Μονάδες επεξεργασίας δεδομένων

για μικροεπεξεργαστές υψηλών επιδόσεων

Διδακτορική διατριβή

Φεβρουάριος 2007

Γιώργος Δημητρακόπουλος

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής
Πανεπιστήμιο Πατρών

Έκδοση 14/2/2006

Περίληψη

Οι μονάδες επεξεργασίας δεδομένων αποτελούν τις βασικές δομικές μονάδες όλων των μικροεπεξεργαστών. Κάποια από τα κυκλώματα αυτής της κατηγορίας υλοποιούν τις βασικές αριθμητικές πράξεις πάνω σε δεδομένα τόσο σταθεράς όσο και κινητής υποδιαστολής, ενώ κάποια άλλα αναλαμβάνουν την αναδιοργάνωση των δεδομένων αυτών για την επιτάχυνση του υπολογισμού. Σε επεξεργαστές ειδικού σκοπού, όπως οι επεξεργαστές πολυμέσων και γραφικών, οι μονάδες επεξεργασίας δεδομένων καταλαμβάνουν περισσότερο από το 30% του ολοκληρωμένου και η αποτελεσματική σχεδίαση τους έχει άμεσο αντίκτυπο στην απόδοση ολόκληρου του συστήματος. Στο μέλλον, αναμένεται πως ακόμα και οι επεξεργαστές γενικού σκοπού, θα είναι εξοπλισμένοι από εξειδικευμένους επιταχυντές, οι οποίοι θα εκτελούν απ' ευθείας σε υλικό σύνθετους αλγορίθμους με μεγάλες υπολογιστικές απαιτήσεις.

Η βάση όλων των προτεινόμενων λύσεων σ' αυτή τη διατριβή είναι η αναλυτική εύρεση ενός εγγενώς απλούστερου αλγορίθμου, ο οποίος θα επιτρέπει την αποτελεσματική υλοποίηση των αντίστοιχων κυκλωμάτων ανεξάρτητα από την τεχνολογία που θα χρησιμοποιηθεί και από τους επιπλέον περιορισμούς που τυχόν θα επιβληθούν στο μέλλον κατά την κατασκευή των κυκλωμάτων αυτών. Η ανάλυση και τα πειραματικά αποτελέσματα που συλλέξαμε βασίζονται τόσο σε υλοποιήσεις σε επίπεδο τρανζίστορ, που είναι η κύρια μέχρι τώρα πρακτική σχεδίασης των μικροεπεξεργαστών υψηλών επιδόσεων, όσο και σε πλήρως αυτοματοποιημένες υλοποιήσεις. Φυσικά, στη δεύτερη περίπτωση η απόδοση των κυκλωμάτων επιβαρύνεται, τόσο σε καθυστέρηση όσο και σε ενέργεια, εξαιτίας των περιορισμών των αυτοματοποιημένων εργαλείων και την αναγκαστική χρήση των προσχεδιασμένων βιβλιοθηκών βασικών πυλών. Η μελέτη που πραγματοποιήσαμε στοχεύει στην πλήρη εξερεύνηση του χώρου λύσεων των κυκλωμάτων αυτών. Η ανάλυση της συμπεριφοράς τους πραγματοποιήθηκε χρησιμοποιώντας τις βέλτιστες καμπύλες της ενέργειας ως προς την καθυστέρηση, οι οποίες αποτελούν τον πιο έγκυρο τρόπο περιγραφής της απόδοσης ενός κυκλώματος.

Τα κυκλώματα που παρουσιάζονται ανήκουν σε τρεις βασικές κατηγορίες. Στην πρώτη ανήκουν οι αθροιστές παράλληλου προθέματος, που χρησιμοποιούν τα κρατούμενα του Ling για την υλοποίηση της δυαδικής πρόσθεσης. Τα κρατούμενα που προτάθηκαν από τον Ling αποτελούν απλοποιημένες μορφές των κλασσικών σχέσεων πρόβλεψης κρατουμένου και χρησιμοποιούνται αυτή τη στιγμή στην πλειοψηφία των εμπορικών επεξεργαστών. Το νέο κύκλωμα, που προτείναμε, αποτελεί ουσιαστικά τη γενίκευση των σχέσεων αυτών, επιτρέποντας την υλοποίηση τους με απλοποιημένες δομές παράλληλου προθέματος, με αποτέλεσμα τη μείωση τόσο της καθυστέρησης όσο και της απαιτούμενης ενέργειας. Η νέα τεχνική οδηγεί σε γρηγορότερα κυκλώματα ανεξάρτητα από τη λογική οικογένεια που θα χρησιμοποιηθεί (στατική ή δυναμική CMOS) και το δένδρο παράλληλου προθέματος που θα επιλεγεί.

