

Κεφάλαιο 4

Τρόποι Υλοποίησης Ενσωματωμένων Συστημάτων

Η μεγάλη ποικιλία του οικοσυστήματος των ενσωματωμένων συστημάτων, που οφείλεται στο ευρύ φάσμα των απαιτήσεων που καλούνται να χρησιμοποιηθούν, αντικατοπτρίζεται και στις πάρα πολλές δυνατότητες υλοποίησης. Σε αντίθεση με τους τυπικούς προσωπικούς ή φορητούς υπολογιστές ή και διακομιστές, που οι μόνες επιλογές περιορίζονται στον τύπο του επεξεργαστή (που όλοι ακολουθούν τη συμβατότητα με την αρχιτεκτονική Intel IA32 ή AMD64), και της κεντρικής πλακέτας συστήματος (που διαφοροποιούνται μόνο ως προς τις υποδοχές επέκτασης), τα ΕΣ μπορούν να υλοποιηθούν με πολλαπλούς τρόπους, ο καθένας από τους οποίους παρουσιάζει τα δικά του ιδιαίτερα χαρακτηριστικά και ταιριάζει καλύτερα σε συγκεκριμένες σχεδιαστικές απαιτήσεις. Σε αυτό το κεφάλαιο περιγράφονται όλοι οι διαφορετικοί τρόποι υλοποίησης, και γίνεται κατανοητό πότε και που θα χρησιμοποιηθεί μια τεχνολογία.

4.1 Εισαγωγή

Η επεξεργασία μέσα στο ενσωματωμένο σύστημα μπορεί να υλοποιηθεί μέσω των διαφόρων μονάδων εξειδικευμένου υλικού. Αυτά υλοποιούνται είτε σε μη προγραμματιζόμενο υλικό (ή αλλιώς σε ολοκληρωμένα κυκλώματα εξειδικευμένης εφαρμογής), είτε σε προγραμματιζόμενο υλικό (ή αλλιώς σε επί του πεδίου προγραμματιζόμενοι συστοιχίες πυλών). Ένας επεξεργαστής ειδικού σκοπού (που αναφέρεται σε ολοκληρωμένα κυκλώματα εξειδικευμένης εφαρμογής) σε αντίθεση με έναν επεξεργαστή γενικού σκοπού είναι σχεδιασμένος σε ψηφιακό κύκλωμα, ώστε να εκτελεί μια συγκεκριμένη υπολογιστική εργασία ή εφαρμογή. Ο επεξεργαστής ειδικού σκοπού ονομάζεται αλλιώς και ως συνεπεξεργαστής (coprocessor) ή επιταχυντής υλικού (hardware accelerator). Ένας

σχεδιαστής ενσωματωμένων συστημάτων μπορεί να επιτύχει διάφορα οφέλη επιλέγοντας να χρησιμοποιήσει ένα επεξεργαστή ειδικού σκοπού, ώστε να υλοποιήσει μια υπολογιστική εργασία.

Πρώτον, η απόδοση των εξειδικευμένων ενσωματωμένων επεξεργαστών είναι ταχύτερη, εξαιτίας των λιγότερων κύκλων ρολογιού, ως αποτέλεσμα ενός εξατομικευμένου μονοπατιού δεδομένων. Επιπλέον, η απόδοση αυτή είναι βελτιωμένη, λόγω των μικρότερων κύκλων ρολογιού των απλούστερων λειτουργικών μονάδων και της απλούστερης λογικής ελέγχου. Δεύτερον, το μέγεθος είναι μικρότερο, εξαιτίας του απλού μονοπατιού δεδομένων και ότι δεν απαιτείται μνήμη προγράμματος, εφόσον η υπολογιστική εργασία είναι υλοποιημένη σε ψηφιακό υλικό και υλοποιείται με συνδυαστικά ή ακολουθιακά κυκλώματα. Τρίτον, η κατανάλωση ενέργειας είναι λιγότερη, εξαιτίας του αποτελεσματικού υπολογισμού, χωρίς να υλοποιούνται περιττές συναρτήσεις.

Όμως, υπάρχουν διάφορα μειονεκτήματα όπως το κόστος που ίσως είναι υψηλότερο, λόγω του υψηλού κόστους έρευνας και ανάπτυξης (NRE). Ταυτόχρονα, ο χρόνος εισαγωγής στην αγορά είναι μεγαλύτερος και μειώνεται η ευελιξία σε σχέση με τους επεξεργαστές γενικού σκοπού.

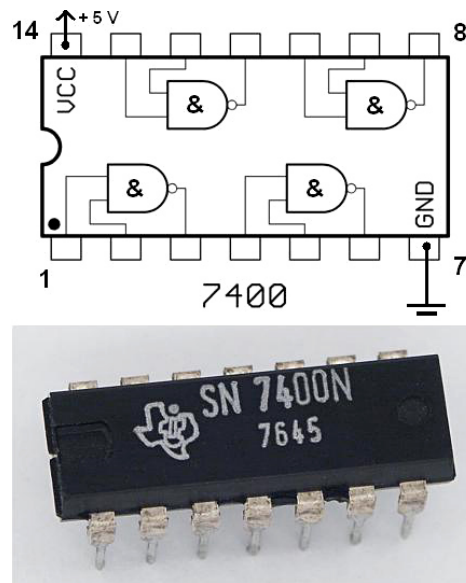
Από την άλλη πλευρά, η τεχνολογία επί του πεδίου προγραμματιζόμενοι συστοιχίες πυλών, προσπαθεί να διατηρήσει τα οφέλη που έχουν οι επεξεργαστές ειδικού σκοπού, αλλά και να βελτιώσει το κόστος, το χρόνο εισαγωγής στην αγορά και την ευελιξία. Το κεφάλαιο ξεκινά με την περιγραφή της τεχνολογίας ολοκληρωμένων κυκλωμάτων εξειδικευμένης εφαρμογής και έπειτα επί του πεδίου προγραμματιζόμενοι συστοιχίες πυλών. Τα πλεονεκτήματα και τα μειονεκτήματα όλων των επιλογών θα αναλυθούν σε επόμενους παραγράφους.

4.2 Οι πρώτες υλοποιήσεις

Λίγα χρόνια μετά την ανακάλυψη του τρανζίστορ, η Texas Instruments κατασκεύασε μια σειρά διακριτών δομοστοιχείων, η οποία απαριθμούσε εκατοντάδες λογικά στοιχεία υλοποιώντας μια πληθώρα λειτουργιών. Η σειρά αυτή ήταν γνωστή ως 7400, από τον αριθμό που φέρει το πρώτο δομικό στοιχείο και αποτελείται από τέσσερις πύλες NAND (Εικόνα 4.1). Αυτά τα στοιχεία ονομάστηκαν TTL (transistor-transistor logic ή λογική συνδεδεμένων τρανζίστορ), είχαν τυποποιημένες συσκευασίες (standard packaging) και μπορούσαν να συνδεθούν εύκολα, ώστε να κατασκευάσουν μεγαλύτερα ψηφιακά συστήματα. Παρείχαν ένα πλήθος από λειτουργίες, όπως ψηφιακές πύλες, φλιπ-φλοπς, μετρητές, απομονωτές, αριθμητικές λογικές μονάδες κτλ. Ενδεικτικό είναι το γεγονός, ότι τα περισσότερα μικρά και μεγάλα υπολογιστικά κέντρα του 1960 και 1970 χρησιμοποίησαν αυτή την οικογένεια.

Εκτός από την οικογένεια 7400 υπήρχε και η οικογένεια διακριτών στοι-

χείων 5400, η οποία είχε τα ίδια ακριβώς μέλη, αλλά παρείχε προδιαγραφές για στρατιωτικές εφαρμογές (π.χ. μεγαλύτερο εύρος θερμοκρασιών λειτουργίας). Εκτός από την Texas Instruments που τα παρείχε αρχικά, και άλλες εταιρίες προσέφεραν συμβατά δομοστοιχεία με την ίδια αρίθμηση και έτσι υπήρχε μια αφθονία στοιχείων TTL. Με τον καιρό δημιουργήθηκαν και εκδόσεις που είχαν ελαφρώς διαφορετικά χαρακτηριστικά (αλλά παρείχαν τις ίδιες ψηφιακές λειτουργίες): εκδόσεις μειωμένης κατανάλωσης ενέργειας (είχαν το γράμμα 'L', όπως 74L00), εκδόσεις με μειωμένες καθυστερήσεις διάδοσης χρησιμοποιώντας διόδους Schottky (είχαν το γράμμα 'S', όπως 74S00), εκδόσεις συνδυασμό των δυο προηγούμενων (είχαν τα γράμματα 'LS', όπως 74LS00), εκδόσεις μεγάλης ταχύτητας (fast και είχαν το γράμμα 'F', όπως 74F00), και εκδόσεις προηγμένης χαμηλής κατανάλωσης ενέργειας (advanced low power, με τα γράμματα 'AL', όπως 74AL00). Τα στοιχεία της TTL λογικής ονομάζονταν συσκευές διαμέσου οπής (through hole devices), γιατί οι ακροδέκτες διέρχονταν από οπές πάνω στην πλακέτα και στη συνέχεια γίνονταν η κόλληση.



Σχήμα 4.1: Η σειρά διακριτών στοιχείων ψηφιακής λογικής 7400 ονομάστηκε έτσι από το πρώτο στοιχείο της σειράς που φέρει τέσσερις πύλες NAND. Εικόνα από wikipedia.org

Η οικογένεια TTL 7400 ή παραλλαγές αυτής, συνεχίζει να υπάρχει ύστερα από 55 χρόνια. Οι μεγαλύτεροι κατασκευαστές συνεχίζουν να προμηθεύουν τα καταστήματα ηλεκτρονικών με δομοστοιχεία, είτε διαμέσου οπής (through hole devices) ή τα πιο σύγχρονα στοιχεία επιφανειακής συγκόλλησης (surface mount). Χρησιμοποιούνται τόσο για την εκπαίδευση, αφού οι φοιτητές μπορούν

να κάνουν πλήθος εργαστηριακών ασκήσεων σε μια προτυποποιημένη πλακέτα (breadboard) για να μάθουν ψηφιακή σχεδίαση, ή για τη συντήρηση παλαιών ψηφιακών συστημάτων (legacy digital systems) ή ως λογική συγκόλλησης (glue logic), προκειμένου να διασυνδεθούν και να επικοινωνήσουν δυο διαφορετικά και μη συμβατά ψηφιακά συστήματα.

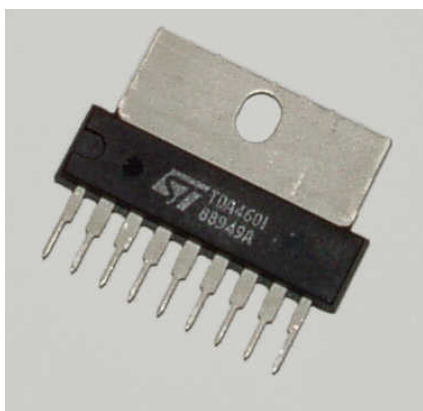
Τα TTL χρησιμοποιούν διπολικά τρανζίστορ και έτσι παρουσιάζουν μια μεγάλη κατανάλωση ενέργειας. Προκειμένου να βελτιστοποιηθεί η κατανάλωση χρησιμοποιήθηκε μια εναλλακτική υλοποίηση, η οποία είναι γνωστή ως CMOS (Complementary metal-oxide-semiconductor ή Τεχνολογία Συμπληρωματικού ημιαγωγού μετάλλου- οξειδίου). Αυτή η τεχνολογία που χρησιμοποιείται ως τις μέρες μας λόγω της χαμηλής κατανάλωσης ενέργειας, αποτελείται από τρανζίστορ n-MOS και p-MOS, όπου η κάθε ψηφιακή λειτουργία υλοποιείται από ένα n και ένα p δίκτυο τρανζίστορ, ώστε το ένα να συμπληρώνει το άλλο. Μια από τις πρώτες οικογένειες της CMOS που αναπτύχθηκε την ίδια περίοδο με την 7400 ήταν η 4000 η οποία ήταν αρκετά πιο αργή από ότι η TTL και έτσι δε βρήκε μεγάλη απήχηση. Όμως, μετά από λίγα χρόνια, το 1980, κάποιοι κατασκευαστές παρουσίασαν μια νέα οικογένεια CMOS που ήταν συμβατή με τη σειρά 7400 με χαμηλότερη κατανάλωση ενέργειας, και από τότε άρχισε η ευρεία υιοθέτηση αυτής της τεχνολογίας.

Από τη στιγμή που παρουσιάστηκαν οι πρώτες τυποποιημένες συσκευές που είχαν παραπάνω από ένα τρανζίστορ, υιοθετήθηκαν κάποιοι ακρωνύμια, ώστε να προσδιορίζουν τον αριθμό των τρανζίστορ που φέρουν. Συσκευές που φέρουν από μια έως είκοσι λογικές πύλες φέρουν τον χαρακτηρισμό SSI (small scale integrated ή μικρής κλίμακας ολοκλήρωσης), ενώ συσκευασίες που περιέχουν είκοσι έως διακόσιες λογικές πύλες ονομάζονται MSI (medium scale integrated ή μεσαίας κλίμακας ολοκλήρωσης). Συσκευασίες που περιέχουν 200 έως 200.000 λογικές πύλες ονομάζονται LSI (large scale integrated ή μεγάλης κλίμακας ολοκλήρωσης) και τέλος από 200.000 λογικές πύλες και πάνω VLSI (very large scale integrated ή πολύ μεγάλης κλίμακας ολοκλήρωσης). Η ονομασία VLSI ήταν και η τελευταία (X)LSI που χρησιμοποιήθηκε, γιατί από τις αρχές του 1980 είχε γίνει σαφές ότι αν συνεχίζονταν η προσθήκη επιθέτων μπροστά από το LSI θα τελείωναν σε λίγα χρόνια οι προσδιορισμοί.

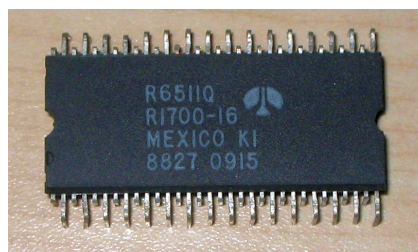
Όλες οι παραπάνω οικογένειες παρέχουν πλήθος ψηφιακών λειτουργιών μέσα στην ίδια συσκευασία, και μπορούν να χρησιμοποιηθούν για την υλοποίηση ενσωματωμένων συστημάτων. Για παράδειγμα συσκευασίες SSI, όπως ο δημοφιλής μετρητής 555, μπορούν να χρησιμοποιηθούν για ένα απλό ψηφιακό σύστημα ξυπνητηριού, έως και VLSI όπου ανήκουν ενσωματωμένοι επεξεργαστές υψηλής απόδοσης.

Ένα στοιχείο, επίσης, που μας ενδιαφέρει στο σχεδιασμό ΕΣ είναι η συσκευασία που περιέχει το ολοκληρωμένο κύκλωμα. Η συσκευασία περιέχει εκτός από την ίδια τη λογική και τους ακροδέκτες, το περίβλημα προστασίας,

τους ενισχυτές και τους αισθητήρες για τη διασύνδεση με άλλες εξωτερικές συσκευές. Όπως και κάθε άλλη επιλογή που συνδέεται με τα ενσωματωμένα συστήματα, έτσι και εδώ υπάρχει μια μεγάλη ποικιλία, η οποία εξαρτάται από την πολυπλοκότητα ή των αριθμό των τρανζίστορ που περιέχει. Οι πιο απλές συσκευές έχουν συσκευασίες ακροδεκτών μονής γραμμής (single in-line package) (Εικόνα 4.2) ή διπλής γραμμής (dual in-line package) (Εικόνα 4.5), και επινοήθηκαν το 1964 στην εταιρία Fairchild R&D. Αυτές οι συσκευασίες είναι εύκολες στη χρήση και μπορούν να τοποθετηθούν, είτε με κόλληση πάνω στην ηλεκτρονική πλακέτα του κυκλώματος, είτε σε ειδική υποδοχή που επιτρέπει την αντικατάσταση (Εικόνα 4.5). Οι συσκευασίες είναι ορθογώνιες, απαραίτητο για τη διευκόλυνση της δρομολόγησης των αγωγών μέσα στην ψηφιακή λογική. Οι τυπικές συσκευασίες DIP περιέχουν 8 ή 14 ή 16 ακροδέκτες και φέρουν τις ονομασίες DIP8, DIP14 και DIP16 αντιστοίχως. Επίσης, υπάρχουν διαφορετικά υλικά κατασκευής (π.χ. πλαστικό ή κεραμικό) ή μικρότερες αποστάσεις ανάμεσα στους ακροδέκτες, οπότε προκύπτουν οι κατηγορίες CDIP (κεραμική συσκευασία), PDIP (πλαστική συσκευασία), SPDIP (1.778 mm απόσταση ακροδεκτών), SDIP (7.62 mm πλάτος). Τέλος, υπάρχουν και συσκευασίες που φέρουν τέσσερις γραμμές ακροδεκτών και ονομάζονται QIP (quad in-line package) (Εικόνα 4.3).



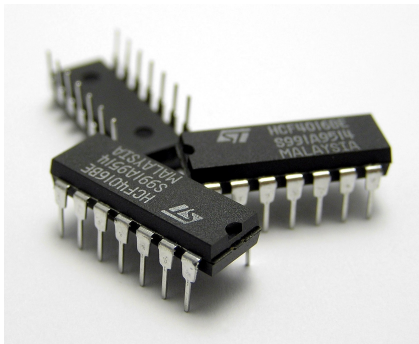
Σχήμα 4.2: Η συσκευασία SIL απο-τελείται από 1 σειρά ακροδεκτών. Εικόνα από wikipedia.org.



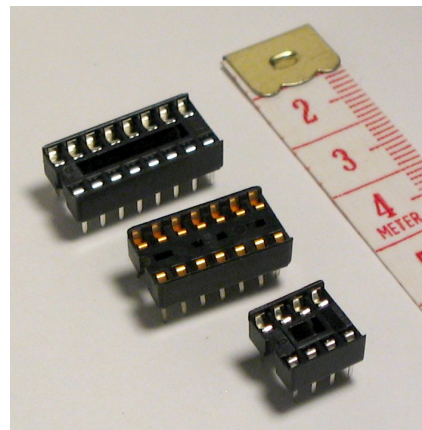
Σχήμα 4.3: Εκτός από τις SIL και DIP υπάρχουν και οι συσκευασίες QIP με τέσσερις σειρές ακροδεκτών. Εικόνα από wikipedia.org.

Οι συσκευασίες DIP και SIL χρησιμοποιούνται για κυκλώματα που απαιτούν σχετικά λίγους ακροδέκτες (έως 64). Όμως, υπάρχουν κυκλώματα που έχουν μεγαλύτερες απαιτήσεις. Μια συνηθισμένη συσκευασία για αυτά είναι η PGA (pin grid array ή πλέγμα πίνακα ακροδεκτών). Αυτή είναι μια ορθογώνια ή τετράγωνη συσκευασία που φέρει τους ακροδέκτες από την κάτω πλευρά του ολοκληρωμένου κυκλώματος. Η τυπική απόσταση των ακροδεκτών είναι 2.54 mm

και μπορεί είτε να καλύπτει όλη την επιφάνεια, είτε να αφήνει μια περιοχή στο κέντρο χωρίς ακροδέκτες (αν δεν υπάρχει ανάγκη διασύνδεσης). Τα ολοκληρωμένα κυκλώματα σε συσκευασία PGA τοποθετούνται πάνω σε ειδικές υποδοχές σε κεντρικές πλακέτες, και μπορούν να κολληθούν από βιομηχανικά ρομπότ (επειδή είναι πάρα πολλοί ακροδέκτες και απαιτείται μεγάλη ακρίβεια) πάνω στην πλακέτα.



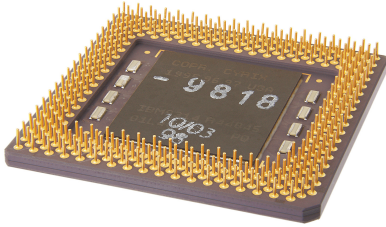
Σχήμα 4.4: Η συσκευασία DIP αποτελείται από 2 σειρές ακροδεκτών. Εικόνα από wikipedia.org.



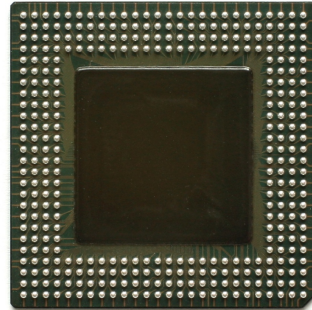
Σχήμα 4.5: Οι συσκευασίες DIP μπορούν να κολληθούν στην πλακέτα ή να τοποθετηθούν σε μια υποδοχή. Εικόνα από wikipedia.org.

Προκειμένου να μειωθεί το κόστος του ολοκληρωμένου κυκλώματος, να μειωθεί το μήκος των ακροδεκτών και οι παρεμβολές, επιτρέποντας με αυτόν τον τρόπο την επίτευξη υψηλότερης συχνότητας λειτουργίας, χρησιμοποιείται η συσκευασία BGA (ball grid array - σφαιρικό πλέγμα πίνακα), στην οποία οι ακροδέκτες είναι σφαίρες συγκόλλησης για την άμεση επιφανειακή κόλληση στην κεντρική πλακέτα του συστήματος (Εικόνα 4.7). Αυτές οι συσκευασίες χρησιμοποιούνται για όλα τα πολύπλοκα ολοκληρωμένα κυκλώματα που έχουν εκατομμύρια τρανζίστορ και απαιτούν υψηλές ταχύτητες επικοινωνίας. Με τη συγκόλληση επιτυγχάνεται και η καλύτερη επαφή του τσιπ με την πλακέτα, η οποία βοηθάει και στη θερμική απαγωγή και ψύξη του επεξεργαστή, ενώ το μειωμένο μήκος των ακροδεκτών ελαττώνει σημαντικά την ηλεκτρομαγνητική επαγωγή, η οποία είναι ένα σημαντικό χαρακτηριστικό για την αποφυγή αλλοίωσης των σημάτων, που συμβαίνει στις PGA συσκευασίες υψηλής συχνότητας. Οι συσκευασίες BGA έχουν το μειονέκτημα ότι απαιτούν εξειδικευμένο ρομποτικό εξοπλισμό ακριβείας για την τοποθέτηση πάνω στην πλακέτα, για αυτό έχουν και το υψηλότερο κόστος.

Τα ενσωματωμένα συστήματα αναλόγως της πολυπλοκότητας χρησιμο-



Σχήμα 4.6: Η συσκευασία PGA αποτελείται από ένα πυκνό πλέγμα ακροδεκτών. Εικόνα από το χρήστη Eric Gaba (Sting) της wikipedia.org.



Σχήμα 4.7: Η συσκευασία BGA αποτελείται αντί για ακροδέκτες από σφαίρες κόλλησης. Εικόνα από wikipedia.org.

ποιούν όλες τις συσκευασίες. Ο σχεδιαστής των ΕΣ θα πρέπει να γνωρίζει όλες αυτές τις συσκευασίες και να μπορεί να επιλέγει την καλύτερη συσκευασία για το πρόβλημά του. Όσο πιο πολύπλοκο είναι το ΕΣ τόσο πιο σύγχρονη συσκευασία θα πρέπει να επιλέξει. Σε περίπτωση που αποφασίζει να σχεδιάσει το δικό του επεξεργαστή για το ΕΣ, τότε ασφαλώς εμφανίζονται και άλλα θέματα σχεδιασμού, που διδάσκονται ή διερευνώνται σε μαθήματα VLSI ή CAD/CAM/CAE.

4.3 Κατηγορίες υλοποίησης

Για πέντε δεκαετίες, η αλματώδης πρόοδος της τεχνολογίας έχει βοηθήσει στην ανάπτυξη και υιοθέτηση των υπολογιστών σε όλους τους τομείς της καθημερινής μας ζωής. Έτσι, αν κοιτάξει κανείς γύρω του θα δει ότι περιτριγυρίζεται από ηλεκτρονικά συστήματα, που εμπεριέχονται για παράδειγμα στις τηλεοράσεις, στα αυτοκίνητα, στους υπολογιστές, στα κινητά τηλέφωνα, σε βομβητές κτλ. Τα περισσότερα από αυτά τα ηλεκτρονικά συστήματα έχουν κυκλώματα που ελέγχουν κάποιες ή όλες τις λειτουργίες της συσκευής.

Στα υπολογιστικά συστήματα γενικού σκοπού, ο σχεδιασμός ή η επιλογή του επεξεργαστή είναι μια ορθογώνια διαδικασία ως προς το λογισμικό και έτσι οι ομάδες είναι τελείως ανεξάρτητες ή ακριβέστερα, συνεργάζονται μόνο οι ομάδες της αρχιτεκτονικής με αυτές που κατασκευάζουν τους μεταγλωττιστές και τους συμβολομεταφραστές. Οι υπόλοιποι προγραμματιστές, απλώς χρησιμοποιούν τα αναπτυξιακά εργαλεία που τους παρέχονται.

Στα ενσωματωμένα συστήματα όμως, πάντα υπάρχει ο συ-σχεδιασμός του

υλικού και του λογισμικού, το οποίο περιγράφεται λεπτομερώς και σε επόμενη παράγραφο. Αρχικά, επιλέγεται ή σχεδιάζεται ο εξειδικευμένος επεξεργαστής που καλύπτει τις απαιτήσεις, ένα αρκετά δύσκολο εγχείρημα που δεν πρέπει να υποτιμάται, μαζί με τις αρχιτεκτονικές λεπτομέρειες που το συνοδεύουν όπως οι κρυφές μνήμες, οι μηχανισμοί συνοχής ή δυνατότητα πολυεπεξεργασίας κτλ. Ο σχεδιασμός του υπόλοιπου συστήματος δεν είναι λιγότερο σημαντικός ή πιο δύσκολος. Υπάρχει αλληλεπίδραση των σχεδιαστικών αποφάσεων ανάμεσα στον επεξεργαστή και στα υπόλοιπα στοιχεία του ενσωματωμένου συστήματος. Για να σχεδιαστεί ένας απλός επεξεργαστής θα πρέπει να εργάζονται δυο ή τρεις μηχανικοί, από έξι έως εννέα μήνες, προκειμένου να δημιουργήσουν μια περιγραφή επιπέδου μεταφοράς καταχωρητών (register transfer level, RTL). Το επόμενο βήμα είναι η κατασκευή, η δοκιμή, η ανάπτυξη λογισμικού κτλ. Κάποιοι πιστεύουν ότι ο πιο δύσκολος στόχος είναι να σχεδιαστεί ένας επεξεργαστής, για αυτό υποστηρίζουν ότι ο σχεδιασμός ενός πολύπλοκου ενσωματωμένου συστήματος ολοκληρώνεται με το σχεδιασμό του επεξεργαστή. Όμως, αυτό είναι λάθος, αφού το πιο σωστό είναι ότι ο ο σχεδιασμός ενός πολύπλοκου ενσωματωμένου συστήματος ξεκινάει με το σχεδιασμό του επεξεργαστή.

