23 | | 23 | |
| UTOPIA I/O | | 102 | 26.15% | 102 | 21.65% |
| SDRAM | <i>Data</i> | 64 | 23.84% | 64 | 19.74% |
| | <i>Address</i> | 14 | | 14 | |
| | <i>Control</i> | 15 | | 15 | |
| SRAM | <i>Data</i> | 64 | 34.35% | 64 | 28.45% |
| | <i>Address</i> | 64 | | 64 | |
| | <i>Control</i> | 6 | | 6 | |
| Leds | | 4 | 1.02% | 16 | 3.4% |
| Header Connector | | 1 | 0.25% | 65 | 13.80% |
| Not connected | | 1 | 0.25% | 6 | 1.27% |
| Σύνολο | | 390 | 100% | 471 | 100% |
| Max I/O | | 404 | | 480 | |

Πίνακας 3.1 ‘Κατανομή των I/O της εκάστοτε FPGA σε σχέση με τα περιφερειακά του συστήματος’

Οι διαφορές των δυο εκδόσεων του PLATO οφείλονται στη διαφορετική FPGA που χρησιμοποιεί η κάθε μια. Όσον αφορά τους πόρους του συστήματος στις δυο εκδόσεις είναι ακριβώς ίδιες ως προς τις θύρες SRAM και UTOPIA, με αποτέλεσμα τα δυο daughterboards να είναι απόλυτα συμβατά και με τις δυο εκδόσεις. Αντίθετα, στις άλλες δυο θύρες, του PCI και της SDRAM, σημειώνονται κάποιες διαφορές λόγω των διαφορετικών ηλεκτρικών χαρακτηριστικών της FPGA, APEX, της έκδοσης PLATO A1.0, σε σχέση με αυτή της έκδοσης PLATO X1.0, που χρησιμοποιεί Virtex.

3.4.1 PCI

Το PCI μπορεί να χαρακτηριστεί με βάση τέσσερις παραμέτρους:

1. Την έκδοσή του. Στη περίπτωση του PLATO χρησιμοποιείται η νεότερη έκδοση 2,2, η οποία υπάρχει στα νεότερα PC, αλλά και στους υπολογιστές Sun.
2. Το data bus, το οποίο υπάρχει σε 32 και 64 bit. Στο PLATO χρησιμοποιούμε data bus των 32 bit.
3. Το επίπεδο τάσης λειτουργίας του, το οποίο στους προσωπικούς υπολογιστές είναι 5 Volt, ενώ στους SUN είναι 3,3 Volt και
4. Τη συχνότητα λειτουργίας του, η οποία μπορεί να είναι 33MHz ή 66MHz.

| <u>PCI</u> | <u>PLATO X1.0</u> | <u>PLATO A1.0</u> |
|--|-------------------|-------------------|
| 5 Volt, 33 MHz | * | ** |
| 5 Volt, 66 MHz | ** | ** |
| 3.3 Volt, 33 MHz | * | * |
| 3.3 Volt, 66 MHz | * | * |
| * Συμβατό χωρίς απαραίτητη τη χρήση της ενδιάμεσης κάρτας. ** Συμβατό με απαραίτητη τη χρήση της ενδιάμεσης κάρτας. | | |

Πίνακας 3.2 Συμβατότητα PLATO-PCI.

Η FPGA που χρησιμοποιείται στο PLATO X1.0, Virtex παρότι έχει Vcco (τάση τροφοδοσίας των IOBs της FPGA) = 3.3 Volt είναι συμβατή με PCI στα 5 Volt και σε συχνότητα λειτουργίας 33 MHz. Για αυτό το λόγο μπορεί να συνδεθεί απευθείας με τα pins του PCI και ανάλογα με το προγραμματισμό της να χειριστεί το PCI στα 5 ή στα 3.3 Volt. Στην περίπτωση των 3.3 Volt υπάρχει η δυνατότητα να λειτουργήσει και στα 66MHz.

Αντίθετα η apex στη έκδοση PLATO A1.0 είναι συμβατή με PCI στα 3.3 και όχι στα 5 Volt και σε συχνότητα λειτουργίας στα 33 και 66 MHz. Για να λειτουργήσει σε 5 Volt PCI χρειάζεται μια πλακέτα ανάμεσα στα slot του PCI της PLATO A1.0 και στο motherboard του υπολογιστή. Στον πίνακα 3.3.1.1 παρουσιάζονται όλοι οι δυνατοί συνδυασμοί ως προς τη λειτουργία του PCI στις δυο εκδόσεις του PLATO.

3.4.2 SRAM.

Η επιλογή των κατάλληλων μνημών προκύπτει από το ερευνητικό πρόγραμμα ‘Αναγνώριση Φωνής σε Πραγματικό Χρόνο’. Σύμφωνα με την αρχιτεκτονική αυτής της εφαρμογής είναι απαραίτητη η χρήση τεσσάρων μνημών, οι οποίες θα χρησιμοποιούνται για LUT [Look up Tables]. Σκοπός τους είναι να μετατρέπουν έναν αριθμό στο λογάριθμο της βάσης του. Και οι τέσσερις μνήμες κατά τη εφαρμογή γράφονται και διαβάζονται παράλληλα. Η κάθε μια από αυτές είναι 64k x 16.

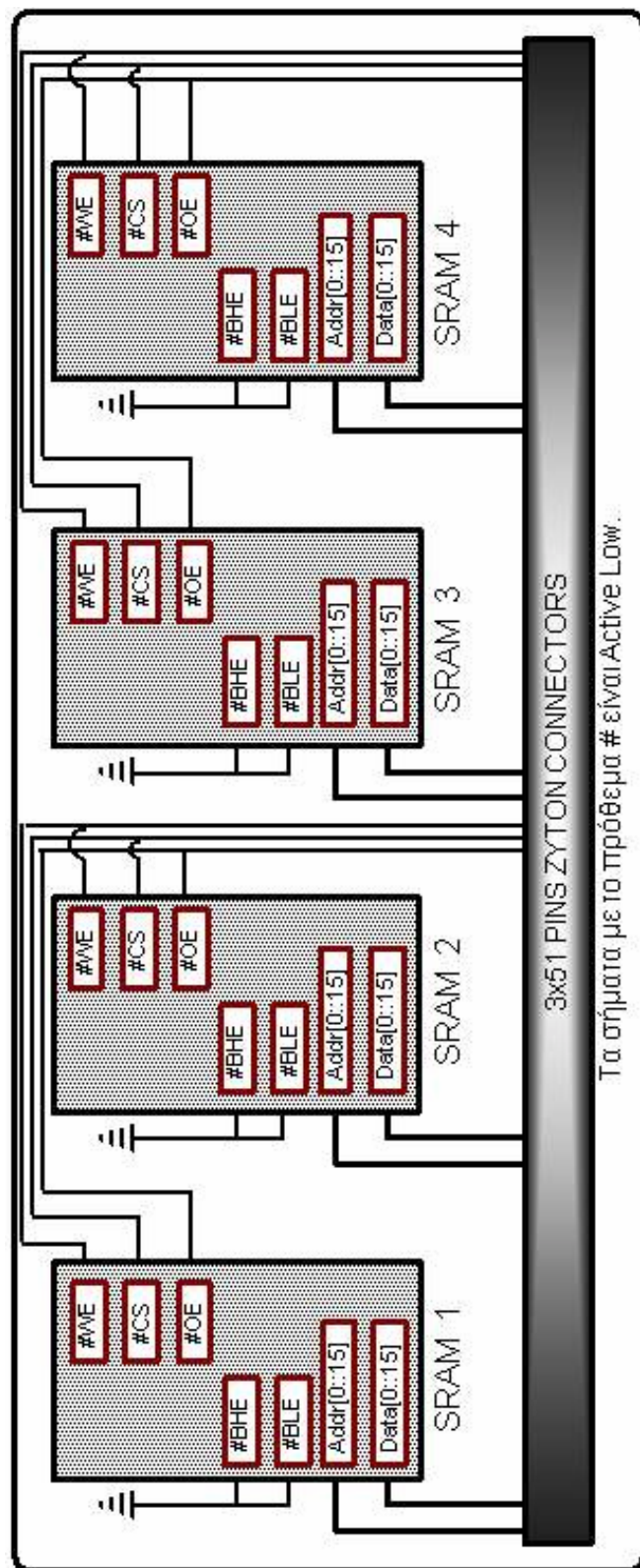
Τα σήματα ελέγχου για κάθε ολοκληρωμένο κύκλωμα είναι:

- #WE [Write Enable],
- #CS [Chip Select],
- #OE [Output Enable],
- #BHE [Byte High Enable],
- #BLE [Byte Low Enable].

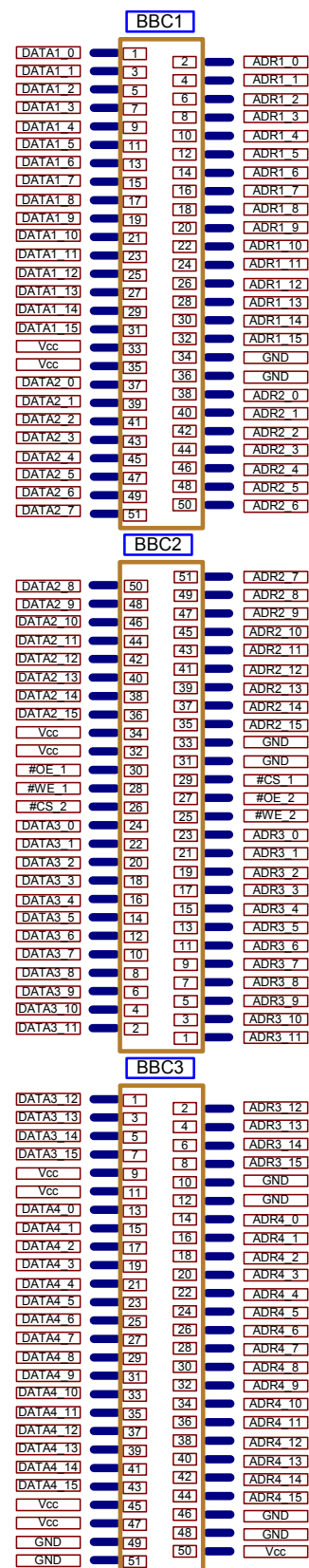
Τα σήματα #BHE και #BLE συνδέθηκαν απευθείας με τη γείωση του κυκλώματος εφόσον οι αριθμοί που χρησιμοποιεί η εφαρμογή είναι 16 bit.

Όπως φαίνεται από τον Πίνακα 3.1 το daughterboard της SRAM χρειάζεται στο σύνολό του 134 σήματα. Οι μνήμες είναι συνδεδεμένες ανά δυο παράλληλα, όπως δείχνει το Σχήμα 3.5. Η κάθε μνήμη έχει 16 pins Address και 16 pins Data καθώς και 3 pins ελέγχου. Η παράλληλη σύνδεσή τους επιτυγχάνεται με τη διασύνδεση των τριών σημάτων ελέγχου δυο μνημών σε κοινό net. Το αποτέλεσμα είναι να χρειάζονται $2[\text{ομάδες μνημών}] * (2 * 16[\text{pins data}] + 2 * 16[\text{pins address}] + 3[\text{control pins}]) = 134 \text{ pins}$, όπως φαίνεται και από το Πίνακα 3.1. Το κάθε chip μνήμης έχει δυο pins Vcc = 3.3 Volt και δυο pins γείωσης. Κάθε ζεύγος pin τροφοδοσίας συνδέεται σε διαφορετικά pins τροφοδοσίας από τους connectors διασύνδεσης των πλακετών. Αυτή η συνδεσμολογία προτιμήθηκε για να ελαχιστοποιηθεί η πιθανότητα ελαττωματικής τροφοδοσίας του chip λόγω κακής επαφής των pins τροφοδοσίας των connectors. Έτσι προκύπτει ότι χρειάζονται το λιγότερο 16 pins για τη συνολική τροφοδοσία του SRAM daughterboard.

Στο Σχήμα 3.6 παρουσιάζεται η τοπολογία των σημάτων σε σχέση με την θύρα επέκτασης.



Σχήμα 3.5 Αρχιτεκτονική του SRAM daughterboard.



Σχήμα 3.6 Θύρα SRAM.

3.4.3 SDRAM.

Η μνήμη που επιλέχθηκε για κύριο αποθηκευτικό χώρο των δεδομένων είναι η 133 MHz unbuffered SDRAM. Για την τοποθέτησή της στην πλατφόρμα ανάπτυξης χρησιμοποιήθηκε το κλασικό DIMM slot που υπάρχει σε όλες τις motherboards των υπολογιστών. Το DIMM slot συνδέθηκε με την FPGA με τρόπο τέτοιο, ώστε να μπορούν να χρησιμοποιηθούν όλες οι μνήμες που κυκλοφορούν στο εμπόριο, καθώς και οι περισσότερες από της δυνατότητες ελέγχου τους. Πιο συγκεκριμένα:

1. Μπορεί να συνδεθεί μνήμη έως και 512 MB συνολικής χωρητικότητας
2. Τα σήματα $CKE[0:1]$ μπορούν να ελεγχθούν από την FPGA. Όταν είναι στο λογικό '1' η SDRAM χρησιμοποιείται κανονικά. Όταν είναι στο λογικό '0' η SDRAM απενεργοποιείται και γίνεται refresh μόνη της.
3. Η FPGA επίσης, μπορεί να ελέγχει τα pins $DQMB[0:7]$. Με αυτά τα pins δίνεται η δυνατότητα στο χρήστη να χειρίζεται το data bus έχοντας μια μάσκα, η μορφή της οποίας καθορίζεται από τα bit $DQMB$.
4. Επίσης, η SDRAM μπορεί να χρονίζεται απευθείας από το PLL της πλατφόρμας ή από την ίδια την FPGA. Αυτό επιλέχθηκε για δυο βασικούς λόγους:
 1. Πολλές SDRAMs έχουν κατώτερη συχνότητα λειτουργίας. Οπότε, αν η συχνότητα της πλατφόρμας είναι 25MHz και η κατώτερη συχνότητα της SDRAM είναι 33MHz, θα μπορεί η εφαρμογή που θα εκτελεί η FPGA, διπλασιάζοντας το ρολόι εσωτερικά, να χειρίζεται την SDRAM σε διπλάσια συχνότητα, δηλαδή 50MHz και η επικοινωνία με τα υπόλοιπα περιφερειακά να γίνεται στα 25MHz.
 2. Η μεγαλύτερη συχνότητα λειτουργίας της SDRAM, σε σχέση με την υπόλοιπη εφαρμογή, μπορεί να φανεί χρήσιμη όταν, για παράδειγμα, η εφαρμογή έχει τη δυνατότητα να παράγει αποτελέσματα προς αποθήκευση με συχνότητα π.χ. 66MHz, και σύμφωνα με την αρχιτεκτονική του συστήματος θα μπορούσε να υπάρχει κάποιου είδους βελτίωση με το 'διάβασμα' των περιεχομένων της μνήμης στη διπλάσια συχνότητα. Για παράδειγμα, είναι ίσως επιθυμητό εκτός του burst mode κατά τη διαδικασία της ανάγνωσης να έχουμε ακόμα

μεγαλύτερο bandwidth στη φάση που θέλουμε να μεταφέρουμε τα αποτελέσματα μέσω του PCI. Μια ακόμα ίσως χρήση θα ήταν να κάνουμε refresh σε μεγαλύτερη συχνότητα. Περισσότερες λεπτομέρειες για αυτή συνδεσμολογία παρουσιάζονται στο κεφάλαιο 3.4.

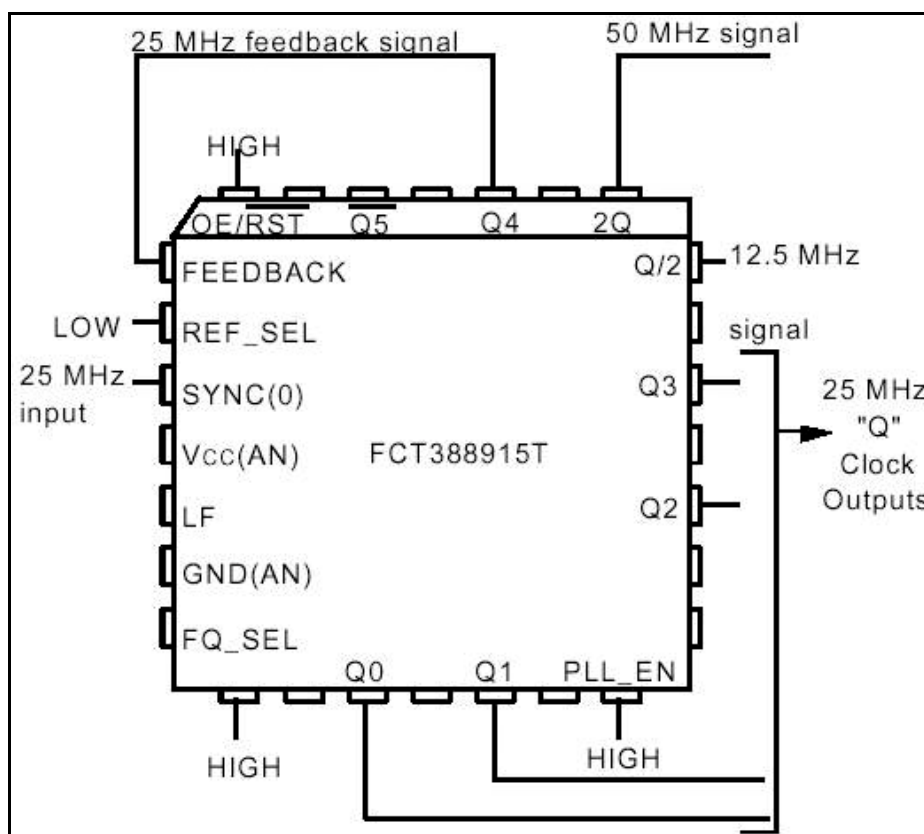
Η συνδεσμολογία της SDRAM με την FPGA διαφέρει στις δυο εκδόσεις. Αυτό οφείλεται στα ηλεκτρικά χαρακτηριστικά της εκάστοτε FPGA σε συνδυασμό με τα ηλεκτρικά χαρακτηριστικά της SDRAM. Τα σήματα της SDRAM #WE, #CAS, #RAS, A[0::13], BA0 και BA1 έχουν, σε σχέση με τα υπόλοιπα σήματα, πολύ μεγαλύτερη χωρητικότητα με αποτέλεσμα να μην έχει τη δυνατότητα η APEX να τα ‘οδηγήσει’, σε αντίθεση με τη Virtex. Το πρόβλημα λύθηκε, ‘περνώντας’ τα σήματα αυτά από έναν buffer στην έκδοση PLATO A1.0.

3.4.4 UTOPIA.

Το daughterboard UTOPIA, σχεδιάστηκε και υλοποιήθηκε στο ITE Ηρακλείου από τον Κύριο Μιχάλη Λιγεράκη στα πλαίσια του ερευνητικού προγράμματος PENED 99. Και οι δυο εκδόσεις του PLATO είναι συμβατές με το daughterboard, έχοντας εκ των προτέρων αποφασίσει το ‘lay out’ του connector διασύνδεσης.

3.5 Χρονισμός της πλατφόρμας

Ο χρονισμός της πλατφόρμας μπορεί να γίνει με δύο τρόπους: Με το ρολόι λειτουργίας του PCI ή με εξωτερικό ρολόι από κάποια πηγή όπως ένας κρύσταλλος ή μια γεννήτρια συχνοτήτων. Και με τους δυο τρόπους το ρολόι δεν συνδέεται απευθείας με την FPGA και την SDRAM. Ο λόγος είναι ότι σε ένα σύστημα το οποίο αποτελείται από μικρότερα υποσυστήματα είναι επιθυμητό να ελαχιστοποιηθεί το skew μεταξύ αυτών. Αν θεωρήσουμε ως σύστημα έναν υπολογιστή, ο οποίος χρησιμοποιεί το PLATO μέσω του PCI, τότε το PLATO είναι υποσύστημά του. Το ίδιο μπορούμε να υποθέσουμε όταν συνδέσουμε το PLATO με κάποια άλλη πλατφόρμα, όπως για παράδειγμα την Pamette.