Στη συνέχεια τα ηλεκτρονικά κυκλώματα σχεδιάζονταν μόνο σε ολοκληρωμένα κυκλώματα εξειδικευμένα για την εφαρμογή (Application Specific Integrated Circuit, ASIC). Τα κυκλώματα αυτά εκτός από τις υψηλές επιδόσεις είχαν και υψηλό κόστος παραγωγής. Σήμερα, το κόστος αυτό δεν έχει μειωθεί, αλλά αυξάνεται συνεχώς εκθετικά ακολουθώντας το νόμο του Moore [23]. Γι' αυτό το λόγο, ερευνητές έχουν υποστηρίξει ότι οι συνθήκες μας οδηγούν σε μια απομάκρυνση από τα ASIC και σε εντατικοποίηση της χρήσης των προγραμματιζόμενων υλοποιήσεων[31], της άλλης εναλλακτικής πορείας υλοποίησης συστημάτων. Οι δύο αυτές εναλλακτικές πορείες υλοποίησης έχουν θετικά και αρνητικά στοιχεία. Συγκεκριμένα, τα ASIC έχουν πολύ χαμηλή ευελιξία (αφού σχεδιάζονται αποκλειστικά για μια εφαρμογή), έχουν πολύ χαμηλή κατανάλωση ενέργειας με υψηλές επιδόσεις και έχουν πολύ μεγάλο χρόνο ανάπτυξης (Σχήμα 4.8). Στην ακριβώς αντίθετη θέση βρίσκονται οι επεξεργαστές γενικού σκοπού, οι οποίοι είναι αρκετά ευέλικτοι (αφού είναι γενικού σκοπού), αλλά υστερούν στις επιδόσεις και συνοδεύονται από μεγάλη κατανάλωση ενέργειας. Ανάμεσα στις δύο αυτές ακραίες θέσεις, υπάρχουν οι ψηφιακοί επεξεργαστές σήματος, οι οποίοι έχουν μεγάλη ευελιξία, χαμηλό χρόνο ανάπτυξης (αφού είναι προγραμματιζόμενοι και εύκολα μπορεί να μεταφραστεί μια εφαρμογή σε εκτελέσιμη μορφή για τον επεξεργαστή), αλλά η κατανάλωση ενέργειας και οι επιδόσεις υστερούν συγκρινόμενες με τα ASIC. Στις ενδιάμεσες θέσεις επίσης υπάρχουν τα FPGA, τα οποία έχουν μια παρόμοια ροή σχεδιασμού με τα ASIC, αλλά αρκετά πιο γρήγορη αφού παρέχουν τη δυνατότητα του προγραμματισμού, και του χαμηλότερου κόστους, γιατί παράγονται κατά μεγάλες ποσότητες.

Έως τώρα αναφερθήκαμε για υλοποιήσεις ενσωματωμένων συστημάτων που είτε φέρουν έναν επεξεργαστή, είτε αποτελούνται από εξειδικευμένο υλικό. Ως τώρα ήταν η εισαγωγή και δεν έχει δοθεί μια λεπτομερή ταξινόμηση. Είναι όμως ευκολονόητο ότι τα ΕΣ μπορούν να υλοποιηθούν με πάρα πολλούς τρόπους, και αυτό φαίνεται και από το μεγάλο πλήθος των ακρωνυμίων υλοποίησης, όπως CPLD, FPGA, ASIC, SOC, PSOC, DSP, ASIP, ASSP, PAL, CPU, MCPU, που θα περιγράψουμε λεπτομερώς στις επόμενες ενότητες.

4.4 Ταξινόμηση της προγραμματίσιμης και μη προγραμματίσιμης επεξεργασίας

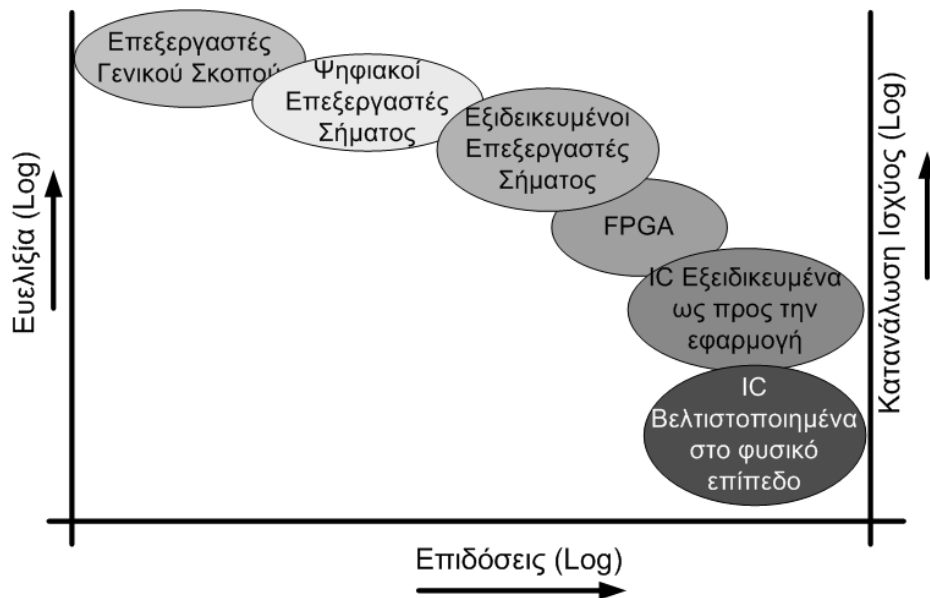
Το Σχήμα 4.8 παρουσιάζει ποικίλες μορφές υπολογισμού που μπορούν να υλοποιήσουν ένα ενσωματωμένο σύστημα. Η ειδίκευση αυξάνεται από τα αριστερά στα δεξιά και η ευελιξία από τα δεξιά στα αριστερά.

Μπορούμε να διαχωρίσουμε ένα ενσωματωμένο σύστημα στην επεξεργασία συστήματος, την επεξεργασία εφαρμογής, και την επιτάχυνση πυρήνων. Ανάλογα με το επιθυμητό επίπεδο κόστους/απόδοσης, μπορούμε να εφαρμόσουμε διάφορους βαθμούς προσαρμογής σε κάθε επίπεδο: γενικής χρήσης, εξαρτώμενοι από το πεδίο ή επεξεργαστές οριζόμενοι από την εφαρμογή, στα υψηλού επιπέδου συστατικά, και προγραμματίσιμη ή μη προγραμματίσιμη επιτάχυνση στους πυρήνες εντατικών υπολογισμών. Στην πραγματικότητα, ένας συνδυασμός αυτών των τεχνικών βρίσκεται συνήθως σε πολλά από τα σημερινά πολύπλοκα ενσωματωμένα συστήματα.

Εξετάζοντας αυτά τα παραδείγματα, είναι σημαντικό να τονισθεί ότι σε πολλές περιπτώσεις οι επιταχυντές υλικού κάνουν τον υπολογισμό περιφερειακά, δηλαδή εκεί που απαιτείται απομακρυσμένα από τον επεξεργαστή, π.χ. εκεί που παράγονται τα δεδομένα ή εκεί που καταναλώνονται για επεξεργασία. Αυτό επιτυγχάνει μείωση της επικοινωνίας και συνεπώς της επιβάρυνσης του δικτύου διασύνδεσης. Βέβαια, υπάρχουν και επιταχυντές που συνεργάζονται στενά με κάποιον επεξεργαστή, π.χ. για την κρυπτογράφηση ή αποκωδικοποίηση video υψηλής ποιότητας. Ασφαλώς, η ανάπτυξη τέτοιων επιταχυντών στενής συνεργασίας με τον επεξεργαστή, απαιτεί ιδιαίτερες δεξιότητες και γνώσεις. Επιπλέον, η ιστορία έχει δείξει ότι η επιτάχυνση υλικού που επιτυγχάνεται με τέτοιους βοηθητικούς πυρήνες επεξεργασίας, είναι συχνά ο πρώτος υποψήφιος για να απορροφηθεί από το λογισμικό, μόλις η ταχύτητα του πυρήνα-επεξεργαστή γίνεται επαρκής για την απαραίτητη απόδοση ή μόλις αυξηθεί ο αριθμός των επεξεργαστών και μπορεί υπολογιστικά να καλυφθούν οι απαιτήσεις. Αυτό δεν είναι παράξενο, αφού ένα λογισμικό αναπτύσσεται και αναβαθμίζεται με μεγαλύτερη ευκολία από ένα υλικό. Ένας άλλος λόγος για να χρη-

σιμοποιηθούν επιταχυντές υλικού σε διεπαφές εισόδου/εξόδου είναι ότι αυτοί είναι ιδιαίτερα αποδοτικοί σε επεξεργασία ροής δεδομένων (data streams), που σε διαφορετική περίπτωση θα έπρεπε να αποθηκευτούν στη μνήμη αυξάνοντας το ενεργειακό ισοζύγιο, αφού οι προσβάσεις στη μνήμη είναι δαπανηρές και άρα η άλλη εναλλακτική αποθήκευσης των δεδομένων στη μνήμη (1 πρόσβαση για εγγραφή) και μετά ανάκληση των δεδομένων για επεξεργασία (1 πρόσβαση για ανάγνωση) και αποθήκευση των νέων δεδομένων στη μνήμη (1 πρόσβαση για εγγραφή), απαιτεί πολλές παραπάνω προσβάσεις, από την άμεση περιφερειακή επεξεργασία και αποθήκευση του αποτελέσματος στη μνήμη (μόνο 1 πρόσβαση για εγγραφή). Προκειμένου ο σχεδιασμός να είναι αποτελεσματικός και σύντομος, θα πρέπει οι επιταχυντές να χρησιμοποιούν προτυποποιημένους διαύλους δεδομένων, όπου έχουν σαφώς ορισμένες και επιβεβαιωμένες διεπαφές και διαγράμματα χρονισμού, ενώ πρέπει να υπάρχει και καλή υποστήριξη τόσο σε block επαναχρησιμοποιήσιμης λογικής, όσο και σε οδηγούς συσκευών. Σε ειδικές συνθήκες μπορούν να δημιουργηθούν νέοι δίαυλοι και νέα πρότυπα επικοινωνίας ανάμεσα στον επιταχυντή και τον επεξεργαστή, αλλά θα πρέπει αυτό να γίνει συνειδητά και με πλήρη επίγνωση όλων των μειονεκτημάτων και πλεονεκτημάτων της επιλογής.

Μέχρι τις αρχές της δεκαετίας του 1980, τα περισσότερα συστήματα κυκλωμάτων λογικής υλοποιούνταν σε μικρής κλίμακας τυποποιημένα ολοκληρωμένα κυκλώματα (large scale integrated LSI circuits), όπως μικροεπεξεργαστές, ελεγκτές διαύλου-E/E, χρονιστές συστημάτων κλπ. Εντούτοις, κάθε σύστημα είχε ακόμα την ανάγκη για 'λογική συγκόλλησης' (glue logic) για να συνδεθεί με μεγαλύτερα κυκλώματα. Για παράδειγμα, απαιτούνταν συνδέσεις πολυπλεξίας σειριακού σε παράλληλο και αντίστροφα ή τροποποίησης επιπέδων λογικής τάσης (12V, 7V, 5V, 3.3V, 1.8V). Τα εξειδικευμένα ολοκληρωμένα κυκλώματα είχαν ως σκοπό να αντικαταστήσουν τη μεγάλη ποσότητα κυκλωμάτων συγκόλλησης και συνεπώς να μειώσουν την πολυπλοκότητα συστημάτων και το κόστος παραγωγής, καθώς επίσης και να βελτιώσουν την απόδοση. Όμως, τα εξειδικευμένα ολοκληρωμένα κυκλώματα ήταν (και είναι) ακριβά για να αναπτυχθούν εκ του μηδενός, ενώ υπήρχαν και μεγάλες χρονικές καθυστερήσεις για την εισαγωγή του προϊόντος στην αγορά λόγω του απαγορευτικά μεγάλου χρόνου σχεδίασης. Επομένως, η προσέγγιση εξειδικευμένου ολοκληρωμένου κυκλώματος ήταν μόνο βιώσιμη για προϊόντα με πολύ μεγάλη ποσότητα (που χαμηλώνει την επίδραση του κόστους σχεδιασμού και ανάπτυξης, NRE). Αντιμετωπίζοντας αυτό το πρόβλημα, η νεοσύστατη εταιρεία Xilinx εισήγαγε στα μέσα της δεκαετίας του '80 (γύρω στα 1984) την τεχνολογία των προγραμματιζόμενων συστοιχιών πυλών στο πεδίο (Field Programmable Gate Arrays, FPGA), ως εναλλακτική λύση των εξειδικευμένων ολοκληρωμένων κυκλωμάτων για την υλοποίηση της λογικής συγκόλλησης. Έτσι, οι επί του πεδίου προγραμματιζόμενες συστοιχίες πυλών αποτέλεσαν μια σημαντική τεχνολογία,



Σχήμα 4.8: Η κάθε πλατφόρμα υλοποίησης έχει διαφορετικά χαρακτηριστικά ως προς τις επιδόσεις, την κατανάλωση ενέργειας και την ευελιξία.

η οποία επέτρεψε στους σχεδιαστές των κυκλωμάτων να υλοποιούν κυκλώματα σε σχετικά μικρό χρονικό διάστημα, αφού δεν απαιτούνταν η διαδικασία της φυσικής χωροθέτησης, της δημιουργίας μασκών και της κατασκευή ολοκληρωμένων κυκλωμάτων.

Το χαμηλό κόστος των επαναπρογραμματιζόμενων δομών έχουν οδηγήσει τους κατασκευαστές στην υιοθέτηση αυτών των δομών ως τις αρχιτεκτονικές υλοποίησης των ενσωματωμένων συστημάτων. Έτσι, οι σχεδιαστές επιλέγουν συνήθως αρχιτεκτονικές με προγραμματιζόμενους επεξεργαστές, ή επεξεργαστές εξειδικευμένους προς την εφαρμογή (Application Specific Instruction Set Processor - ASIP). Όμως, υπάρχουν και απλά ενσωματωμένα που μπορούν να υλοποιηθούν με 4bit ή 8bit μικροελεγκτές, όπως τον ATMEGA328P. Θα περιγράψουμε τους μικροελεγκτές, όπως και την πιο εξελιγμένη μορφή τους (μικροεπεξεργαστές) στην επόμενη παράγραφο.

4.5 Μικροελεγκτές και μικροεπεξεργαστές

Μερικές φορές ένα ενσωματωμένο σύστημα επιτελεί μια απλή λειτουργία που διαβάζει τιμές από αισθητήρες και ενεργοποιεί αντίστοιχες συσκευές δράσης. Τέτοια συστήματα δεν έχουν υψηλές υπολογιστικές απαιτήσεις και μπορούν να χρησιμοποιήσουν μικροελεγκτές ή μικροεπεξεργαστές. Ο μικροελεγ-

κτής (microcontroller, mC) είναι ένα ολοκληρωμένο κύκλωμα που μπορεί να παρέχει αρκετά περιορισμένες υπολογιστικές ικανότητες, αφού συνήθως λειτουργεί σε ταχύτητες κάποιων Mhz και έχει κάποια KB μνήμης RAM, εντούτοις παρέχει αρκετά χαρακτηριστικά που είναι χρήσιμα σε συγκεκριμένες περιπτώσεις. Αυτό το στοιχείο είναι παρόμοιο με έναν επεξεργαστή (ή μικροεπεξεργαστή), δηλαδή έχει την ικανότητα να εκτελεί αριθμητικές και λογικές πράξεις, να μεταφέρει δεδομένα εντός και εκτός του ολοκληρωμένου κυκλώματος και να μπορεί να εκτελέσει ένα πρόγραμμα (αρχιτεκτονική αποθηκευμένου προγράμματος). Τα σημαντικότερα χαρακτηριστικά του μικροελεγκτή είναι τα παρακάτω: Ο μικροελεγκτής έχει χαμηλό κόστος, που είναι πολύ σημαντικό για ενσωματωμένα συστήματα που πρόκειται να κατασκευαστούν σε χιλιάδες ή εκατομμύρια κομμάτια. Επίσης, περιέχει έναν αριθμό από διάφορες ενσωματωμένες λειτουργικές μονάδες (ή περιφερειακές συσκευές) στο ίδιο υπόστρωμα πυριτίου με τον επεξεργαστή. Αυτές οι μονάδες παρέχουν χρήσιμες δυνατότητες για τα ΕΣ, όπως μέτρηση χρόνου με χρονιστές (timers), μέτρηση συμβάντων με μετρητές (counters), υποστήριξη διάφορων τρόπων επικοινωνίας, όπως σειριακή (RS232 ή USB) ή άλλων προτύπων, και υποστήριξη καλύτερης αξιοπιστίας με χρονομετρητές επιτηρητών (watchdog timers), οι οποίοι αν παρατηρήσουν κάποια δυσλειτουργία επανεκκινούν το σύστημα. Επιπρόσθετα, ο μικροελεγκτής περιέχει πάνω στο ίδιο chip, επαναπρογραμματιζόμενη μνήμη που φέρει το αποθηκευμένο πρόγραμμα και είναι είτε τύπου EEPROM, είτε FLASH, και έτσι δε χρειάζεται η αποθήκευση του προγράμματος σε ένα εξωτερικό μέσο όπως ένα δίσκο. Ακόμη, ο μικροελεγκτής παρέχει άμεση πρόσβαση και χειρισμό σε ακροδέκτες για ψηφιακή και αναλογική επεξεργασία ή για να οδηγήσει φορτία ή transistor ενίσχυσης ρεύματος, ενώ έχει και εξειδικευμένες εντολές χειρισμού bit. Ένα ολοκληρωμένο κύκλωμα που έχει όλα τα παραπάνω, ονομάζεται μικροελεγκτής. Η ολοκλήρωση όλων αυτών των υπομονάδων στο ίδιο τσιπ, έχει ως συνέπεια τη δυνατότητα λειτουργίας του με ελάχιστες εξωτερικές απαιτήσεις και έτσι μπορεί να τοποθετηθεί εύκολα σε ΕΣ με μικρό μέγεθος και με χαμηλή απαίτηση ενέργειας. Για παράδειγμα, ένα απλό ΕΣ που βασίζεται σε έναν επεξεργαστή ATmega328 AVR μπορεί να λειτουργήσει με την προσθήκη 2 αντιστάσεων, 2 πυκνωτών, ενός σταθεροποιητή τάσης και ενός κρυστάλλου σταθερής συχνότητας (με συνολικό κόστος υλικού για παραγγελία 1000 κομματιών στα 2\$).

Πολύ σημαντικό στοιχείο είναι η ευκολία πρόσβασης στους ακροδέκτες του ολοκληρωμένου κυκλώματος, αφού μπορούν να συνδεθούν αισθητήρια ή συσκευές δράσης και να χρησιμοποιηθούν με απλές εντολές ανάγνωσης ή εγγραφής. Οι μικροελεγκτές μπορεί να παρέχουν μόνο ρεύμα λίγων mA για οδήγηση στους ακροδέκτες, αλλά με κατάλληλες διατάξεις (π.χ. τρανζίστορ που ενισχύουν την ικανότητα οδήγησης και ελέγχου), μπορούν να χειριστούν συσκευές με μεγάλες απαιτήσεις σε ρεύματα οδήγησης (όπως κινητήρες) ή με κατάλλη-

λες αντιστάσεις να διαβάσουν αισθητήρια που λειτουργούν σε μεγαλύτερες τάσεις. Ασφαλώς, απαιτείται προσοχή στην επιλογή των κατάλληλων βοηθητικών ηλεκτρονικών στοιχείων, διαφορετικά μπορούν να καταστραφούν οι ακροδέκτες του μικροεπεξεργαστή.

Αξίζει να σημειωθεί ότι, εκτός από τους μικροελεγκτές, υπάρχουν και οι μικροεπεξεργαστές (microprocessor, mP) που έχουν παρόμοια στοιχεία, αλλά και κάποιες σημαντικές διαφορές. Οι μικροεπεξεργαστές είναι και αυτοί ολοκληρωμένα κυκλώματα στα οποία εμπεριέχονται πιο σύνθετα δομοστοιχεία από τους mC με περισσότερες λειτουργικές μονάδες και με μεγάλες υπολογιστικές ικανότητες. Για παράδειγμα μπορεί να έχουν 3 αριθμητικές λογικές μονάδες (ALU), 2 μονάδες πράξεων πραγματικών αριθμών (FPU), 2 μονάδες πράξεων διανυσματικών στοιχείων (SIMD), ελεγκτές DMA, ελεγκτές video και ήχου και πολλά άλλα. Αυτά είναι τα επεξεργαστικά στοιχεία των σημερινών προσωπικών υπολογιστών και ασφαλώς έχουν πολλαπλάσιο κόστος από τους mC. Δε μπορούν να λειτουργήσουν με προσθήκη απλών ηλεκτρονικών στοιχείων όπως οι mC, και απαιτείται μια κεντρική πλακέτα (motherboard) πάνω στην οποία τοποθετούνται αυτοί και τα παθητικά και ενεργητικά ηλεκτρονικά στοιχεία υποστήριξης.

4.6 Επεξεργαστές Ειδικής Εφαρμογής

Οι σημερινές ενσωματωμένες πολυμεσικές εφαρμογές, όπως η αποκωδικοποίηση επίγειου τηλεοπτικού σήματος υψηλής ανάλυσης (Full HD 1080p, 4K, 8K) που χρησιμοποιεί κωδικοποίηση MPEG4 ή H264, απαιτούν υψηλή υπολογιστική ισχύς και πολύ συγκεκριμένη λειτουργικότητα. Οι απαιτήσεις για απόδοση, ισχύς, κόστος ή μέγεθος για διάφορες εφαρμογές δεν μπορούν να αντιμετωπιστούν αποτελεσματικά με τη χρήση μικροεπεξεργαστών 8bit. Μπορούν να χρησιμοποιηθούν επεξεργαστές ειδικού σκοπού (ASIC), αλλά έχουν υψηλό κόστος κατασκευής και δεν είναι προγραμματιζόμενοι. Μια λύση είναι να χρησιμοποιήσουμε επεξεργαστές συνόλου εντολών που είναι ειδικοί σε μια εφαρμογή ή μια συγκεκριμένη περιοχή εφαρμογών. Αυτοί οι επεξεργαστές συνόλου εντολών ονομάζονται επεξεργαστές συνόλου εντολών ειδικής εφαρμογής (application-specific instruction-set processors - ASIP), οι οποίοι μπορούν να προγραμματιστούν γράφοντας προγράμματα σε γλώσσα υψηλού επιπέδου (όπως η C). Με αυτόν τον τρόπο έχουμε ευελιξία, ενώ η απόδοση και οι άλλοι περιορισμοί ικανοποιούνται αποτελεσματικά. Ο επεξεργαστής συνόλου εντολών ειδικής εφαρμογής είναι ένας προγραμματιζόμενος επεξεργαστής, ο οποίος είναι βελτιστοποιημένος για την εκτέλεση μιας συγκεκριμένης κλάσης εφαρμογών με κοινά χαρακτηριστικά, όπως η ψηφιακή επεξεργασία σήματος, τηλεπικοινωνίες κτλ. Ένα χαρακτηριστικό αυτού του τύπου επεξεργαστή, είναι ότι

ο σχεδιαστής μπορεί να βελτιστοποιήσει το μονοπάτι δεδομένων ανάλογα με τις απαιτήσεις της κλάσης εφαρμογών. Με άλλα λόγια, ο επεξεργαστής μπορεί να περιέχει καταχωρητές ειδικού σκοπού και διαύλους για την ταχεία ολοκλήρωση των υπολογισμών. Επίσης, ο σχεδιαστής μπορεί να προσθέσει λειτουργικές μονάδες ειδικού σκοπού για να εκτελέσουν ορισμένες κοινές απαιτητικές λειτουργίες σε λιγότερους κύκλους ρολογιού και να απαλείψει κάποιες άλλες μονάδες οι οποίες είναι περιττές.

Έκτος των βασικών λειτουργικών μονάδων, ο επεξεργαστής μπορεί να περιέχει μονάδες ελέγχου ειδικού σκοπού, για να εκτελέσουν κοινές και συνδυασμένες λειτουργίες σε λιγότερους κύκλους, όπως η εκτέλεση λειτουργιών πολλαπλασιασμού και πρόσθεσης σε μια μόνο εντολή. Με τη χρήση του επεξεργαστή ειδικής εφαρμογής μέσα στο ενσωματωμένο σύστημα, υπάρχουν πολλά οφέλη: Ευελιξία, γρήγορη ανάπτυξη και αποσφαλμάτωση, ενώ συγχρόνως η χρήση του συγκεκριμένου επεξεργαστή συνδέεται με πολύ καλή απόδοση για τη συγκεκριμένη κλάση εφαρμογών, μικρότερο μέγεθος από ένα γενικού τύπου επεξεργαστή και συνεπώς μικρότερη κατανάλωση με μικρότερο κόστος. Ως προς τους μηχανικούς που σχεδιάζουν και κατασκευάζουν αυτούς τους επεξεργαστές, απαιτείται μεγάλο κόστος στην έρευνα και ανάπτυξη τόσο του υλικού, όσο και του λογισμικού υποστήριξης, όπως είναι οι μεταγλωττιστές και οι συμβολομεταφραστές.

4.7 Επεξεργαστές Ψηφιακού Σήματος

Ένα παράδειγμα επεξεργαστή, από την κλάση επεξεργαστών ASIP, είναι οι επεξεργαστές ψηφιακού σήματος (Digital Signal Processor - DSP). Ένα DSP είναι ένας επεξεργαστής σχεδιασμένος για την ψηφιακή επεξεργασία ενός μεγάλου όγκου δεδομένων. Η προέλευση αυτού του μεγάλου όγκου δεδομένων είναι σε μορφή ψηφιακού σήματος, όπως οι εικόνες από μια ψηφιακή κάμερα, ένα πλαίσιο ήχου (ή φωνής) μέσω μιας συσκευής τηλεφωνίας διαδικτύου (Voice Over Internet Protocol, VOIP), φίλτρα, δρομολογητές δικτύου κτλ.

Συνήθως, ένα DSP περιέχει πολλούς καταχωρητές, πολλαπλά τμήματα μνήμης, πολλαπλασιαστές και άλλες αριθμητικές μονάδες. Επίσης, τα DSP περιέχουν ειδικές εντολές για ψηφιακή επεξεργασία σήματος, όπως το φιλτράρισμα σήματος, μετασχηματισμούς ή συνδυασμός αυτών. Επιπλέον, ένα DSP περιέχει συχνά αριθμητικές μονάδες για την υλοποίηση απαιτητικών λειτουργιών, όπως οι πράξεις πολλαπλασιασμού και πρόσθεσης ή ολίσθησης και πρόσθεσης, οι οποίες υλοποιούνται σε υλικό και εκτελούνται πολύ ταχύτερα από μια υλοποίηση λογισμικού σε έναν μικροεπεξεργαστή. Για παράδειγμα, ο επεξεργαστής DSP περιέχει μονάδες ειδικού σκοπού, όπως μια μονάδα πολλαπλασιασμού-πρόσθεσης (multiply-accumulate unit, MAC), η οποία μπορεί να εκτελεί μια

πράξη σαν τη $t = t + M[i] * k$ με μια μόνο εντολή. Επίσης, ένα DSP μπορεί να επιτρέψει τη παράλληλη εκτέλεση μερικών λειτουργιών. Για παράδειγμα, επειδή τα προγράμματα DSP συχνά χειρίζονται μεγάλους πίνακες δεδομένων, ένα DSP μπορεί να περιέχει ένα εξειδικευμένο υλικό ώστε να ανακαλεί ακολουθιακές θέσεις μνήμης παράλληλα με άλλες πράξεις, έτσι ώστε να επιταχύνει ακόμα περισσότερο την εκτέλεση. Τέλος, ένα DSP μπορεί να περιέχει έναν αριθμό ενσωματωμένων περιφερειακών συσκευών, που είναι χρήσιμες για την επεξεργασία σήματος πάνω σε ένα ολοκληρωμένο κύκλωμα. Για παράδειγμα, μια συσκευή DSP πιθανώς να περιέχει μετατροπείς σήματος αναλογικό σε ψηφιακό, ψηφιακό σε αναλογικό, χρονοιστές, μετρητές συμβάντων, ελεγκτές άμεσης πρόσβασης μνήμης, σειριακές θύρες επικοινωνίας κλπ.

4.8 Ολοκληρωμένα Κυκλώματα Εξειδικευμένης Εφαρμογής

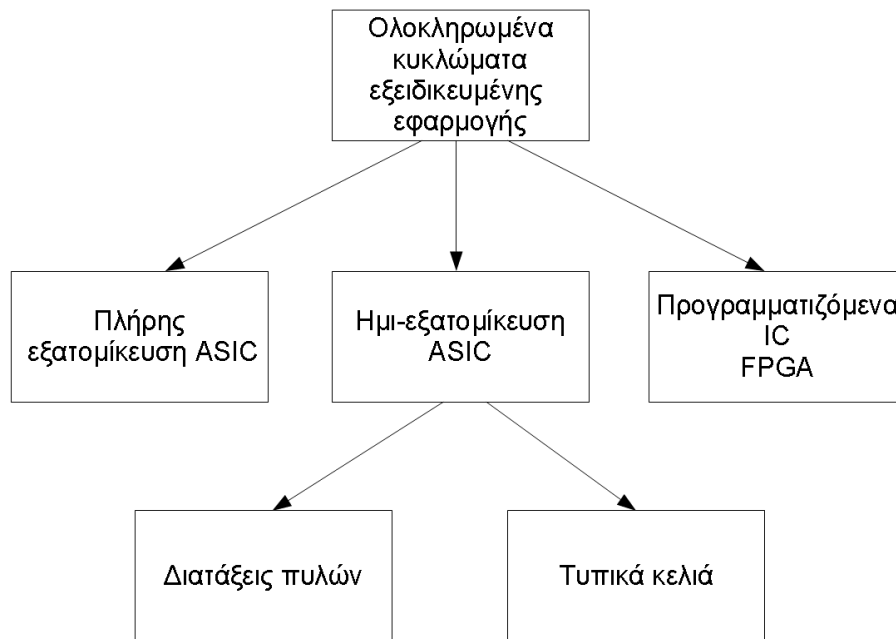
Τα ολοκληρωμένα κυκλώματα εξειδικευμένης εφαρμογής (Application-specific integrated circuits - ASICs) αναφέρονται σε εκείνα τα ολοκληρωμένα κυκλώματα που κατασκευάζονται ειδικά για συγκεκριμένες υπολογιστικές εργασίες. Με άλλα λόγια, το κάθε ολοκληρωμένο κύκλωμα κάνει μια συγκεκριμένη εργασία γρήγορα, αλλά ενδεχόμενα είναι ακατάλληλο για άλλη δουλειά έστω και αν εκείνη έχει πάνω από 90% ομοιότητα με την πρώτη. Για παράδειγμα, ένα κύκλωμα ASIC που χρησιμοποιείται για την εξόρυξη νομισμάτων bitcoin χρησιμοποιώντας συναρτήσεις σύντοψης, είναι παντελώς ακατάλληλο για την εξόρυξη νομισμάτων σε οποιοδήποτε άλλο νόμισμα ή ακόμη και για υπολογισμό συναρτήσεων σύντοψης για άλλα θέματα. Στο ASIC οι λειτουργίες που επιτελεί κατασκευάζονται μόνιμα πάνω στο υπόστρωμα και δε μπορούν να μεταβληθούν.

Εντούτοις, υπάρχουν περιπτώσεις που προτιμάται η χρήση ενός ASIC ως προς έναν μικροεπεξεργαστή ή μικροελεγκτή, ακόμη και αν δεν παρέχει καθόλου ευελιξία. Στα πλεονεκτήματα που στηρίζουν την επιλογή του ASIC, συγκαταλέγονται τα παρακάτω: αυξανόμενη ταχύτητα, μικρή κατανάλωση ενέργειας, χαμηλό κόστος (για τη μαζική παραγωγή), καλύτερη διασφάλιση ορθής λειτουργίας, καλύτερος έλεγχος των χαρακτηριστικών E/E και περισσότερο συμπαγή σχεδιασμό πλακέτας (λιγότερο σύνθετο PCB).

Η επιλογή ενός ASIC συνδέεται και με σημαντικά μειονεκτήματα, όπως μακροχρόνια σχεδίαση, δυσκολία στη ρύθμιση της γραμμής παραγωγής του ASIC στα εργοστάσια κατασκευής (απαιτούνται αρκετές δεκάδες εβδομάδες), ακριβή διαδικασία για χαμηλού όγκου παραγωγή, πολύ υψηλό κόστος ανάπτυξης (υψηλή επένδυση στα εργαλεία CAD και σε σταθμούς εργασίας) και τέλος

η λειτουργία του είναι δεσμευμένη και δε μπορεί να τροποποιηθεί.

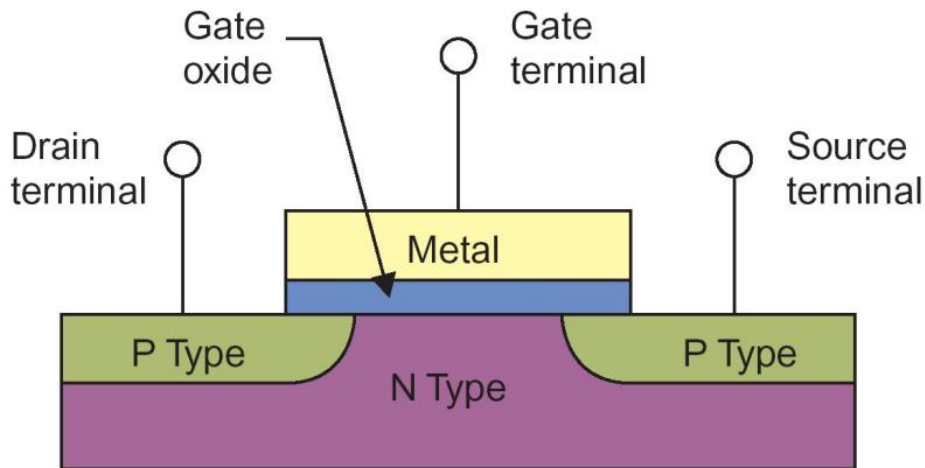
Τα ολοκληρωμένα κυκλώματα εξειδικευμένης εφαρμογής μπορούν να ταξινομηθούν σε τρεις κατηγορίες: η πλήρους εξατομίκευση ASIC (full-custom), η ημι-εξατομίκευση ASIC (semicustom) και επί του πεδίου προγραμματιζόμενων ολοκληρωμένων κυκλωμάτων (field programmable IC), όπως φαίνεται στο Σχήμα 4.9. Η τελευταία κατηγορία αναφέρεται μερικές φορές ως επί του πεδίου προγραμματιζόμενοι συστοιχίες πυλών (field programmable gate arrays, FPGA), που θα την εξετάσουμε διεξοδικά στην επόμενη ενότητα.



Σχήμα 4.9: Κατηγορίες ολοκληρωμένων κυκλωμάτων εξειδικευμένης εφαρμογής.

Πριν αναλύσουμε τις παραπάνω κατηγορίες πρέπει να σημειώσουμε ότι ένα ολοκληρωμένο κύκλωμα (integrated circuit) ή αλλιώς τσιπ (chip) είναι μια συσκευή ημιαγωγών (semiconductor device) που αποτελείται από ένα σύνολο συνδεδεμένων τρανζίστορ και άλλων συσκευών. Υπάρχει ένα πλήθος διαφορετικών διαδικασιών και τεχνολογιών για να κατασκευαστούν ημιαγωγοί, η πιο δημοφιλής είναι ο συμπληρωματικός ημιαγωγός μετάλλου-οξειδίου (complementary metal oxide semiconductor - CMOS [32]). Έτσι, οι ημιαγωγοί (semiconductors) αποτελούνται από πολλά στρώματα μετάλλου, οξειδίου του πυριτίου και απλού πυριτίου, όπως φαίνεται στο Σχήμα 4.10. Τα κατώτατα στρώματα σχηματίζουν τα τρανζίστορ. Τα μεσαία στρώματα σχηματίζουν λογικά συστατικά ή λογικές πύλες. Τα ανώτερα στρώματα συνδέουν αυτά τα λογικά συστατικά με αγωγούς.

Ένας τρόπος για να δημιουργηθούν αυτά τα στρώματα είναι η τοποθέτηση φωτοευαίσθητων χημικών ουσιών στην επιφάνεια του τσιπ και έπειτα η είσοδος του φωτός μέσω των масκών για να τροποποιηθούν οι περιοχές των χημικών ουσιών. Ένα σύνολο масκών ονομάζεται συχνά κατάστρωση (layout).



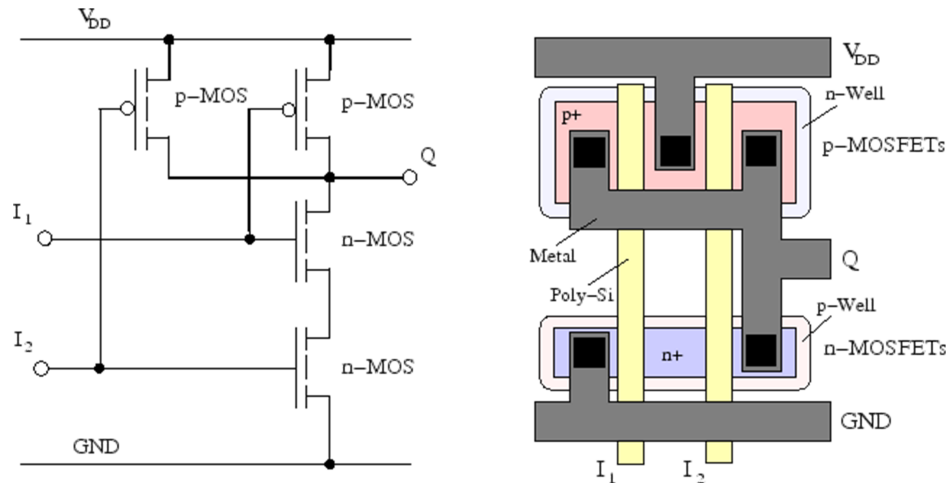
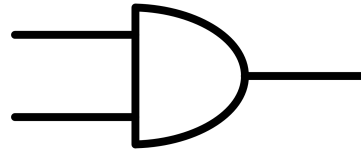
Σχήμα 4.10: Ένα τρανζίστορ CMOS αποτελείται από πολλά στρώματα υλικού. Εικόνα από wikipedia.org.

Συνεπώς, για να αποκτήσουμε ένα ολοκληρωμένο κύκλωμα που υλοποιεί μια συγκεκριμένη υπολογιστική εργασία, πρέπει τελικά να κατασκευαστούν όλα τα στρώματα ή αλλιώς η κατάστρωση. Το ζητούμενο είναι ποιος κατασκευάζει κάθε στρώμα και πότε. Η απάντηση στο ερώτημα αυτό βρίσκεται στις τρεις κατηγορίες που θα αναφέρουμε παρακάτω.

ASIC πλήρους εξατομίκευσης

Σε ένα ASIC πλήρους εξατομίκευσης, υλοποιούνται όλα τα επίπεδα στο πυρίτιο με την πλήρη δημιουργία όλων των στρωμάτων масκών, όπως φαίνεται στο Σχήμα 4.11.

Έτσι, βελτιστοποιούμε όλα τα στρώματα για μια συγκεκριμένη ψηφιακή υλοποίηση του ενσωματωμένου συστήματος. Τέτοια βελτιστοποίηση περιλαμβάνει την τοποθέτηση των τρανζίστορ για να ελαχιστοποιηθούν τα μήκη διασύνδεσης, κλιμάκωση των τρανζίστορ (δηλαδή βέλτιστη επιλογή μεγέθους τρανζίστορ) για να βελτιστοποιηθούν οι μεταδόσεις σημάτων και συνδέσεις μεταξύ των τρανζίστορ μέσω καλωδίων χωρίς να τέμνονται άλλα καλώδια ή τρανζίστορ. Μόλις ολοκληρώσουμε όλες τις μάσκες ή αλλιώς δημιουργήσουμε μια πλήρη κατάστρωση, στέλνουμε τις προδιαγραφές των масκών σε έναν προμηθευτή που κατασκευάζει τα ολοκληρωμένα κυκλώματα. Τα πλεονεκτήματα



Σχήμα 4.11: Σε μια σχεδίαση πλήρους - εξατομίκευσης πρέπει να ορίζεται κάθε στρώμα υλικού.

των σχεδιασμών πλήρης εξατομίκευσης περιλαμβάνουν την υψηλότερη απόδοση, την αποτελεσματική ισχύ και το μικρότερο μέγεθος ψηφίδων ή κύβων (die). Όμως, τα κύρια μειονεκτήματα είναι ο μεγαλύτερος χρόνος σχεδίασης, η υψηλότερη πολυπλοκότητα και κόστος, μαζί με τον υψηλότερο κίνδυνο αποτυχίας. Αυτή η επιλογή σχεδιασμού έχει μόνο νόημα όταν δεν είναι διαθέσιμες ούτε οι βιβλιοθήκες ούτε οι πυρήνες πνευματικής ιδιοκτησίας IP, ή όταν απαιτούνται πολύ υψηλές αποδόσεις. Επανειλημμένως, λίγα έργα είναι πραγματικά 'πλήρης - εξατομίκευσης' λόγω του πολύ υψηλού κόστους και του απαγορευτικού αργού χρόνου εισαγωγής στην αγορά. Οι περισσότερες από τις εργασίες πλήρης - εξατομίκευσης συσχετίζονται με την παραγωγή βιβλιοθήκης κελιών ή δευτερεύοντα μέρη ενός πλήρους σχεδιασμού. Παραδείγματα ολοκληρωμένων κυκλωμάτων πλήρης - εξατομίκευσης βρίσκονται σε συστήματα χαμηλών χρόνων απόκρισης (π.χ. αυτοκίνητα, ηλεκτρονικά συστήματα αεροσκαφών) ή ειδικών απαιτήσεων (π.χ. στρατιωτικές εφαρμογές ή διαστημικά ηλεκτρονικά). Παραδοσιακά, οι μικροεπεξεργαστές και οι μνήμες ήταν αποκλειστικά πλήρης - εξατομίκευσης, αλλά η βιομηχανία στρέφεται όλο και περισσότερο σε τεχνικές ημι-εξατομίκευσης ASIC.

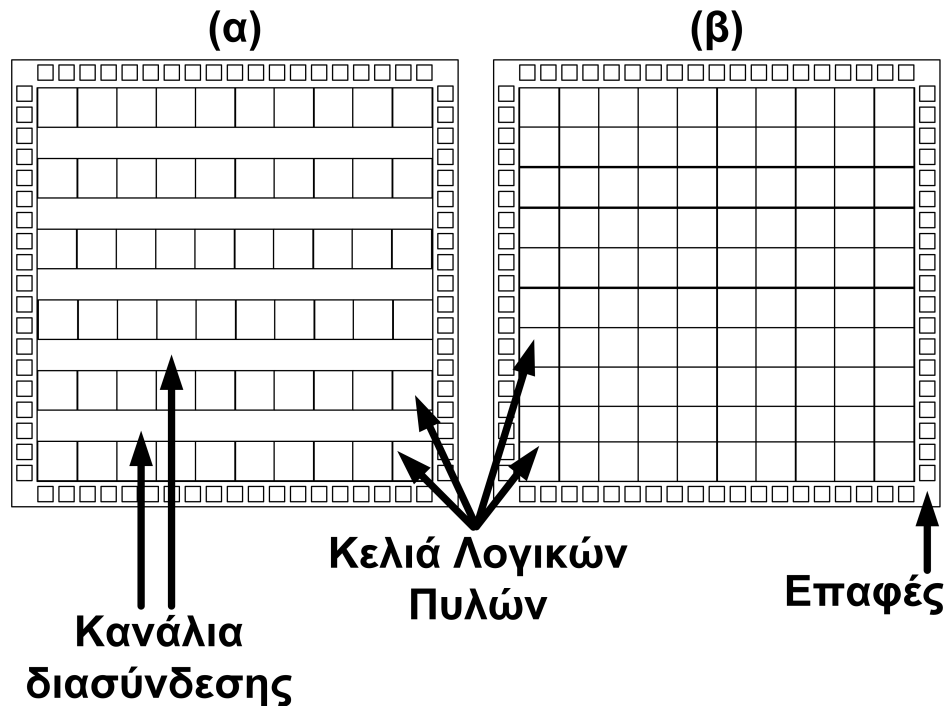
4.8.1 ASIC Ημι-Εξατομίκευσης

Το κόστος έρευνας και ανάπτυξης για πλήρη εξατομικευμένη σχεδίαση ASIC είναι πολύ μεγάλο και έτσι χρησιμοποιείται μόνο σε περιπτώσεις που κάποιο ASIC θα πουληθεί σε εκατομμύρια κομμάτια, όπως είχαμε συζητήσει στο πρώτο κεφάλαιο. Προκειμένου να μειωθεί το υπερβολικά υψηλό κόστος της πλήρους εξατομίκευσης στα περισσότερα έργα, έχει αναπτυχθεί μια ευρεία ποικιλία προσεγγίσεων σχεδιασμού για να μειωθεί ο χρόνος σχεδίασης, να μειωθεί ο χρόνος εισαγωγής προϊόντος στην αγορά, να μειωθούν τα κόστη και να αυτοματοποιηθούν οι διαδικασίες. Αυτές οι προσεγγίσεις συνήθως καλούνται ημι-εξατομίκευση (semicustom). Οι σχεδιασμοί ημι-εξατομίκευσης υλοποιούνται σε επίπεδο λογικών πυλών. Υπό αυτήν την έννοια, χάνουν μερικώς την ευελιξία που ήταν διαθέσιμη από την πλήρους - εξατομίκευσης - δηλαδή καταβάλλεται το κόστος για μια πολύ ευκολότερη τεχνική σχεδίασης. Οι λύσεις ημι-εξατομίκευσης μπορούν να ταξινομηθούν περαιτέρω σε διατάξεις πυλών (gate arrays) και τυπικά κελιά (standard cell).

Διατάξεις Πυλών: Οι διατάξεις πυλών ASIC αποτελούνται από συνεχείς διατάξεις τρανζίστορ των τύπων p και n . Ο προμηθευτής πυριτίου παρέχει προσχεδιασμένα πλακίδια βάσης (master or base wafers), τα οποία περιέχουν ένα σύνολο μασκών προκαθορισμένων πυλών και έπειτα διαμορφώνονται σύμφωνα με τις πληροφορίες διασύνδεσης που παρέχονται από τον πελάτη, δηλαδή ο σχεδιαστής συνδέει τις πύλες για να υλοποιήσει ένα συγκεκριμένο κύκλωμα. Επομένως, ο σχεδιαστής παίρνει τις εξατομικευμένες πληροφορίες που ορίζουν τις συνδέσεις μεταξύ των τρανζίστορ σε μια διάταξη πυλών. Παρόλο που μια διάταξη πυλών προτυποποιεί το τσιπ σε επίπεδο γεωμετρίας, τυπικά η αλληλεπίδραση χρήστη ακόμα υφίσταται σε λογικό επίπεδο. Η απεικόνιση, από τρανζίστορ σε πύλες, εκτελείται μέσω ενός ειδικού εργαλείου CAD (computer aided design, σχεδίαση υποβοηθούμενη από τον υπολογιστή). Η διάταξη πυλών χρησιμοποιεί μια βιβλιοθήκη συστατικών και μακροεντολών που μειώνουν το χρόνο ανάπτυξης. Δύο κύριοι τύποι διατάξεων πυλών μπορούν να αναφερθούν: με κανάλι (channeled) και χωρίς κανάλι (channelless), όπως φαίνεται στο Σχήμα 4.12.

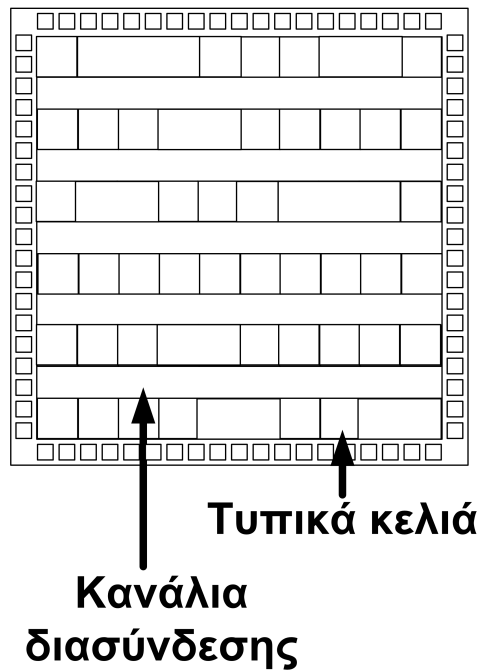
Σε μια διάταξη πυλών με κανάλι, οι διασυνδέσεις σχεδιάζονται μέσα στα προκαθορισμένα διαστήματα (κανάλια) μεταξύ των γραμμών των λογικών κελιών. Σε μια διάταξη πυλών χωρίς κανάλι, γίνεται αντιληπτό ότι δεν υπάρχουν κανάλια σύνδεσης. Δηλαδή, οι συνδέσεις σχεδιάζονται με τα ανώτερα στρώματα μεταλλικών, στην κορυφή των λογικών κελιών. Και στις δύο περιπτώσεις, μόνο μερικά στρώματα μασκών (τα ανώτερα) πρέπει να σχεδιαστούν και να κατασκευαστούν, για τη συγκεκριμένη εφαρμογή.

Τυπικά Κελιά ASIC: Τα τυπικά κελιά είναι λογικά συστατικά, όπως πύλες, πολυπλέκτες, αθροιστές, flip-flops, που σχεδιάζονται προηγουμένως και απο-



Σχήμα 4.12: Δύο αρχιτεκτονικές διατάξεων πυλών: διατάξεις πυλών channeled και channelless.

θηκεύονται σε μια βιβλιοθήκη. Έτσι, τα τρανζίστορ μέσα σε ένα κελί είναι προ-σχεδιασμένα, αλλά δεν έχει προσδιοριστεί η τοποθέτηση των κελιών. Ο σχεδιαστής για να υλοποιήσει μια συγκεκριμένη εφαρμογή πρέπει να επιλέξει ποια κελιά θα χρησιμοποιήσει και στην συνέχεια ασχολείται με την τοποθέτηση και την σύνδεση τους. Για αυτό μια σχεδίαση δημιουργείται χρησιμοποιώντας τις επιλεγμένες βιβλιοθήκες κελιών ως είσοδοι σε ένα σύστημα CAD: ο κώδικας περιγραφής μιας γλώσσας περιγραφής υλικού (hardware description language - HDL) συνθέτεται και επιλέγονται τα κατάλληλα κελιά. Έπειτα, ένα εργαλείο CAD μετατρέπει αυτόματα τη σχεδίαση σε μια κατάστρωση τσιπ (δηλαδή, τοποθέτηση και σύνδεση μεταξύ των κελιών). Οι σχεδιασμοί τυπικών κελιών οργανώνονται στο τσιπ, ως γραμμές κελιών σταθερού ύψους, όπως φαίνεται στο Σχήμα 4.13. Μαζί με τα συστατικά κελιών λογικού επιπέδου, τα συστήματα τυπικών κελιών προσφέρουν συναρτήσεις υψηλού επιπέδου όπως πολλαπλασιαστές και διατάξεις μνήμης. Με αυτό τον τρόπο, επιτρέπεται η χρήση προκαθορισμένων συστατικών υψηλού επιπέδου για την ολοκλήρωση της σχεδίασης.



Σχήμα 4.13: Περίγραμμα ενός τυπικού κελιού.

4.8.2 Σχεδιαστική Ροή στα ASIC

Τα βήματα σε μια παραδοσιακή σχεδιαστική ροή ASIC είναι:

- Εισαγωγή σχεδιασμού (design entry): εισαγωγή σχεδιασμού χρησιμοποιώντας είτε μια γλώσσα περιγραφής υλικού όπως οι VHDL, Verilog ή ένα σχηματικό συντάκτη (schematic editor).
- Λογική σύνθεση (logic synthesis): από την γλώσσα περιγραφής υλικού εξάγεται μια λίστα διασυνδέσεων netlist, δηλαδή μια περιγραφή των λογικών κελιών και των συνδέσεων τους.
- Διαμερισμός συστήματος (system partitioning): διαιρεί ένα μεγάλο σύστημα σε τμήματα μεγέθους ASIC.
- Προσομοίωση συμπεριφοράς (behavioral simulation): ελέγχει την λειτουργία των κυκλωμάτων.
- Χωροθέτηση (floorplanning): τακτοποιεί τα διαφορετικά τμήματα του κυκλώματος στο τσιπ.
- Τοποθέτηση (placement): τοποθετεί τις θέσεις των κελιών σε ένα τμήμα.