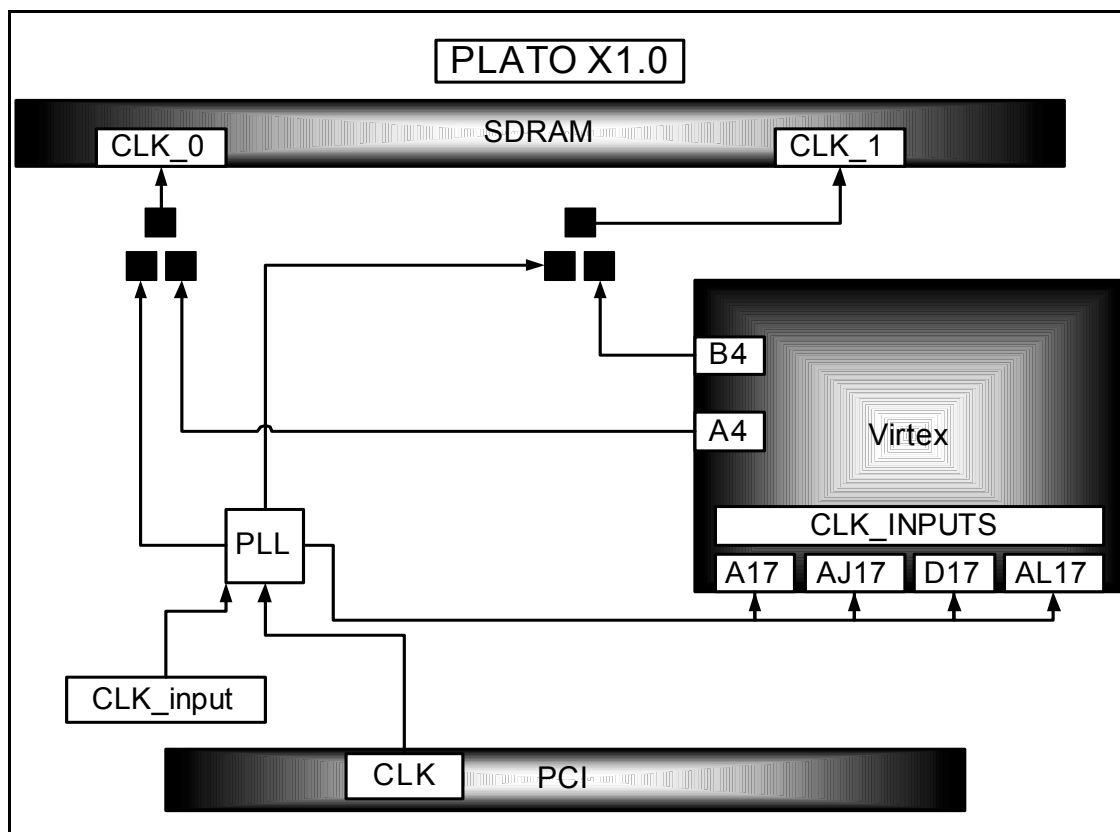


Σχήμα 3.7 Παράδειγμα συνδεσμολογίας του PLL.

Η ελαχιστοποίηση του Board-to-Board skew επιτυγχάνεται με τη χρήση ενός PLL [Phase Locked Loop]. Αυτό το ολοκληρωμένο κύκλωμα έχει δύο εισόδους για ρολόι, στη μια εκ των οποίων έχει συνδεθεί το ρολόι από το PCI και στην άλλη το ρολόι από την εξωτερική είσοδο συχνότητας της πλατφόρμας όπως φαίνεται από το Σχήμα 3.8. Μια από τις εξόδους του ολοκληρωμένου κυκλώματος ενώνεται με την

είσοδο του feedback του ίδιου, για να επιτευχθεί η ελαχιστοποίηση της διαφοράς φάσης του ρολογιού εισόδου με αυτό της εξόδου. Ανάλογα με το ποιο σήμα εξόδου συνδέσουμε στο feedback του PLL επιτυγχάνεται διπλασιασμός ή υποδιαίρεση της συχνότητας εξόδου. Και στις δυο εκδόσεις του PLATO υπάρχει η δυνατότητα να παραχθούν και οι τρεις συχνότητες.

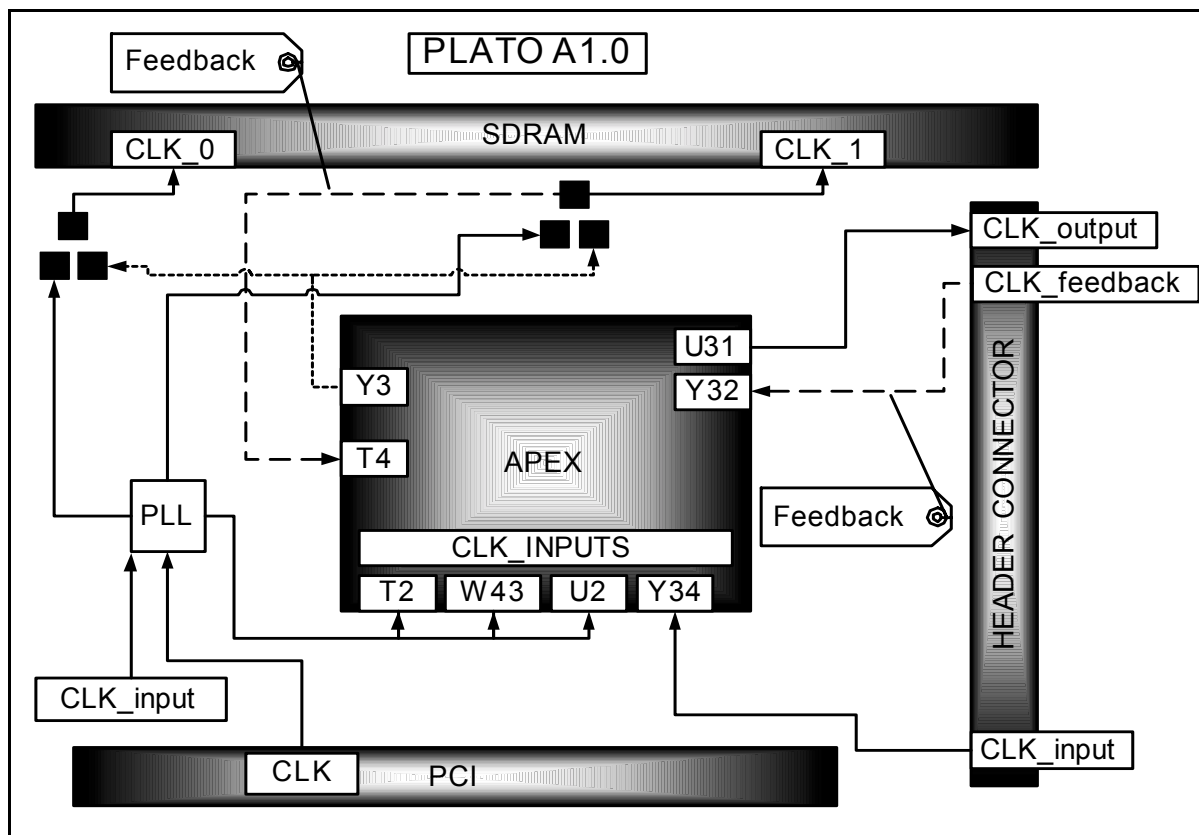
Ένας επίσης βασικός λόγος της χρήσης του PLL είναι τα ηλεκτρικά χαρακτηριστικά των εξαρτημάτων του PLATO. Οι FPGAs και στις δυο εκδόσεις έχουν τέσσερα pins εισόδου για ρολόι καθώς επίσης και η SDRAM. Η σύνδεση όλων αυτών με ένα σήμα εισόδου συχνότητας ίσως να μην μπορούσε να επιτρέψει τη λειτουργία της πλατφόρμας σε υψηλές συχνότητες. Εκμεταλλευόμενοι τα πολλά pins εξόδου του PLL αντιστοιχήσαμε σε κάθε μια από τις εξόδους του κάποια από τα pins εισόδου των άλλων εξαρτημάτων. Όπως φαίνεται και από τα Σχήματα 3.8 και 3.9 χρησιμοποιήθηκε ένα pin του PLL για να χρονίσουμε την FPGA και δυο pins για το χρονισμό της SDRAM.



Σχήμα 3.8 Χρονισμός της έκδοσης PLATO X1.0.

Πιο αναλυτικά, ο χρονισμός της έκδοσης PLATO X1.0 περιγράφεται από το Σχήμα 3.8. Η Virtex εσωτερικά έχει τέσσερα DLL [Delay Lock Loop] για την

υποδιαίρεση και το πολλαπλασιασμό της συχνότητας, καθώς και για την αλλαγή της φάσης, γεγονός που μας δίνει τη δυνατότητα να χρονίσουμε την SDRAM με τη χρήση των I/O pins A4 και B4.



Σχήμα 3.9 Χρονισμός της έκδοσης PLATO X1.0.

Στη PLATO A1.0 το βασικό κύκλωμα χρονισμού παραμένει το ίδιο. Η σημαντική διαφορά σε σχέση με το PLATO X1.0 οφείλεται στους διαφορετικούς πόρους της APEX. Η APEX διαθέτει τέσσερα PLL, δυο εκ των οποίων έχουν pins για την διασύνδεσή τους με τα περιφερειακά του ολοκληρωμένου κυκλώματος. Διαθέτουν ξεχωριστά pins για το Clk_Output, Clk_Input και Clk_Feedback. Στο PLATO X1.0 το ρολόι που 'οδηγεί' την SDRAM παρέχεται από κοινά I/O της FPGA. Στην περίπτωση του PLATO A1.0, η SDRAM 'οδηγείται' από το Clk_Output ενός εσωτερικού PLL και παίρνει ανάδραση στο Clk_Feedback, μετά το jumper της επιλογής του χρονισμού της SDRAM από το PLL της πλατφόρμας ή την APEX. Με αυτό το τρόπο ελαχιστοποιούμε το skew που προκαλείται από τα IOBs του package της APEX και τους αγωγούς του συγκεκριμένου net.

Η ίδια λογική ακολουθήθηκε κατά την διασύνδεση της APEX με τον Header Connector της πλακέτας. Σε μελλοντικές εφαρμογές είναι πιθανό να χρησιμοποιηθεί η επιπλέον θύρα του PLATO A1.0 για την σύνδεσή του με κάποιο άλλο σύστημα. Σε αυτή την περίπτωση είναι πιθανό να συγχρονιστούν τα δυο συστήματα με βάση το ρολόι του ενός. Αν λοιπόν το PLATO A1.0 ‘δίνει’ ρολόι στο άλλο σύστημα, τότε χρησιμοποιεί το Clk_Output και Clk_Feedback ενός εσωτερικού PLL της APEX, όπως φαίνεται και από το Σχήμα 3.9. Αν το εξωτερικό σύστημα δίνει ρολόι στο PLATO A1.0, τότε χρησιμοποιείται το Clk_Input ενός PLL της APEX.

3.6 Προγραμματισμός της FPGA.

Οι FPGAs και στις δυο εκδόσεις του PLATO είναι ISP (In System Programming, σε αυτή την περίπτωση με τη πτώση της τροφοδοσίας χάνονται και τα δεδομένα τους). Στο PLATO X1.0, εκτός από τη θύρα για τον προγραμματισμό του, υπάρχουν οι θέσεις για να τοποθετηθούν μελλοντικά δυο μνήμες (Proms) σε σειρά, για τον άμεσο προγραμματισμό του με το 'power up'. Στο PLATO A1.0, επειδή υπήρχε πρόβλημα χώρου στην πλακέτα, προτιμήθηκε να κατασκευαστεί μελλοντικά μια επιπρόσθετη πλακέτα που θα προσαρμόζεται στη θύρα προγραμματισμού. Σε αυτό το υποσύστημα θα υπάρχει μια μνήμη FLASH με το bit αρχείο του προγραμματισμού της APEX, καθώς και ένας μικροελεγκτής ή μια μικρή FPGA, η οποία θα υλοποιεί την κατάλληλη FSM για το προγραμματισμό. Στη μεν περίπτωση του PLATO X1.0, ο προγραμματισμός γίνεται από τον προγραμματιστή της Xilinx, MultiLINX, στη δε περίπτωση του PLATO A1.0 από το MasterBlaster της Altera.

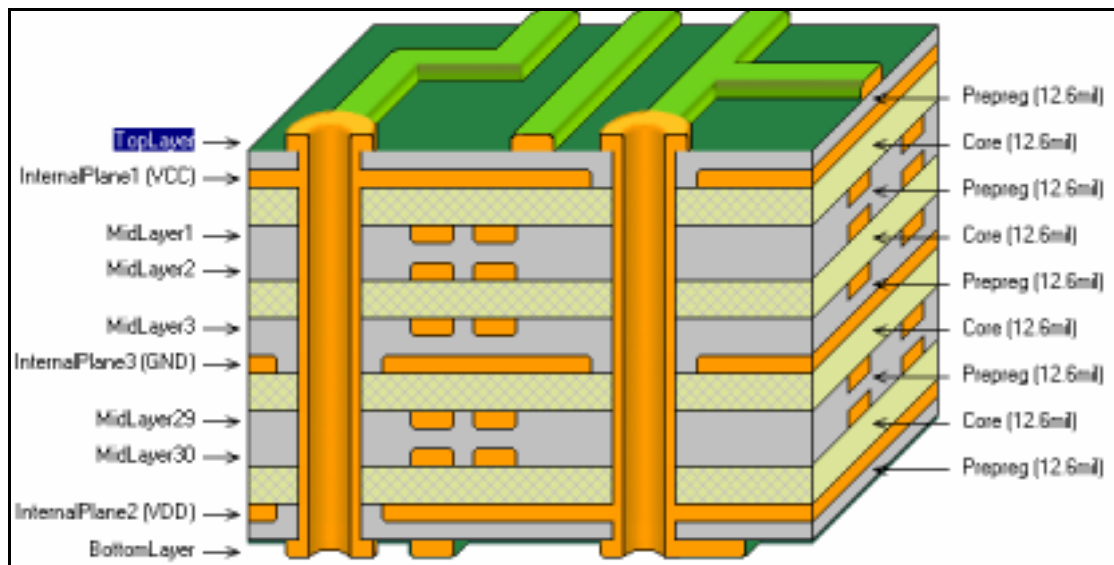
ΚΕΦΑΛΑΙΟ 4^ο

4 Σχεδίαση Πολυεπίπεδου Τυπωμένου Κυκλώματος.

Αυτό το κεφάλαιο αναφέρεται στη διαδικασία μελέτης και σχεδίασης ενός πολυεπίπεδου τυπωμένου κυκλώματος, καθώς και στις ιδιαιτερότητές του. Θα εξηγηθούν όροι και μεθοδολογίες που θα πρέπει να ακολουθήσει ένας μηχανικός για να έχει τη δυνατότητα να σχεδιάζει ένα πολύπλοκο σύστημα σε μια σχετικά περιορισμένη επιφάνεια. Θα αναφερθούν επίσης και κάποιες σημαντικές μέθοδοι που επηρεάζουν εξαιρετικά το χρόνο σχεδίασης, καθώς και τη δυνατότητα ελέγχου των σημάτων.

Σημείωση: PCB είναι τα αρχικά της φράσης Printed Circuit Board, δηλαδή Τυπωμένο Ηλεκτρονικό Σχέδιο-Πλακέτα.

Η τομή μιας πολυεπίπεδης πλακέτας φαίνεται στο Σχήμα 4.1.



Σχήμα 4.1 Τομή μιας πολυεπίπεδης πλακέτας.

4.1 Γενικά χαρακτηριστικά.

Ένα πολυεπίπεδο τυπωμένο κύκλωμα έχει σημαντικές διαφορές και αυξημένη πολυπλοκότητα σε σχέση με ένα κύκλωμα δυο επιπέδων. Για αυτό το λόγο είναι απαραίτητο να δοθεί μεγαλύτερη προσοχή στη ‘ρύθμιση’ του εργαλείου κατά τη διαδικασία σχεδιασμού, αλλά και στη κατανόηση διάφορων εννοιών από το σχεδιαστή, όπως πχ, Plane, power plane, nested power plane, clearance, grid κτλ. Η μεθοδολογία που θα ακολουθηθεί κατά τη διαδικασία της σχεδίασης είναι ιδιαίτερα σημαντική, εφόσον κάποιες παράμετροι, όπως το πλήθος των σημάτων ή το πάχος των αγωγών, έχουν διαφορετικές προδιαγραφές σε σχέση με μια πλακέτα δυο επιπέδων. Λόγω του πλήθους των σημάτων και των παραμέτρων που περιγράφουν, αλλά και καθορίζουν επακριβώς μια πολυεπίπεδη πλακέτα, είναι πολύ σημαντικό, για την αποφυγή σφαλμάτων, να υπάρχει μία συγκεκριμένη διαδικασία ολοκλήρωσης της κάθε φάσης. Για παράδειγμα, δεν είναι δυνατό να μπούμε στο στάδιο της σχεδίασης, αν δεν έχουμε αποφασίσει το πλήθος των επιπέδων ή το πλάτος των αγωγών. Οι φάσεις της σχεδίασης είναι:

1. Μελέτη της συμβατότητας των εξαρτημάτων, καθώς και των διάφορων παθητικών παρελκόμενων που πιθανώς θα χρειάζονται για τη σωστή λειτουργία τους, όπως κυκλώματα απόζευξης με πυκνωτές και ferrites ή αντιστάσεις.
2. Επιλογή κατάλληλου package των εξαρτημάτων, ανάλογα με τις απαιτήσεις της σχεδίασης (π.χ. BGA, TSOP).
3. Κατασκευή του netlist της σχεδίασης.
4. Σχεδίαση του footprint των εξαρτημάτων.
5. Μελέτη του ιδανικού πάχους των αγωγών, των via, του clearance, του grid κτλ. σύμφωνα με το package των εξαρτημάτων, αλλά και των προδιαγραφών της εταιρίας κατασκευής του τυπωμένου κυκλώματος.
6. Μελέτη της τοπολογίας των εξαρτημάτων σύμφωνα με τη γενικότερη αρχιτεκτονική του συστήματος.
7. Σχεδίαση του τυπωμένου κυκλώματος.

8. Έλεγχος της σχεδίασης με DRC.
9. Παραγωγή των αρχείων περιγραφής της σχεδίασης (Gerber files) σύμφωνα με τις προδιαγραφές της εταιρίας κατασκευής του τυπωμένου κυκλώματος.

Η ολοκλήρωση κάθε φάσης δεν συνεπάγεται και την ορθότητά της. Αποφάσεις που παίρνονται στις πρώτες φάσεις της σχεδίασης είναι πολύ πιθανό να επηρεάσουν αρνητικά κάποιες από τις φάσεις που έπονται, σε βαθμό τέτοιο που να μην είναι εφικτή η ολοκλήρωσή τους. Αυτά τα φαινόμενα είναι δυνατό να προκαλέσουν την καθυστέρηση του έργου. Για να αποφευχθεί κάτι τέτοιο καλό θα ήταν ο σχεδιαστής να ακολουθήσει τους παρακάτω δυο κανόνες.

1. Η σχεδίαση της κάθε φάσης πρέπει να γίνεται, λαμβάνοντας υπόψη κατά πόσο είναι εφικτή η ολοκλήρωση των επόμενων, με βάση τις επιλογές που κάνουμε. Για παράδειγμα, αν το package ενός εξαρτήματος είναι τέτοιο, ώστε να μην έχουμε τη δυνατότητα να το κολλήσουμε στο PCB, ο σχεδιαστής πρέπει να το σκεφτεί και να το αποφύγει στη δεύτερη φάση.
2. Διατήρηση της κατάλληλης μεθοδολογίας σχεδίασης της κάθε φάσης και κυρίως του netlist, όπως θα δούμε και παρακάτω, για τον γρήγορο επανασχεδιασμό της.