- Διασύνδεση (routing): δημιουργεί συνδέσεις μεταξύ των κελιών και των τμημάτων.
- Εξαγωγή (extraction): προσδιορίζει την αντίσταση και την χωρητικότητα των διασυνδέσεων και υπολογίζει τις καθυστερήσεις για τους σκοπούς της προσομοίωσης.
- Φυσική προσομοίωση (physical simulation): ελέγχει την λειτουργία του κυκλώματος μετά την συμπερίληψη των καθυστερήσεων που δημιουργούνται από τα φορτία διασύνδεσης.
- Έλεγχος κανόνων σχεδίασης (design rule check - DRC): ελέγχει ότι η κατάστρωση του κυκλώματος συμμορφώνεται με τις προδιαγραφές των κανόνων σχεδίασης. Τα εργαλεία DRC κυμαίνονται από μια απλή φυσική εξέταση χώρου μέχρι και σύνθετους ελέγχους.

4.8.3 Επί του Πεδίου Προγραμματιζόμενοι Συστοιχίες Πυλών (FPGA)

Μέχρι τώρα είδαμε ολοκληρωμένα κυκλώματα εξειδικευμένης εφαρμογής που τα κυκλώματά τους είναι σταθερά και μόνιμα. Δηλαδή, κυκλώματα που εκτελούν μια συγκεκριμένη λειτουργία και μόλις κατασκευαστούν δεν μπορούν να αλλάξουν την λειτουργικότητά τους. Επίσης, το κόστος ανάπτυξης για να κατασκευάσουμε ένα ολοκληρωμένο κύκλωμα (δηλαδή, να δημιουργήσουμε μια κατάστρωση και μάσκες) μπορεί να είναι πάρα πολύ ακριβό για να γίνει απόσβεση, εάν ο αριθμός των κυκλωμάτων που εκτιμάμε πως θα πουληθούν είναι μικρός. Επιπλέον, η κατασκευή ενός ολοκληρωμένου κυκλώματος είναι ένα ρίσκο, εφόσον μετά την κατασκευή ανακαλύψουμε ότι ένα ολοκληρωμένο κύκλωμα δεν λειτουργεί σωστά στο σύστημα στόχου εξαιτίας των προβλημάτων κατασκευής ή εσφαλμένου αρχικού σχεδιασμού. Συνεπώς, θα θέλαμε μια λογική συσκευή που θα μας επιτρέψει να υλοποιήσουμε την λειτουργικότητα του συστήματος πάνω στην συσκευή αυτή και να μην απαιτείται από μας να κατασκευάσουμε το ολοκληρωμένο κύκλωμα. Γι αυτό το λόγο υπάρχουν εμπορικές προγραμματιζόμενες λογικές συσκευές, τις οποίες μπορούμε να προγραμματίσουμε επί τόπου (ή επί του πεδίου εφαρμογής) στο εργαστήριό μας. Εδώ, ο όρος πρόγραμμα δεν αναφέρεται στην συγγραφή λογισμικού το οποίο εκτελείται σε ένα μικροεπεξεργαστή, αλλά σημαίνει διαμόρφωση λογικών κυκλωμάτων και διασύνδεση διακοπών ώστε να υλοποιήσουμε ένα συγκεκριμένο επιθυμητό κύκλωμα. Συνεπώς, οι προγραμματιζόμενες λογικές συσκευές έχουν το πλεονέκτημα ότι μπορούν να αλλάξουν την λειτουργία των κυκλωμάτων σύμφωνα με τις ανάγκες της εφαρμογής που θέλουμε να υλοποιήσουμε. Η αλλαγή ασφαλώς

υλοποιείται από τον προγραμματιστή ή σχεδιαστή του υλικού, και με τη μεταφόρτωση του παραγόμενου αρχείου περιγραφής, που ονομάζεται bitstream.

Η τεχνολογία FPGA μας επιτρέπει να αγοράσουμε ένα προκατασκευασμένο ολοκληρωμένο κύκλωμα που θα περιέχει όλα τα στρώματα λογικής και διασύνδεσης, πριν ακόμα υλοποιήσουμε την λειτουργικότητα της εφαρμογής μας. Τα στρώματα υλοποιούν ένα προγραμματιζόμενο κύκλωμα, όπου ο προγραμματισμός έχει μια σημασία χαμηλού επιπέδου παρά ένα πρόγραμμα λογισμικού. Έτσι, ο προγραμματισμός που λαμβάνει χώρα, αποτελείται από τη δημιουργία ή καταστροφή συνδέσεων ανάμεσα στα καλώδια, τα οποία συνδέουν πύλες είτε καταστρέφοντας ('καίγοντας') μια ασφάλεια ή θέτοντας ένα bit σε ένα προγραμματιζόμενο διακόπτη ή πολυπλέκτη. Υπάρχουν μικρές συσκευές που λέγονται προγραμματιστές (programmers), όπως το USB Blaster της Altera ή το JTAG Programmer της Xilinx, οι οποίες συνδέονται με έναν υπολογιστή γραφείου και εκτελούν τον προγραμματισμό που αναφέραμε. Τέλος, υπάρχουν πολλές προγραμματιζόμενες λογικές συσκευές, αλλά η πιο δημοφιλής είναι η επί του πεδίου προγραμματιζόμενες συστοιχίες πυλών (Field Programmable Gate Arrays - FPGA). Οι συσκευές FPGAs μπορούν να υλοποιήσουν ένα οποιοδήποτε σχεδιασμό υλικού, όπως ένα ολόκληρο πυρήνα επεξεργαστή (processor core), μαζί με τα περιφερειακά εντός του ίδιου ολοκληρωμένου κυκλώματος. Επίσης, αυτές οι συσκευές είναι ειδικά κατάλληλες για ταχεία προτυποποίηση ενός τμήματος υλικού το οποίο τελικά θα υλοποιηθεί αργότερα σε ένα ASIC. Παρακάτω δίνουμε μια σύντομη ιστορική ανασκόπηση και περιγράφουμε την βασική αρχιτεκτονική των FPGA.

Ο βασικός σκοπός της τεχνολογίας FPGA είναι η επίτευξη ικανοποιητικής απόδοσης στην υλοποίηση του κυκλώματος, σε σύγκριση με τα κυκλώματα εξειδικευμένου σκοπού, ενώ παράλληλα εξακολουθούν να παρέχουν το πλεονέκτημα της ευελιξίας όπως εμφανίζουν οι επεξεργαστές γενικού σκοπού. Επίσης, σε σχέση με τα κυκλώματα εξειδικευμένου σκοπού, η τεχνολογία FPGA έχει διάφορα πλεονεκτήματα όπως: ταχύς χρόνος για την εισαγωγή προϊόντος στην αγορά, χαμηλό μη-επαναλαμβανόμενο κόστος κατασκευής, δεν απαιτείται καμία μάσκα κατασκευής, ιδανικές για προτυποποίηση και τέλος γίνονται σχεδιαστικές αλλαγές χωρίς κόστος. Έτσι, διαμορφώνοντας μια συσκευή με ένα καινούργιο κύκλωμα, τα λάθη σχεδίασης μπορούν εύκολα να διορθωθούν, να προστεθούν νέα χαρακτηριστικά ή η λειτουργία του υλικού μπορεί εύκολα να απεικονιστεί σε άλλες εφαρμογές. Φυσικά, τα ολοκληρωμένα κυκλώματα FPGA σε σχέση με τα ASICs κοστίζουν περισσότερο ανά τσιπ για να εκτελέσουν μια συγκεκριμένη λειτουργία και έτσι δεν είναι καλοί υποψήφιοι για σχεδιασμούς που στοχεύουν σε μεγάλη παραγωγή.

Επειδή οι επαναπρογραμματιζόμενες αρχιτεκτονικές, όπως τα FPGA και τα PLD (programmable logic devices, προγραμματίσιμες συσκευές λογικής) έχουν αποκτήσει ιδιαίτερη δημοτικότητα και πολλαπλές χρήσεις, ακολουθεί μια λε-

πτομερή περιγραφή στο επόμενο κεφάλαιο.

4.9 Επαναδιαμορφώσιμη λογική PLD και FPGA

Όπως αναλύσαμε προηγουμένως, μια τυπική τεχνολογία υλοποίησης των ΕΣ, είναι τα ASIC. Όμως, σε μερικές περιπτώσεις (παραδείγματος χάριν, σε αυτές όπου ο αναμενόμενος όγκος προϊόντων δεν δικαιολογεί τα έξοδα πλήρους σχεδιασμού και τις δαπάνες NRE μιας τυπικής σχεδίασης ASIC), προτιμώνται άλλες τεχνολογίες, όπως τα FPGAs και τα PLDs. Το κόστος μερικών συστατικών NRE (όπως τα σύνολα μασκών) αυξάνεται με αντίστροφη αναλογία ως προς τα μεγέθη χαρακτηριστικών γνωρισμάτων VLSI, και αυτά γίνονται θεμελιώδεις περιοριστές στην οικονομική βιωσιμότητα ενός συγκεκριμένου σχεδιασμού σε έναν δεδομένο όγκο συστημάτων. Αυτό σημαίνει ότι σημείο ισορροπίας για τη χρήση της εναλλακτικής τεχνολογίας (όπως η επαναδιαμορφώσιμη λογική) μετατοπίζεται, οπότε όλο και περισσότερα κυκλώματα δε δικαιολογούν μια πλήρη επένδυση σε μια γραμμή παραγωγής VLSI μικρού μεγέθους τρανζίστορ. Οι σχεδιαστές αυτών των κυκλωμάτων επιλέγουν άλλες εναλλακτικές αρχιτεκτονικές, όπως FPGA και PLD.

Τα FPGAs και PLDs (συχνά ομαδοποιούνται με τον κοινό όρο επαναδιαμορφώσιμη λογική), επιτρέπουν στους σχεδιαστές να επιτύχουν υψηλή απόδοση παρόμοια με τα τυπικά τυποποιημένα δομοστοιχεία που υπάρχουν στην αγορά.

Η επαναδιαμορφώσιμη λογική ανταλλάσσει το υψηλότερο κόστος του υλικού (και συχνά την ταχύτητα, αλλά όχι πάντα) με τις χαμηλότερες δαπάνες σχεδίου και τη γρηγορότερη είσοδο στην αγορά. Δηλαδή, το κόστος αγοράς ενός ολοκληρωμένου κυκλώματος FPGA είναι αρκετά μεγάλο, ως προς ένα IC παρόμοιου αριθμού πυλών. Πχ. ένα ολοκληρωμένο κύκλωμα FPGA με περίπου 1.2 δισεκατομμύρια τρανζίστορ XC5VFX30T-1FFG665C, FPGA Virtex-5 κοστίζει το 2015 πάνω από 1000 ευρώ, ενώ ένα αρκετά πολύπλοκο IC που έχει 1.4 δισεκατομμύρια τρανζίστορ με τον επεξεργαστή CPU INTEL CORE I7-4790K 4.00GHZ LGA1150, κοστίζει 400 ευρώ.

Αν και το κόστος ανά τρανζίστορ στα FPGA είναι σχεδόν τριπλάσιο, εντούτοις το FPGA IC είναι πολύ πιο ευέλικτο και μπορεί να προσαρμοστεί σε κάθε πρόβλημα. Από αυτή την άποψη, είναι μια καλή αντιστοιχία με τα χαμηλού όγκου πωλήσεων προϊόντα. Πιο πρόσφατα, οι σχεδιαστές είχαν αρχίσει να χρησιμοποιούν τα FPGAs και PLDs για να υλοποιήσουν δυναμικά επαναδιαμορφώσιμους πυρήνες υπολογισμού, όπου αναλόγως τις ανάγκες του προβλήματος φορτώνεται κατ' απαίτηση η κατάλληλη διαμόρφωση, και έτσι το υπόστρωμα υλικού ανταποκρίνεται καλύτερα στο πρόβλημα. Για παράδειγμα, έστω ένας υπολογιστικός πυρήνας εκτίμησης κίνησης έχει ρυθμιστεί στο υλικό να επεξεργάζεται block 32x32 pixels. Θα μπορούσε να ζητηθεί μια καλύτερη ποιότητα

στο τελικό αποτέλεσμα με το να επαναδιαμορφωθεί για να επεξεργάζεται block 16x16. Η επαναδιαμόρφωση κατά το χρόνο εκτέλεσης μπορεί να χρησιμοποιηθεί για να αλλάξει τη συμπεριφορά του κυκλώματος, είτε ως προς την απόδοση, είτε ως προς την κατανάλωση. Βέβαια, η επαναδιαμόρφωση μπορεί να γίνει μόνο σε συγκεκριμένα FPGA και μόνο για συγκεκριμένα προβλήματα, στα οποία η εφαρμογή μπορεί να τμηματοποιηθεί, οπότε να επαναδιαμορφωθεί μόνο το κατάλληλο τμήμα και όχι όλο το ολοκληρωμένο κύκλωμα. Πάντως, η χρήση των FPGA βρίσκει απήχηση και στα συστήματα που εισέρχονται πρώτα στην καταναλωτική αγορά (first batch), τα οποία συνήθως έχουν προβλήματα και ίσως δεν ανταποκρίνονται πλήρως στις προσδοκίες των χρηστών. Με την επαναδιαμορφώσιμη λογική, μπορούν να γίνουν ενημερώσεις ή τροποποιήσεις με στόχο τη βελτίωση. Επίσης, μια εταιρία όσο περιμένει την ολοκλήρωση της παραγωγικής διαδικασίας για τα ASIC, μπορεί να χρησιμοποιεί FPGA τα οποία βασίζονται στην ίδια περιγραφή HDL, βοηθώντας την επίτευξη ενός γρήγορου ρυθμού διείσδυσης στην καταναλωτική αγορά (αν και με υψηλότερο κόστος τελικού προϊόντος αφού το FPGA IC έχει πολλαπλάσιο κόστος παραγωγής από το ASIC IC).

Για να εξεταστούν όλοι οι τύποι εφαρμογών, εξαιτίας των αυξήσεων στην πυκνότητα κυκλωμάτων μια πρόσφατη τεχνική είναι η ιδέα του συστήματος σε ένα προγραμματίσιμο τσιπ (System on a Programmable Chip, SoPC). Τα SoPCs συνδυάζουν την επαναδιαμορφώσιμη λογική με ένα σταθερό μη επαναδιαμορφώσιμο επεξεργαστή και ένα συνδυασμό σταθερών περιφερειακών και ελεγκτών μνήμης. Αυτό συνδυάζει το καλύτερο από τους δυο κόσμους: οι μονάδες του επεξεργαστή και τα μπλοκ λογικής των περιφερειακών μπορούν να εφαρμοστούν σαν μία μακροεντολή, με τον τελευταίας τεχνολογίας σχεδιασμό ASIC, και η επαναδιαμορφώσιμη λογική μπορεί να χρησιμοποιηθεί για την υλοποίηση είτε λογικής συγκόλλησης, είτε απαιτητικών πυρήνων επεξεργασίας. Η ιδέα πίσω από το SoPC είναι να παρέχει ένα συστατικό το οποίο ταιριάζει παντού (one-size-fits-all) και να υποστηρίζει αποτελεσματικά το λογισμικό μέσα από τον παραδοσιακό πυρήνα. Την ίδια στιγμή παρέχει ένα εύκολο μονοπάτι για την περιορισμένου βαθμού ευελιξία που υπάρχει σε επεξεργαστές, χωρίς να αυξάνει το κόστος του ολοκληρωτικού σχεδιασμού ASIC.

Η πλειοψηφία των σχεδιαστών FPGA/PLD προσφέρει αυτή την στιγμή SoPC τα οποία περιλαμβάνουν δημοφιλείς πυρήνες επεξεργαστών. Για παράδειγμα η Altera προσφέρει συστατικά βασισμένα στο ARM στην γραμμή Excalibur και η Xilinx προσφέρει chips βασισμένα σε τεχνολογία PPC στην οικογένεια Virtex-II-Pro (περισσότεροι από 4 επεξεργαστές ανά FPGA). Οι τελευταίες πλακέτες της Xilinx με την αρχιτεκτονική Zynq7000 φέρουν τετραπύρηνους επεξεργαστές ARM A9 MPCore, ενώ της Altera με την αρχιτεκτονική Cyclone V SOC, φέρουν διπύρηνους επεξεργαστές ARM Cortex A9.

4.9.1 Βασική Αρχιτεκτονική FPGA

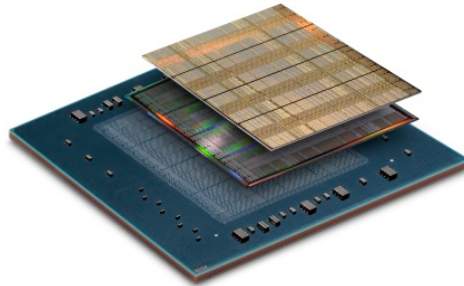
Σε γενικές γραμμές τα FPGA μπορούν να θεωρηθούν ως προγραμματιζόμενες συσκευές που παρέχουν μια μεγάλη πυκνότητα ολοκλήρωσης και ευελιξία. Η βασική αρχιτεκτονική των FPGA αποτελείται από τρία συστατικά, ανεξαρτήτως της εταιρίας που τα κατασκευάζει:

- τη λογική μονάδα (logic block),
- τη διασύνδεση (routing) και
- τη μονάδα E/E (I/O block).

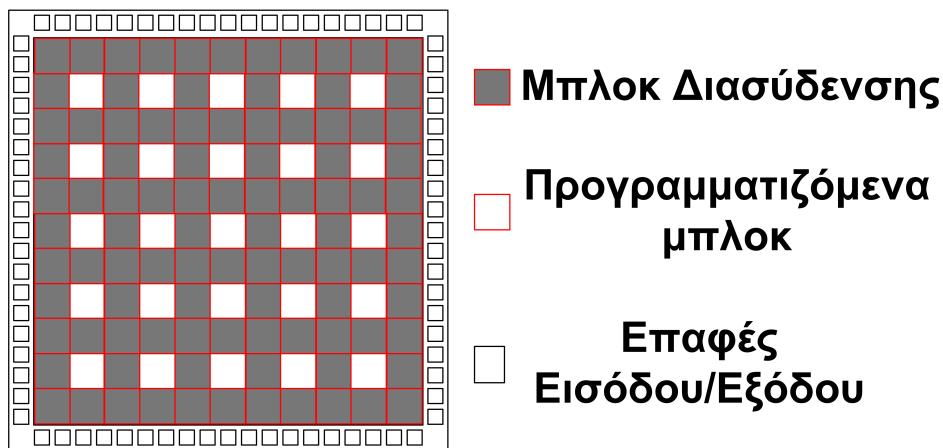
Τα FPGA αποτελούνται από ένα δισδιάστατο πλέγμα προγραμματιζόμενων λογικών μονάδων που μπορούν να διασυνδεθούν μεταξύ τους καθώς επίσης και με τις προγραμματιζόμενες μονάδες E/E μέσω κάποιου είδους προγραμματιζόμενης αρχιτεκτονικής διασύνδεσης. Τα τελευταία χρόνια γίνονται προσπάθειες ανάπτυξης τρισδιάστατων ολοκληρωμένων κυκλωμάτων στα FPGA, τα οποία αποτελούνται από την κατακόρυφη συσσώρευση πολλαπλών IC που συνδέονται μέσω κατακόρυφων διαμπερών οπών. Το πρώτο τρισδιάστατο FPGA, που διατέθηκε για πώληση στους καταναλωτές υψηλών απαιτήσεων, ήταν το Virtex-7 2000T (Εικόνα 4.14) που παρουσιάστηκε το 2011. Είχε 6.8 δισεκατομμύρια τρανζίστορ με συνέπεια να διαθέτει πάνω από 2 εκατομμύρια λογικά κελιά ή το ισοδύναμο 20 εκατομμυρίων λογικών πυλών. Για την κατασκευή χρησιμοποιήθηκαν 4 ολοκληρωμένα κυκλώματα FPGA Virtex, το ένα πάνω από το άλλο, με πάνω από 10.000 κατακόρυφες οπές διασύνδεσης. Εκτός από το Virtex-7 2000T, που είναι ομογενές τρισδιάστατο ολοκληρωμένο κύκλωμα, αφού και τα 4 επίπεδα φέρουν ίδια FPGA, η Xilinx παρουσίασε το 2012 μια ετερογενή αρχιτεκτονική που περιέχει 2 chip FPGA και ένα ολοκληρωμένο κύκλωμα πομποδέκτη 8 καναλιών και 28Gbps, με όνομα H580T. Όπως και τα τρισδιάστατα ολοκληρωμένα κυκλώματα ASIC, έτσι και τα FPGA έχουν πολύ υψηλό κόστος, εξαιτίας της δυσκολίας παραγωγής τους, αφού απαιτείται να ευθυγραμμιστούν χιλιάδες οπές διασύνδεσης σε όλα τα επίπεδα μεγέθους κάποιων χιλιάδων τρανζίστορ. Για αυτό και τα περισσότερα ολοκληρωμένα κυκλώματα είναι δισδιάστατα και αυτό δε φαίνεται να αλλάζει τα επόμενα χρόνια.

Όλα τα FPGA έχουν την ίδια τυπική αρχιτεκτονική, ανεξαρτήτως της εταιρίας που τα κατασκευάζει. Στο Σχήμα 4.15 φαίνεται μια γενική αρχιτεκτονική FPGA.

Για να γίνει η υλοποίηση ενός κυκλώματος σε ένα FPGA πρέπει να προγραμματιστεί η κάθε βασική λογική μονάδα του, ώστε να υλοποιεί ένα μικρό μέρος της λογικής, ενώ ταυτόχρονα κάθε μια από τις μονάδες E/E να ενεργεί είτε ως μονάδα εισόδου, είτε ως εξόδου, ανάλογα με τις απαιτήσεις του κυκλώματος. Η προγραμματιζόμενη διασύνδεση ρυθμίζεται για να υλοποιεί τις



Σχήμα 4.14: Ένα από τα πρώτα τρισδιάστατα FPGA, το Virtex-7 2000T, παρουσιάστηκε το 2011 από τη Xilinx και είχε 6.8 δισεκατομμύρια τρανζίστορ. Εικόνα από www.eetimes.com.



Σχήμα 4.15: Τυπική Αρχιτεκτονική FPGA.

απαραίτητες συνδέσεις τόσο ανάμεσα στις βασικές λογικές μονάδες, όσο και ανάμεσα στις βασικές λογικές μονάδες και τις μονάδες E/E. Η λειτουργική πολυπλοκότητα των βασικών λογικών μονάδων μπορεί να ποικίλει από μια απλή λογική συνάρτηση 2-εισόδων έως πολύπλοκες αριθμητικές λειτουργίες πολλών bit. Η επιλογή της κοκκιότητας (granularity) της βασικής λογικής μονάδας εξαρτάται από το πεδίο της κάθε εφαρμογής. Η χρησιμοποιούμενη τεχνολογία προγραμματισμού καθορίζει και τη μέθοδο αποθήκευσης της πληροφορίας προγραμματισμού. Η επιλογή της συγκεκριμένης τεχνολογίας έχει σημαντική επίδραση τόσο στην καταλαμβανόμενη επιφάνεια όσο και στην απόδοση του FPGA. Όπως αναφέρθηκε, τα FPGA είναι προγραμματιζόμενα και επαναδιαμορφούμενα πολλών χρήσεων, δηλαδή μπορούν να επαναδιαμορφώνονται συνεχώς. Θα πρέπει λοιπόν να αποθηκεύεται κάπου αυτό το πρόγραμμα, και για αυτό χρησιμοποιούνται διάφορες τεχνολογίες.

Οι κυριότερες και επικρατέστερες τεχνολογίες αποθήκευσης των δεδομένων προγραμματισμού, είναι η Στατική Μνήμη Τυχαίας Προσπέλασης (SRAM), η αντι-ασφάλεια (antifuse) και η μη-πτητική (non-volatile). Η επιλογή μιας εκ των προαναφερθέντων τεχνολογιών βασίζεται κυρίως στο υπολογιστικό περιβάλλον, στο οποίο πρόκειται να χρησιμοποιηθεί το FPGA, και στις δυνατότητες του εκάστοτε ολοκληρωμένου κυκλώματος FPGA. Κάποιες τεχνολογίες επιτρέπουν τη διατήρηση του προγραμματισμού ακόμη και αν σταματήσει η παροχή και επανέλθει το ρεύμα (όπως π.χ. η Flash), ενώ κάποιες άλλες είναι πτητικές και απαιτούν σε κάθε εκκίνηση λειτουργίας τον επαναπρογραμματισμό του FPGA.

Οι προγραμματιζόμενες συσκευές FPGA που διατίθενται στην αγορά διαφέρουν στον τύπο τεχνολογίας προγραμματισμού, στην αρχιτεκτονική της βασικής λογικής μονάδας καθώς και την δομή της αρχιτεκτονικής διασύνδεσης. Στις επόμενες παραγράφους θα περιγράψουμε τα τρία συστατικά της αρχιτεκτονικής FPGA και τις τεχνολογίες προγραμματισμού.

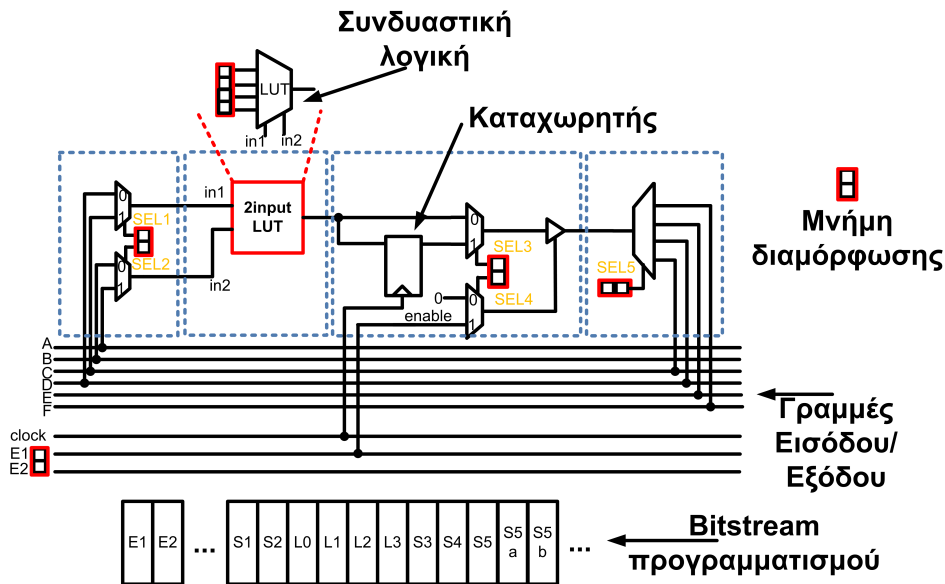
4.9.2 Αρχιτεκτονική της Λογικής Μονάδας

Η λογική μονάδα των FPGA είναι υπεύθυνη για την υλοποίηση σε επίπεδο πύλης της λειτουργίας που απαιτείται για κάθε εφαρμογή. Η λογική μονάδα χαρακτηρίζεται από την εσωτερική δομή και την κοκκοποίηση της. Η εσωτερική δομή καθορίζει τις διάφορες κατηγορίες λογικής τις οποίες μπορεί να υλοποιήσει, ενώ η κοκκοποίηση ορίζει το μέγεθος της συνάρτησης την οποία μπορεί να πραγματοποιήσει. Η λειτουργικότητα της λογικής μονάδας υλοποιείται ελέγχοντας τις συνδέσεις ορισμένων βασικών λογικών πυλών ή χρησιμοποιώντας πίνακες αναζήτησης (lookup-table - LUT). Η λειτουργικότητα αυτή έχει σημαντική επίπτωση καθώς αυξάνει το μέγεθος της λογικής μονάδας και είναι πιθανόν αυτό να μην αξιοποιείται πλήρως, με αποτέλεσμα να υπάρχει απώλεια τόσο στην επιφάνεια όσο και στην καταναλισκόμενη ενέργεια.