4.2 Επιλογή του πλήθους των επιπέδων.

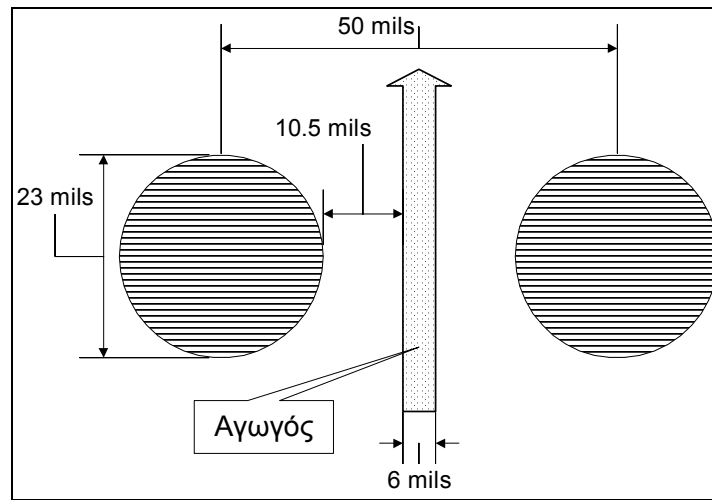
Η επιλογή των επιπέδων του τυπωμένου κυκλώματος απορρέει από τις προδιαγραφές του συστήματος. Στο στάδιο της σχεδίασης της αρχιτεκτονικής δεν είναι ακόμα γνωστό το ακριβές πλήθος των επιπέδων. Αυτή η παράμετρος υπολογίζεται, συνδυάζοντας διάφορα στοιχεία όπως:

1. Το επιθυμητό μέγεθος της πλακέτας σε σχέση με το πλήθος των σημάτων που θα υπάρχουν, σύμφωνα με την αρχιτεκτονική του συστήματος. Σε μια πλακέτα δυο επιπέδων και προκαθορισμένης έκτασης δεν είναι δυνατό να μπορούμε να σχεδιάσουμε πάνω από κάποιο αριθμό αγωγών. Αν θέλουμε και άλλα σήματα, είναι απαραίτητο να αυξήσουμε τον αριθμό των επιπέδων ή την επιφάνεια της πλακέτας.
2. Το footprint των εξαρτημάτων που θα χρησιμοποιηθούν στη σχεδίαση, σε συνδυασμό με τις δυνατότητες της σειράς παραγωγής του τυπωμένου κυκλώματος.
3. Το κοστολόγιο της κατασκευής.

Ο καθοριστικός παράγοντας του πλήθους των επιπέδων στο PLATO X1.0 και στο PLATO A1.0, ήταν το package της εκάστοτε FPGA, σε συνδυασμό με τους περιορισμούς της κατασκευάστριας εταιρίας, Intracom. Στη μεν έκδοση PLATO X1.0 χρησιμοποιήθηκε η FPGA Virtex XCV1000 σε package BGA-560 5 γραμμών, στη δε PLATO A1.0 χρησιμοποιήθηκε η FPGA Apex 20K400E σε package BGA-650 5 και 6 γραμμών, γεγονός που μας οδήγησε στη κατασκευή τυπωμένων κυκλωμάτων 8 και 10 επιπέδων αντίστοιχα.

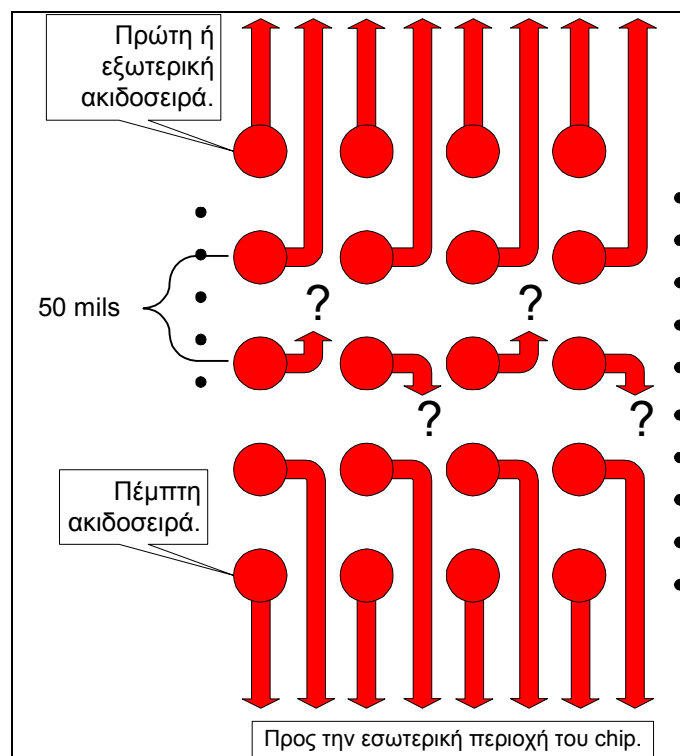
Πιο συγκεκριμένα, το package της Virtex αποτελείται από 560 pins, τα οποία είναι διατεταγμένα σε 5 παράλληλες σειρές με τρόπο τέτοιο που το κάθε pin να απέχει από τα γειτονικά του 50 mils ή αλλιώς, περίπου 2 χιλιοστά. Η ιδιαιτερότητα του package, παρουσιάζεται όταν προσπαθήσουμε να μεταφέρουμε τα σήματα της FPGA μέσω αγωγών προς τα περιφερειακά εξαρτήματα της πλατφόρμας. Σύμφωνα με τις δυνατότητες κατασκευής των τυπωμένων κυκλωμάτων της κατασκευάστριας εταιρίας, το ελάχιστο πλάτος αγωγού είναι 6 mils με clearance 7 mils (Clearance, είναι η απόσταση, που πρέπει να έχει το υλικό ενός σήματος, από οποιοδήποτε άλλο. Για παράδειγμα η απόσταση ενός αγωγού από έναν άλλο.). Αν λάβουμε υπόψη μας

και τα data sheets της Xilinx για το footprint της Virtex, που προτείνουν 23 mils ελάχιστη διάμετρο για το pad του κάθε pin, τότε καταλήγουμε στη συνδεσμολογία που περιγράφεται από το Σχήμα 4.2.1.



Σχήμα 4.2 Ελάχιστες πλάτος αγωγού σύμφωνα με τις προδιαγραφές τις INTRACOM.

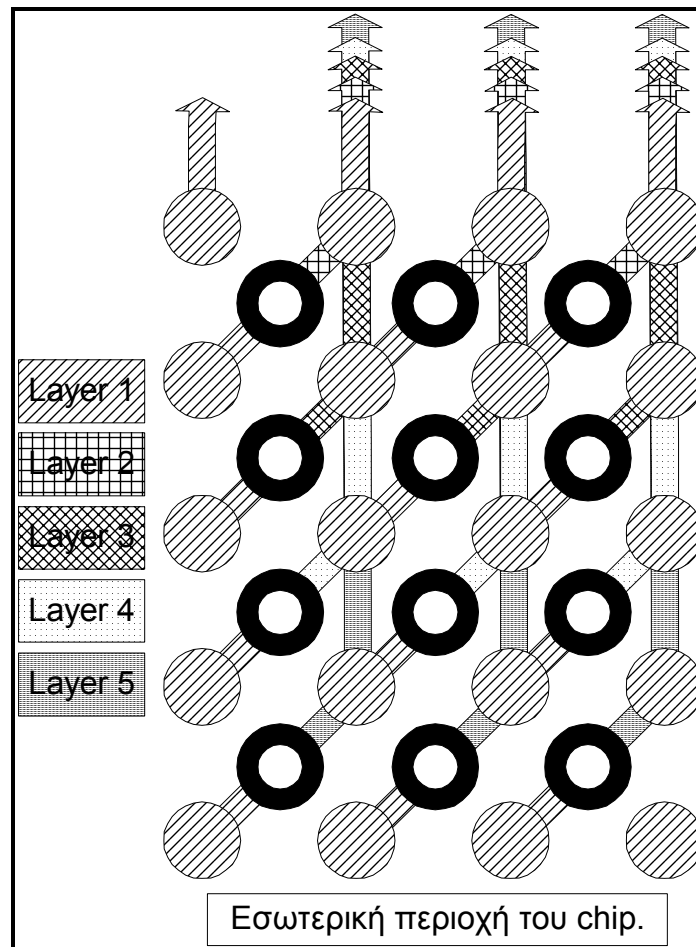
Όπως φαίνεται από το Σχήμα 4.2, είναι αδύνατο να σχεδιαστούν παραπάνω από ένας αγωγοί στο πάνω επίπεδο του τυπωμένου κυκλώματος, ανάμεσα από δύο γειτονικά pads του footprint της FPGA, ενώ παράλληλα να διατηρήσουμε τις προδιαγραφές σχεδίασης. Για αυτό το λόγο, ο αγωγός και το clearance σε όλη τη σχεδίαση έχουν πλάτος 8 mils. Έχοντας διαπιστώσει ότι μόνο ένας αγωγός μπορεί να σχεδιαστεί σε αυτό το σημείο, δημιουργείται πρόβλημα στη δρομολόγηση των σημάτων των εσωτερικών pads της FPGA, όπως περιγράφεται και γραφικά στο Σχήμα 4.3.



Σχήμα 4.3 Πρόβλημα με το routing του BGA footprint.

Είναι φανερό ότι τα σήματα της μεσαίας σειράς του BGA footprint της Virtex δεν έχουν πρόσβαση προς τα περιφερειακά εξαρτήματα της πλατφόρμας, αν υποθέσουμε ότι δεν έχουμε στη διάθεσή μας κάποιο άλλο επίπεδο, εκτός από το πάνω, όπου γίνεται και η κόλληση της FPGA. Ακόμα και οι δυο εσωτερικές σειρές του footprint που κατευθύνουν τα σήματά τους προς την εσωτερική περιοχή του chip, είναι εξαιρετικά απίθανο να δρομολογηθούν προς τις άλλες περιοχές της πλατφόρμας, λόγω της πυκνότητας των σημάτων και των αντίστοιχων νία που απαιτούνται. (Συνολικά περί τα 200 Vias και 200 traces, σε μια περιοχή που δεν ξεπερνά τα 2 cm²). Σε αυτό το συμπέρασμα καταλήξαμε επίσης και μετά από πολλούς πειραματισμούς σχεδίασης με τη βοήθεια του autorouter του Protel ή χωρίς.

Για να μπορέσουμε λοιπόν να λύσουμε το πρόβλημα, θα έπρεπε να τοποθετήσουμε νίας για όλα τα σήματα της FPGA δίπλα ακριβώς από τα αντίστοιχα pads, δίνοντας με αυτό το τρόπο τη δυνατότητα σε κάθε σήμα να έχει πρόσβαση σε όλα τα επίπεδα. Αυτή η τακτική είναι ο ιδανικότερος τρόπος σχεδιασμού τυπωμένου κυκλώματος που έχει BGA εξαρτήματα ή ακόμα και FPGA (είναι όπως και το BGA με τη μόνη διαφορά ότι τα pads του footprint τοποθετούνται σε απόσταση 30 mils μεταξύ τους), όπως φαίνεται και από το Σχήμα 4.2.3.



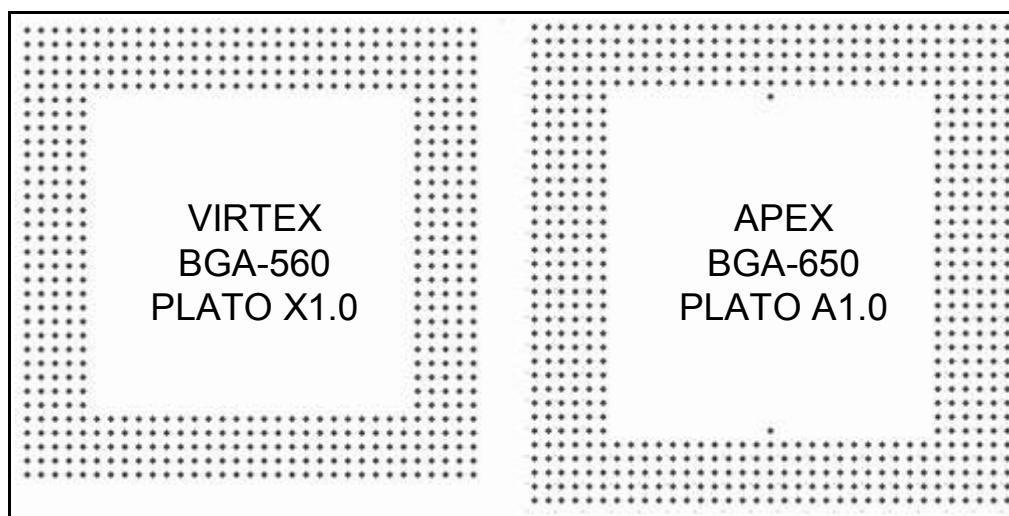
Σχήμα 4.4 Τοποθέτηση vias ανάμεσα σε τέσσερα γειτονικά pads.

Ο ελάχιστος αριθμός επιπέδων που χρειάζονται για το routing της FPGA είναι θεωρητικά τέσσερα. Αυτό προκύπτει αν προσέξουμε στο Σχήμα 4.4 τη δεύτερη εξωτερική σειρά από pads. Θα μπορούσε να μην έχει vias και να χρησιμοποιεί το top (Layer 1) επίπεδο για τα δικά της σήματα. Σε μια όμως τόσο αυστηρά προκαθορισμένη σχεδίαση ο autorouter του Protel δεν συμπεριφέρεται ικανοποιητικά, με αποτέλεσμα να μην ολοκληρώνει ποτέ το routing πάνω από 60% και με μια καθυστέρηση της τάξεως της μιάμισης ώρας με χρήση ενός υπολογιστή Pentium III στα 560MHz.

Είναι φανερό από τη παραπάνω μελέτη ότι η FPGA για το PLATO X1.0 χρειάζεται το λιγότερο 5 επίπεδα, για να έχει τη δυνατότητα να συνδεθεί με τα περιφερειακά της. Αν προσθέσουμε και άλλα δυο επίπεδα για την τροφοδοσία και τη γείωση της πλακέτας, τότε φτάνουμε στα 7 επίπεδα και τελικά καταλήγουμε στα 8, αφού τα πολυεπίπεδα τυπωμένα κυκλώματα κατασκευάζονται πάντα σε ζυγό αριθμό επιπέδων. Μια πλακέτα 7 επιπέδων σε σχέση με μια των 8 έχει

μηδενική επιβάρυνση ως προς το κόστος κατασκευής και την πολυπλοκότητα. Τελικά, καταλήγουμε στο συμπέρασμα ότι για την έκδοση PLATO X1.0 χρειαζόμαστε 2 επίπεδα για την τροφοδοσία και 6 επίπεδα για τα υπόλοιπα σήματα.

Σε αντίθεση με το PLATO X1.0, το PLATO A1.0 χρειάστηκε 10 επίπεδα για την κατασκευή του. Αυτά τα δύο παραπάνω επίπεδα δεν οφείλονται στο footprint της FPGA (BGA-650). Όπως φαίνεται και από το Σχήμα 4.5, η σημαντική διαφορά του footprint της APEX είναι ότι στις δυο κάθετες πλευρές της έχει 6 σειρές από pads. Για να γίνει routing στα σήματα σε αυτές τις περιοχές αρκούν 6 επίπεδα, σύμφωνα με τη προηγούμενη μελέτη. Αν λοιπόν χρειαζόμασταν δυο επίπεδα για τροφοδοσία και γείωση, τότε δεν θα υπήρχε πρόβλημα να γίνει η σχεδίαση σε ένα τυπωμένο κύκλωμα 8 επιπέδων. Η ιδιαιτερότητα του chip οφείλεται στη τροφοδοσία του. Χρειάζεται μια τροφοδοσία στα 3.3 Volt για τα IOBs και μια στα 1.8 Volts για τα CLBs. Οπότε, έχοντας τρία επίπεδα για τις τροφοδοσίες και τη γείωση και έξι επίπεδα για τα σήματα, συνολικά χρειάζονται 9 επίπεδα, που σύμφωνα με τον κανόνα του ζυγού πλήθους επιπέδων σε πολυεπίπεδα τυπωμένα κυκλώματα γίνονται 10.



Σχήμα 4.5 Τα footprint των FPGA.

Η διαδικασία που περιγράψαμε σε αυτό το κεφάλαιο αποτελεί ταυτόχρονα και ένα πολύ καλό παράδειγμα για την τεκμηρίωση της μεθοδολογίας που αναφέραμε στη κεφάλαιο 4.1. Είναι φανερό ότι αν δεν είχαμε υπολογίσει σωστά το πλήθος των επιπέδων της πλατφόρμας, πιθανόν δε θα μπορούσαμε να ολοκληρώσουμε τη σχεδίαση της. Για την ολοκλήρωση αυτής της φάσης, ήταν απαραίτητο ο σχεδιαστής να λάβει υπόψη του τους εξής παράγοντες:

- 1) Τη συνδεσμολογία της πλατφόρμας.
- 2) Τις δυνατότητες της κατασκευάστριας εταιρίας.
- 3) Τις προδιαγραφές του footprint των ολοκληρωμένων κυκλωμάτων.
- 4) Τις δυνατότητες του εργαλείου σχεδίασης.
- 5) Τα ηλεκτρικά χαρακτηριστικά των ολοκληρωμένων κυκλωμάτων.

4.3 Επιλογή και χρήση παθητικών εξαρτημάτων και αγωγών.

Η αρχιτεκτονική που παρουσιάστηκε στο 3^ο κεφάλαιο, περιγράφει τη συνδεσμολογία μεταξύ των βασικών εξαρτημάτων της πλατφόρμας. Το σύστημα, στο σύνολό του, περιλαμβάνει πάνω από 140 εξαρτήματα στην έκδοση PLATO X1.0, τα περισσότερα από τα οποία είναι παθητικά. Δηλαδή πυκνωτές, αντιστάσεις, connectors, αλλά και άλλα εξαρτήματα, που σαν βασική τους λειτουργία έχουν να ‘ρυθμίζουν’ τα ενεργά στοιχεία του κυκλώματος ή να τους παρέχουν τις κατάλληλες συνθήκες λειτουργίας.

Ανάλογα με το σκοπό που εξυπηρετούν στο κύκλωμα τα παθητικά εξαρτήματα, μπορούμε να τα κατατάξουμε σε δυο κατηγορίες:

- Πυκνωτές, αντιστάσεις, φερριτικοί πυρήνες κτλ.
- Εξαρτήματα όπως Board-to-Board connectors ή slots π.χ. για SDRAM.

4.3.1 Πυκνωτές, αντιστάσεις & Φερριτικοί Πυρήνες.

Στην αρχιτεκτονική που παρουσιάστηκε στο 3^ο κεφάλαιο δεν περιγράφονται τα περιφερειακά εξαρτήματα των ολοκληρωμένων κυκλωμάτων. Η διαδικασία της επιλογής αυτών των εξαρτημάτων γίνεται στη πρώτη φάση της σχεδίασης, όπως περιγράφεται και στην παράγραφο 4.1. Στη συγκεκριμένη περίπτωση, στο PLATO, τέτοια εξαρτήματα χρειάστηκαν στην FPGA, στην SDRAM, στο PLL και στα σήματα της FPGA που είναι για το προγραμματισμό της.

Μια λανθασμένη επιλογή της τιμής για τα εξαρτήματα αυτά μπορεί να είναι και ο λόγος για τη δυσλειτουργία της πλατφόρμας. Για αυτό το λόγο, κάθε ολοκληρωμένο κύκλωμα συνοδεύεται από αντίστοιχες πληροφορίες που αφορούν στη συνδεσμολογία, στην τιμή και μερικές φορές στο package του περιφερειακού εξαρτήματος. Σε πλατφόρμες, όπου οι εφαρμογές τους χρησιμοποιούν ρολόι της τάξεως των 15 MHz και πάνω, πρέπει να ληφθούν υπόψη και κάποια άλλα στοιχεία, όπως η απόσταση τοποθέτησης των περιφερειακών εξαρτημάτων από το ολοκληρωμένο κύκλωμα και ο χρόνος απόκρισής τους.