Η λογική μονάδα που χρησιμοποιείται στις FPGA μπορεί να γενικευτεί, με την κλωνοποίηση σε γειτονικές θέσεις, όπως φαίνεται στο Σχήμα 4.16.

Γενικά, το βασικό λογικό στοιχείο περιέχει κάποια μορφή μιας προγραμματιζόμενης συνδυαστικής λογικής, ένα flip-flop και κάποια γρήγορη λογική κρατούμενου για να μειώσει το χώρο και τις καθυστερήσεις. Στο γενικό μας λογικό στοιχείο, η έξοδος του στοιχείου είναι επιλέξιμη ανάμεσα στην έξοδο της συνδυαστικής λογικής ή την έξοδο του flip-flop. Επίσης, στο Σχήμα 4.16 φαίνεται κάποια μορφή προγραμματισμού ή μνήμη διαμόρφωσης για να ελέγξει την έξοδο του πολυπλέκτη, και συνεπώς την έξοδο του προγραμματίσιμου κελιού.

Όσον αφορά το τμήμα συνδυαστικής λογικής της λογικής μονάδας, έχουν χρησιμοποιηθεί διαφορετικές μέθοδοι υλοποίησης, όπως οι πίνακες αναζήτησης, οι πολυπλέκτες και οι συνδυασμοί λογικών πυλών. Έτσι, ο πιο κοινός τρόπος υλοποίησης της συνδυαστικής λογικής είναι ο πίνακας αναζήτησης (LUT).

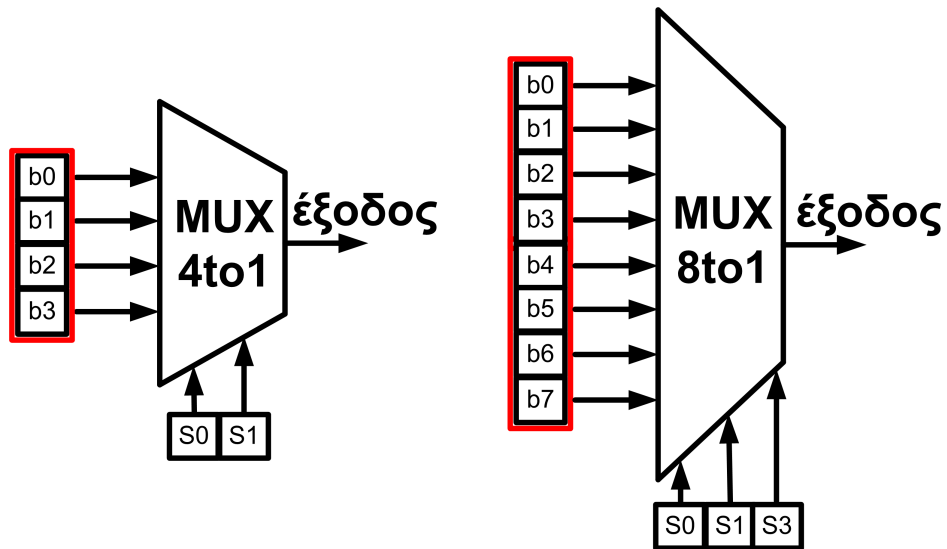


Σχήμα 4.16: Ένα FPGA κελί αποτελείται από στοιχεία μνήμης, πολυπλέκτες και αποκωδικοποιητές.

Στο Σχήμα 4.17 φαίνεται πως υλοποιείται ένας πίνακας αναζήτησης των 2 ή των 3 εισόδων.

Όπως φαίνεται, υπάρχει μια σειρά προγραμματιζόμενων κελιών μνήμης με έναν πολυπλέκτη, ο οποίος επιλέγει την έξοδο ενός συγκεκριμένου κελιού μνήμης. Ο πίνακας αναζήτησης λειτουργεί ως μνήμη με n γραμμές διευθύνσεων και 2^n θέσεις μνήμης. Για να υλοποιήσουμε μια συγκεκριμένη λογική συνάρτηση με την μνήμη, φορτώνεται ο πίνακας αληθείας της συνάρτησης στην μνήμη. Για παράδειγμα, σε ένα πίνακα αναζήτησης LUT αν θέσουμε $b_7b_6b_5b_4b_3b_2b_1b_0 = 10000000$, τότε υλοποιείται μια συνάρτηση AND 3 εισόδων. Υπενθυμίζουμε ότι ο πίνακας αναζήτησης παράγει το λογικό 1 όταν όλες οι γραμμές διευθύνσεων (με ονόματα s_0, s_1, s_2) έχουν λογικό 1 και γι' αυτό το κελί μνήμης της θέσης 7 ($s_0, s_1, s_2 = 1, 1, 1$) είναι το μόνο κελί που αποθηκεύει την τιμή 1. Οι λογικές μονάδες των οικογενειών XC3000 και XC4000 της εταιρείας Xilinx βασίζονται στους πίνακες αναζήτησης. Ένας πολυπλέκτης 3 εισόδων, μπορεί να υλοποιήσει οποιαδήποτε συνάρτηση 3 εισόδων με την τοποθέτηση της κατάλληλης σειράς bit στις εισόδους. Αν αντιγραφεί το ίδιο κύκλωμα πολλές φορές, τότε θα υπάρχει η δυνατότητα εξόδου παραπάνω από ένα bit, ώστε να δημιουργούνται στην έξοδο ολόκληρες λέξεις.

Παρόλο που οι περισσότερες προγραμματιζόμενες συσκευές FPGA χρησιμοποιούν πίνακες αναζήτησης για συνδυαστική λογική, υπάρχουν εναλλακτικές αρχιτεκτονικές βασισμένες σε πολυπλέκτες. Η βασική ιδέα του πολυπλέκτη



Σχήμα 4.17: Πίνακας αναζήτησης δύο και τριών εισόδων που βρίσκεται σε ένα κελί FPGA.

είναι ότι για να υλοποιήσει διαφορετικές λογικές συναρτήσεις συνδέει κάθε μια από τις εισόδους της στις κατάλληλες γραμμές. Για παράδειγμα, θεωρούμε ένα πολυπλέκτη 2 με γραμμή επιλογής εισόδου s , εισόδους a και b και έξοδο $f = sa + s'b$. Θέτοντας το σήμα b σε λογικό 0, ο πολυπλέκτης μπορεί να υλοποιήσει την συνάρτηση AND $f = sa$. Επίσης, θέτοντας το σήμα a σε λογικό 1, δίνει έξοδο τη συνάρτηση OR $f = s + b$. Έτσι, συνδέοντας ένα αριθμό από πολυπλέκτες και από βασικές λογικές πύλες, μπορεί να κατασκευαστεί ένα λογικό στοιχείο το οποίο μπορεί να υλοποιεί ένα μεγάλο αριθμό λογικών συναρτήσεων. Ένα παράδειγμα οικογένειας FPGA που στηρίζεται σε πολυπλέκτες για την υλοποίηση της λογικής μονάδας είναι αυτό της ACT3 της εταιρείας Actel.

Από την άλλη μεριά, η πολυπλοκότητα της λογικής μονάδας ποικίλει από μια πολύ μικρή μονάδα η οποία μπορεί να υλοποιήσει μια συνάρτηση 3-εισόδων, έως μια δομή η οποία είναι μια 4-bit αριθμητική λογική μονάδα (δηλαδή, πάνω από 8 εισόδους). Τόσο το μέγεθος, όσο και η πολυπλοκότητα των βασικών μονάδων, αναφέρεται συχνά ως κοκκοποίηση του τελευταίου. Με άλλα λόγια, το κριτήριο της κοκκοποίησης αναπαριστά τη μικρότερη μονάδα από την οποία αποτελείται η προγραμματιζόμενη διάταξη. Η επιλογή της κοκκοποίησης της λογικής μονάδας εξαρτάται από το πεδίο εφαρμογής στο οποίο θα χρησιμοποιηθεί το FPGA, και έχει σημαντική επίπτωση στο χρόνο που χρειάζεται για την επαναδιαμόρφωση, και είναι ιδιαίτερα κρίσιμο ειδικά για συστήματα που η διαμόρφωση γίνεται κατά την διάρκεια λειτουργίας του FPGA.

Όλες οι επαναδιαμορφούμενες πλατφόρμες διακρίνονται ανάλογα με την

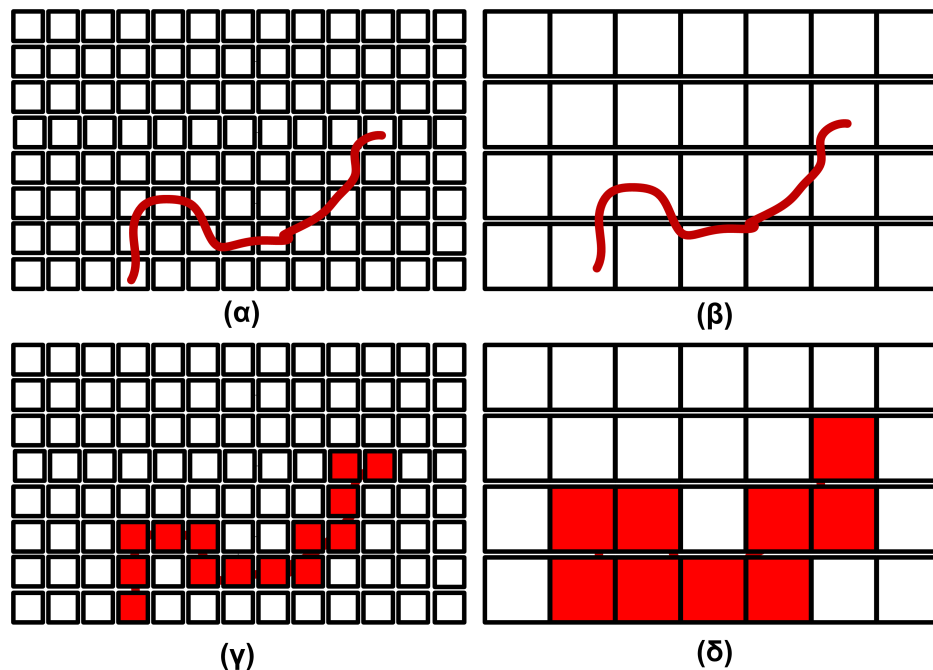
κοκκοποίηση τους σε λεπτόκοκκες και χοντρόκοκκες. Στις λεπτόκοκκες αρχιτεκτονικές, η βασική λογική μονάδα αποτελείται από ένα συνδυαστικό δίκτυο και μερικούς καταχωρητές. Μια διάταξη από τέτοιες λογικές μονάδες μπορεί να υλοποιήσει πολύ μικρούς υπολογισμούς και επομένως απαιτεί περισσότερα bit δεδομένων κατά την διάρκεια του προγραμματισμού. Η κοκκοποίηση αυτού του τύπου είναι περισσότερο αποδοτική για λειτουργίες ελέγχου, ενώ οι μονάδες που χαρακτηρίζονται ως χοντρόκοκκες και οι οποίες έχουν μεγαλύτερες αριθμητικές ικανότητες, είναι περισσότερο χρήσιμες για λειτουργίες μεταφοράς δεδομένων μέσω των διόδων δεδομένων (datapath).

Το Σχήμα 4.18 παρουσιάζει τη σημασία της κοκκοποίησης. Συγκεκριμένα στα σχήματα 4.18(α) και 4.18(β) εμφανίζονται δυο FPGA με μικρή και μεγάλη κοκκοποίηση, πάνω στα οποία θα απεικονιστεί μια λειτουργία, που αφαιρετικά εμφανίζεται με μια κόκκινη γραμμή. Το FPGA με τη μικρή κοκκοποίηση θα μπορέσει να αντιστοιχίσει καλύτερα αυτή τη λειτουργία σε αντίθεση με το (β) αφού στην πρώτη περίπτωση (Σχήμα 4.18(γ)) θα απαιτηθούν 14/104 κελιά (ποσοστό χρήσης 14%) και στη δεύτερη περίπτωση (Σχήμα 4.18(δ)) θα απαιτηθούν 9/28 κελιά (ποσοστό χρήσης 32%). Η χρήση παραπάνω κελιών, εκτός του ότι μειώνει τα διαθέσιμα επαναπρογραμματιζόμενα κελιά για άλλες λειτουργίες, έχει ως αποτέλεσμα και την αύξηση της κατανάλωσης ενέργειας. Στο παράδειγμα αυτό αρκετά αφαιρετικά παρουσιάστηκε η έννοια της κοκκοποίησης. Στην πραγματικότητα, το κάθε κελί έχει πεπερασμένα στοιχεία μνήμης και υλοποίησης λειτουργικών πράξεων, οπότε ένα κελί με διπλάσια επιφάνεια μάλλον θα έχει και διπλάσια χωρητικότητα μνήμης και υλοποίησης λογικών συναρτήσεων με παραπάνω παραμέτρους. Συνεπώς, θα χρησιμοποιούνταν αρκετά λιγότερα κελιά, οπότε στην πραγματικότητα η διαφορά δε θα είναι τόσο μεγάλη. Τέλος, όσο μικρότερη είναι η κοκκοποίηση τόσο πιο πολύπλοκη είναι η διασύνδεση των κελιών.

4.9.3 Αρχιτεκτονική Διασύνδεσης

Η αρχιτεκτονική διασύνδεσης πραγματοποιείται χρησιμοποιώντας διακόπτες (switches), οι οποίοι προγραμματίζονται για να υλοποιούν τις συνδέσεις (δηλαδή να συνδέουν ή να αποσυνδέουν γραμμές επικοινωνίας). Η μέθοδος με την οποία δημιουργούνται οι συνδέσεις ανάμεσα στις βασικές λογικές μονάδες έχει σημαντική επίπτωση στα χαρακτηριστικά της αρχιτεκτονικής του FPGA. Με βάση την τοποθέτηση των βασικών λογικών μονάδων και των πόρων διασύνδεσης, οι διατάξεις προγραμματιζόμενων πυλών είναι δυνατόν να διακριθούν στις ακόλουθες κατηγορίες:

- Αρχιτεκτονική νησίδας,
- Αρχιτεκτονική βασιζόμενη σε γραμμές και



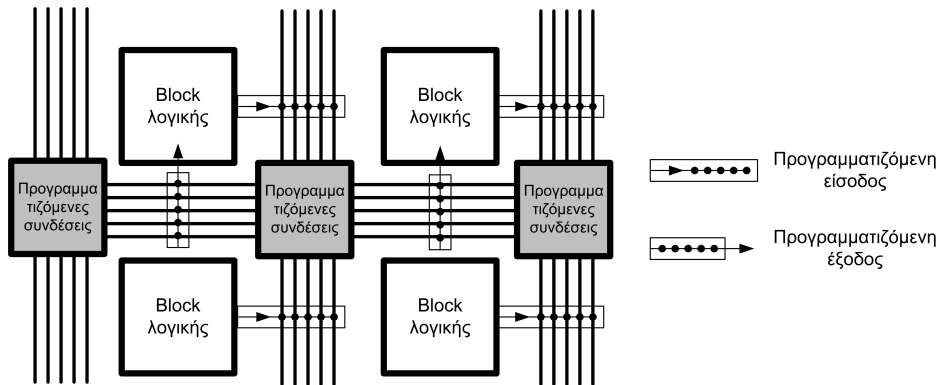
Σχήμα 4.18: Όσο μεγαλώνει η κοκκοποίηση σε ένα FPGA, τόσο δεν αξιοποιείται πλήρως το ολοκληρωμένο κύκλωμα, με συνέπεια να υπάρχει απώλεια στην επιφάνεια και στην καταναλισκόμενη ενέργεια. Στο παράδειγμα μας, η ίδια λειτουργία (που απεικονίζεται αφαιρετικά με μια κόκκινη γραμμή), σε δυο FPGA με μικρή και μεγάλη κοκκιότητα, χρησιμοποιεί 14% της επιφάνειας (γ) έναντι 32% (δ) στο IC μεγάλης κοκκιότητας.

- Ιεραρχική αρχιτεκτονική.

Παρακάτω παρουσιάζουμε τα βασικά χαρακτηριστικά της κάθε μιας από τις προαναφερθείσες αρχιτεκτονικές.

Αρχιτεκτονική Νησίδας

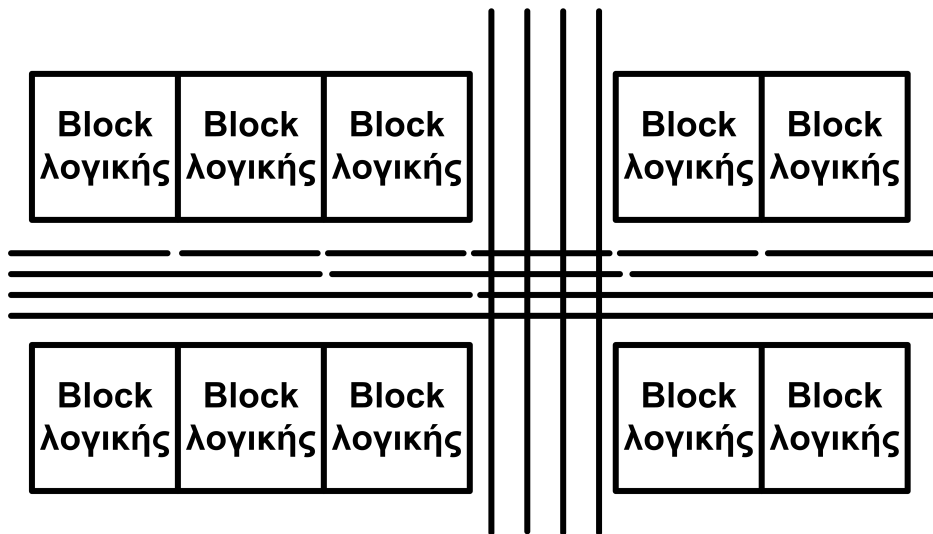
Η αρχιτεκτονική νησίδας αποτελείται από μια διάταξη προγραμματιζόμενων λογικών μονάδων με κατακόρυφα και οριζόντια προγραμματιζόμενα κανάλια διασύνδεσης. Η βασική ιδέα της συγκεκριμένης αρχιτεκτονικής παρουσιάζεται στο Σχήμα 4.19. Ο αριθμός των τμημάτων των καλωδίων που υπάρχουν στο εσωτερικό του καναλιού προσδιορίζει τους διαθέσιμους πόρους για τη διασύνδεση. Οι ακροδέκτες της λογικής μονάδας έχουν πρόσβαση στο κανάλι διασύνδεσης διαμέσου του κουτιού διασύνδεσης (connection box). Οι οικογένειες XC4000 και XC3000 της Xilinx είναι παραδείγματα του συγκεκριμένου τύπου αρχιτεκτονικής.



Σχήμα 4.19: Αρχιτεκτονική FPGA τύπου νησίδας.

Αρχιτεκτονική βασισμένη σε Γραμμές

Όπως υποδηλώνει το όνομα, στη συγκεκριμένη αρχιτεκτονική οι λογικές μονάδες είναι τοποθετημένες σε γραμμές και διαχωρίζονται μεταξύ τους με οριζόντια κανάλια διασύνδεσης. Η αρχιτεκτονική αυτή παρουσιάζεται στο Σχήμα 4.20.



Σχήμα 4.20: Αρχιτεκτονική FPGA βασισμένη σε γραμμές.

Τα μονοπάτια διασύνδεσης στο εσωτερικό του καναλιού χωρίζονται σε ένα ή περισσότερα τμήματα, το μήκος των οποίων μπορεί να κυμαίνεται από το πλάτος ενός ζεύγους βασικών λογικών στοιχείων, έως και ολόκληρο το μήκος του καναλιού. Τα τμήματα αυτά ενδέχεται να ενώνονται στα άκρα τους χρησιμοποιώντας προγραμματιζόμενους διακόπτες για τη δημιουργία γραμμών με-

γαλύτερου μήκους. Πέρα από αυτά τα κανάλια, υπάρχουν και εκείνα που βρίσκονται τοποθετημένα κάθετα ανάμεσα στις λογικές μονάδες, τα οποία παρέχουν τις απαραίτητες συνδέσεις ανάμεσα στα οριζόντια κανάλια και τα κατακόρυφα τμήματα της διασύνδεσης. Το μήκος των συγκεκριμένων τμημάτων στα κανάλια προσδιορίζεται από τον συμβιβασμό που γίνεται ανάμεσα στον αριθμό των μονοπατιών, την αντίσταση που εμφανίζουν οι διακόπτες διασύνδεσης και τη χωρητικότητα των συγκεκριμένων τμημάτων. Ένα παράδειγμα οικογένειας FPGA που στηρίζεται στην αρχιτεκτονική βασιζόμενη σε γραμμές είναι αυτό της ACT3 που παράγει η εταιρεία Actel.

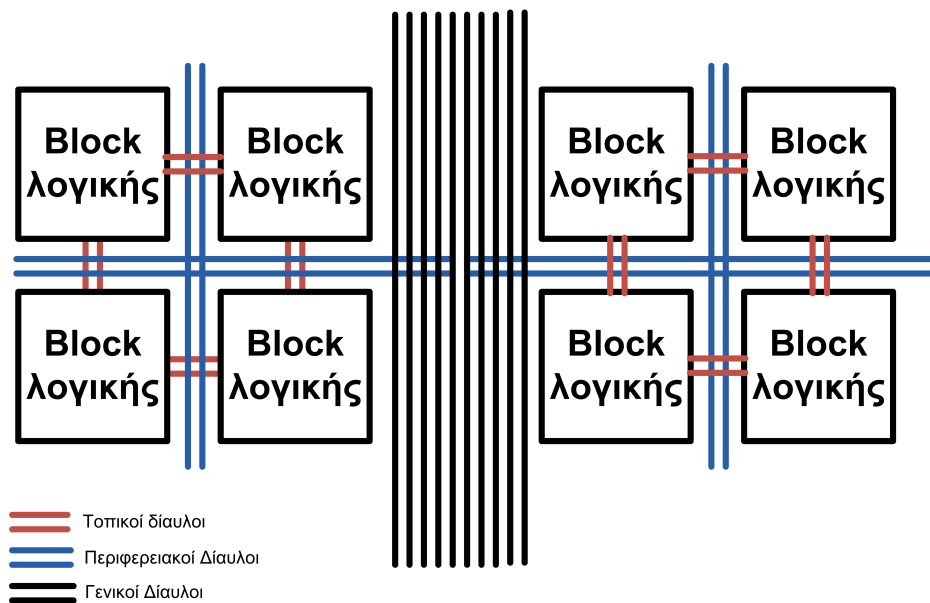
Ιεραρχική Αρχιτεκτονική

Οι πλειονότητα των λογικών σχεδιασμών παρουσιάζουν κάποια τοπικότητα στις συνδέσεις, η οποία υποδηλώνει μια ιεραρχία στην τοποθέτηση και διασύνδεση των συνδέσεων ανάμεσα στις λογικές μονάδες. Η ιεραρχική αρχιτεκτονική στα FPGA (Εικόνα 4.21) προσπαθεί να εκμεταλλευτεί το συγκεκριμένο χαρακτηριστικό, ώστε οι διατάξεις των προγραμματιζόμενων πυλών να εμφανίζουν μικρότερες καθυστερήσεις διασύνδεσης και μια πιο προβλέψιμη συμπεριφορά χρονισμού. Η συγκεκριμένη αρχιτεκτονική δημιουργείται συνδέοντας λογικές μονάδες σε συστοιχίες, οι οποίες στη συνέχεια συνδέονται αναδρομικά για τη δημιουργία μιας ιεραρχικής δομής. Ο αριθμός των διακοπών διασύνδεσης από τους οποίους πρέπει να περάσει το σήμα καθορίζει τη ταχύτητα του δικτύου. Η ιεραρχική αρχιτεκτονική μειώνει το πλήθος των διακοπών που απαιτούνται για τη δημιουργία μεγάλων μονοπατιών διασύνδεσης, με αποτέλεσμα η διάταξη να λειτουργεί σε μεγαλύτερη ταχύτητα.

4.9.4 Αρχιτεκτονική της Μονάδας E/E

Σε σχέση με τις αρχιτεκτονικές λογικής μονάδας και διασύνδεσης, η αρχιτεκτονική E/E όπως φαίνεται στο Σχήμα 4.22 είναι παρόμοια στις περισσότερες οικογένειες FPGA. Τα τμήματα E/E έχουν τριστάθεις απομονωτές (tri-state buffers) για τις εξόδους και απομονωτές εισόδων για τις εισόδους. Το σήμα επίτρεψης τριστάθειας, το σήμα εξόδου και τα σήματα εισόδου μπορούν να καταχωριστούν ξεχωριστά μέσα στο τμήμα E/E ή να μην καταχωριστούν με βάση πως είναι προγραμματισμένο το τμήμα E/E.

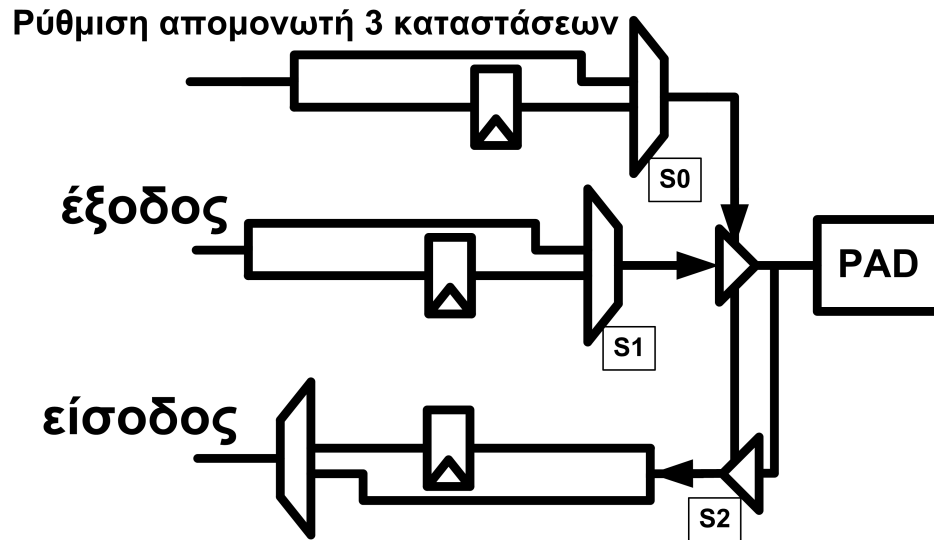
Όλοι οι ακροδέκτες εισόδου/εξόδου του ολοκληρωμένου κυκλώματος μπορούν να προγραμματιστούν να είναι είτε είσοδοι, είτε έξοδοι, είτε να μη συνδέονται πουθενά. Κατά τον προγραμματισμό του FPGA, ο σχεδιαστής δηλώνει με το αρχείο περιορισμών (constraints file) ποιοι ακροδέκτες χρησιμοποιούνται και σε ποιες θύρες αντιστοιχούν στο δομοστοιχείο της σχεδιαστικής κορυφής (top level design).