Πιο αναλυτικά:

SDRAM

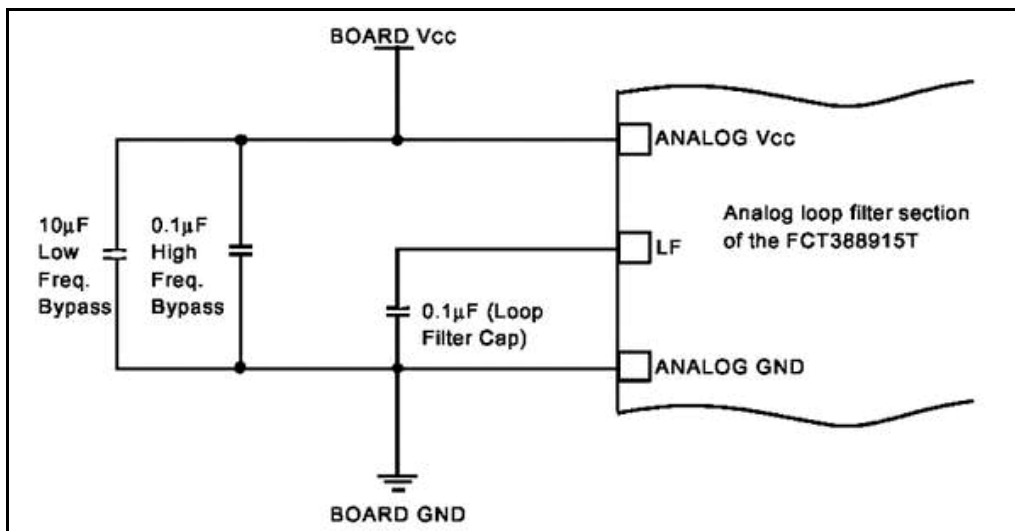
Στην SDRAM τοποθετήθηκαν 16 αποζευκτικοί πυκνωτές των $0.10\mu\text{F}$ ο καθένας.

SRAM

Για κάθε SRAM τοποθετήθηκαν δυο αποζευκτικοί πυκνωτές των $0.10\mu\text{F}$, ένας για το κάθε pin τροφοδοσίας του ολοκληρωμένου.

PLL

Το PLL, που χρησιμεύει στο χρονισμό της πλατφόρμας, είναι ιδιαίτερα ευαίσθητο στο θόρυβο της τροφοδοσίας. Για αυτό το λόγο είναι απαραίτητο, ειδικά σε υψηλές ταχύτητες, να υπάρχει κατάλληλη συνδεσμολογία, όπως περιγράφεται στο Σχήμα 4.6, που λειτουργεί ως φίλτρο για την αποκοπή του θορύβου στη τροφοδοσία. Αυτή η διάταξη θα πρέπει να βρίσκεται όσο γίνεται πιο κοντά στα αντίστοιχα pins του PLL, ώστε να υπάρχει όσο το δυνατό γρηγορότερη απόκριση.



Σχήμα 4.6 Διάγραμμα του φίλτρου στο PLL

Εκτός από τη παραπάνω διάταξη, για κάθε pin τροφοδοσίας του ολοκληρωμένου τοποθετήθηκε ένας αποζευκτικός πυκνωτής $0.1\mu\text{F}$.

FPGA

Τα κύρια χαρακτηριστικά που πρέπει να ληφθούν υπόψη στην απόζευξη ενός ολοκληρωμένου κυκλώματος είναι η συχνότητα λειτουργίας του και οι απαιτήσεις του σε ρεύμα. Στη εναλλαγή του ρολογιού από το λογικό «0» στο λογικό «1» και στα επόμενα λίγα nanoseconds γίνονται οι εναλλαγές των σημάτων από το «0» στο «1» ή και αντίστροφα. Σε αυτό το χρονικό παράθυρο οι απαιτήσεις του ολοκληρωμένου σε ρεύμα είναι αυξημένες αφού τα transistor μεταβαίνουν από τη μια κατάσταση στην άλλη. Αυτό έχει ως αποτέλεσμα να παρουσιάζονται «εκρηκτικές» απαιτήσεις σε ρεύμα στην περιοχή του τυπωμένου κυκλώματος, όπου βρίσκονται τα pins τροφοδοσίας του εν λόγω ολοκληρωμένου κυκλώματος. Για να καλυφθούν αυτές οι ανάγκες χρησιμοποιούμε τους αποζευκτικούς πυκνωτές. Η συχνότητα λειτουργίας, αλλά και οι απαιτήσεις του ολοκληρωμένου κυκλώματος σε ρεύμα είναι οι σημαντικότεροι παράγοντες που καθορίζουν την τιμή, το πλήθος και το package των αποζευκτικών πυκνωτών που θα χρησιμοποιηθούν.

Όσο πιο μεγάλη είναι η συχνότητα λειτουργίας του ολοκληρωμένου κυκλώματος, τόσο μεγαλύτερη ταχύτητα απόκρισης πρέπει να έχει ο αποζευκτικός πυκνωτής. Ένας μικρός πυκνωτής πολύ κοντά στα pins τροφοδοσίας του ολοκληρωμένου κυκλώματος είναι σε θέση να προσφέρει λίγο ρεύμα, αλλά σε πολύ μικρό χρονικό διάστημα, πράγμα που τον κάνει ιδανικό σε μεγάλες συχνότητες. Όσο η συχνότητα μικραίνει, τόσο μικραίνει και η απαίτηση του ολοκληρωμένου κυκλώματος σε ταχύτητα απόκρισης. Ένας μεσαίος σε χωρητικότητα πυκνωτής, που έχει μικρότερη ταχύτητα απόκρισης, είναι κατάλληλος σε αυτές τις περιπτώσεις, εφόσον μπορεί να αποζεύξει περισσότερα pins τροφοδοσίας λόγω του μεγαλύτερου φορτίου του και να τοποθετηθεί σε μεγαλύτερη απόσταση από το σημείο τροφοδοσίας του ολοκληρωμένου κυκλώματος.

Με βάση τα παραπάνω είναι εμφανές ότι το βασικότερο που χρειάζεται να προσδιορίσουμε είναι η συχνότητα λειτουργίας του κάθε ολοκληρωμένου κυκλώματος. Στη περίπτωση της SRAM και του PLL, ακολουθώντας τα data sheets προσδιορίσαμε την τιμή και το πλήθος των αποζευκτικών πυκνωτών. Στην περίπτωση όμως της FPGA δεν υπάρχει καμία εκτίμηση ούτε για την απαίτηση του ολοκληρωμένου κυκλώματος σε ρεύμα, αλλά ούτε και για τη συχνότητα λειτουργίας του. Η συχνότητα και το ρεύμα εξαρτώνται άμεσα από την εφαρμογή που θα υλοποιεί η πλατφόρμα και άρα η FPGA. Για παράδειγμα στα 3.3 Volt τροφοδοσίας, σε μια εφαρμογή που καταλαμβάνει το 80% της χωρητικότητας της FPGA και λειτουργεί

στα 15 MHz μπορεί να καταναλώνει 150 mA, ενώ στα 30 MHz 600 mA και μία εφαρμογή που καταλαμβάνει το 12% στα 10 MHz μπορεί να καταναλώνει 100 mA και στα 160 MHz 800 mA.

Εφόσον η κατανάλωση της FPGA εξαρτάται από την εφαρμογή, θεωρήσαμε ότι θα πρέπει να κάνουμε απόζευξη σε όλες τις πιθανές συχνότητες των εφαρμογών, θεωρώντας ότι αυτές καταλαμβάνουν το 100% της χωρητικότητας της FPGA. Για παράδειγμα στην έκδοση PLATO X1.0 τοποθετήσαμε:

- Για τις πολύ αργές συχνότητες χρησιμοποιήθηκε ένας μεγάλος πυκνωτής 3300 μF , ο οποίος τοποθετήθηκε σε μια άκρη της πλακέτας.
- Για μεσαίες συχνότητες χρησιμοποιήθηκαν τέσσερις πυκνωτές τανταλίου των 10 mF, σε απόσταση μικρότερη των δυο εκατοστών.
- Για τις πολύ υψηλές συχνότητες χρησιμοποιήθηκαν 40 πυκνωτές 0.01 μF για τα Vcco, σε απόσταση μικρότερη του ενός εκατοστού από το ολοκληρωμένο κύκλωμα και άλλοι 36 πυκνωτές 0,1 μF για τα Vccint, επίσης πολύ κοντά.

Σημείωση: Vcco ονομάζονται τα pins τροφοδοσίας των IOBs, ενώ Vccint τα pin τροφοδοσίας των CLBs της Virtex. Οι απαιτήσεις της FPGA στις δυο διαφορετικές περιοχές διαφέρουν, γεγονός που μας αναγκάζει να τις μελετήσουμε ξεχωριστά. Αντίστοιχες διαδικασίες ακολουθήσαμε και στην έκδοση PLATO A1.0 με Apex.

4.3.2 Επιλογή του Package των Εξαρτημάτων.

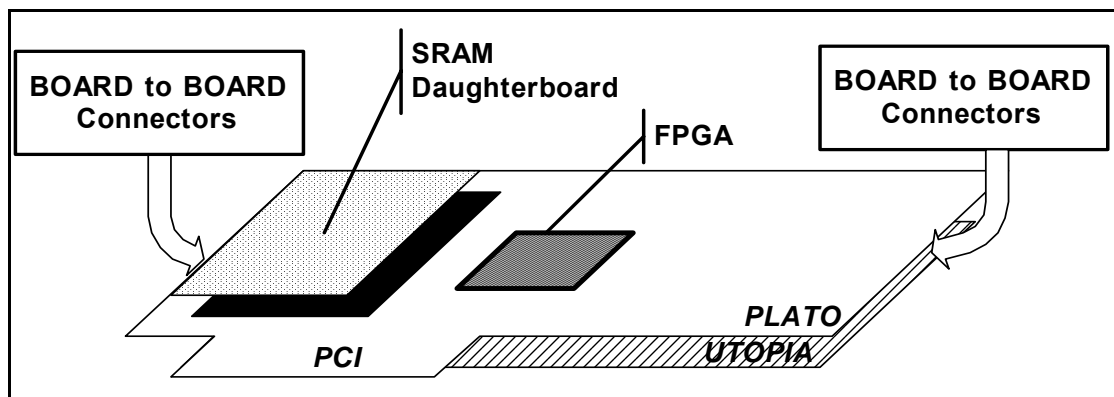
Τα εξαρτήματα γενικά μπορούμε να τα κατατάξουμε σε δυο μεγάλες κατηγορίες. Τα 'Trough Hole' και τα SMD-Surface Mound Device. Τα "Trough Hole" έχουν pins, τα οποία διαπερνούν τη πλακέτα για να κολληθούν στην επάνω ή και τη κάτω πλευρά της, ενώ τα SMD έχουν επαφές για να γίνεται η κόλλησή τους στην πλευρά που γίνεται και η τοποθέτηση του εξαρτήματος. Η βασική τους διαφορά είναι ότι τα "Trough Hole" είναι σχετικά δύσκολο να κατασκευαστούν σε πολύ μικρό package, σε αντίθεση με τα SMD που είναι δυνατό να φτάσουν και σε διαστάσεις μικρότερες του ενός χιλιοστού.

Από την παραπάνω μελέτη προέκυψαν 81 πυκνωτές, 76 από τους οποίους έπρεπε να τοποθετηθούν σε μια περιοχή πάρα πολύ κοντά στη Virtex. Για να μπορέσουμε να τους τοποθετήσουμε σε μία επιφάνεια περίπου 20 cm² έπρεπε να επιλεγθεί αρκετά μικρό package. Στη δική μας περίπτωση λοιπόν, όπου ήταν αναγκαίο να τοποθετήσουμε πολλά εξαρτήματα σε περιορισμένο χώρο επιλέξαμε τα SMD.

Αυτή η διευκόλυνση που μας παρέχουν τα SMD εξαρτήματα είναι ιδιαίτερα πολύτιμη στη δική μας περίπτωση και πιο συγκεκριμένα στην περιοχή της σχεδίασης περιμετρικά της FPGA. Όπως φαίνεται στο Σχήμα 4.7, περιμετρικά του footprint της FPGA υπάρχουν πολλοί αποζευκτικοί πυκνωτές. Αν είχαμε επιλέξει "Trough Hole" footprint για τους πυκνωτές, θα δημιουργούσαμε μια συστοιχία από κυλίνδρους στην περίμετρο, αλλά και στο εσωτερικό της FPGA. Με αυτό τον τρόπο θα περιορίζαμε πολύ τα περίπου 400 σήματα που θα πρέπει να δρομολογηθούν από το εσωτερικό της FPGA στα περιφερειακά της πλακέτας.

4.3.3 Board to Board connectors.

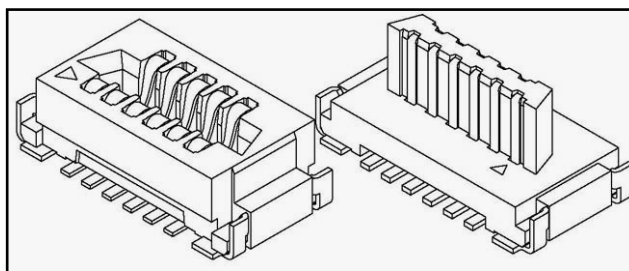
Όπως περιγράφεται και στο 3^ο κεφάλαιο η αρχιτεκτονική του συστήματος έχει χωριστεί σε τρία μικρότερα υποσυστήματα, κάθε ένα από τα οποία υλοποιείται σε μια πλακέτα. Οι πλακέτες μεταξύ τους τοποθετούνται παράλληλα, όπως φαίνεται και γραφικά στο Σχήμα 4.8 και ενώνονται με ειδικά εξαρτήματα που ονομάζονται 'Board to Board' connectors.



Σχήμα 4.8 Συνδεσμολογία των daughterboards με τη mainboard.

Η τοπολογία αλλά και το πλήθος των σημάτων από τη μια πλακέτα στη άλλη είναι τα σημαντικότερα κριτήρια με τα οποία έγινε η επιλογή αυτών των εξαρτημάτων. Το πλήθος των σημάτων ήταν 134 και 89 pin στο daughterboard της SRAM και UTOPIA αντίστοιχα. Αν προσθέσουμε και τα απαραίτητα pins για την τροφοδοσία των ολοκληρωμένων κυκλωμάτων, τότε το πλήθος των pins αυξάνονται κατά 19 και 36 αντίστοιχα. Τα εξαρτήματα 'Trough Hole' με τέτοιο πλήθος

ακροδεκτών έχουν διαστάσεις εξαιρετικά μεγάλες για το μέγεθος της πλατφόρμας ανάπτυξης. Για αυτό το λόγο επιλέχθηκαν SMD εξαρτήματα. Οι Board to Board connectors που τελικά χρησιμοποιήθηκαν είναι οι Zyton, της εταιρίας Molex (Γραφική απεικόνιση στο Σχήμα 4.9). Παράγονται σε μεγάλη ποικιλία ως προς το πλήθος των pins π.χ.(9, 11, 15, 21, 25, 31, 41, 51, 69) και το βασικό τους ιδίωμα είναι ότι έχουν ασύμμετρες συστοιχίες από pins.



Σχήμα 4.9 Board-to-Board connectors [ZYTTON].

Στη μια πλακέτα τοποθετείται ο ‘αρσενικός’ και στην άλλη ο ‘θηλυκός’. Στη δική μας εφαρμογή χρησιμοποιήθηκαν οι σειρές των 51 και των 69 pins. Εξαιτίας του μεγάλου πλήθους των επιθυμητών σημάτων από το mainboard στο daughterboard, τοποθετήσαμε παραπάνω από ένα εξάρτημα στην κάθε θύρα επέκτασης. Το πλήθος των συνδέσεων μεταξύ των πλακετών φαίνεται στον παρακάτω πίνακα.

| | I/O | Power | Σύνολο. | Συνδυασμός. |
|--------|-----|-------|---------|-------------|
| SRAM | 134 | 19 | 153 | 3 x 51 Pins |
| UTOPIA | 102 | 36 | 138 | 2 x 69 Pins |

Πίνακας 4.1 Χρήση των connectors για την διασύνδεση των υποσυστημάτων.

4.4 Δημιουργία netlist.

Με τον όρο 'netlist' ονομάζουμε την πληροφορία που περιγράφει τα εξαρτήματα ενός συστήματος αλλά και τη λεπτομερή συνδεσμολογία μεταξύ αυτών. Στη προκειμένη περίπτωση όταν αναφέρουμε τον όρο αυτό, θα εννοούμε την πλήρη συνδεσμολογία των εξαρτημάτων της πλατφόρμας ανάπτυξης, περιλαμβανομένων και των παθητικών εξαρτημάτων.

Το netlist συνήθως περιγράφεται μέσω κάποιας τυποποιημένης γλώσσας. Η γλώσσα που χρησιμοποιήθηκε και στις δυο εκδόσεις του PLATO είναι η Protel, ομώνυμη με το εργαλείο που χρησιμοποιήσαμε για τη σχεδιάσή τους. Η δομή αυτής της γλώσσας είναι αρκετά απλή, ώστε να μπορεί να περιγραφεί πλήρως στην επόμενη παράγραφο. Η απλότητά της είναι το βασικότερο προσόν της γεγονός, που όπως θα δούμε στη συνέχεια, μας βοήθησε στο να τυποποιήσουμε τη διαδικασία παραγωγής του μεγαλύτερου μέρους του netlist.

Η δομή της θα μπορούσε να χωριστεί σε δυο μεγάλα τμήματα. Το πρώτο, που ασχολείται με τη δήλωση των εξαρτημάτων της πλακέτας, αλλά και του footprint του, και το δεύτερο που περιγράφει τη συνδεσμολογία των εξαρτημάτων που δηλώθηκαν στο πρώτο μέρος.

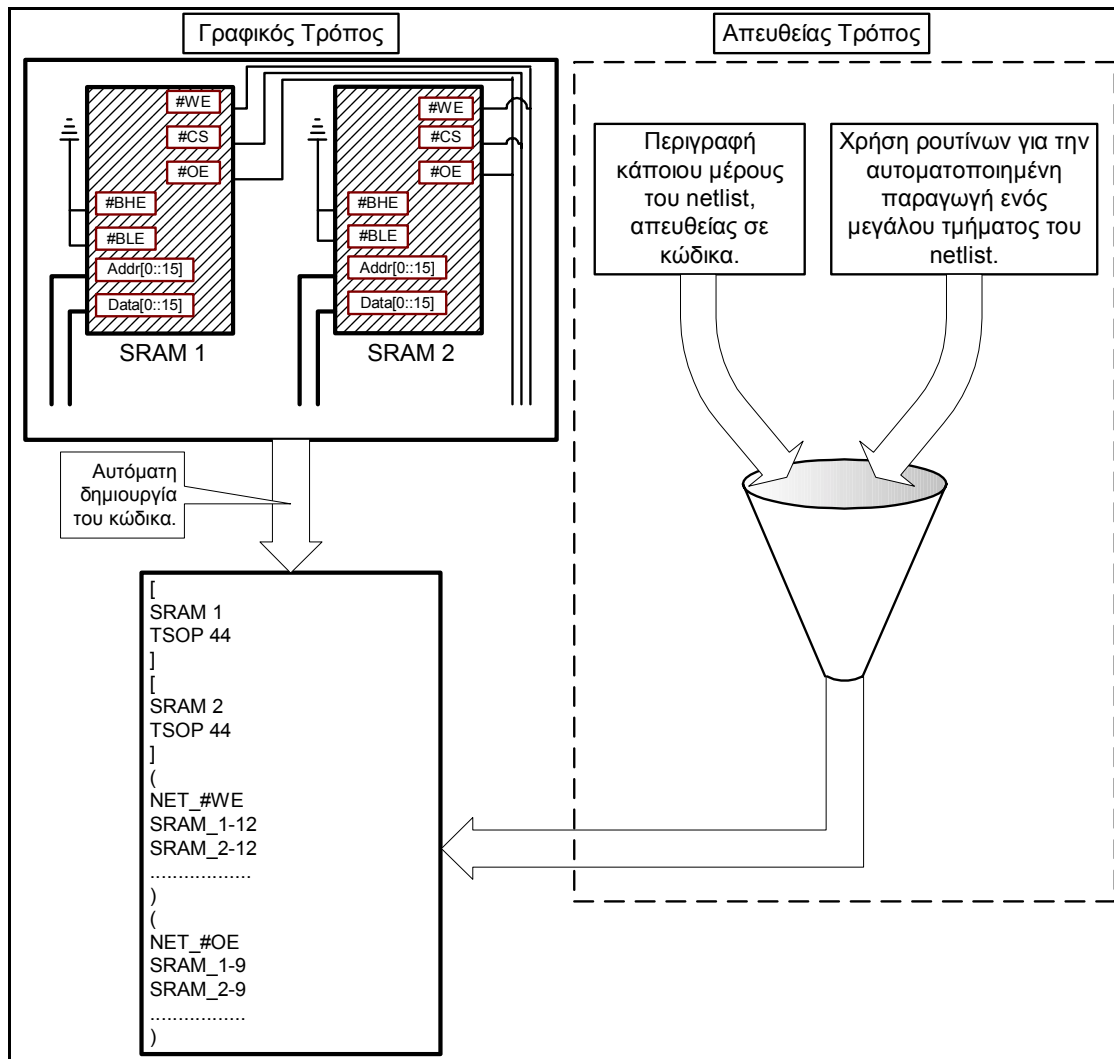
| |
|---|
| <i>Περιγραφή της γλώσσας περιγραφής netlist «Protel».</i> |
| <p><u>Μέρος Πρώτο:</u> «Περιγραφή των εξαρτημάτων».</p> <div style="border: 1px solid black; padding: 10px; margin: 5px 0;"> <ul style="list-style-type: none"> Δήλωση του ονόματος του εξαρτήματος. Δήλωση του ονόματος της βιβλιοθήκης που περιγράφει το footprint του εξαρτήματος. Προαιρετικά σχόλια, για τη περιγραφή του εξαρτήματος. </div> <p>Επανάληψη για κάθε εξάρτημα ξεχωριστά.</p> |
| <p><u>Μέρος Δεύτερο:</u> «Περιγραφή της συνδεσμολογίας των εξαρτημάτων».</p> <div style="border: 1px solid black; padding: 10px; margin: 5px 0;"> <ul style="list-style-type: none"> Δήλωση του ονόματος του net. Δήλωση όλων των pad που ανήκουν στο net αυτό. </div> <p>Επανάληψη για κάθε net ξεχωριστά.</p> |

Το netlist μπορεί να δημιουργηθεί με δυο τρόπους. Το γραφικό τρόπο και με την απευθείας περιγραφή του με κώδικα της παραπάνω μορφής. Ο κάθε τρόπος έχει τα πλεονεκτήματα και τα μειονεκτήματά του. Τα περισσότερα εργαλεία σχεδιασμού,

όπως το Protel, υποστηρίζουν τη γραφική διαδικασία. Με αυτή τη μέθοδο ο σχεδιαστής ενώνει γραφικά τα pads των εξαρτημάτων που ανήκουν στο ίδιο net. Όταν ολοκληρωθεί όλη η αρχιτεκτονική με αυτό το τρόπο, το εργαλείο αυτόματα δημιουργεί τον αντίστοιχο κώδικα για το netlist. Αυτός ο κώδικας είναι και η είσοδος για την επόμενη βαθμίδα σχεδίασης του εργαλείου, το routing. Το βασικό πλεονέκτημα αυτής της μεθόδου είναι ότι ο σχεδιαστής έχει καλύτερο έλεγχο της συνδεσμολογίας των εξαρτημάτων, έχοντας στη διάθεσή του τη γραφική απεικόνιση του συστήματος. Το μειονέκτημα αυτής της μεθόδου είναι ότι σε κάποιο πιθανό λάθος στη δημιουργία του netlist, η διόρθωσή του είναι εξαιρετικά χρονοβόρα, ιδιαίτερα σε τέτοιας κλίμακας συστήματα όπως του PLATO, όπου τα nets είναι περίπου 600. Κάτι τέτοιο θα μπορούσε να συμβαίνει αν με την αλλαγή ενός σήματος στο netlist αναδρομικά προκληθούν και άλλες αλλαγές στο σύστημα.

Ο δεύτερος τρόπος, εφόσον δεν είναι γραφικός, δεν μας παρέχει την άνεση του οπτικού ελέγχου της συνδεσμολογίας του κυκλώματος, αλλά μας προσφέρει τη δυνατότητα, λόγω της απλότητάς του, να αυτοματοποιήσουμε τη διαδικασία παραγωγής του netlist με τον ακόλουθο τρόπο.

Αν παρατηρήσουμε την αρχιτεκτονική του 3^{ου} κεφαλαίου θα προσέξουμε ότι η πλειοψηφία των nets είναι ένα σύνολο από δυο pads, ένα I/O της FPGA και ένα κάποιου άλλου εξαρτήματος. Γνωρίζοντας ότι τα I/O της FPGA δεν διαφέρουν μεταξύ τους ως προς το τρόπο λειτουργίας τους, έχουμε την άνεση να τα χρησιμοποιούμε χωρίς διάκριση για τα nets των περιφερειακών εξαρτημάτων. Δημιουργώντας τρεις λίστες, μια με όλα τα διαθέσιμα I/O της FPGA, μια με τα αντίστοιχα pads των περιφερειακών εξαρτημάτων και τέλος μια με τα κατάλληλα ονόματα των nets, έχουμε όλη την πληροφορία που χρειάζεται για τη δημιουργία του μεγαλύτερου μέρους του netlist. Εισάγοντας αυτές τις τρεις λίστες σε ένα σχετικά απλό πρόγραμμα σε C παίρνουμε ως έξοδο το κατάλληλο κώδικα για το netlist. Το υπόλοιπο τμήμα του netlist το παράγουμε με απευθείας κώδικα. Αυτό το τμήμα είναι κυρίως οι δηλώσεις των εξαρτημάτων και η υπόλοιπη συνδεσμολογία του συστήματος. Πρακτικά μπορούμε να αναφέρουμε ότι το netlist για την έκδοση PLATO A1.0 είναι περίπου 4800 γραμμές, εκ των οποίων μόνο οι 1900 γράφτηκαν απευθείας σε κώδικα. Μια γραφική απεικόνιση των δυο μεθόδων φαίνεται στο Σχήμα 4.10.



Σχήμα 4.10 Μέθοδοι δημιουργίας του Netlist.

Δυο βασικά προτερήματα της αυτοματοποιημένης παραγωγής του κώδικα είναι η ταχύτητα επιδιόρθωσης σφαλμάτων και η δυνατότητα ελέγχου του netlist. Επιδιορθώνοντας τις τρεις λίστες, Netlist name, Component 1 και Component 2, και ελέγχοντας τη πληροφορία τους σχετικά εύκολα, παράγουμε το νέο κώδικα για το netlist μέσα σε ελάχιστο χρόνο.

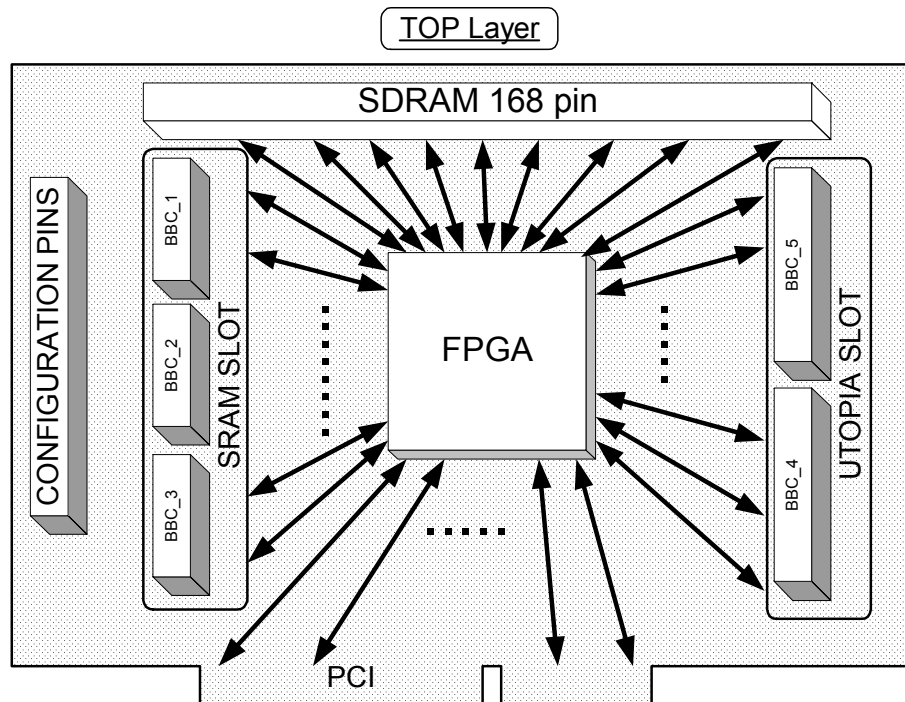
4.5 Placement, Rules, Route and DRC.

Σε αυτή την ενότητα θα αναφερθούμε στη διαδικασία σχεδίασης του τυπωμένου κυκλώματος. Πιο συγκεκριμένα, θα περιγράψουμε με ποιο τρόπο μπορεί να επηρεαστεί η ποιότητα και η ταχύτητα της σχεδίασης από διάφορους παράγοντες, όπως η σωστή τοποθέτηση των εξαρτημάτων σε σχέση με το κατάλληλο netlist, την επιλογή των κατάλληλων παραμέτρων και κανόνων σχεδίασης, καθώς και με ποιο τρόπο μπορούμε να ελέγξουμε τη σχεδίαση ως προς την ορθότητά της. Έχοντας μελετήσει και ολοκληρώσει τις προηγούμενες φάσεις της σχεδίασης είναι πολύ πιθανό να έχουμε ήδη καταλήξει σε κάποιες παραμέτρους, όπως το πλήθος των επιπέδων ή τη γενικότερη τοπολογία των εξαρτημάτων πάνω στην επιφάνεια σχεδίασης. Παρόλα αυτά, είναι σημαντικό αυτή η φάση να εξεταστεί εκτενέστερα, περιγράφοντας κρίσιμα σημεία που είναι δυνατό να στοιχίσουν στην αποτελεσματικότητα του εργαλείου κατά τη διαδικασία της σχεδίασης.

4.5.1 Τοποθέτηση εξαρτημάτων & δημιουργία netlist.

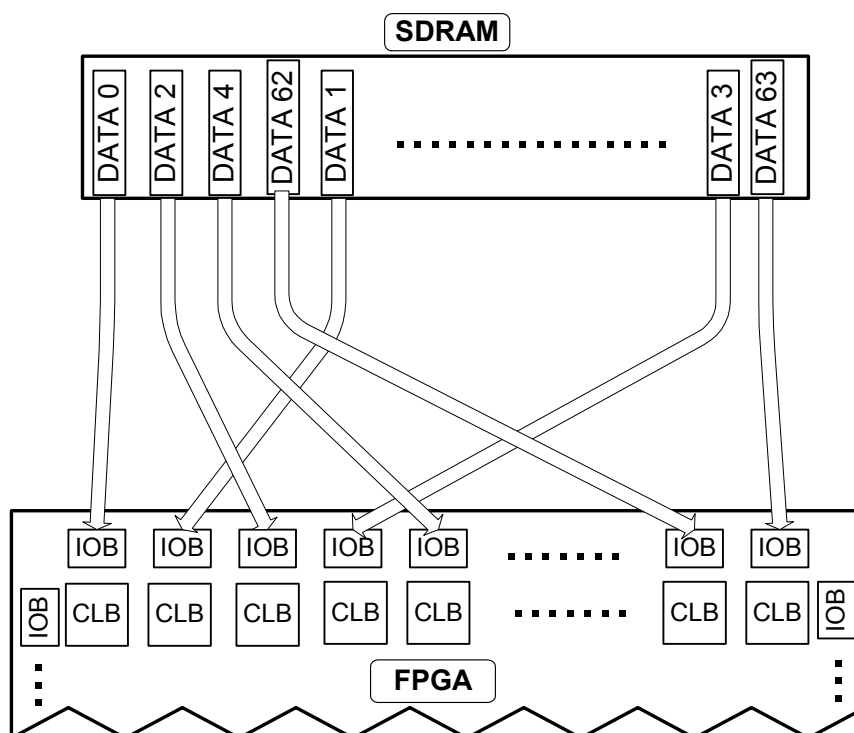
Η επιλογή της κατάλληλης τοπολογίας των εξαρτημάτων έχει άμεση σχέση με τη διασύνδεση των daughterboards με την mainboard, αλλά και με τη δομή του netlist. Πρώτιστος στόχος είναι να τοποθετηθούν τα εξαρτήματα με τρόπο τέτοιο, ώστε με το κατάλληλο netlist να μεταφέρονται τα σήματα σχεδόν παράλληλα από τη μια περιοχή στην άλλη. Αν οι κατευθύνσεις των σημάτων τέμνονται, το εργαλείο σχεδίασης θα αναγκαστεί να μεταφέρει διαδοχικά τον αγωγό του κάθε net από το ένα επίπεδο στο άλλο με τρόπο τέτοιο, ώστε να δημιουργήσει γεφυρώσεις, πάνω ή κάτω από τις οποίες θα σχεδιαστούν τα εγκάρσια σήματα. Αυτές οι εναλλαγές επιπέδου για κάθε σήμα μεταφράζονται σε αύξηση του πλήθους των vias και άρα της απαιτούμενης επιφάνειας για την δρομολόγηση των υπόλοιπων σημάτων επιβαρύνοντας με αυτό το τρόπο και το κόστος της κατασκευής, αλλά και ελαχιστοποίηση της μέγιστης δυνατής συχνότητας λειτουργίας της πλατφόρμας. Πρακτικά, σε πειραματικές απόπειρες σχεδίασης με σκόπιμη τοποθέτηση κάποιων σημάτων εγκάρσια στην αρχιτεκτονική του PLATO X1.0, παρατηρήσαμε ότι

αυξάνεται εξαιρετικά ο χρόνος σχεδίασης από το router του εργαλείου που χρησιμοποιήσαμε. Εκτός του μεγάλου χρόνου υλοποίησης παρατηρήθηκε και αύξηση του μήκους των αγωγών περίπου του 20% των σημάτων σε σχέση με τα υπόλοιπα nets.



Σχήμα 4.11 Τοπολογία και διασύνδεση της FPGA με τα περιφερειακά της.

Για αυτούς του λόγους έπρεπε η τοπολογία των εξαρτημάτων να είναι τέτοια, ώστε τα σήματα να μη τέμνονται. Γνωρίζοντας ότι στην γενική αρχιτεκτονική του PLATO μια μεγάλη FPGA ενώνεται με τέσσερα βασικά υποσυστήματα, το PCI, το UTOPIA, την SDRAM και τις SRAMs τοποθετήσαμε την FPGA στο κέντρο της πλατφόρμας και σε ακτινωτή διάταξη τα τέσσερα περιφερειακά όπως φαίνεται στο Σχήμα 4.11. Αυτή η τοπολογία είναι η ιδανικότερη και προκύπτει αν κατά τη δημιουργία του netlist φροντίσουμε, όπου είναι δυνατό, να σχεδιάσουμε τα nets του κάθε εξαρτήματος, μεταξύ τους όσο γίνεται παράλληλα. Πιο συγκεκριμένα, στην SRAM θύρα μπορούσαμε, χωρίς κανένα περιορισμό, να σχεδιάσουμε σχεδόν παράλληλα τα σήματα από την FPGA προς την θύρα, εφόσον συνδέουμε το connector με την FPGA. Το netlist του connector της SRAM μπορούμε χωρίς κανένα περιορισμό να το σχεδιάσουμε ακριβώς όπως μας διευκολύνει στη δρομολόγηση των σημάτων.

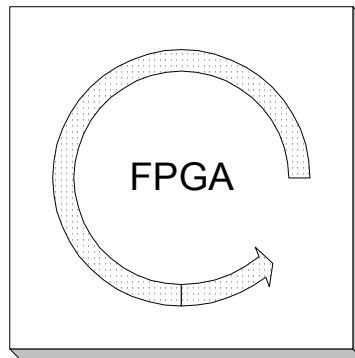


Σχήμα 4.12 Δημιουργία του Netlist σύμφωνα με το VLSI της FPGA, για αποδοτικότερο routing της εφαρμογής.

Αντίθετα όμως, στο PCI και στην SDRAM δεν μπορέσαμε να ακολουθήσουμε την ίδια τεχνική σε όλη την έκταση της συνδεσμολογίας. Αυτό οφείλεται στο γεγονός ότι προτιμήσαμε να κάνουμε κάποιου είδους βελτιστοποίηση όσον αφορά την σχεδίαση των εφαρμογών στο εσωτερικό της FPGA. Όλα τα περιφερειακά εξαρτήματα της FPGA, έχουν κάποια σήματα που ανήκουν στον ίδιο δίαυλο. Για παράδειγμα οι SRAM έχουν το δίαυλο δεδομένων και το δίαυλο της διεύθυνσης. Το PCI έχει το δίαυλο AD[0::31], που είναι για τα δεδομένα και τις διευθύνσεις και η SDRAM έχει τα AD[0::12] και Data[0::63]. Η βελτιστοποίηση που αναφέραμε έχει να κάνει με το γεγονός ότι κάθε ένας από αυτούς τους διαύλους πρέπει να χρησιμοποιείται σαν μια οντότητα και όχι σαν μια ομάδα από σήματα. Η κάθε εφαρμογή μέσα στην FPGA χειρίζεται αυτούς τους διαύλους μέσω ενός υποσυστήματός της. Η λογική αυτού του υποσυστήματος συνήθως τοποθετείται στην περιοχή που βρίσκεται πολύ κοντά στα pins που χρησιμοποιούνται από την εφαρμογή, ώστε να επιτευχθεί υψηλότερη συχνότητα λειτουργίας. Αν λοιπόν από τη δημιουργία του netlist, τα σήματα του διαύλου της SDRAM, Data[0::63] είχαν τοποθετηθεί σε απομακρυσμένα μεταξύ τους σημεία της FPGA, δεν θα μπορούσε η αντίστοιχη λογική του υποσυστήματος να τοποθετηθεί αρκετά κοντά σε όλα τα

σήματα. Για αυτό το λόγο φροντίσαμε, μέσω του netlist, να τοποθετήσουμε τα σήματα των διαύλων σε διαδοχικά pins της FPGA, όπως φαίνεται γραφικά στο Σχήμα 4.12.

Για να διευκολύνουμε τη διαδικασία της δημιουργίας του netlist αναφέραμε στο κεφάλαιο 4.4 τη μεθοδολογία της περιγραφής του με απευθείας κώδικα. Το μόνο που χρειάζεται να κάνουμε είναι να δημιουργήσουμε τρεις λίστες. Η μια με τα ονόματα των nets και οι άλλες δυο με τα ονόματα των pins των δυο εξαρτημάτων αντίστοιχα. Αυτό που χρειαζόταν λοιπόν να κάνουμε, ήταν να τοποθετήσουμε τα ονόματα των pins της FPGA σε μια λίστα καθώς διανύουμε αριστερόστροφα την περίμετρό της, όπως περιγράφεται από το Σχήμα 4.13. Η σειρά τοποθέτησης των pins της εκάστοτε FPGA στην περίμετρο του package της δεν είναι ενδεικτικός της σειράς με την οποία παρουσιάζονται και στο VLSI της, η οποία και μας ενδιαφέρει. Για αυτό το λόγο, στη μεν έκδοση του PLATO X1.0 (με τη Virtex της εταιρίας Xilinx), η λίστα αυτή δημιουργήθηκε με τη χρήση του FPGA Editor του Foundation 3.1, ενώ στη περίπτωση του PLATO A1.0 με τη χρήση του Quartus.