Σχήμα 4.21: Ιεραρχική δομή FPGA, στην οποία φαίνονται τα τοπικά και τα γενικά κανάλια διασύνδεσης.

Σε σύγχρονες αρχιτεκτονικές FPGAs, μπορεί να βρεθούν μια ευρεία ποικιλία πρόσθετων χαρακτηριστικών που βελτιώνουν αλλά και περιπλέκουν σημαντικά αυτή τη βασική δομή E/E. Για παράδειγμα, τα τμήματα E/E της αρχιτεκτονικής Xilinx Virtex-4 παρέχουν τα παρακάτω χαρακτηριστικά:

- περισσότερες από 50 παραλλαγές προτύπων σηματοδότησης E/E μερικές από τις οποίες περιλαμβάνουν τα χαρακτηριστικά όπως:
 - ψηφιακά ελεγχόμενη αντίσταση για να εξαλείψει την ανάγκη για τον τερματισμό αντιστάσεων, ώστε να αντιμετωπίζει τις επιδράσεις των γραμμών μετάδοσης και
 - διαφορική σηματοδότηση για να βελτιώσει την ακεραιότητα των σημάτων
- διπλασιασμό του ρυθμού δεδομένων που καταχωρεί σε πολλαπλούς κόμβους για παρουσίαση και λήψη δεδομένων σε και από την εσωτερική λογική και
- προγραμματιζόμενες καθυστερήσεις εισόδων.



Σχήμα 4.22: Τυπική αρχιτεκτονική εισόδου-εξόδου των FPGA.

4.9.5 Τεχνολογίες Προγραμματισμού

Τόσο το μπλοκ λογικής, όσο και οι πόροι διασύνδεσης από τους οποίους αποτελείται το FPGA πρέπει να προγραμματιστούν, ώστε να υλοποιούν την απαιτούμενη συνάρτηση. Τα περιεχόμενα της λογικής μονάδας μπορούν να προγραμματιστούν για την πραγματοποίηση της λογικής συνάρτησης αυτού, ενώ οι διακόπτες διασύνδεσης προγραμματίζονται για τον έλεγχο των συνδέσεων μεταξύ των λογικών μονάδων. Υπάρχει ένα πλήθος διαφορετικών μεθόδων για την αποθήκευση της πληροφορίας προγραμματισμού, οι οποίες εκτείνονται από την ευμετάβλητη μέθοδο της SRAM, έως την μη αναστρέψιμη τεχνολογία αντι-ασφάλειας (antifuse). Το μεγαλύτερο τμήμα της επιφάνειας ενός FPGA καταλαμβάνεται κυρίως από την περιοχή των επαναδιαμορφώσιμων συστατικών. Επομένως, η επιλογή της μεθόδου προγραμματισμού επηρεάζει σημαντικά την επιφάνεια του FPGA. Ένας επιπλέον παράγοντας, ο οποίος πρέπει να λαμβάνεται υπόψη είναι το πόσες φορές απαιτείται η επαναδιαμόρφωση του FPGA. Έτσι για τα FPGA στα οποία η τεχνολογία προγραμματισμού τους βασίζεται στην αντι-ασφάλεια μπορούν να προγραμματιστούν μόνο μία φορά, ενώ αντίθετα όσα χρησιμοποιούν SRAM δεν εμφανίζουν κάποιο παρόμοιο περιορισμό, και μπορούν να προγραμματίζονται πολύ συχνά.

SRAM

Στη συγκεκριμένη μέθοδο προγραμματισμού η διαμόρφωση αποθηκεύεται σε κελιά μνήμης SRAM. Στην περίπτωση που το δίκτυο διασύνδεσης υλοποιεί-

ται με τη χρήση των τρανζίστορ διέλευσης (pass-transistors), τότε τα κελιά της SRAM ελέγχουν εάν το τρανζίστορ άγει ή όχι. Επιπλέον, όταν οι λογικές μονάδες αποτελούνται από έναν πίνακα αναζήτησης τότε η λογική αποθηκεύεται στα κελιά SRAM. Η συγκεκριμένη μέθοδος προγραμματισμού παρουσιάζει το μειονέκτημα της προσωρινής αποθήκευσης της πληροφορίας. Αποτέλεσμα είναι πως η διαμόρφωση πρέπει να φορτώνεται στο FPGA κάθε φορά που το τελευταίο ανατροφοδοτείται. Όσα συστήματα βασίζονται στη τεχνολογία αυτή συχνά χρησιμοποιούν μια μόνιμη εξωτερική συσκευή αποθήκευσης των δεδομένων (π.χ. μνήμη FLASH ή EEPROM). Η χρήση της μεθόδου SRAM απαιτεί τουλάχιστον 5 τρανζίστορ ανά κελί. Λόγω του σχετικά μεγάλου μεγέθους των κελιών μνήμης, μεγάλο μέρος της επιφάνειας του FPGA καταλαμβάνεται για την αποθήκευση της διαμόρφωσης. Το βασικό πλεονέκτημα που έχει η χρήση της μεθόδου προγραμματισμού που στηρίζεται στη τεχνολογία της SRAM είναι η δυνατότητα επαναχρησιμοποίησης της συσκευής για την υλοποίηση διαφορετικών εφαρμογών, απλά φορτώνοντας κάθε φορά την κατάλληλη διαμόρφωση. Το χαρακτηριστικό αυτό έκανε τη συγκεκριμένη τεχνολογία αποθήκευσης πληροφορίας στα FPGA αρκετά δημοφιλή για την υλοποίηση επαναδιαμορφώσιμων συστημάτων, τα οποία στοχεύουν στην επίτευξη κερδών στην απόδοση προσαρμόζοντας την υλοποίηση των συναρτήσεων στις ανάγκες της συγκεκριμένης εφαρμογής.

Αντι-ασφάλεια

Στη μέθοδο προγραμματισμού που στηρίζεται στη τεχνολογία SRAM, η πληροφορία αποθηκεύεται ελέγχοντας την κατάσταση των κελιών μνήμης. Αντίθετα η τεχνολογία της αντι-ασφάλειας (antifuse) χρησιμοποιεί προγραμματιζόμενες συνδέσεις, των οποίων η αντίσταση μεταβάλλεται με την εφαρμογή υψηλής τάσης. Στη μη-προγραμματισμένη κατάσταση η αντίσταση των συνδέσεων είναι της τάξης των μερικών GΩ, επομένως μπορεί να θεωρηθεί ως ανοιχτό κύκλωμα. Με την εφαρμογή μιας σχετικά μεγάλης τιμής τάσης, συμβαίνει ένα φυσικό φαινόμενο που ονομάζεται τήξη. Το αποτέλεσμα αυτού είναι η δημιουργία μιας αντίστασης της τάξης των μερικών Ohm (Ω) κατά μήκος της συσκευής, υλοποιώντας με τον τρόπο αυτό μια σύνδεση. Το πλεονέκτημα της συγκεκριμένης μεθόδου είναι πως το μέγεθος του προγραμματιζόμενου στοιχείου είναι της τάξης του μεγέθους ενός περάσματος, με αποτέλεσμα να πετυχαίνεται σημαντική μείωση στην επιφάνεια συγκρινόμενη με τα FPGA που στηρίζονται σε SRAM. Αντίστοιχα, η αντίσταση μεταξύ των στοιχείων είναι της τάξης των μερικών Ohm (Ω) και είναι πολύ μικρότερη από την αντίσταση των τρανζίστορ μετάβασης τα οποία χρησιμοποιούνται ως διακόπτες διασύνδεσης στη μέθοδο SRAM. Η συγκεκριμένη τεχνική προγραμματισμού δεν έχει προσωρινή μορφή, και επομένως δεν απαιτεί την ύπαρξη εξωτερικής συσκευής για την αποθήκευση

της πληροφορίας διαμόρφωσης κατά το διάστημα διακοπής της τροφοδοσίας. Το μειονέκτημα της τεχνολογίας αυτής, σε αντίθεση με ότι συμβαίνει στην τεχνολογία SRAM, είναι πως πιθανά λάθη που συμβαίνουν κατά το στάδιο του σχεδιασμού δεν μπορούν να διορθωθούν αφού η διαδικασία διαμόρφωσης είναι μη αναστρέψιμη.

EPROM, EEPROM και FLASH

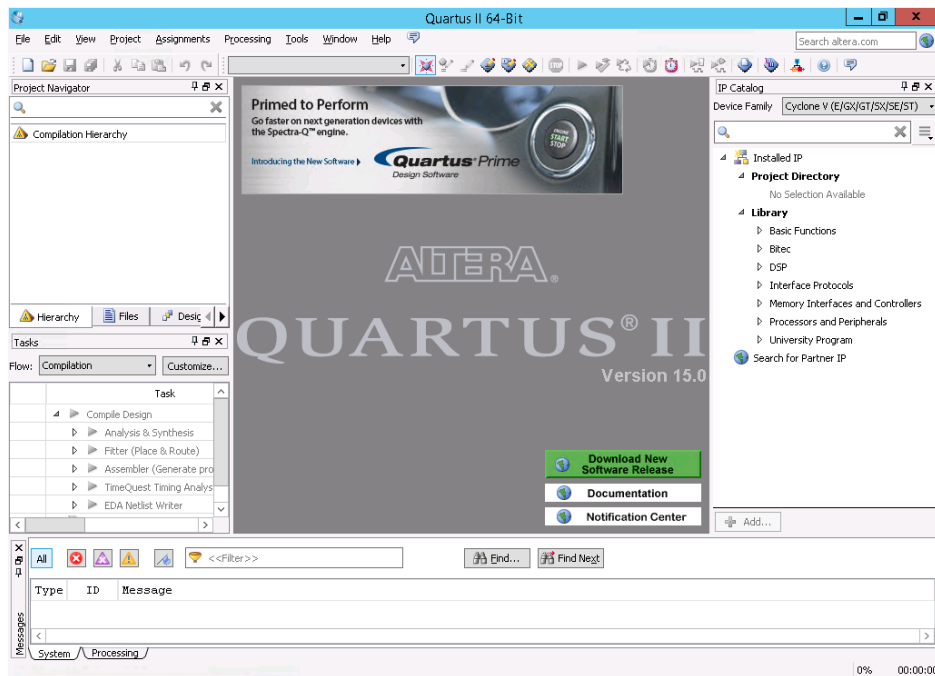
Η κατηγορία των τεχνολογιών προγραμματισμού που εμφανίζουν μόνιμο χαρακτήρα χρησιμοποιεί τις ίδιες τεχνικές με εκείνες των EPROM, EEPROM και Flash μνημών. Η συγκεκριμένη μέθοδος βασίζεται σε ένα ειδικό τρανζίστορ με δύο πύλες, εκ των οποίων η μία είναι πλωτή πύλη, ενώ η άλλη πύλη επιλογής. Όταν κατά μήκος του τρανζίστορ διέλθει κάποιο ισχυρό ρεύμα, το φορτίο παγιδεύεται στην πλωτή πύλη αυξάνοντας τη τάση κατωφλίου του τρανζίστορ. Στη κανονική λειτουργία τα προγραμματισμένα τρανζίστορ μπορούν να θεωρηθούν ως ανοιχτά κυκλώματα, ενώ τα υπόλοιπα να ελεγχθούν χρησιμοποιώντας τις πύλες επιλογής. Η φόρτιση στην πλωτή πύλη θα παραμείνει ακόμη και κατά τη διακοπή της τροφοδοσίας. Η πλωτή φόρτιση μπορεί να απομακρυνθεί εκθέτοντας την πύλη σε υπεριώδες φως για την περίπτωση των EPROM, και σε ηλεκτρικό ρεύμα στις περιπτώσεις των EEPROM και Flash. Οι συγκεκριμένες τεχνικές καλύπτουν το κενό που υπάρχει ανάμεσα στις τεχνολογίες προγραμματισμού SRAM και αντι-ασφάλειας, παρέχοντας την ευστάθεια που εμφανίζει η αντι-ασφάλεια με τον επαναπρογραμματισμό που παρουσιάζει η SRAM. Η αντίσταση των διακοπτών διασύνδεσης είναι μεγαλύτερη από εκείνη της αντι-ασφάλειας, ενώ ο προγραμματισμός είναι περισσότερο πολύπλοκος και χρονοβόρος σε σύγκριση με αυτόν που απαιτείται για την τεχνολογία SRAM.

4.9.6 Γενική Σχεδιαστική Ροή

Η πολυπλοκότητα των FPGAs κάνει απαραίτητη την χρήση ειδικού λογισμικού για την σχεδίαση και υλοποίηση αλγορίθμων ή εφαρμογών απευθείας στο υλικό. Συνεπώς, κάθε εταιρία παρέχει το δικό της λογισμικό προγραμματισμού που μπορεί να εκμεταλλευτεί βέλτιστα τις δυνατότητες των συσκευών της. Οι Εικόνες 4.23 και 4.24 παρουσιάζουν τα εργαλεία για τις 2 πιο δημοφιλείς εταιρίες, Altera και Xilinx. Για αυτό το λόγο χρησιμοποιούμε μια σχεδιαστική ροή. Αυτή δεν είναι τίποτα άλλα από μια σειρά διαδικασιών με την χρήση εργαλείων CAD (Computer Aided Design), ώστε τελικά να έχουμε την ολοκληρωμένη σχεδίαση. Τα σημερινά εργαλεία έχουν αυτοματοποιήσει τη σχεδιαστική ροή. Ο σχεδιαστής απλώς εισάγει τις κατάλληλες πληροφορίες και το περιβάλλον ανάπτυξης εκτελεί ένα-προς-ένα όλα τα εργαλεία με την κατάλληλη σειρά, αναφέροντας τα αποτελέσματα, τις μετρήσεις, τα λάθη, τα προβλήματα,

ΚΕΦΑΛΑΙΟ 4. ΥΛΟΠΟΙΗΣΗ ΕΣ

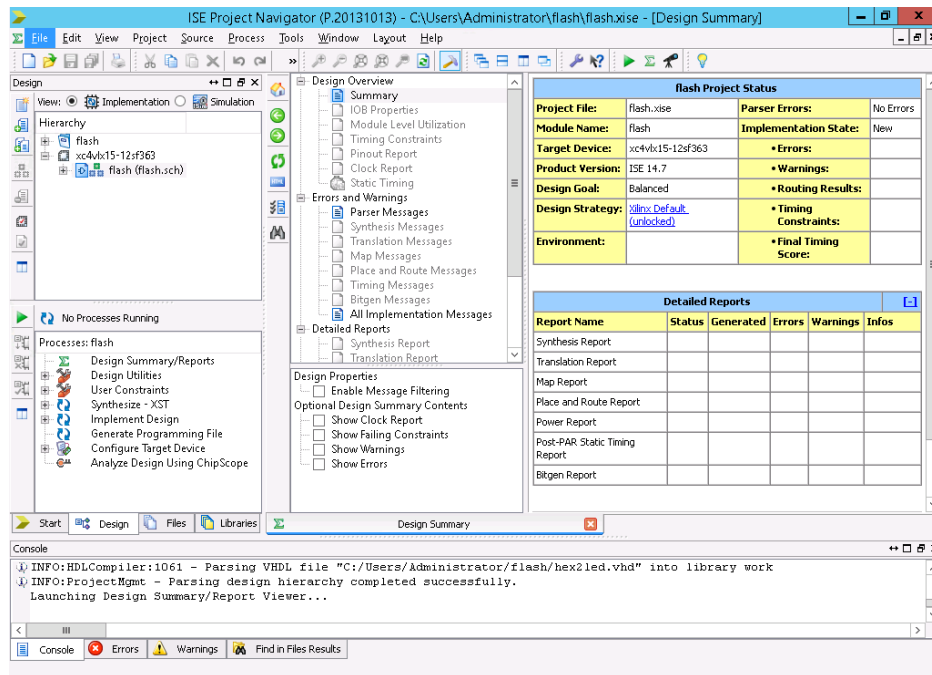
τις προειδοποιήσεις και οποιαδήποτε άλλη χρήσιμη πληροφορία.



Σχήμα 4.23: Η εταιρία Altera παρέχει το Quartus II ως ένα ολοκληρωμένο εργαλείο ανάπτυξης για τα δικά της FPGA.

Τα τυπικά βήματα της σχεδιαστικής ροής που βρίσκουμε σε όλα τα περιβάλλοντα που στοχεύουν τα FPGA είναι τα εξής:

- Εισαγωγή σχεδιασμού (design entry): δημιουργία των αρχείων σχεδιασμού χρησιμοποιώντας ένα σχηματικό συντάκτη (schematic editor) ή γλώσσα περιγραφής υλικού (hardware description language, HDL), όπως οι γλώσσες Verilog και VHDL.
- Σύνθεση σχεδιασμού (design synthesis): μια διαδικασία που μετατρέπει από ένα υψηλό επίπεδο αφαίρεσης λογικής (όπως Verilog ή VHDL) σε ένα χαμηλότερο επίπεδο αφαίρεσης λογικής χρησιμοποιώντας μια βιβλιοθήκη. Υπάρχουν πολλά εργαλεία για την σύνθεση όπως Synplify, Synopsys, κλπ
- Απεικόνιση (mapping): μια διαδικασία που απεικονίζει σε κάθε λογικό στοιχείο ένα συγκεκριμένο φυσικό στοιχείο που υλοποιεί πραγματικά τη λογική συνάρτηση σε μια διαμορφώσιμη συσκευή.



Σχήμα 4.24: Η εταιρία Xilinx παρέχει το ISE και το Vivado ως ολοκληρωμένα εργαλεία ανάπτυξης για τα δικά της FPGA.

- Τοποθέτηση (place) και σύνδεση (route): τοποθετεί λογικά στοιχεία σε συγκεκριμένες θέσεις του στόχου τσιπ FPGA και συνδέει λογικά στοιχεία. Αυτό το στάδιο γίνεται συνήθως με εκάστοτε εργαλεία της εταιρείας FPGA, όπως της Xilinx, Actel, Altera, κλπ
- Παραγωγή προγράμματος (program generation): παράγεται ένα αρχείο bit-stream για προγραμματισμό της συσκευής.
- Προγραμματισμός συσκευής (device programming): μεταφόρτωση του αρχείου bit-stream στο FPGA.
- Επιβεβαίωση σχεδιασμού (design verification): χρησιμοποιείται η προσομοίωση για να ελέγξει τις λειτουργίες. Η προσομοίωση μπορεί να γίνει σε διαφορετικά επίπεδα. Η λειτουργική ή προσομοίωση συμπεριφοράς δεν λαμβάνει υπόψη τις καθυστερήσεις συστατικών ή της διασύνδεσης. Η προσομοίωση χρονισμού χρησιμοποιεί τις πληροφορίες καθυστέρησης που εξάγονται από το κύκλωμα. Επίσης, παράγονται άλλες αναφορές για να ελέγξουν άλλα αποτελέσματα εφαρμογής, όπως η μέγιστη συχνότητα, η καθυστέρηση και η χρήση των πόρων.

Τα στάδια απεικόνιση, τοποθέτηση και σύνδεσης αναφέρονται κοινά ως υλοποίηση σχεδιασμού.

4.10 Σύστημα πάνω σε ολοκληρωμένο κύκλωμα-SoC

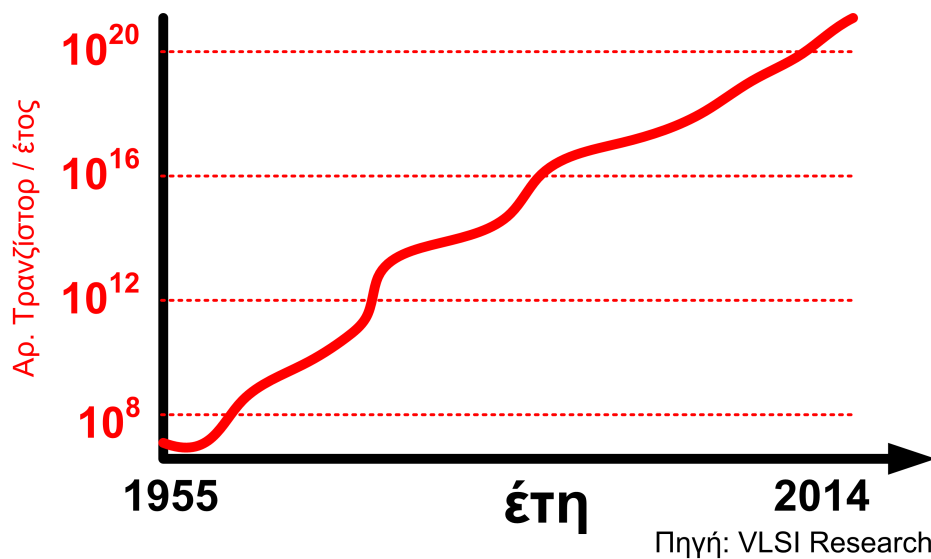
Η βιομηχανία ημιαγωγών βρίσκεται σε ανθηρούς καιρούς, με παγκόσμιες πωλήσεις το έτος πάνω από 300 δισεκατομμύρια αμερικάνικα δολάρια- από τα οποία το 30% προέρχεται από τους μικροεπεξεργαστές, τα DSP, τους μικροελεγκτές, και τα περιφερειακά chips που μπορούν να προγραμματισθούν. Αναλυτικά στοιχεία υπάρχουν από την Ομοσπονδία Κατασκευαστών Ημιαγωγών¹, η οποία διατηρεί ένα αρκετά ενημερωμένο αρχείο πωλήσεων τόσο ανά γεωγραφική περιοχή, όσο και συνολικά. Εκεί φαίνεται η συνεχόμενη αύξηση και από τα 261 εκατομμύρια το 1976, οι συνολικές πωλήσεις έφτασαν 300 δισεκατομμύρια το 2015. Όπως αναγράφεται στην αρχική σελίδα της Ομοσπονδίας, μόνο οι αμερικάνικες εταιρίες πούλησαν 71 εκατομμύρια ολοκληρωμένα κυκλώματα το 2014 ή 200 IC για κάθε Αμερικάνο πολίτη.

Μαζί με αυτές τις τεράστιες ευκαιρίες αγοράς, ωστόσο, έχει επέλθει και μια αύξηση στην πολυπλοκότητα του σχεδιασμού συστημάτων. Εκτιμάται ότι μέχρι το τέλος του 2015 τα αναμενόμενα ποσά τρανζίστορ για λύσεις σε τυπικά συστήματα σε chip (SoC) θα είναι πάνω από 3 τρισεκατομμύρια, με αντίστοιχες αναμενόμενες ταχύτητες ρολογιού πάνω από 100 GHz, και τις πυκνότητες τρανζίστορ να φτάνουν τα 660 εκατομμύρια τρανζίστορ/τετρ. εκατοστό. Σύμφωνα με την εταιρία VLSI Research, ο αριθμός των τρανζίστορ που κατασκευάστηκαν το 2014, έχει ξεπεράσει τα 10^{18} (Εικόνα 4.25), που είναι ένα αστρονομικό νούμερο (25 φορές μεγαλύτερο από το συνολικό αριθμό των αστεριών στο γαλαξία μας, και 75 φορές μεγαλύτερος από το συνολικό εκτιμώμενο αριθμό γαλαξιών στο σύμπαν). Η αύξηση του αριθμού των τρανζίστορ ακολουθεί το νόμο του Moore, του διπλασιασμού του αριθμού των τρανζίστορ στην ίδια επιφάνεια κάθε 18 μήνες.

Παράλληλα, αυτή η αύξηση στην πολυπλοκότητα προκαλεί μια αύξηση σε απώλεια ενέργειας, κόστος, και στον χρόνο 'σχεδιασμού της αγοράς'. Στην επιστημονική κοινότητα συζητιέται ότι τα προϊόντα υπολογιστών θα εξελιχθούν τελικά από μεγάλα, γενικού σκοπού, απρόσωπα δυσκίνητα συστήματα σε φορητά, προσωπικά, ευέλικτα, στοχευμένα στην αγορά προϊόντα. Η εξατομίκευση, η ευελιξία και η γρήγορη απόκριση στην αγορά θα υπαγορεύσει μια 'απότομη αλλαγή' προσέγγιση σχεδιασμού. Αυτό φαίνεται και στα πιο πρόσφατα στατιστικά που δημοσίευσε τον Οκτώβριο του 2015 η Gartner στα οποία

¹<http://www.semiconductors.org/>

φαίνεται η πτώση κατά 9.2% των πωλήσεων των προσωπικών υπολογιστών από όλους τους κατασκευαστές και η πολλαπλάσια άνοδος πωλήσεων των κινητών τηλεφώνων. Το συμπέρασμα είναι ότι υπάρχει μια έντονη τάση της μετακίνησης της προτίμησης από μεγάλες συσκευές (π.χ. υπολογιστές γραφείου) σε μικρότερες (κινητά) και σε λίγα χρόνια σε ακόμη μικρότερες (π.χ. φορέσιμες συσκευές, όπως τα γυαλιά της google με τις κατάλληλες εφαρμογές [33], [34]). Αυτό δείχνει ότι και η ανάγκη για βέλτιστα και αποδοτικά ενσωματωμένα συστήματα συνεχώς θα αυξάνεται, όπως και θα αυξάνονται και οι πιέσεις που αφορούν τα προϊόντα, με το πιο χαρακτηριστικό το χρόνο τοποθέτησης στην αγορά.



Σχήμα 4.25: Ο συνολικός αριθμός τρανζίστορ που παράχθηκαν το 2014 είναι 250×10^{18} , ένα αστρονομικό νούμερο που αν συγκριθεί με τον αριθμό των αστεριών στον γαλαξία μας τον ξεπερνάει κατά 25 φορές.

Ο χρόνος τοποθέτησης του προϊόντος στην αγορά για νέες πλατφόρμες δε θα μετριέται πλέον σε χρόνια, αλλά σε μήνες ή εβδομάδες. Ο κύκλος σχεδιασμού πρέπει να μειωθεί ή να γίνει το στενό πέρασμα για μελλοντικές εξελίξεις και να καθορίσει ποιες εταιρίες θα επιζήσουν και ποιες όχι. Οι αυστηρές απαιτήσεις στο time-to-market, ζευγαρωμένες με επιπρόσθετους περιορισμούς σχεδίασης όπως η ευελιξία σχεδίασης, το κόστος, οι απαιτήσεις πραγματικού χρόνου και ο άκαμπτος παράγοντας προτύπου (π.χ. μέγεθος, βάρος και απώλεια ενέργειας) αντιπροσωπεύουν μια αξιολάβαστη πρόκληση που οι σχεδιαστές των υφιστάμενων συστημάτων πρέπει να ξεπεράσουν. Σήμερα, η ανάγκη για μια ιδανική αλλαγή κατεύθυνσης στην μέθοδο σχεδιασμού έχει γίνει ακόμη περισσότερο απαιτητική, αλλά και δύσκολη.

Όσον αφορά τον όγκο πωλήσεων, οι ενσωματωμένοι επεξεργαστές έχουν ξε-

περάσει τους επεξεργαστές H/Y κατά πολύ. Μια μεγάλη έκταση από εφαρμογές, ειδικά στον τομέα των ασύρματων και φορητών συσκευών, έχουν χαρακτηριστεί από μια τεράστια απαίτηση για ενσωματωμένους επεξεργαστές. Δεδομένου μιας εκτιμημένης εξέλιξης στην τεχνολογία IC σήμερα, ένα ενσωματωμένο SoC πραγματικού χρόνου θα βρει ακόμη περισσότερες κατάλληλες εφαρμογές στο μέλλον, με την επαναχρησιμοποίηση των IP να παρουσιάζεται ως μια μέθοδος για βελτίωση της παραγωγικότητας.

Τα τρέχοντα δεδομένα πυκνότητας ισχύος δείχνουν ότι η απόδοση τείνει να διπλασιάζεται κάθε 18 μήνες. Η θερμική εξαγωγή των ολοκληρωμένων κυκλωμάτων σε κάποιες περιπτώσεις μπορεί να φτάσει τα επίπεδα εκείνα της πυρηνικής ισχύος (250 watts/τετρ. εκατοστό), αν δε ακολουθηθούν οι βέλτιστες πρακτικές εξοικονόμησης ενέργειας και σχεδιασμού. Με μια τέτοια αύξηση σε απώλεια ενέργειας, τα κόστη που σχετίζονται με τις ανάγκες συσκευασίας και θερμικής εξαγωγής μπορούν να μειώσουν οποιαδήποτε πλεονάσματα επιτυγχάνονται με υψηλότερες πυκνότητες τρανζίστορ. Ο θόρυβος και τα ζητήματα σύζευξης θα λύνονται επίσης πιο δύσκολα όσο οι συχνότητες αυξάνονται. Σαν αποτέλεσμα ένα καταναμημένο σύστημα που περιέχει αρκετούς μικρότερους, αργότερους, ενεργειακά αποδοτικούς επεξεργαστές θα γίνει μια προτιμότερη επιλογή για ένα σύστημα μεμονωμένου επεξεργαστή με πρόβλημα απώλειας ενέργειας που λύνεται δύσκολα. Ήδη έχουν προταθεί ως λύση στο πρόβλημα δημιουργίας συστοιχιών μεγάλων επεξεργαστικών δυνατοτήτων, η χρήση των επεξεργαστών ARM που μπορεί να είναι πιο αργοί από τους επεξεργαστές της Intel, αλλά είναι πιο αποδοτικοί στην ενέργεια [35], και για αυτό αποτελούν τη de-facto επιλογή στα έξυπνα κινητά τηλέφωνα.

Όσο η τεχνολογία των IC εξελίσσεται από μικρό-επίπεδα σε υποδεέστερα μικρό-επίπεδα και από αυτά σε επίπεδα στοιχειώδους δομής, το κόστος παραγωγής συνεχώς αυξάνεται εκθετικά. Μια γραμμή παραγωγής για ολοκληρωμένα κυκλώματα με μέγεθος 24nm, κοστίζει αρκετά δισεκατομμύρια δολάρια [36]. Έτσι, οι σχεδιαστές ολοκληρωμένων κυκλωμάτων θα τείνουν να παράγουν ένα chip κατάλληλο για περισσότερες από μια εφαρμογές. Αυτά θα συνδυάζονται με το γεγονός ότι η ευελιξία επίδοσης και η ευελιξία εφαρμογής είναι τόσο βασικές για την επιτυχία ενός ενσωματωμένου συστήματος, οπότε συμπεραίνουμε ότι ένα σύστημα με υψηλό βαθμό αναδιαμόρφωσης μπορεί να μην είναι ασυνήθιστο στο κοντινό μέλλον.

Ο συ-σχεδιασμός υλικού-λογισμικού αναφέρεται σε ένα παράλληλο και συνεργατικό σχεδιασμό υλικού και λογισμικού σε ένα σύστημα. Μια διαδικασία συ-σχεδιασμός συνήθως περιλαμβάνει τέσσερις κύριες εργασίες: επιλογή αρχιτεκτονικής, διαμοιρασμό HW/SW, χρονικό προγραμματισμό και σύνθεση επικοινωνίας. Η διαδικασία σχεδιασμού ρέει σε ένα μοντέλο καταρράκτη, που συχνά ξεκινά με ένα σύνολο λεπτομερειών.

Όπως επισημάνθηκε και στην εισαγωγή του βιβλίου, οικονομικοί παράγο-

ντες στο σχεδιασμό VLSI έχουν οδηγήσει την ενσωμάτωση σε επίπεδα στα οποία κυκλώματα που προηγουμένως χρειαζόταν για να υλοποιηθούν ολόκληρες πλακέτες FPGA και πολλαπλά τσιπ, τώρα ενσωματώνονται σε έναν ενιαίο μόνο ολοκληρωμένο σύστημα (chip), με την ονομασία System-on-Chip (SoC). Το σύστημα SoC παρουσιάζει αντιφατικές προδιαγραφές. Αφ' ενός παρέχει αρκετές ευκαιρίες για ελαχιστοποίηση του κόστους, και αφετέρου συνεπάγεται έναν υψηλότερο βαθμό εξειδίκευσης ο οποίος μειώνει το συνολικό όγκο κάθε επιμέρους συστατικού. Με άλλα λόγια, ο συνδυασμός δύο γενικού σκοπού συστατικών είναι λιγότερο γενικός από κάθε συστατικό ξεχωριστά. Δεδομένου ότι ενσωματώνουμε περισσότερα συστατικά το προκύπτει κομμάτι του πυριτίου γίνεται πιο εξειδικευμένο και αναμένονται ενδεχόμενες μειώσεις του όγκου του (σχετικά με τη δυνατότητα εφαρμογής του σε διάφορα προϊόντα). Επιπλέον, με την αύξηση που προκύπτει στις δαπάνες πολυπλοκότητας και σχεδιασμού, υποδηλώνεται ότι οι σχεδιαστές πρέπει να βρουν νέους τρόπους να επαναχρησιμοποιήσουν το αποτέλεσμα στο σχεδιασμό, και ενθαρρύνεται η τακτική της επαναχρησιμοποίησης του σχεδίου και των έτοιμων προς χρήση (plug-and-play) συστατικών. Πρέπει επίσης να εξεταστούν οι ατέλειες της παραγωγικής και κατασκευαστικής διαδικασίας, θέματα που προκαλούν προβλήματα στα SoC. Μια άλλη σημαντική επίπτωση της επιλογής SoC είναι η αυξανόμενη σημασία και δυσκολία που υπάρχει, ώστε να εξασφαλιστεί η ορθότητα των συστημάτων. Αυτό περιλαμβάνει την λεπτομερή παρουσίαση, την προσομοίωση, την επικύρωση, την επαλήθευση και τη δοκιμή.

Η έλλειψη καλής τεκμηρίωσης είναι συνήθως ένα από τα μεγάλα εμπόδια στην επαναχρησιμοποίηση του πρωτοτύπου σχεδιασμού. Η επαναχρησιμοποίηση επιτυγχάνεται με τους σκληρούς και μαλακούς πυρήνες πνευματικής ιδιοκτησίας (intellectual property, IP). Η επαναχρησιμοποίηση πυρήνων, είτε σκληρών είτε μαλακών, υπονοεί ένα χαρτοφυλάκιο που περιέχει στοιχεία καθορισμένα με σαφήνεια, με τον προσεκτικό χαρακτηρισμό του συγχρονισμού, της λειτουργίας, και της στρατηγικής επαλήθευσης.

Οι μαλακοί (soft) πυρήνες είναι διαθέσιμοι είτε ως RTL (register transfer level, περιγραφή επιπέδου επικοινωνίας καταχωρητών), είτε ως τεχνολογικά ανεξάρτητες λίστες δικτύων διασύνδεσης (netlists). Εκτός από τους μαλακούς πυρήνες, υπάρχουν και οι σκληροί πυρήνες. Οι σκληροί πυρήνες έχουν ήδη τοποθετηθεί και δρομολογηθεί για μια συγκεκριμένη VLSI διαδικασία. Σήμερα, οι περισσότερες IP προσφέρονται είτε για μαλακής είτε για σκληρής μορφής πυρήνες. Η χρήση των πυρήνων βελτιώνει σε μεγάλο ποσοστό την παραγωγικότητα, όμως θέτει και σημαντικές προκλήσεις στην επικύρωση των συστημάτων, στην εφαρμογή των αλλαγών, στην προσαρμογή του πυρήνα σε διαφορετικούς διαύλους ή στις μεταδόσεις σημάτων, στην εφαρμογή μιας νέας τεχνολογικής διαδικασίας και στο συντονισμό τους με ποικίλους κατασκευαστές πυριτίου.

Η εικονική σύνθετη επαναχρησιμοποίηση είναι αυτή που εγγυάται έναν

πλήρη διαχωρισμό των εξουσιοδοτημένων και των ενσωματωμένων ρόλων. Ένα εικονικό συστατικό είναι ένα πλήρως χαρακτηρισμένο, ελεγμένο, διαμορφωμένο στοιχείο. Αυτό συνεπάγεται την υιοθέτηση μιας αυστηρής προδιαγραφής των διεπαφών επικοινωνίας, μιας αυστηρά ορισμένης τεκμηρίωσης και έναν ξεκάθαρο διαχωρισμό συστημάτων. Αρχικά, αυτό λειτουργεί ενάντια στην αντιστοιχία χρόνος-αγορά και την απόδοση, και απαιτούνται αρκετές αναθεωρήσεις του σχεδιασμού, ώστε να επιτευχθεί ένα καλό αποτέλεσμα.

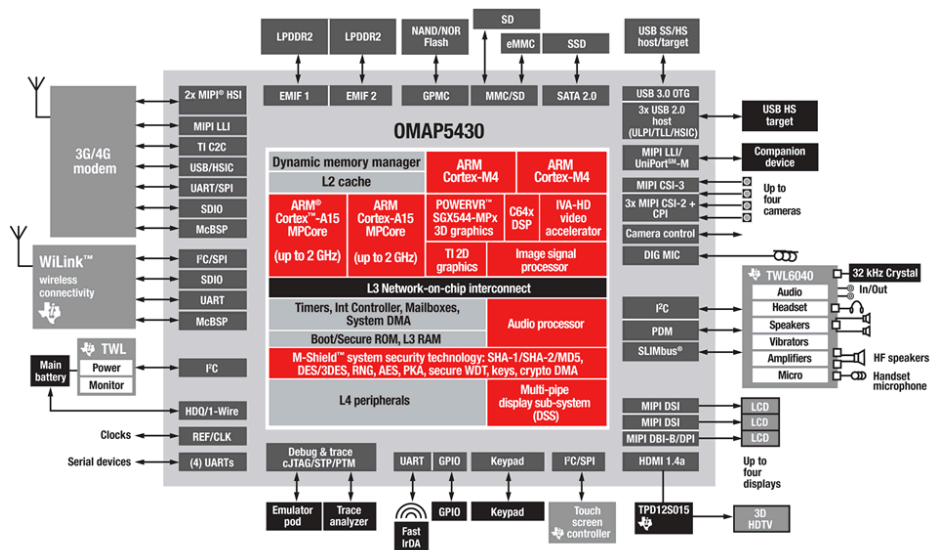
4.10.1 Πρώτο Παράδειγμα για SoC

Σε αυτή τη παράγραφο παρουσιάζουμε ένα σύστημα SoC που χρησιμοποιείται σε εφαρμογές πολυμέσων φορητών συσκευών. Ο κόσμος των κινητών τηλεπικοινωνιών μας δίνει ένα ενδιαφέρον παράδειγμα μιας ετερογενούς SoC πλατφόρμας πολυπρογραμματισμού. Η τέταρτη γενιά (4G) των ασύρματων τεχνολογιών θα προσφέρει σύντομα σε όλους αρκετή χωρητικότητα για εφαρμογές πολυμέσων σε συσκευές κυψελωτού δικτύου, όπως βιντεοκλήση, αναγνώριση λόγου και εφαρμογές ηλεκτρονικού εμπορίου. Σε αυτό το σενάριο, η πλατφόρμα του υλικού πρέπει να προσφέρει τόσο ταχύτητα όσο και ευελιξία, δεδομένου ότι θα πρέπει να υποστηρίζει μεγάλη ποικιλία δυνητικά εξελιξιμων προτύπων και εφαρμογών. Ταυτόχρονα, τα κόστη των τερματικών συσκευών (κινητά και PDA) και η περιορισμένη ηλεκτρική ισχύς, κάνουν σαφή την ανάγκη για μια πολύ ειδικά σχεδιασμένη ενσωματωμένη λύση. Η πλατφόρμα OMAP της Texas Instruments (Σχήμα 4.26) καλύπτει πολλές από αυτές τις απαιτήσεις, και παρέχεται με διάφορες ρυθμίσεις. Για παράδειγμα, η συσκευή OMAP1510 έχει ένα διπύρηνο SoC περιλαμβάνει ένα επεξεργαστή 200MHz, με πυρήνα C55x DSP (μέγιστη απόδοση 400 MIPS), και ένα 175MHz με πυρήνα ARM9 (μέγιστη απόδοση 210 MIPS). Μαζί με τους δυο πυρήνες το σύστημα ενσωματώνει ακόμα αρκετές τοπικές συστοιχίες μνήμης (local banks) τοπικής μνήμης (ένα 192-KB frame buffer για video, περισσότερο από 64 KB για μνήμη δεδομένων και 12 KB I-cache για το DSP, και 16 KB I-cache και 8 KB D-cache για τον ARM) και διάφορες άλλες I/O συσκευές.

Ο συνδυασμός ενός DSP και ενός πυρήνα γενικής χρήσης είναι ένας καλός συμβιβασμός κόστους, απόδοσης, αλλά και απαίτησης ρεύματος. Ο ARM διαχειρίζεται και ελέγχει τον κώδικα του συστήματος (λειτουργικό σύστημα, περιβάλλον χρήστη, χειριστές διακοπών (interrupt handlers) κ.ο.κ.). Το DSP σαφώς πολύ καλύτερο για την επεξεργασία σήματος βάσης σε πραγματικό χρόνο και σε χειρισμό πολυμέσων (όπως βίντεο). Δοκιμάζοντας να εκτελέσουμε ολόκληρη την εφαρμογή σε έναν ταχύτερο RISC επεξεργαστή θα είχαμε σημαντικά προβλήματα με την κατανάλωση ενέργειας, αλλά και με το μέγεθος του κώδικα για το σύστημα. Η εκτέλεση Λειτουργικού συστήματος και κώδικα συστήματος σε C55x DSP είναι απλά πέρα των δυνατοτήτων του.

Από την σκοπιά της ανάπτυξης εφαρμογής μια ετερογενής διπύρνηνης πλατφόρμας θέτει σημαντικές προκλήσεις. Καταρχάς, όπως αναφέρθηκε, οι σχεδιαστές πρέπει να αποφασίζουν στατικά, τι θα εκτελείται σε DSP και τι σε ARM. Όταν το κατορθώσουν αυτό, για να απλοποιήσουν το σχεδιασμό, ο συνδυασμός αλυσίδας εργαλείων πρέπει να εξάγει την αρχιτεκτονική του DSP επιταχυντή και να την παρουσιάσει μέσα σε ένα φιλικό προς τον χρήστη περιβάλλον (API), έτσι ώστε η κύρια εφαρμογή να μπορεί να προγραμματιστεί ανεξάρτητα από το DSP. Αυτή η εφαρμογή γραφικού περιβάλλοντος DSP (ονομάζεται γέφυρα DSP στο περιβάλλον TI) παρέχει μηχανισμούς για να ξεκινήσει και να ελέγξει εργασίες στο DSP, να μετακινήσει δεδομένα από και προς το DSP, και να ανταλλάξει μηνύματα. Με αυτό τον τρόπο, οι λειτουργίες που εκτελούνται στον DSP είναι προσβάσιμες διαμέσου ενός περιβάλλοντος κλήσης απομακρυσμένης βοήθειας που θυμίζει τις γνωστές συναρτήσεις C.

TI OMAP5430 SoC



Σχήμα 4.26: Η αρχιτεκτονική υλικού TI OMAP. Το διάγραμμα δείχνει ένα παράδειγμα (τον TI OMAP 5430 SoC) ένα ετερογενή DSP μικροεπεξεργαστή στο ίδιο υπόστρωμα πυριτίου Αυτό συμπεριλαμβάνει ένα ARM CORTEX A15 32-bit RISC πυρήνα και μικροεπεξεργαστές ARM-Cortex-M4, μαζί με μνήμη και καθιερωμένα περιφερειακά. Παρότι το OMAP είχε κυρίως σχεδιαστεί για ασύρματες εφαρμογές πολυμέσων (όπου το τηλεπικοινωνιακό κομμάτι τρέχει στο DSP και η υπόλοιπη εφαρμογή τρέχει σε ARM), είναι εύκολο να γενικευτεί η ιδέα και σε άλλες περιοχές που έχουν ένα μείγμα επεξεργασίας σήματος και απαιτήσεις γενικής χρήσης (ή πεπαλαιωμένο λειτουργικό). Εικόνα από ti.com

Από την πλευρά του DSP, οι προγραμματιστές συνήθως υλοποιούν μεμονωμένες εφαρμογές μέσα από μια ξεχωριστή συλλογή, αλυσίδας εργαλείων. Είναι ακόμη συνηθισμένο για το κατασκευαστή του συστήματος να δίνει ένα σετ από τυποποιημένες DSP βιβλιοθήκες που υλοποιούν τις πιο κοινές λειτουργίες (φίλτρα, μεταμορφώσεις, κωδικοποιητές, κ.α.), έτσι δεν απασχολεί τους προγραμματιστές εφαρμογών το καθόλου φιλικό προς τον χρήστη DSP. Ένα παράδειγμα προγράμματος του OMAP είναι η εφαρμογή ενός πρότυπου video κλήσης 3G σε ένα κινητό τερματικό. Η video κλήση περιλαμβάνει έναν αποκωδικοποιητή πολυμέσων (για να συμπιέζει τη μεταδιδόμενη φωνή και εικόνα που θα σταλεί και να αποσυμπιέζει τις εισερχόμενες ροές) και ορισμένα πρωτόκολλα μετάδοσης (εξασφαλίζουν τη μεταφορά πολυμέσων για να υπάρχει η ακολουθία των πακέτων στο κανάλι, ένα πρωτόκολλο για να διαχειρίζεται την κλήση, και ένα πρωτόκολλο ζεύξης για να περιγράψει το περιεχόμενο). Σε αυτή την περίπτωση, συμπίεση και αποσυμπίεση υλοποιούνται στον DSP (χρησιμοποιώντας κάποιες αφιερωμένες οδηγίες του C55x σύμφωνα με το codec), όπου ο ARM τρέχει τα πρωτόκολλα συμπληρωματικά στο κώδικα του συστήματος και στο περιβάλλον του χρήστη.

Τα ενσωματωμένα συστήματα υπολογισμού πρέπει να αντιμετωπίσουν τους περιορισμούς του κόστους, της κατανάλωσης ενέργειας, και της απόδοσης. Εάν επικρατούσε μία απαίτηση σχεδίου, η ζωή θα ήταν πολύ ευκολότερη για τους σχεδιαστές ενσωματωμένων συστημάτων που —θα μπορούσαν να χρησιμοποιούν σωστά τυποποιημένες αρχιτεκτονικές με εύκολα μοντέλα προγραμματισμού. Αλλά επειδή οι τρεις περιορισμοί πρέπει να αντιμετωπισθούν ταυτόχρονα, οι σχεδιαστές ενσωματωμένων συστημάτων πρέπει να προσαρμόσουν το υλικό και τις αρχιτεκτονικές λογισμικού, ώστε να ανταποκρίνεται στις ανάγκες των εφαρμογών. Το εξειδικευμένο υλικό βοηθάει ώστε να αντιμετωπισθούν οι απαιτήσεις απόδοσης για χαμηλότερη κατανάλωση ενέργειας στο λιγότερο δυνατό κόστος από ένα γενικής χρήσης σύστημα. Συνήθως, τα ενσωματωμένα συστήματα υπολογισμού είναι συχνά ετερογενείς πολυεπεξεργαστές με πολλαπλές CPUs και στοιχεία επεξεργασίας (PEs). Στο συσχεδιασμό εξειδικευμένων επεξεργαστικών μονάδων (processing elements, PE) και επεξεργαστών, τα στοιχεία επεξεργασίας PEs καλούνται γενικά επιταχυντές. Αντίθετα, ο συνεπεξεργαστής ελέγχεται από τη μονάδα εκτέλεσης μιας ΚΜΕ.

Ο συ-σχεδιασμός υλικού/λογισμικού είναι μια συλλογή των τεχνικών που οι σχεδιαστές χρησιμοποιούν για να τους βοηθήσουν τη δημιουργία αποδοτικών συστημάτων ειδικής εφαρμογής. Εάν δεν γνωρίζει κάποιος τίποτα για τα χαρακτηριστικά της εφαρμογής, είναι δύσκολο να ξέρει πώς να βελτιστοποιήσει το σχεδιασμό του συστήματος. Αλλά εάν γνωρίζει κάποιος την εφαρμογή, ως σχεδιαστής, όχι μόνο μπορεί να προσθέσει χαρακτηριστικά στο υλικό και το λογισμικό που το κάνουν να τρέχει γρηγορότερα χρησιμοποιώντας λιγότερη ενέργεια, αλλά μπορεί επίσης να αφαιρέσει στοιχεία υλικού και λογισμι-

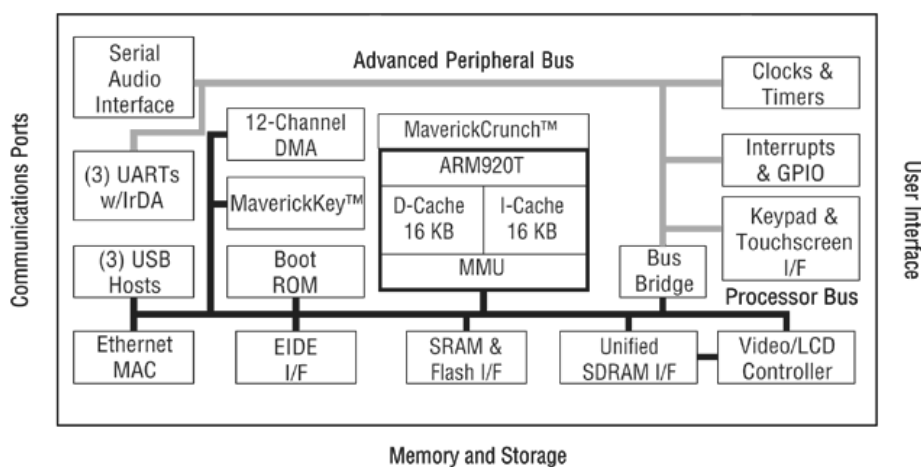
κού που δεν βοηθούν την εφαρμογή. Το να αφαιρούνται περιττά στοιχεία είναι συχνά τόσο σημαντικό, όσο το να προσθέτονται νέα χρήσιμα στοιχεία. Όπως το όνομα υπονοεί, ο συ-σχεδιασμός υλικού/λογισμικού σημαίνει σχεδιασμό από κοινού αρχιτεκτονικές υλικού και λογισμικού, ώστε να επιτευχθούν οι στόχοι όσον αφορά το κόστος, την ενέργεια και την απόδοση.

Ο συ-σχεδιασμός είναι μια ριζικά διαφορετική μεθοδολογία σε σχέση με τις στρωματικές αφαιρέσεις, οι οποίες έχουν χρησιμοποιηθεί στον υπολογισμό γενικής χρήσης. Επειδή ο συ-σχεδιασμός προσπαθεί να βελτιστοποιήσει πολλά διαφορετικά μέρη του συστήματος συγχρόνως, κάνει εκτενή χρήση εργαλείων για την ανάλυση, αλλά και για τη βελτιστοποίηση σχεδίου. Όλο και περισσότερο, ο συ-σχεδιασμός υλικού/λογισμικού χρησιμοποιείται και για να σχεδιάσει συστήματα. Παραδείγματος χάριν, οι κεντρικοί υπολογιστές μπορούν να βελτιωθούν με μερικές εξειδικευμένες εφαρμογές από τις λειτουργίες του σωρού του λογισμικού τους.

4.10.2 Δεύτερο Παράδειγμα για SOC

Τα SoCs ενσωματώνουν τα στοιχεία που προέρχονται από διάφορους προμηθευτές. Παραδείγματος χάριν, ο επεξεργαστής ήχου Cirrus Logic Maverick EP9312 (Εικόνα 4.27). Αυτό είναι ένα SoC που ενσωματώνει έναν επεξεργαστή ARM920T, μαζί με όλες τις λειτουργίες για να επιτρέψει τις εφαρμογές Jukebox-καταγραφής ψηφιακής μουσικής (εκτέλεση-καταγραφή). Το υπόλοιπο των συστατικών SoC περιλαμβάνει την ικανότητα επεξεργασίας σήματος, τις επικοινωνίες, την αποθήκευση, και τη διεπαφή με τον χρήστη. Στο συγκεκριμένο παράδειγμα, το τσιπ περιλαμβάνει τα στοιχεία για τις τυποποιημένες διασυνδέσεις, καθώς επίσης και τα ιδιόκτητα στοιχεία για τις πιο εξατομικευμένες λειτουργίες. Το τελευταίο σύνολο στοιχείων εμπεριέχει έναν προγραμματίσιμο μαθηματικό επεξεργαστή για τους ακέραιους αριθμούς, φίλτρο δεκαδικής υποδιαστολής και το συστατικό κλειδώματος τύπου Maverick για να διαχειρίζεται τις καταστάσεις ασφάλειας για την διαχείριση ψηφιακών δικαιωμάτων.