Σχήμα 4.13 Αριστερόστροφη επιλογή των I/O της FPGA για την διασύνδεσή της με τα περιφερειακά.

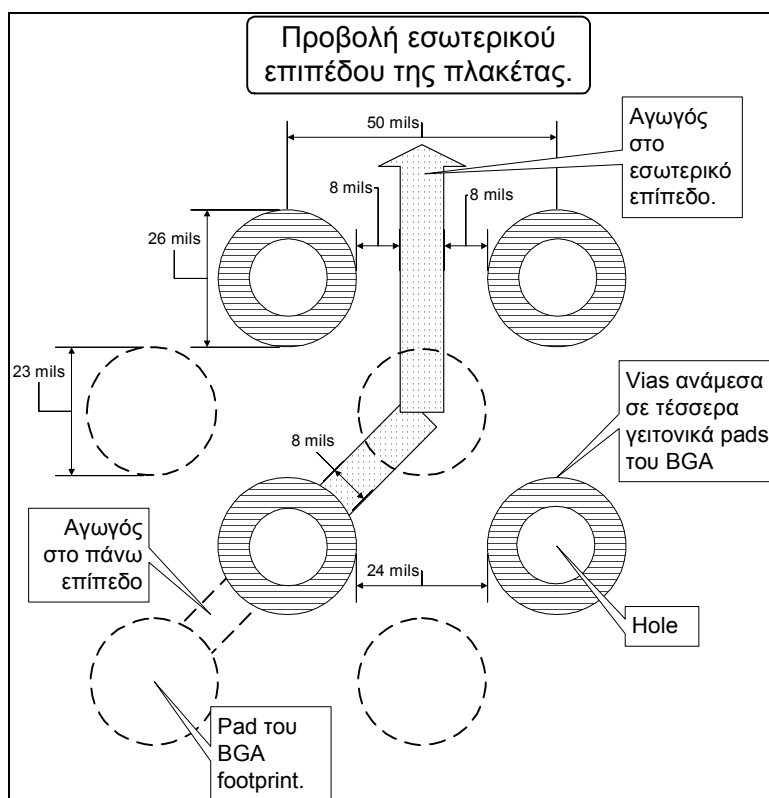
4.5.2 Κανόνες σχεδίασης.

Οι κανόνες σχεδίασης ενός συστήματος μεταβάλλονται ανάλογα με το ίδιο το σύστημα, τις δυνατότητες της κατασκευάστριας εταιρείας του τυπωμένου κυκλώματος και τα εξαρτήματα που χρησιμοποιούνται σε αυτό. Σκοπός τους είναι να παρέχουν στο εργαλείο σχεδίασης τις απαραίτητες παραμέτρους και προδιαγραφές του συστήματος που προέκυψαν από τη μελέτη των προηγούμενων φάσεων. Κάποιοι

από τους βασικούς κανόνες αφορούν στο πλάτος και το μήκος των αγωγών, στη διάμετρο των vias και των pads, στο clearance κτλ.

Πιο αναλυτικά στη δική μας περίπτωση έχουμε:

- Η διάμετρος των vias, στη περιοχή της FPGA, είναι 26 mils με 12 mils διάκενο. Ο λόγος που επιλέχθηκε αυτό το via είναι γιατί ο χώρος που δημιουργείται ανάμεσα σε τέσσερα γειτονικά pads του BGA footprint των FPGA είναι περιορισμένος. Η σειρά παραγωγής της Intracom δεν μπορεί να κατέβει κάτω από τα 12 mils τρύπα.

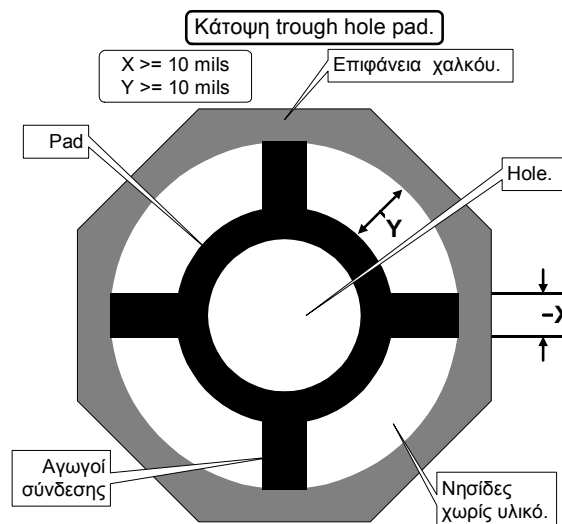


Σχήμα 4.14 Αποστάσεις των Vias και των αγωγών στη περιοχή των FGAs.

- Η διάμετρος των vias είναι 40 mils με 20 mils hole σε όλη την υπόλοιπη έκταση της πλακέτας, εκτός της περιοχής της FPGA.
- Το πλάτος των αγωγών και το clearance σύμφωνα και με το Σχήμα 4.14 είναι 8 mils. Παρά το γεγονός ότι η κατασκευάστρια εταιρεία είχε τη δυνατότητα να κατασκευάσει αγωγό μέχρι και 6 mils, δεν χρειάστηκε, εφόσον δεν θα μπορούσαμε με αυτό το πλάτος αγωγού να μειώσουμε τα επίπεδα της πλακέτας. Έτσι, προτιμήθηκε να μοιράσουμε την απόσταση μεταξύ δυο

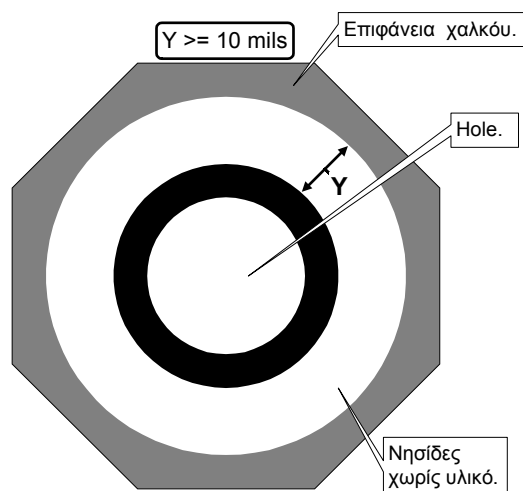
γειτονικών vias 24 mils, των FPGAs στο πλάτος του αγωγού και στο clearance ($24/3=8$ mils).

Οι ενώσεις των σημάτων με τα επίπεδα της πλακέτας γίνονται με Thermal συνδέσεις. Αυτές οι συνδέσεις υπάρχουν για την ασφάλεια του κυκλώματος κατά τη συγκόλληση των εξαρτημάτων πάνω στη πλακέτα. Αυτό συμβαίνει για παράδειγμα στα pads της SDRAM, που θα πρέπει να ενωθούν στο 'power plane' επίπεδο τροφοδοσίας της πλακέτας. Η περιοχή ανάμεσα στον κύλινδρο του pad στο ύψος του επιπέδου με το υλικό του επιπέδου πρέπει να καλυφθεί με χαλκό, ώστε να επιτευχθεί ένωση. Το πρόβλημα παρατηρείται κατά τη διάρκεια της κόλλησης του εξαρτήματος. Η θερμότητα που χρησιμοποιείται για να θερμανθούν τα pads και να λιώσει ο κασσίτερος, επιτυγχάνοντας έτσι την κόλληση, μεταφέρεται εξ επαφής και στο υλικό του power plane. Υπάρχει λοιπόν ο κίνδυνος με την υπερθέρμανση της συγκεκριμένης περιοχής του plane και λόγω της θερμικής διαφοράς με την γύρω περιοχή να προκληθεί στρέβλωση του χαλκού του επιπέδου τροφοδοσίας και τοπική αποκόλληση των επιπέδων της πλακέτας. Στις thermal ενώσεις η σύνδεση του κυλίνδρου (pad) με το επιθυμητό επίπεδο τροφοδοσίας επιτυγχάνεται με την τοποθέτηση δυο ή τεσσάρων αγωγών, αφήνοντας με αυτό το τρόπο δυο ή τέσσερις τοξωτές νησίδες αντίστοιχα χωρίς υλικό, όπως φαίνεται και στο Σχήμα 4.15. Έτσι, η θερμότητα της κόλλησης δεν μεταφέρεται στο υλικό του επιπέδου και αποφεύγονται ανεπιθύμητες στρεβλώσεις και αποκολλήσεις των επιπέδων.



Σχήμα 4.15 Thermal pad.

- Η περιοχή ανάμεσα στον κύλινδρο ενός via η ενός pad και του υλικού του επιπέδου, γνωστή και σαν ‘Annular Ring’ ή ‘Donald’, πρέπει από τις προδιαγραφές της Intracom να είναι το λιγότερο 10 mils. Στο Σχήμα 4.16 το μέγεθος αυτό εκφράζεται με το ‘Y’.



Σχήμα 4.16 Annular Ring.

- Μια επιπλέον παράμετρος σχεδίασης σε μια πολυεπίπεδη πλακέτα είναι η σειρά με την οποία διατάσσονται τα επίπεδα τροφοδοσίας με τα επίπεδα σημάτων. Η βασική αρχή σε μια πλακέτα με δυο επίπεδα τροφοδοσίας ή παραπάνω από δυο είναι να τοποθετούνται τα επίπεδα τροφοδοσίας (power planes) όσο γίνεται πιο απομακρυσμένα μεταξύ τους, ώστε να μην λειτουργούν σαν πυκνωτές, λόγω της μικρής τους απόστασης. Στις εκδόσεις του PLATO προτιμήθηκε η εξής διάταξη.

| Επίπεδα. | PLATO X1.0 | | PLATO A1.0 | |
|----------|------------|----------|------------|----------|
| 1 | Top | Σήματα | Top | Σήματα |
| 2 | Internal 1 | 3.3 Volt | Internal 1 | 3.3 Volt |
| 3 | Internal 2 | Σήματα | Internal 2 | Σήματα |
| 4 | Internal 3 | Σήματα | Internal 3 | Σήματα |
| 5 | Internal 4 | Σήματα | Internal 4 | Σήματα |
| 6 | Internal 5 | Σήματα | Internal 5 | 0 Volt |
| 7 | Internal 6 | 0 Volt | Internal 6 | Σήματα |
| 8 | Bottom | Σήματα | Internal 7 | Σήματα |
| 9 | NA | NA | Internal 8 | 1.8 Volt |
| 10 | NA | NA | Bottom | Σήματα |

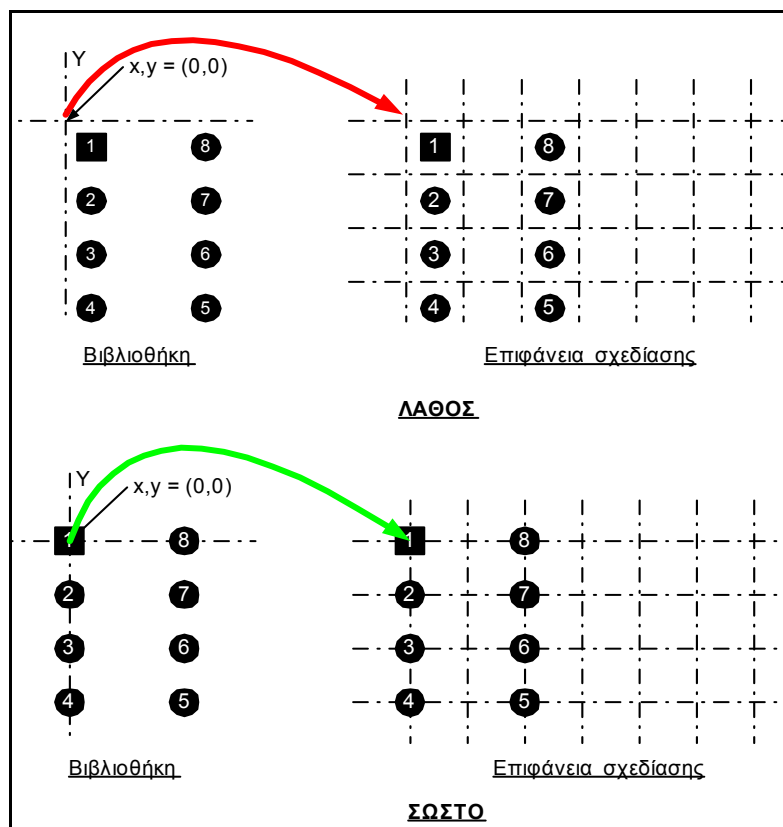
- ο Ένα άλλο χαρακτηριστικό, που υπάρχει η δυνατότητα να ελεγχθεί, είναι το μέγιστο μήκος των σημάτων. Κατά τη διάρκεια της σχεδίασης, αν το εργαλείο δεν έχει τη δυνατότητα να ικανοποιήσει αυτόν το κανόνα θα ολοκληρώσει τη δρομολόγηση όλων των σημάτων και στο τέλος θα δώσει αναφορά για το τελικό μήκος των σημάτων αυτών. Τέτοιοι κανόνες μπορούν να φανούν πολύ χρήσιμοι σε κρίσιμα nets, όπως αυτό του ρολογιού. Και στις δυο εκδόσεις του PLATO ο αγωγός του ρολογιού ρυθμίστηκε έτσι, ώστε να είναι όσο τα $2/3$ της διαγωνίου της πλακέτας, θεωρώντας εμπειρικά αυτό το νούμερο ιδανικό από πλευράς υλοποίησης και λειτουργίας.
- ο Επίσης, το κάθε επίπεδο της πλακέτας ορίστηκε έτσι, ώστε οι αγωγοί που θα βρίσκονται επί του επιπέδου αυτού να έχουν κάθετη τοπολογία σε σχέση με τους αγωγούς των δύο εκατέρωθεν επιπέδων, για να αποφευχθούν φαινόμενα ρεύματος εξ επαγωγής.

4.5.3 Δρομολόγηση σημάτων (Routing).

Το στάδιο της δρομολόγησης των σημάτων ή αλλιώς routing είναι το πιο κρίσιμο στάδιο της σχεδίασης. Όλες οι προηγούμενες φάσεις είναι η προεργασία για την επιτυχή ολοκλήρωση του routing. Υπάρχουν δύο τρόποι με τους οποίους μπορεί να υλοποιηθεί. Σε μικρής κλίμακας και πολυπλοκότητας συστήματα συνηθίζεται και είναι χρήσιμο να γίνεται απευθείας από το σχεδιαστή, χωρίς τη χρήση κάποιου router εργαλείου. Αντίθετα, σε πολυεπίεδα τυπωμένα κυκλώματα, αυξημένης πολυπλοκότητας όπως το PLATO, είναι απαραίτητη η χρήση routing προγράμματος, εφόσον το πλήθος των σημάτων είναι της τάξεως των μερικών εκατοντάδων. Στο εμπόριο υπάρχουν πολλά προγράμματα για το routing και αρκετές εταιρείες που ασχολούνται μόνο με την παραγωγή κατάλληλων routing αλγορίθμων. Παρόλα αυτά, η διαφορά στην απόδοση τους, κυρίως ως προς το χρόνο ολοκλήρωσης της σχεδίασης, μπορεί να επηρεασθεί εξαιρετικά από τις ρυθμίσεις του σχεδιαστή.

Η αντίληψη της αρχής με την οποία λειτουργεί ο router είναι ο σημαντικότερος παράγοντας στη φάση αυτή. Η επιφάνεια της σχεδίασης έχει τη μορφή πλέγματος ή αλλιώς grid (διακριτό σύστημα συντεταγμένων), στα διακριτά σημεία του οποίου επιτρέπεται η σχεδίαση και τοποθέτηση εξαρτημάτων, αλλά και οποιωνδήποτε άλλων σημμάτων που είναι απαραίτητα, όπως αγωγοί, pads, vias κτλ. Κάθε ένα από τα

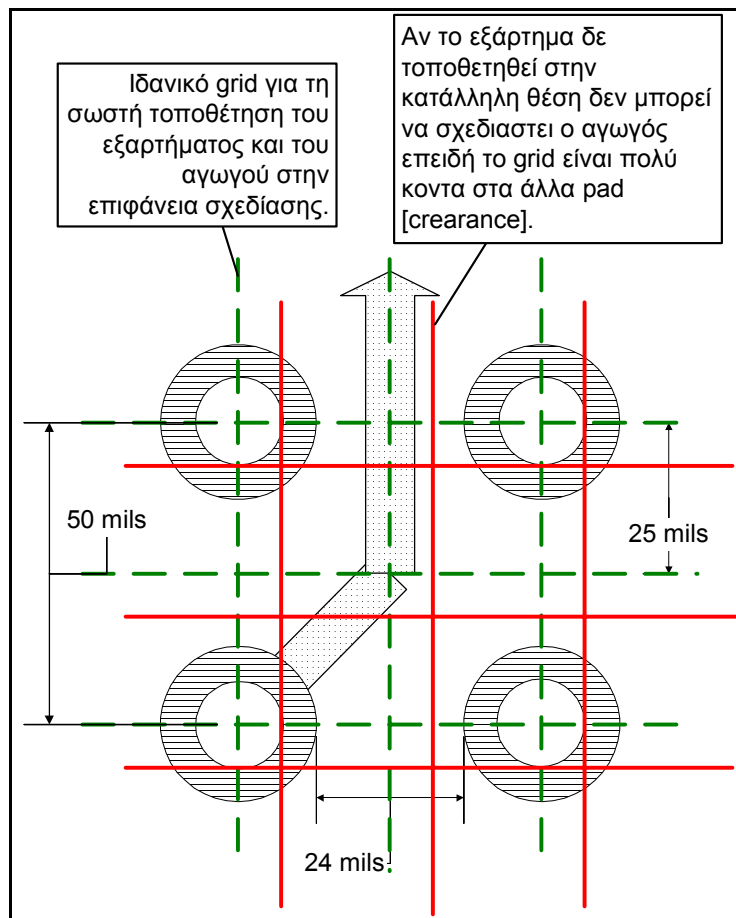
παραπάνω είναι σχεδιασμένο στις βιβλιοθήκες του εργαλείου, όπως τα εξαρτήματα ή είναι ήδη έτοιμο από το ίδιο το πρόγραμμα, όπως είναι οι αγωγοί και τα pads. Το βασικό τους χαρακτηριστικό, είναι ότι όλα σχεδιάζονται σε ένα σύστημα συντεταγμένων X-Y, σαν και αυτό της επιφάνειας σχεδίασης. Όσο πιο πυκνό είναι αυτό το grid της επιφάνειας σχεδίασης, τόσο περισσότερες πιθανές θέσεις τοποθέτησης υπάρχουν σε μια σταθερής έκτασης επιφάνεια, και ανάλογα όσο πιο αραιό είναι τόσο λιγότερες.



Σχήμα 4.17 Αποθήκευση των εξαρτημάτων σύμφωνα με το σημείο αναφοράς (0,0).

Το σημαντικό στοιχείο κατά την τοποθέτηση των εξαρτημάτων είναι το σημείο αναφοράς τους. Όπως προαναφέρθηκε, όλα τα εξαρτήματα βρίσκονται σχεδιασμένα με ακρίβεια σε κατάλληλες βιβλιοθήκες του εργαλείου. Περιγράφονται και αυτά σε ένα σύστημα συντεταγμένων X-Y. Η τοποθέτησή τους στην επιφάνεια εργασίας και σχεδίασης της πλατφόρμας ανάπτυξης γίνεται με βάση το σημείο αναφοράς της βιβλιοθήκης του εξαρτήματος $x,y = (0,0)$. Αυτό το σημείο ταυτίζεται με το σημείο τοποθέτησης στην επιφάνεια σχεδίασης. Αν το εξάρτημα κατασκευαστεί σωστά μεν, αλλά το σημείο αναφοράς $x,y=(0,0)$ της βιβλιοθήκης δεν είναι στο πρώτο pad του

εξαρτήματος, όπως συνηθίζεται, αλλά σε κάποιο άλλο τυχαίο σημείο, τότε το πιο πιθανό είναι τα pads του να μην συμπίπτουν με το grid της επιφάνειας σχεδίασης, όπως περιγράφεται γραφικά από το Σχήμα 4.17 Σε αυτή την περίπτωση, το πρόγραμμα σχεδίασης θα το ενώσει με το κατάλληλο net παρά το γεγονός ότι το pad δεν βρίσκεται πάνω στο grid. Το πρόβλημα παρουσιάζεται όταν το εξάρτημα έχει πυκνό footprint σε σχέση με το grid, όπως των δυο FPGAs που έχουν BGA package. Σε αυτή την περίπτωση είναι πολύ πιθανό λόγω της μικρής απόστασης μεταξύ των pads, 50 mils, να μην υπάρχει χώρος για να τοποθετηθεί ούτε ένας αγωγός ανάμεσά τους, όπως φαίνεται και από το Σχήμα 4.18.



Σχήμα 4.18 Επιλογή του κατάλληλου Grid και τοποθέτηση σε αυτό των εξαρτημάτων.

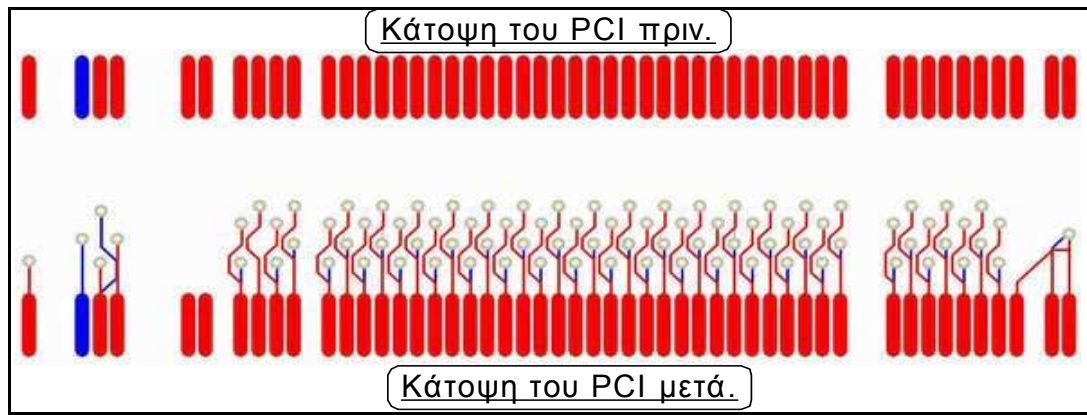
Κατά τη διαδικασία του routing οι αλγόριθμοι λειτουργούν έχοντας σαν δεδομένα τα δυο σημεία που πρέπει να ενώσουν και τα διακριτά σημεία της επιφάνειας μεταξύ των οποίων μπορούν 'κινηθούν' και να τοποθετήσουν διάφορα εξαρτήματα. Όσο πιο πολλά είναι τα σημεία αυτά, τόσο πιο πολλές είναι οι δυνατές διαδρομές που μπορούν να σχεδιάσουν. Εκ πρώτης όψεως, κάτι τέτοιο είναι πλεονέκτημα στη σχεδίαση, γιατί προσφέρεται μεγαλύτερη ακρίβεια, αλλά και η

δυνατότητα να παρακαμφθούν διάφορα εμπόδια που πιθανόν να εμποδίζουν στο routing. Το τίμημα αυτής της περίπτωσης είναι ότι οι πολλαπλές επιλογές στοιχίζουν στη κατανάλωση της υπολογιστικής ισχύος του υπολογιστή, ο οποίος αναλαμβάνει το routing, με αποτέλεσμα να προστίθεται επιπλέον χρόνος στην ολοκλήρωση της σχεδίασης (χρειάζονται περισσότερες πράξεις για να γίνει η επιλογή μιας διαδρομής όταν οι πιθανές διαδρομές είναι πολλές, παρά όταν είναι λίγες).

Παρότι και οι δυο παραπάνω συλλογισμοί είναι σωστοί, έρχονται σε σύγκρουση μεταξύ τους. Το ιδανικότερο θα ήταν να μπορούσαμε να είχαμε τα πλεονεκτήματα και των δυο περιπτώσεων. Δηλαδή, αρκετά μικρό grid για να έχουμε ακρίβεια και ταυτόχρονα αρκετά μεγάλο για γρήγορη επεξεργασία των δεδομένων. Στο PLATO, το ιδανικό ήταν να ευθυγραμμιστούν όλα τα εξαρτήματα και να βρεθεί το μέγιστο δυνατό grid που να τα εξυπηρετεί όλα. Για παράδειγμα, το μέγιστο δυνατό grid για το BGA footprint των FPGAs ήταν 25 mils, όπως φαίνεται και στο Σχήμα 4.18. Κάνοντας την ίδια διαδικασία και στα άλλα εξαρτήματα της πλατφόρμας και φροντίζοντας το grid να είναι ακέραια υποδιαίρεση του grid που ήδη έχει βρεθεί από τη μελέτη των προηγούμενων εξαρτημάτων, καταλήξαμε στο τελικό grid = 5 mils. Το σχέδιο θα μπορούσε να σχεδιαστεί και με 25 mils grid μιας και τα άλλα εξαρτήματα θα μπορούσαν να γίνουν rout, αλλά δεν θα υπήρχε αρκετά μεγάλη ακρίβεια στη σχεδίαση από ότι αποδείχτηκε σε πειραματικά routing.

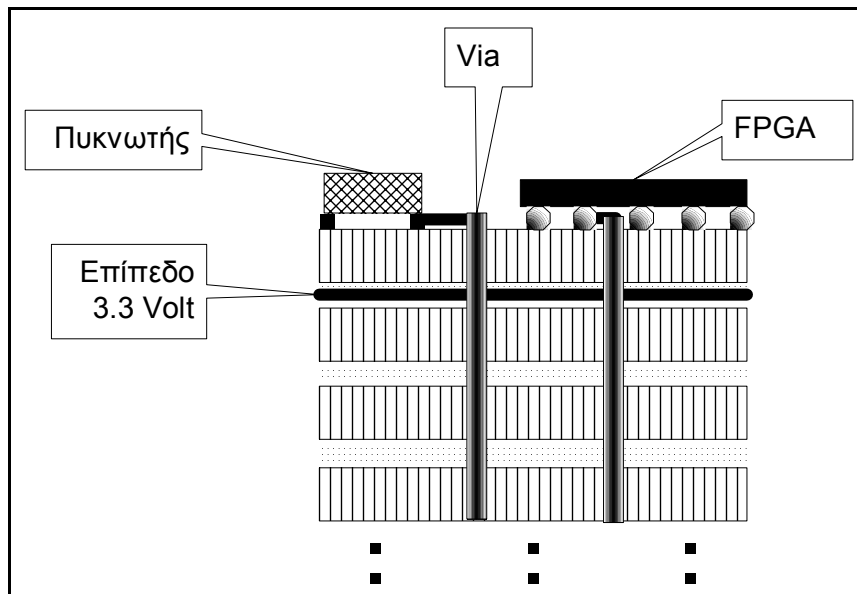
Εκτός της επιλογής του κατάλληλου grid, αλλά και της κατασκευής των βιβλιοθηκών των εξαρτημάτων με εξαιρετικά μεγάλη ακρίβεια είναι πιθανό να χρειαστεί και κάποια επιπλέον παρέμβαση από το σχεδιαστή. Αυτό συνήθως γίνεται αντιληπτό από τις δοκιμαστικές προσπάθειες για routing. Στο PLATO παρατηρήθηκε ότι υπήρχαν κάποιες περιοχές ιδιαίτερα δύσκολες στο routing, όπως το PCI και οι πυκνωτές.

Στο PCI, παρότι τα σήματα ήταν σχετικά λίγα, το γεγονός ότι δεν ήταν διαθέσιμα σε όλα τα επίπεδα ανάγκαζε τον router να τοποθετεί vias. Επειδή όμως το component για το PCI έχει δυο όψεις απόλυτα ευθυγραμμισμένες ως προς τη κάτοψή τους, τα vias του πάνω επιπέδου και του κάτω επιπέδου δημιουργούσαν εξαιρετικά μεγάλη πυκνότητα. Το αποτέλεσμα ήταν να μην μπορούν να δρομολογηθούν επιτυχώς το 10% των σημάτων. Το πρόβλημα λύθηκε, σχεδιάζοντας με το χέρι πριν το στάδιο του autorouting μια διάταξη από nets και vias όπως φαίνεται στο Σχήμα 4.19.



Σχήμα 4.19 Manual routing του PCI.

Αντίστοιχα συνέβη και με τους αποζευκτικούς πυκνωτές υψηλών συχνοτήτων. Σύμφωνα με τη μελέτη στο κεφάλαιο 4.3.1 πρέπει να τοποθετηθούν σε πολύ μικρή απόσταση από την FPGA για τη σωστή τους λειτουργία. Επειδή είναι SMD, το πιο πιθανό είναι να χρησιμοποιηθούν νίσιες για την ένωσή τους με το επίπεδο των 3.3 Volt ή 1.8 Volt και το GND plane. Η απόσταση του ενός πόλου του πυκνωτή με το κατάλληλο κοντινότερο pin τροφοδοσίας της FPGA πρέπει να είναι όσο γίνεται πιο μικρή. Αν αναλάβανε ο autorouter την συνδεσμολογία του με το κατάλληλο plane, θα χρησιμοποιούσε αρκετά μακρύ αγωγό. Για αυτό το λόγο, με την ίδια τακτική συνδέσαμε όλους τους πυκνωτές στο κατάλληλο plane πριν τον autorouter στη διάταξη που φαίνεται στο Σχήμα 4.20, πετυχαίνοντας με αυτό το τρόπο την μικρότερη δυνατή απόσταση.



Σχήμα 4.20 Manual routing των πυκνωτών υψηλής συχνότητας των FPGAs.

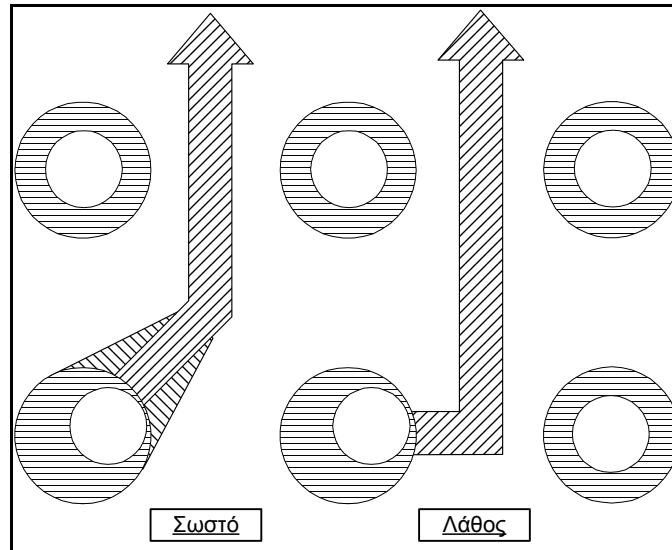
Με τις κατάλληλες ρυθμίσεις του εργαλείου και εφαρμόζοντας τους παραπάνω κανόνες και διαδικασίες, η διαφορά που παρατηρήθηκε στο χρόνο του routing είναι εξαιρετικά μεγάλη. Στον παρακάτω πίνακα περιέχονται οι χρόνοι του routing για τις δυο εκδόσεις του PLATO, καθώς και το ποσοστό ολοκλήρωσης της σχεδίασης. Και οι δύο σχεδιάσεις υλοποιήθηκαν με τη χρήση του εργαλείου Protel 99 SE σε υπολογιστή Pentium III στα 550 MHz, 256 MB μνήμη και με εγκατεστημένο το λειτουργικό NT4 Workstation στην έκδοση PLATO X1.0 και NT4 Server στην έκδοση PLATO A1.0.

| | Χρόνος routing για το PLATO X1.0 | | Χρόνος routing για το PLATO A1.0 | |
|--|-------------------------------------|------------|-------------------------------------|------------|
| <u>Πριν τη μελέτη του κεφαλαίου 4.5.3.</u> | > 2 Ώρες | Ολοκλήρωση | <i>Δεν υπάρχουν στοιχεία.</i> | |
| | | <75% | | |
| <u>Μετά τη μελέτη του κεφαλαίου 4.5.3.</u> | 09:52 Λεπτά | Ολοκλήρωση | 16:15 | Ολοκλήρωση |
| | | 99.8% | Λεπτά | 100% |

Η διαδικασία που ακολουθείται κατά την κατασκευή πολυστοματικών τυπωμένων κυκλωμάτων πολύ περιεκτικά είναι:

1. Δημιουργία των δυεπίπεδων πλακετών (περιλαμβάνουν ανά δυο τα διαδοχικά επίπεδα της πλακέτας).
2. Συγκόλληση των πλακετών.
3. Επικασσιτέρωση.
4. Διάνοιξη των τρυπών.

Κατά την διάνοιξη των τρυπών υπάρχει ο κίνδυνος να μετακινηθεί το τρυπάνι διάτρησης, λόγω της μηχανικής υποστήριξης και να μην δημιουργήσει την τρύπα στο ιδανικό σημείο, ώστε η τρύπα και το υλικό να είναι δυο ομόκεντροι κύλινδροι. Αν λοιπόν η τρύπα διανοιχτεί προς το σημείο από όπου ‘ξεκινάει’ ο αγωγός, μπορεί το υλικό στο σημείο της ένωσης μεταξύ του νία με τον αγωγό να είναι πάρα πολύ λίγο ή καθόλου, όπως περιγράφεται γραφικά και στο Σχήμα 4.21.



Σχήμα 4.21 Teardrops.

Αυτός ο κίνδυνος μπορεί να αποφευχθεί με την τοποθέτηση teardrops. Το teardrop είναι η διεύρυνση κατά κάποιο τρόπο του αγωγού λίγο πριν συνδεθεί με το νία. Η προϋπόθεση για να μπορεί να σχεδιαστεί όμως, είναι να υπάρχει ένα ευθύγραμμο τμήμα αγωγού αμέσως μετά τη σύνδεσή του με το νία. Αυτή η διεύρυνση δεν έχει τη δυνατότητα να σχεδιαστεί από το Protel και προστέθηκε από την εταιρεία κατασκευής κατόπιν συνεννόησης. Από τη δική μας πλευρά όμως θα έπρεπε να υπάρχει η παραπάνω προϋπόθεση. Για αυτό το λόγο μετά το routing έπρεπε να επέμβουμε με το χέρι στη σχεδίαση για να κάνουμε τη διόρθωση που περιγράφεται στο Σχήμα 4.21.

4.6 DRC.

Ο έλεγχός της σχεδίασης που έπεται του autorouting ονομάζεται DRC (Design Rule Check). Το DRC μπορεί να είναι ενεργό κατά τη διάρκεια του routing, ώστε να ολοκληρωθεί το σχέδιο, πληρώντας κάποιους κανόνες. Αν στη πορεία δεν μπορεί να ολοκληρώσει κάποιους από αυτούς, τότε ανάλογα με τον κανόνα θα συνεχίσει το routing δρομολογώντας τα σήματα, μη τηρώντας τον αντίστοιχο κανόνα. Στο τέλος της σχεδίασης θα δώσει πλήρη αναφορά των σημείων της σχεδίασης που δεν πληρούν τους κανόνες και ποιοι κανόνες είναι αυτοί.

Οι κανόνες που χρειάστηκαν στη σχεδίαση του PLATO είναι:

1. Clearance = 8 mils.

2. Μικρότερη γωνία αγωγού = 45^0 .
3. Τοπολογία routing των nets = Η συντομότερη σε συνολικό μήκος.
4. Διαστάσεις vias: Εξωτερική διάμετρος = 40 mils, διάμετρος τρύπας = 20 mils.
5. Ελάχιστο πλάτος αγωγού = 8 mils.
6. Ελάχιστο Annular Ring ή Donald = 8 mils σε όλη τη πλατφόρμα, εκτός των vias κάτω από τις FPGAs που είναι 7 mils.
7. Σε όλη την πλακέτα thermal συνδέσεις με τα planes με τέσσερις αγωγούς των 10 mils έκαστος εκτός των συνδέσεων των εξαρτημάτων τροφοδοσίας που ήταν 20 mils.
8. Solder mask = 3 mils.
9. Μέγιστο μήκος του net χρονισμού = 4000 mils.
10. Μέγιστο μήκος όλων των υπόλοιπων nets = 5000 mils.
11. Ελάχιστη απόσταση μεταξύ των εξαρτημάτων = 10 mils.
12. Δεν επιτρέπονται βραχυκυκλώματα μεταξύ διαφορετικών nets.
13. Έλεγχος της ολοκλήρωσης της δρομολόγησης του κάθε net.

Στη δική μας περίπτωση ο κανόνας που δεν μπορούσε να εκπληρωθεί ήταν το μέγιστο μήκος των σημάτων του χρονισμού της πλατφόρμας. Όπως προαναφέραμε επιθυμητό μήκος ήταν τα 2/3 της διαγωνίου της πλατφόρμας. Σε αυτή τη περίπτωση, μετά το στάδιο του routing και με χειρονακτική επέμβαση ξανασχεδιάστηκε το εν λόγω net, αποκτώντας συγκριτικά με το σύνολο των σημάτων, αποδεκτό μήκος.

4.7 Output Files.

Τα αρχεία εξόδου ποικίλουν ως προς το τύπο τους και την ακρίβεια περιγραφής του τυπωμένου κυκλώματος. Ο δημοφιλέστερος τύπος είναι τα Gerber files. Το βασικό στην παραγωγή των αρχείων εξόδου είναι η συμβατότητα τους με τις απαιτήσεις της κατασκευάστριας εταιρείας, καθώς και η απαραίτητη ακρίβεια για την αλάνθαστη περιγραφή του κυκλώματος.

Στη δική μας πλατφόρμα ανάπτυξης, η οποία κατασκευάστηκε στην Intracom, περιγράφηκε με Gerber files. Η ακρίβεια περιγραφής των συντεταγμένων ήταν με 4 δεκαδικά ψηφία.

ΚΕΦΑΛΑΙΟ 5^ο

5 Υλοποίηση και Δοκιμή Συστημάτων.

Και οι δυο εκδόσεις του PLATO, καθώς και τα daughterboards κατασκευάστηκαν από την εταιρία Intracom. Οι FPGAs κολλήθηκαν με τη χρήση κατάλληλου εργαλείου από τον Κύριο Μιχάλη Λιγεράκη του ΙΤΕ (Ινστιτούτο Τεχνολογίας και Έρευνας), στο Ηράκλειο Κρήτης. Τα υπόλοιπα εξαρτήματα του PLATO X1.0 τοποθετήθηκαν από τον Κύριο Μάρκο Κιμιωνή, μέλος ΕΔΤΠ του Πολυτεχνείου Κρήτης και του PLATO A1.0 τοποθετήθηκαν στο ΙΤΕ, όπου γίνεται και ο έλεγχος της λειτουργίας στα πλαίσια του ερευνητικού προγράμματος ΠΕΝΕΔ 99.

Ο έλεγχος της λειτουργίας του PLATO X1.0 έγινε στο Πολυτεχνείο Κρήτης το πρώτο εξάμηνο του 2001. Για τον έλεγχο της λειτουργίας του σχεδιάστηκαν δυο εφαρμογές.

Εφαρμογή 1^η. Shift Register.