Τα SoCs είναι σύνθετα ολοκληρωμένα κυκλώματα. Παραδείγματος χάριν, το EP93121 περιλαμβάνει 5.7 εκατομμύρια τρανζίστορ σε τεχνολογία CMOS 0,25 μm , παρέχοντας υψηλή ταχύτητα (200 MHz) και χαμηλή στιγμιαία κατανάλωση ισχύος (1,5 W) συγκρινόμενος με έναν γενικής χρήσης επεξεργαστή. Μια τυπική εφαρμογή για το EP9312 είναι το οικιακό σύστημα αναπαραγωγής τραγουδιών, για το οποίο χρειάζεται να δημιουργηθούν, να διαχειριστούν, και να αποσυμπιεστούν πολλαπλές ροές συμπιεσμένων ακουστικών στοιχείων σε όλο το σπίτι για ποικίλους ψηφιακούς φορείς μουσικής (π.χ., ένας φορητός προσωπικός φορέας, ένα κέντρο ψυχαγωγίας, ένα στερεοφωνικό συγκρότημα αυτοκινήτων).



Σχήμα 4.27: Παράδειγμα ενός εμπορικού SoC. Το διάγραμμα παρουσιάζει το SoC Cirrus Logic EP9312 chip. Αυτό περιλαμβάνει όλα τα συστατικά ενός χαρακτηριστικού SoC: ένας τριανταδύαμιτος πυρήνας RISC (ο ARM920T), τυποποιημένες περιφερειακές μονάδες (ήχος, UARTs, USB, Ethernet), μνήμη διεπαφής (SRAM, flash, SDRAM), διεπαφές βίντεο και συσκευές εισόδου, ιδιότητα στοιχείου IP (το τμήμα ασφάλειας κλειδώματος τύπου Maverick), και τους διαύλους διασύνδεσης.

4.10.3 Δομοστοιχεία Πνευματικής Ιδιοκτησίας

Η μεγάλη 'υπόσχεση' της τεχνικής επαναχρησιμοποίησης του σχεδιασμού στην τεχνολογία SoC, είναι να καταστήσει τα τμήματα υλικού τόσο ανταλλάξιμα όσο τα παξιμάδια βίδας και τα μπουλόνια στην κατασκευή των φυσικών αγαθών. Οι εταιρείες κατασκευής μικροεπεξεργαστών και σχεδιασμού περιβάλλοντος δεν μπορούν να σχεδιάσουν από την αρχή ολοκληρωμένα SoCs για κάθε νέο προϊόν. Αυτό οδηγεί στην κεντρική ιδέα της επαναχρησιμοποίησης των συστατικών SoC, συχνά αποκαλούμενης ως στοιχεία πνευματικής ιδιοκτησίας (ή στοιχεία IP). Με τον όρο επαναχρησιμοποίηση (όρος δανεισμένος από την περιοχή της τεχνολογίας λογισμικού), η κοινότητα SoC δείχνει ότι είναι ικανή να υιοθετήσει την περιγραφή των στοιχείων του υλικού στα διάφορα σχέδια με ελάχιστες ή καθόλου αλλαγές. Τα στοιχεία IP μπορεί να είναι οτιδήποτε από τα περιφερειακά αναλογικά, και τους ελεγκτές διαύλων, ως και τις γέφυρες που υλοποιούνται από τους πυρήνες μικροεπεξεργαστών. Οι εταιρείες μικροεπεξεργαστών μιλούν πολύ για τη μέθοδο επαναχρησιμοποίησης, αλλά μέχρι τώρα αγωνίζονται ακόμα να βρουν έναν αποδοτικό τρόπο για να θέσουν τη μέθοδο σε εφαρμογή. Η χορήγηση αδειών IP από τους τρίτους προκαλεί νομικά ζητήματα και προβλήματα συμβατότητας, προσθέτοντας περισσότερες προκλήσεις στην

επαναχρησιμοποίηση IP. Το σημαντικότερο πρόβλημα με την επαναχρησιμοποίηση IP σχετίζεται με την έλλειψη τυποποιημένων διεπαφών, μεθοδολογίας των συστημάτων προσομοίωσης, και κοινής στρατηγικής επικύρωσης.

Η προσωπική επαναχρησιμοποίηση είναι παραδοσιακή στις ομάδες σχεδιασμού ASIC. Στη μέθοδο αυτή τα μέρη ενός σχεδίου επαναχρησιμοποιούνται για τις διαδοχικές εκδόσεις. Το πλεονέκτημα έγκειται στον γεγονός ότι δεν απαιτείται ειδική υποδομή, εξ άλλου η δεδομένη μέθοδος δεν ανέρχεται σε κλιμακούμενη σειρά στις μεγαλύτερες ομάδες ή στα διαφορετικά σχέδια, και συνήθως υπονοεί την επαναχρησιμοποίηση της ομάδας του σχεδιασμού.

Η επαναχρησιμοποίηση πρωτοτύπου (source-πηγή) παρέχει μια υπάρχουσα περιγραφή RTL (επίπεδο μεταφοράς καταχωρητών), ή λίστα δικτύου, ή περιγραφή για την αυτοδύναμη επανεκκίνηση ενός νέου σχεδίου. Η αποτελεσματικότητά της εξαρτάται από την ποιότητα του κώδικα, της τεκμηρίωσης, της ακολουθίας ελέγχου, και της διαθεσιμότητας του αρχικού σχεδιαστή.

4.10.4 Πλατφόρμες σχεδίου

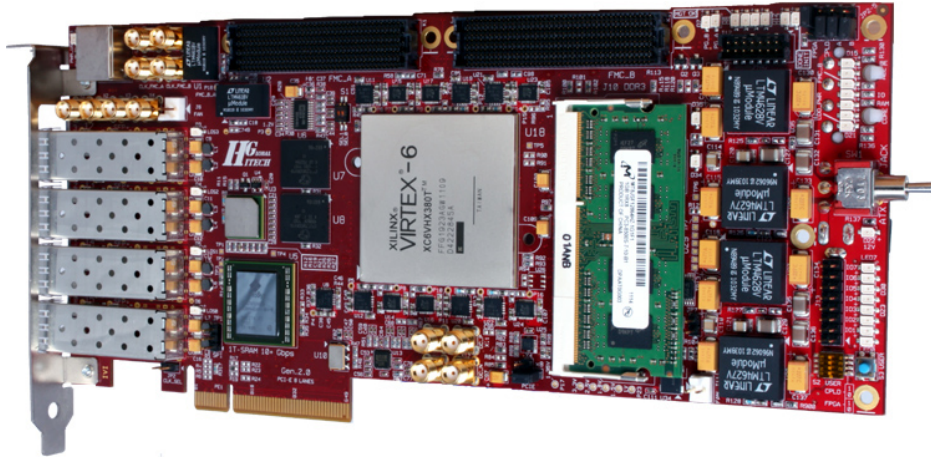
Ο συ-σχεδιασμός υλικού/λογισμικού μπορεί να χρησιμοποιηθεί για να σχεδιάσει συστήματα από την αρχή ή για να δημιουργήσει συστήματα που υλοποιούνται σε μια υπάρχουσα πλατφόρμα. Η χρήση της ΚΜΕ και μιας αρχιτεκτονικής επιταχυντών είναι μια κοινή πλατφόρμα συσχεδιασμού. Μια ποικιλία διαφορετικών CPUs μπορεί να χρησιμοποιηθεί για να φιλοξενήσει τον επιταχυντή. Ο επιταχυντής μπορεί να εφαρμόσει πολλές διαφορετικές λειτουργίες, επιπλέον μπορεί να εφαρμοστεί χρησιμοποιώντας οποιαδήποτε από τις αρκετές τεχνολογίες λογικής. Αυτές οι επιλογές επηρεάζουν το χρόνο σχεδιασμού, την κατανάλωση ενέργειας, και άλλα σημαντικά χαρακτηριστικά του συστήματος.

Η πλατφόρμα συσχεδιασμού θα μπορούσε να εφαρμοστεί σε οποιοδήποτε από τις πολλές διαφορετικές τεχνολογίες σχεδιασμού.

Μια πιθανή υλοποίηση είναι ένα σύστημα βασισμένο σε Η/Υ με τον επιταχυντή να στεγάζεται σε μια κάρτα που συνδέεται με το PC bus (Εικόνα 4.28). Η πλακέτα αυτή μπορεί να χρησιμοποιήσει ένα συνηθισμένο τσιπ ή έναν FPGA για να υλοποιήσει τον επιταχυντή. Αυτό το είδος του συστήματος είναι σχετικά ογκώδες και συχνότερα χρησιμοποιείται για ανάπτυξη ή για πολύ χαμηλής έντασης εφαρμογές. Ένας συνηθισμένος τυπωμένος πίνακας κυκλωμάτων, χρησιμοποιεί είτε ένα FPGA, είτε ένα συνηθισμένο ενσωματωμένο κύκλωμα για τον επιταχυντή. Η κάρτα επέκτασης FPGA απαιτεί περισσότερη σχεδιαστική εργασία από ένα σύστημα βασισμένο σε Η/Υ, αλλά έχει αποτελέσματα σε ένα χαμηλότερου κόστους και αρκετά ποιο δυνατό επεξεργαστικά σύστημα.

Μια εναλλακτική και η πιο πρόσφατη υλοποίηση είναι μια πλατφόρμα FPGA που περιλαμβάνει μια ΚΜΕ και μια δομή FPGA σε ένα ενιαίο τσιπ, όπως η σειρά Zynq7000 της Xilinx (Εικόνα 4.29). Αυτά τα τσιπ είναι ακριβότερα από

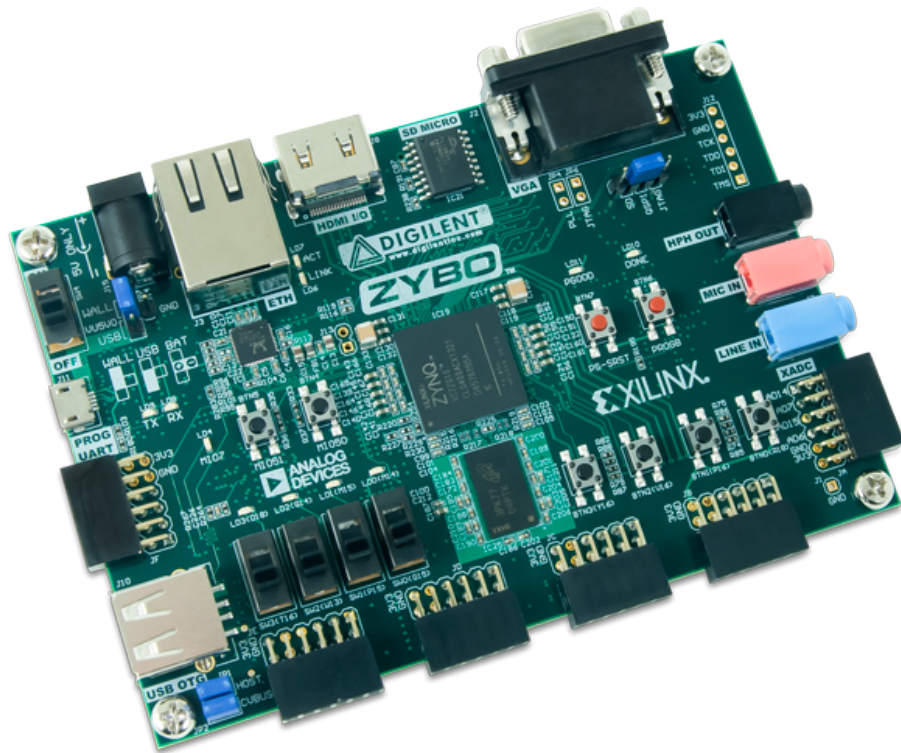
τα συνηθισμένα τσιπ αλλά παρέχουν μια εφαρμογή ενιαίου τσιπ με μία ή περισσότερες CPUs και μια τυπική επαναδιαμορφώσιμη λογική.



Σχήμα 4.28: Ο επιταχυντής σε ένα Ενσωματωμένο σύστημα, μπορεί να είναι μια κάρτα FPGA που συνδέεται στο διάλυτο του συστήματος. Εδώ φαίνεται η Virtex 6 FPGA κάρτα της Xilinx με διεπαφή PCI Express. Εικόνα από xilinx.com.

Ένας επιταχυντής υλικού είναι ένα συνηθισμένο ολοκληρωμένο κύκλωμα, που υλοποιεί μια εξειδικευμένη λειτουργία, με όσο το δυνατόν μικρότερη κατανάλωση ενέργειας και υψηλές επιδόσεις. Πολλά ενσωματωμένα συστήματα σε τσιπ systems-on-chips (SoC) χρησιμοποιούν επιταχυντές για διάφορες λειτουργίες.

Ο συνδυασμός μιας ΚΜΕ με έναν ή περισσότερους επιταχυντές είναι η απλούστερη μορφή ετερογενούς πλατφόρμας ενσωματωμένων συστημάτων. Η ΚΜΕ καλείται συχνά host (ξενιστής). Η ΚΜΕ επικοινωνεί με τον επιταχυντή μέσω των δεδομένων και των καταχωρητών ελέγχου που βρίσκονται μέσα σε αυτόν. Αυτοί οι καταχωρητές επιτρέπουν στην ΚΜΕ να ελέγξουν τη λειτουργία του επιταχυντή και για να του δώσουν εντολές. Η ΚΜΕ και ο επιταχυντής μπορούν επίσης να επικοινωνήσουν μέσω της κοινής μνήμης. Εάν ο επιταχυντής πρέπει να λειτουργήσει σε έναν μεγάλο όγκο δεδομένων, είναι συνήθως αποδοτικότερο να αφήσει τα στοιχεία στη μνήμη και να έχει τον επιταχυντή να διαβάζει και να γράφει μνήμη άμεσα, παρά να έχει μεταφορά των δεδομένων της ΚΜΕ από τη μνήμη στους καταχωρητές του επιταχυντή και αντίστροφα. Η ΚΜΕ και ο επιταχυντής συγχρονίζουν τις ενέργειές τους. Περισσότερες και γενικότερες πλατφόρμες είναι επίσης δυνατές. Μπορούμε να χρησιμοποιήσουμε διάφορες CPUs, σε αντίθεση με τον ενιαίο επεξεργαστή που είναι συνδεδεμένος με τους επιταχυντές. Μπορούμε να χρησιμοποιήσουμε πολλαπλούς διαύλους. Επιπλέον, μπορούμε να δημιουργήσουμε ένα πιο σύνθετο σύστημα μνήμης που



Σχήμα 4.29: Στα ενσωματωμένα συστήματα μπορεί να χρησιμοποιηθεί κάποια αρχιτεκτονική FPGA που περιέχει και έναν ή περισσότερους επεξεργαστές. Εδώ φαίνεται η αναπτυξιακή πλατφόρμα Xilinx Zybo, η οποία περιέχει έναν διπύρηνο επεξεργαστή ARM Cortex-A9 στα 650 Mhz και επαναδιαμορφώσιμη FPGA με 28000 λογικά κελιά.

να παρέχει διαφορετικούς τύπους προσβάσεων στα διαφορετικά μέρη του συστήματος. Ο συ-σχεδιασμός τέτοιων τύπων συστημάτων είναι περισσότερο δύσκολος, ιδιαίτερα όταν δεν κάνουμε τις υποθέσεις για τη δομή της πλατφόρμας.

4.10.5 Πολυεπεξεργασία σε Chip

Πολλοί σχεδιαστές, οι οποίοι είναι σχετικοί με την πληροφορική γενικού σκοπού σχετίζουν την πολυεπεξεργασία μόνο με τους διακομιστές υψηλών επιδόσεων, τις μηχανές βάσεων δεδομένων ή τους σταθμούς εργασίας για επιστημονικούς λόγους και τις διεργασίες heavy-duty ή μετατροπή σε 3D γραφικά ή πρόγνωση του καιρού. Στην πραγματικότητα η πολυεπεξεργασία στα ενσωματωμένα συστήματα είναι πολύ κοινή, παρόλο που συχνά παίρνει πολλές διαφορετικές μορφές. Η πιο επιτυχημένη μορφή της πολυεπεξεργασίας (SMP)

στην οποία το σύστημα εξυπηρετεί πολλούς επεξεργαστές, είναι μέσω μια μορφής μνήμης, η οποία διαμοιράζεται (υπολογιστικά συστήματα διαμοιραζόμενης μνήμης). Στην SMP αρχιτεκτονική, οι διεργασίες χρονοδρομολογούνται ατομικά σε κάθε επεξεργαστή από το λειτουργικό σύστημα, το οποίο είναι συχνά υπεύθυνο για την μετανάστευση των διεργασιών κατά μήκος του επεξεργαστή, για να εξισορροπήσει την υπερχείλιση της μηχανής.

Στα ενσωματωμένα η πιο ευρέως υιοθετημένη μορφή πολυεπεξεργασίας είναι η ετερογενής, στην οποία κατεξοχήν διαφορετικοί επεξεργαστές βρίσκονται στο ίδιο σύστημα, αλλά ο καθένας μπορεί να θεωρηθεί ως ένα υποσύστημα (με μια ιδιωτική μνήμη), το οποίο δεν επηρεάζεται από τα άλλα στοιχεία του προγράμματος. Με άλλα λόγια, διαφορετικά λειτουργικά συστήματα (ή *microkernels*) ελέγχουν τους ατομικούς επεξεργαστές ο καθένας από τους οποίους τρέχει ένα προκαθορισμένο, κατανεμημένο σύνολο διεργασιών. Η επικοινωνία μεταξύ των επεξεργαστών επιτυγχάνεται μέσα από μια καλά καθορισμένη μνήμη η οποία μοιράζεται και χειρίζεται τον συγχρονισμό των μηχανισμών, παρόμοια με τον τρόπο με τον οποίο χειριζόμαστε και άλλες μηχανές DMA. Σε αυτές τις διαμορφώσεις, είναι κοινό να ορίζουμε έναν επεξεργαστή ως τον κύριο επεξεργαστή του συστήματος. Ο κύριος αυτός επεξεργαστής ελέγχει την δραστηριότητα των διεργασιών πάνω στους άλλους επεξεργαστές (σκλάβους), στέλνοντας τις κατάλληλες εντολές ελέγχου στους *microkernels* οι οποίοι τρέχουν πάνω στους σκλάβους.

Μια άλλη κοινή μορφή πολυεπεξεργασίας εκμεταλλεύεται τον έμφυτο παραλληλισμό των δεδομένων στην εφαρμογή. Αν μπορούμε να γράψουμε την εφαρμογή σε μορφή SPMD (το ίδιο πρόγραμμα σε πολλά δεδομένα, *single program multiple data*), μπορούμε να χρησιμοποιήσουμε πολλαπλούς ίδιους επεξεργαστές με ιδιωτικές μνήμες δεδομένων για να επεξεργασθούμε ανεξάρτητα περιοχές με προσωπικά δεδομένα. Μπορούμε να βρούμε παραδείγματα από τις πολλές εφαρμογές SPMD οι οποίες χειρίζονται δισδιάστατες εικόνες (π.χ. ιατρική απεικόνιση) ή άλλες συνηθισμένες ροές δεδομένων (τηλεπικοινωνίες, ήχο και βίντεο). Αυτή η προσέγγιση δεν είναι καινούργια, αλλά χρονολογείται στα μέσα τις δεκαετίας του 1970. Συνεχίστηκε μέχρι και στην δεκαετία του 1980 και συμπεριλάμβανε την εκπληκτική ανάπτυξη των επεξεργαστών οι οποίοι σαφέστατα κατασκευάστηκαν για να καθορίζονται παράλληλα (όπως η *Inmos's Transputers*) και δοκιμάστηκαν για πολύ καιρό πριν εγκαταλειφθούν τελείως.

Συμμετρική Πολυεπεξεργασία

Μέχρι πρόσφατα το SMP ήταν ακριβό και συνήθως ήταν αποτελεσματικό για συγκεκριμένα προβλήματα, και το οποίο το έκανε απαγορευτικά ακριβό για οτιδήποτε άλλο εκτός από ακριβούς σταθμούς υψηλών προδιαγραφών και δια-

κομιστές. Τα τελευταία χρόνια οι επεξεργαστές SMP γίνονται πιο κοινοί και αρκετοί κατασκευαστές ενσωματωμένων συστημάτων προσφέρουν πολυπύρρηνα συστήματα (όπως dual ή quad cores ARMS ή MIPS ή PPC). Με αυτό το σενάριο τα ενσωματωμένα συστήματα γίνονται εμπορεύσιμα σαν τα SMP και τα ενσωματωμένα SMP συστήματα γίνονται οικονομικά πιο βιώσιμα. Η ανάπτυξη του λογισμικού στα ενσωματωμένα πολυπύρρηνα συστήματα είναι μια πρόκληση. Ήδη η βέλτιστη χρήση των παράλληλων διακομιστών που έχουν πολλούς πόρους είναι μια δύσκολη διαδικασία. Τα ΕΣ με τους πολύ λιγότερους πόρους και τις ακόμη περισσότερες απαιτήσεις κάνουν το έργο πολύ πιο δύσκολο, με λιγότερες δυνατότητες. Για παράδειγμα πολλές φορές απαιτείται η συνέπεια της μνήμης στα ΕΣ να υλοποιείται από μηχανισμούς λογισμικού ή υλικού των σχεδιαστών και όχι από κάποιο λειτουργικό σύστημα.

Ετερογενής Πολυεπεξεργασία

Εκτός από τη συμμετρική πολυεπεξεργασία, στα ΕΣ συναντάμε και την ασύμμετρη πολυεπεξεργασία ή ετερογενής πολυεπεξεργασία. Αυτή είναι η πιο κοινή μορφή πολυεπεξεργασίας στα ενσωματωμένα SoCs σήμερα. Η πρώτη ερώτηση η οποία μας έρχεται στο μυαλό είναι η εξής: δεδομένου ότι ο γενικός σκοπός των επεξεργαστών είναι διαθέσιμος, γιατί θα πρέπει να χρησιμοποιήσουμε διαφορετικούς επεξεργαστές για κάθε μια διεργασία; Οι λόγοι είναι: οι εκτιμήσεις του κόστους, της απόδοσης και της προσαρμογής. Το να χρησιμοποιηθούν διαφορετικοί επεξεργαστές για διαφορετικές διεργασίες επιτρέπει στους σχεδιαστές να επιλέξουν την πιο αποτελεσματική ως προς το κόστος εναλλακτική για κάθε στάδιο της εφαρμογής. Για παράδειγμα, το DSP και οι επεξεργαστές γενικού σκοπού έχουν ευδιάκριτα διαφορετικά σύνολα χαρακτηριστικών τα οποία τους κάνουν να ταιριάζουν καλύτερα σε διαφορετικού τύπου φόρτου εργασίας. Σε μια εφαρμογή, η οποία περιλαμβάνει και DSP και συστατικά υπολογισμού γενικού σκοπού, χωρίζοντας τις δύο διεργασίες και χρησιμοποιώντας ετερογενές σύστημα είναι πολύ αποτελεσματικότερο ως προς το κόστος από το να δοκιμάσουμε να επιβάλλουμε την ανάδραση σε ολόκληρη την εφαρμογή σε ένα παραδοσιακό γενικού σκοπού SMP (πολύ ακριβό) ή σε ένα μέρος ενός DSP (πολύ άκαμπτο).

Δυστυχώς, η ετερογενής πολυεπεξεργασία δεν είναι δωρεάν. Το μεγαλύτερο εμπόδιο έρχεται από την απουσία ενός διαισθητικού προτύπου προγραμματισμού. Οι διεργασίες θα πρέπει να είναι ρητά χωρισμένες για τους μεμονωμένους επεξεργαστές, η εσωτερική επικοινωνία και ο συγχρονισμός πρέπει να αντιμετωπιστούν χειροκίνητα, τα εργαλεία και τα δυαδικά συστήματα σύνταξης είναι διαφορετικά, και η αποσφαλμάτωση είναι αδέξια.

Παρά τις εγγενείς δυσκολίες στο πρότυπο προγραμματισμού, διάφορες τάσεις βιομηχανίας ωθούν έντονα στην κατεύθυνση της ετερογενούς πολυεπεξερ-

γασίας. Πολλοί προμηθευτές και πρωτοβουλίες σε βιομηχανικό επίπεδο υποστηρίζουν μερικούς μικροελεγκτές και συνδυασμούς DSP που στοχεύουν σε μερικές από τις πιο ελπιδοφόρες περιοχές αγοράς. Παραδείγματος χάριν, και το PCA της Intel και οι γραμμές προϊόντων της Texas Instruments OMAP ανήκουν σε αυτήν την κατηγορία νέων πλατφορμών SoC. Μια από τις σημαντικότερες κατευθυντήριες δυνάμεις πίσω από την υιοθέτηση αυτών των μορφών αρχιτεκτονικής προέρχεται από τις αντιπαραβαλλόμενες απαιτήσεις. Εκτιμώντας ότι οι εφαρμογές και τα λειτουργικά συστήματα παίρνουν έναν μακροχρόνιο χρόνο να μεταναστεύσουν στις νέες αρχιτεκτονικές, τα νέα προϊόντα ωθούν συνεχώς το σχεδιασμό με περισσότερες απαιτήσεις και ένα γρηγορότερο ποσοστό υιοθέτησης. Ο συνδυασμός ενός συμβατικού πυρήνα μικροεπεξεργαστή (όπως ARM) με έναν αποδοτικά προσανατολισμένο στον προγραμματισμό επιταχυντή (όπως πολλές σύγχρονες μηχανές DSP ή FPGA), υπόσχεται να παρέχει την ευελιξία που χρειάζεται για να αντιμετωπιστούν και τα δύο σύνολα απαιτήσεων. Πρέπει όμως να περάσουν πολλά στάδια μέχρι η ετερογενής πολυεπεξεργασία να γίνει κάτι συνηθισμένο και τυπικό.

4.11 Ανασκόπηση του Κεφαλαίου

Σε αυτό το κεφάλαιο περιγράψαμε τους τρόπους υλοποίησης των ενσωματωμένων συστημάτων. Αφού έγινε μια μικρή ιστορική αναδρομή, αναφερθήκαμε στους μικροελεγκτές, στους επεξεργαστές ειδικών εφαρμογών, σε ολοκληρωμένα κυκλώματα και σε FPGA. Ιδιαίτερα περιγράψαμε τα FPGA και τα στοιχεία που αποτελούνται, αφού είναι η πιο δημοφιλής πλατφόρμα ανάπτυξης πολύπλοκων ενσωματωμένων συστημάτων. Τέλος, αναφερθήκαμε στα συστήματα πάνω στο ίδιο υπόστρωμα πυριτίου και δώσαμε μια τυπική σχεδιαστική ροή.