Η εφαρμογή αυτή είχε σκοπό να ελεγχθούν σε πραγματικές συνθήκες όλα τα I/O της FPGA με όλα τα περιφερειακά της, ώστε να εξακριβωθεί αν κατά την συγκόλλησή της προκλήθηκαν βραχυκυκλώματα μεταξύ των pins της, λόγω της τήξης του κασσίτερου, ο οποίος βρίσκεται στα pads της. Έπρεπε να φτιαχτεί μια εφαρμογή με την οποία θα μπορούσαμε να δούμε αν το επίπεδο τάσης σε ένα pin της FPGA επηρέαζε το επίπεδο τάσης σε κάποιο άλλο pin. Για αυτό το σκοπό σχεδιάστηκε ένας κυκλικός καταχωρητής ολίσθησης (Circular Shift Register) 390 θέσεων, όσα και τα I/O της FPGA, ο οποίος αρχικά ήταν αρχικοποιημένος με μηδενικά εκτός του πρώτου bit που ήταν με '1'. Κάθε ένα από αυτά τα bit είναι συνδεδεμένο και με ένα pin εξόδου της FPGA. Δίνοντας στη συνέχεια ρολόι στο κύκλωμα, το λογικό '1' της πρώτης θέσης μεταφέρεται στη δεύτερη θέση και το μηδενικό του τελευταίου bit μεταφέρεται στο πρώτο και ου το καθεξής. Αν δώσουμε

συνεχόμενο ρολόι στο κύκλωμα και παρατηρήσουμε σταθερά ένα από τα σήματα εξόδου στον παλμογράφο, αυτό που θα περιμένουμε να δούμε είναι διαδοχικούς παλμούς, με χρονική διαφορά μεταξύ τους $389/f$, όπου f είναι η συχνότητα λειτουργίας της εφαρμογής. Αν σε αυτό το χρονικό διάστημα παρουσιαστεί ένας παλμός, τότε αυτό σημαίνει ότι στην πλακέτα μας υπάρχει βραχυκύκλωμα μεταξύ του pin που ελέγχουμε και ενός άλλου, το οποίο μεταφέρει τον παλμό του στο pin ελέγχου τη στιγμή που περνάει το λογικό '1' του καταχωρητή ολίσθησης από τη δική του θέση. Αν παρουσιαστούν δυο παλμοί, τότε έχουν βραχυκυκλωθεί τρία pins μεταξύ τους και ου το καθεξής.

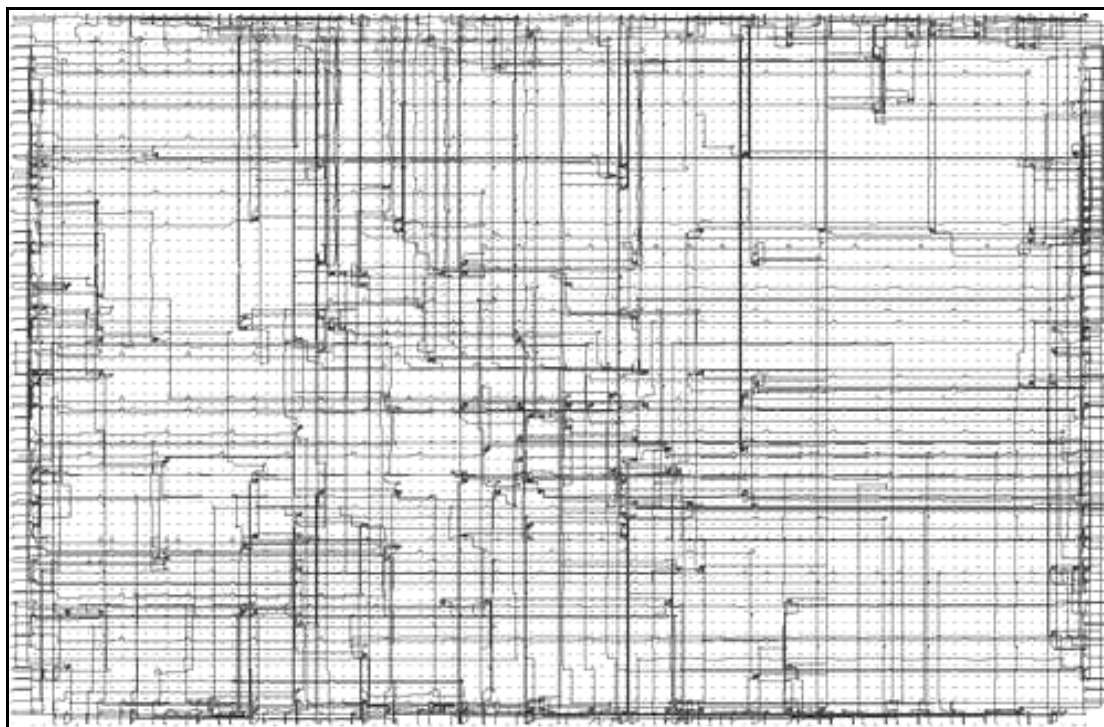
Ο κώδικας για αυτή την εφαρμογή παρουσιάζεται στη συνέχεια.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity Shift_1 is
  port (
    RESET    : in STD_LOGIC;
    CLK      : in STD_LOGIC;
    OK       : out STD_LOGIC_VECTOR (0 to 389)
  );
end Shift_1;

architecture Shift_1_arch of Shift_1 is
  signal reg: STD_LOGIC_VECTOR (0 to 389);
begin
  process (CLK, RESET)
  begin
    if RESET='1' then
      reg(0) <= '1';
      reg(1 to 389) <= (others=>'0');
    elsif (CLK'event and CLK='1') then
      reg(0) <= reg(389);
      reg(1 to 389) <= reg(0 to 388);
    end if;
    OK <= reg;
  end process;
end Shift_1_arch;
```

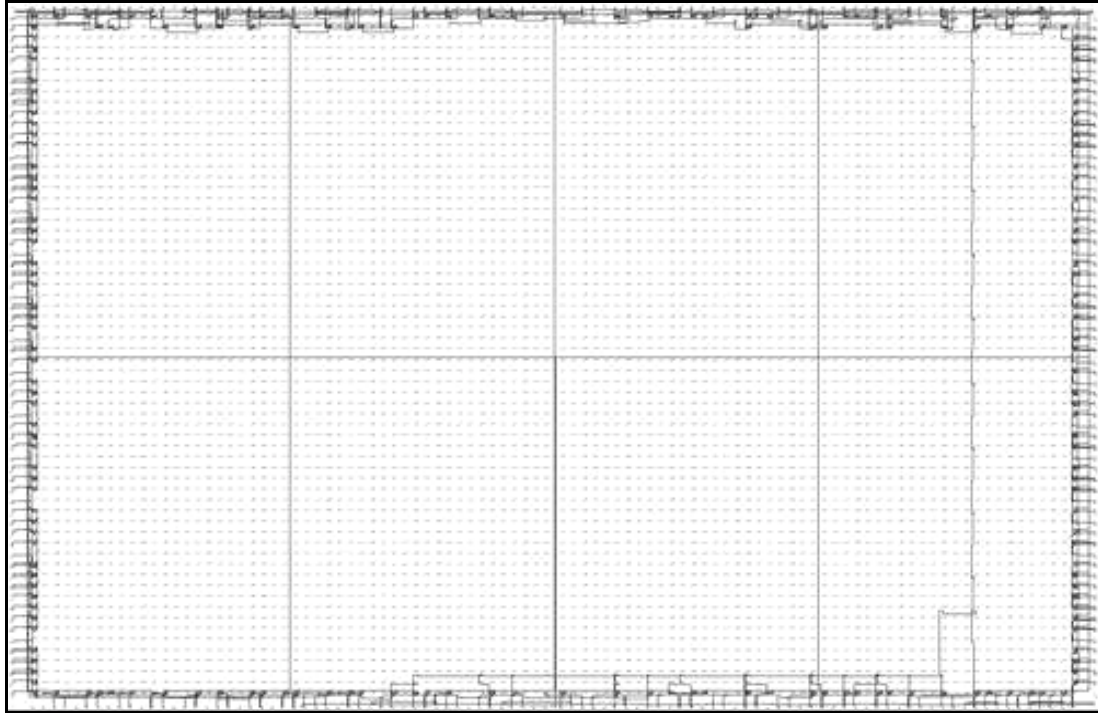
To implementation της εφαρμογής αυτής χωρίς κανένα χωρικό constrain παρουσιάζεται στο Σχήμα 5.1. Σε αυτή την περίπτωση η σχεδίαση έχει μέγιστη συχνότητα λειτουργίας 83MHz και το routing χρειάστηκε 99 δευτερόλεπτα.



Σχήμα 5.1 Circular Shift Register, χωρίς UCF.

Βάζοντας χωρικά constraints και αναγκάζοντας τα bit του shift register να δρομολογούνται διαδοχικά σε όλα τα περιφερειακά της πλακέτας, όχι σύμφωνα με τα IOBs, έτσι όπως αυτά παρουσιάζονται στο VLSI της FPGA, αλλά διαδοχικά από το ένα εξάρτημα στο άλλο, χρειάστηκαν 36 δευτερόλεπτα να γίνει routing και η μέγιστη συχνότητα λειτουργίας έφτασε τα 303 MHz. Το αποτέλεσμα του implementation φαίνεται γραφικά στο Σχήμα 5.2.

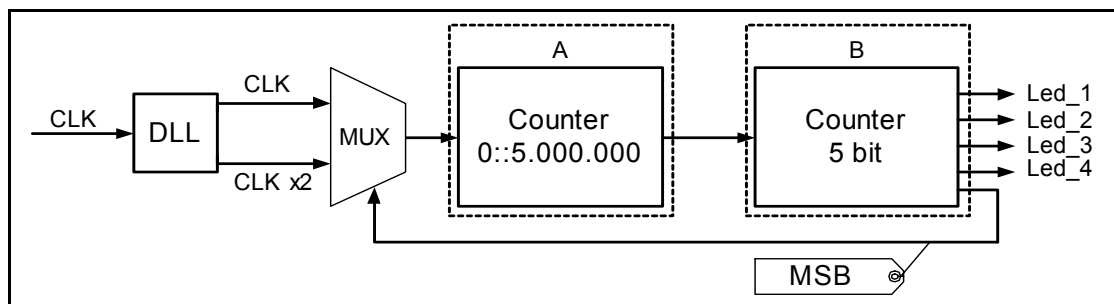
Αυτή η εξαιρετικά μεγάλη διαφορά οφείλεται στο γεγονός ότι προβλέφθηκε από το στάδιο της δημιουργίας του netlist η βέλτιστη λύση, ως προς την ταχύτητα της εφαρμογής και την ευκολία του εργαλείου, να κάνει routing την αρχιτεκτονική στην FPGA. (Βλέπε Κεφάλαιο 4, Σχήμα 4.11).



Σχήμα 5.2 Circular Shift Register, με UCF.

Εφαρμογή 2^η. Counter.

Η εφαρμογή αυτή είχε ως σκοπό να ελεγχθεί σε πραγματικές συνθήκες η λειτουργία των DLL της Virtex, καθώς και να εξεταστούν θέματα κατανάλωσης, σε σχέση με τη συχνότητα λειτουργίας και το μέγεθος της αρχιτεκτονικής της εφαρμογής.



Σχήμα 5.3 Αρχιτεκτονική του μετρητή.

Το ρολόι διπλασιάζεται μέσω του DLL, ενώ υπάρχει και ένας counter (A) που μετράει από το μηδέν και δίνει ένα παλμό κάθε φορά που φτάνει στο 5.000.000 στη επόμενη βαθμίδα, η οποία είναι ένας counter 5 bit (B). Όταν ο counter B γίνει '00001' το ρολόι λειτουργίας του counter A διπλασιάζεται, οπότε και του B, με

αποτέλεσμα να παρατηρούμε στην έξοδο διαδοχικές μετρήσεις του counter με την κανονική και τη διπλάσια ταχύτητα. Λόγω της μεγάλης καθυστέρησης που προκαλείται από το counter A, τα τέσσερα LSB του counter B είναι ορατά στα Leds της πλατφόρμας.

Η κατανάλωση του συστήματος παρατηρήθηκε ορθώς ότι με την αύξηση της συχνότητας λειτουργίας ήταν μεγαλύτερη.

ΚΕΦΑΛΑΙΟ 6^ο

6 Συμπεράσματα.

Στα πλαίσια αυτής της διπλωματικής εργασίας σχεδιάστηκε μια νέα επαναπρογραμματιζόμενη πλατφόρμα, το PLATO. Τα παρόντα αποτελέσματα είναι πολύ ενθαρρυντικά, από την πλευρά της λειτουργικότητάς της αλλά και από το τρόπο που πλησιάζει στο χώρο των ενεργών δικτύων σε σχέση με τις μέχρι τώρα προσεγγίσεις. Στα επόμενα χρόνια είναι πολύ πιθανό να αποδειχτεί εξαιρετικά χρήσιμη στην ανάπτυξη εφαρμογών και στη συλλογή πολυτίμων συμπερασμάτων, καλύπτοντας ένα σχετικά ανεξερεύνητο ερευνητικό πεδίο από πλευράς Hardware.

Εκτός της ερευνητικής περιοχής των Ενεργών Δικτύων, η αρχιτεκτονική και οι πόροι που προσφέρονται στη πλατφόρμα ανάπτυξης PLATO, την καθιστούν κατάλληλη για ανάπτυξη εφαρμογών όπως επεξεργασίας σήματος και εικόνας με τη χρήση των θυρών επέκτασης SRAM και UTOPIA.

Με αφορμή αυτής της διπλωματικής εργασίας, αποκτήθηκε εμπειρία στη σχεδίαση της αρχιτεκτονικής πολύπλοκων συστημάτων βασισμένων σε FPGAs, καθώς και στην σχεδίαση πολυστρωματικών τυπωμένων κυκλωμάτων. Αποκτήθηκε βαθιά γνώση της παρούσας τεχνολογίας των FPGAs των δυο εταιριών Xilinx και Altera.

Παράρτημα Α

ATM: Asynchronous Transfer Mode.

Clearance: Είναι η ελάχιστη απόσταση των υλικών (vias, pads, αγωγοί κτλ.) της πλακέτας που δεν ανήκουν στο ίδιο net.

DRC: Design Rule Check.

Footprint ενός εξαρτήματος: Είναι το σχέδιο πάνω στο οποίο θα τοποθετηθεί το εξάρτημα. Το footprint είναι ένα σύνολο από Pads το μέγεθος των οποίων, καθώς και η μεταξύ τους απόσταση και τοπολογία περιγράφεται από την κατασκευάστρια εταιρία του εξαρτήματος στα αντίστοιχα data sheets.

Grid: Ονομάζουμε το διακριτό σύστημα συντεταγμένων XY πάνω στο οποίο γίνεται η σχεδίαση.

IOB: Input Output Block.

Mil: Είναι μονάδα μέτρησης μήκους και είναι 100mils=2.54χιλιοστά.

Net: Ένα σύνολο από pads που ενώνονται μεταξύ τους.

Netlist: Ονομάζουμε την πληροφορία που περιγράφει τα εξαρτήματα ενός συστήματος, αλλά και τη λεπτομερή συνδεσμολογία μεταξύ αυτών. Αποτελείται από την περιγραφή των εξαρτημάτων και όλα τα Nets.

Pad: Είναι τα σημεία της πλακέτας πάνω στα οποία γίνονται οι συγκολλήσεις των pins των εξαρτημάτων. Για τα εξαρτήματα 'Through Hole', έχουν τη μορφή σωλήνα, του οποίου η εσωτερική διάμετρος ονομάζεται hole. Αυτή η τρύπα διαπερνάται από τα pins του αντίστοιχου εξαρτήματος. Για τα SMD εξαρτήματα, τα pads είναι μια τετράγωνη ή κυκλική επιφάνεια πάνω στη οποία ακουμπάνε τα pins του εξαρτήματος και συγκολλούνται με την τήξη κασσίτερου.

Plane: Είναι ένα επίπεδο το οποίο αποτελείται από μια επιφάνεια χαλκού, η οποία επεκτείνεται σε ολόκληρη την επιφάνεια του επιπέδου, εκτός από τα σημεία όπου αυτό τέμνεται από Pads ή Vias.

SMD εξάρτημα: Surface Mount Device. Εξάρτημα επιφανειακής συγκόλλησης.

Through Hole εξάρτημα: Εξάρτημα του οποίου τα pins διαπερνούν την πλακέτα από το πάνω επίπεδο μέχρι το κάτω. Η συγκόλληση τους μπορεί να γίνει και στα δυο επίπεδα, εφόσον χρησιμοποιούνται Through Hole pads.

Vccint: Pin τροφοδοσίας για τα CLBs.

Vcco: Pin τροφοδοσίας για τα IOBs.

Via: Είναι το σημείο στο οποίο γίνεται η αλλαγή του επιπέδου ενός σήματος. Έχει τη μορφή σωλήνα, του οποίου η εσωτερική διάμετρο είναι το Hole του Via. Τα vias συνήθως επεκτείνονται από το πάνω επίπεδο έως το κάτω της πλακέτας, μεταφέροντας το αντίστοιχο σήμα σε όλα τα ενδιάμεσα επίπεδα. Σε εξαιρετικές περιπτώσεις είναι δυνατό να σχηματιστεί μόνο μεταξύ κάποιων επιπέδων και των ενδιάμεσών τους, γεγονός που αυξάνει κατά πολύ το κόστος κατασκευής.

Παράρτημα Β.

Παρακάτω ακολουθεί το πρόγραμμα που χρησιμοποιήθηκε για να παράγουμε το μεγαλύτερο μέγεθος του κώδικα του netlist. Χρησιμοποιούνται τρία αρχεία με την κατάλληλη πληροφορία για τα ονόματα των nets και τα δυο pads των εξαρτημάτων που ενώνει.

```
#include <stdio.h>
#include <string.h>
char *read_line(FILE *f);
write_line(FILE *f,char *strg);
void main( void )
{
    FILE *f1,*f2,*f3,*f4;
    char line1[40],line2[40],line3[40];
    int count_lines=0;
    if( f1 = fopen( "net.txt", "r" ) ) == NULL )
        printf( "Problem opening the file\n" );
    if( f2= fopen( "cmp1.txt", "r" ) ) == NULL )
        printf( "File could not be opened\n" );
    if( f3= fopen( "cmp2.txt", "r" ) ) == NULL )
        printf( "File could not be opened\n" );
    if( f4= fopen( "out.txt", "w+" ) ) == NULL )
        printf( "File could not be opened\n" );
    while(!feof(f1))
    {
        count_lines++;
        fputs("\n",f4);
        fgets(line1,40,f1);
        fgets(line2,40,f2);
        fgets(line3,40,f3);
        fputs(line1,f4);
        fputs(line2,f4);
        fputs(line3,f4);
        fputs("\n",f4);
    }
    fclose( f1 );
    fclose( f2 );
    fclose( f3 );
    fclose( f4 );
    printf("Einai %d\n",count_lines);
    getchar();
}
```

ΑΝΑΦΟΡΕΣ.

Βιβλιογραφία:

VHDL For Designers, Stefan Sjöholm & Lennard Lindh

World Wide Web

FPGAs

www.xilinx.com [Συλλογή πληροφορίας από τα data sheets του site.]

1. Virtex data sheet [ds003.pdf v2.5]
2. Για τη τροφοδοσία της Virtex. [xapp158.pdf v1.3]
3. Configuration of Virtex [xapp138.pdf v1.0]
4. Configuration of Proms [ds026.pdf v2.7]
5. Configuration of Proms [ds027.pdf v3.1]

www.altera.com [Συλλογή πληροφορίας από τα data sheets του site.]

1. Apex data sheet [ds_apex.pdf v2.06]
2. Configuration of Apex [an116.pdf v1.03]
3. MasterBlaster [ds_masterbl.pdf v1.01]
4. PLL of Apex [an115.pdf]

Connectors & Παρελκόμενα

www.molex.com

www.shadownet.com/hwb/menu_Connector.html

rswww.com

SRAMs

www.cypress.com

www.motorola.com

www.idt.com

Βιβλιοθήκες εργαλείου σχεδίασης.

www.protel.com