Η δεύτερη κατηγορία αναφέρεται σε κυκλώματα αναδιάταξης των δεδομένων που είναι αποθηκευμένα μέσα στους καταχωρητές του επεξεργαστή. Η αποδοτική αναδιάταξη των δεδομένων καταλήγει να είναι σε πολλούς αλγορίθμους (κρυπτογραφία, ψηφιακή επεξεργασία

σήματος, πολυμέσα) τόσο αναγκαία όσο και η γρήγορη υλοποίηση των βασικών αριθμητικών πράξεων, αλλά και η ταχεία επικοινωνία με τη μνήμη. Η προσπάθεια μας εστιάστηκε στην αποδοτική υλοποίηση μιας γενικής εντολής αναδιάταξης δεδομένων, στοχεύοντας σε όσο το δυνατόν ταχύτερες υλοποιήσεις. Όλες οι εκδοχές που προτείναμε στηρίζονται σε μια νέα μορφή δικτύων ταξινόμησης, η οποία μας επιτρέπει να παρέχουμε λύσεις που είναι σημαντικά πιο αποδοτικές σε σχέση με τις ήδη υπάρχουσες. Τα κυκλώματα που προτείνουμε κατασκευάζονται με τη χρήση ενός μόνο κελιού υπολογισμού (διαφορετικό για κάθε δίκτυο ταξινόμησης) και διατηρούν μια πλήρως κανονική δομή. Το στοιχείο αυτό, συμβάλλει, πέρα από τη βελτίωση της απόδοσης, στην αποτελεσματικότερη χωροθέτηση του κυκλώματος και στη μείωση των αρνητικών επιδράσεων των γραμμών διασύνδεσης.

Η τελευταία κατηγορία κυκλωμάτων αναφέρεται σε κυκλώματα που χρησιμοποιούνται για την υλοποίηση της πρόσθεσης αριθμών κινητής υποδιαστολής. Τα κυκλώματα που προτείνουμε χρησιμοποιούνται στα πιο κρίσιμα στάδια, από πλευράς καθυστέρησης, του υπολογισμού του αθροίσματος και αφορούν στην πρόσθεση των μεγεθών και στην κανονικοποίηση του αποτελέσματος. Αρχικά, περιγράφουμε μια εναλλακτική προσέγγιση για την υλοποίηση των αθροιστών μεγέθους των αριθμών κινητής υποδιαστολής. Οι νέες μονάδες εκμεταλλεύονται την αναπαράσταση συμπληρώματος ως προς ένα και τις γρήγορες μονάδες υπολογισμού του κρατουμένου, που βασίζονται στην τεχνική παράλληλου προθέματος. Προτείνουμε μια ενοποιημένη μεθοδολογία για το πως μπορούμε να παράγουμε δομές παράλληλου προθέματος ανεξάρτητα από το μέγεθος της λέξης εισόδου, ενώ καταφέρνουμε να ενώσουμε για πρώτη φορά τις απλοποιημένες σχέσεις κρατουμένου του Ling με την πρόσθεση αριθμών που ακολουθούν την αναπαράσταση συμπληρώματος ως προς ένα. Στη συνέχεια, περιγράφεται ένας νέος απλός τρόπος για την υλοποίηση της πρόβλεψης και της μέτρησης των προπορευόμενων μηδενικών που εμφανίζονται στα αποτελέσματα των πράξεων αριθμών κινητής υποδιαστολής. Με τη χρήση των νέων κυκλωμάτων η κανονικοποίηση του αποτελέσματος μπορεί να πραγματοποιηθεί σε λιγότερο χρόνο και με σημαντικά μικρότερη ενέργεια.

Ευχαριστίες

Πολλοί είναι οι άνθρωποι που συνέβαλλαν άμεσα ή έμμεσα στην ολοκλήρωση των μεταπτυχιακών μου σπουδών και την παρουσίαση αυτής της διδακτορικής διατριβής.

Καταρχήν θα ήθελα να ευχαριστήσω τον Καθ. Δημήτρη Νικολό για την ελευθερία που μου προσέφερε να ασχοληθώ με διάφορα θέματα που κατά καιρούς με απασχολούσαν ακολουθώντας τα μάλλον εύπλαστα ερευνητικά μου ενδιαφέροντα. Παρόλες τις αντιρρήσεις που δικαίως κατά καιρούς εξέφραζε, θεωρώ πως ουσιαστικά συνέβαλλε στο να με παρακινήσει ακόμη περισσότερο. Άλλωστε, είναι γνωστός σε όλα τα μέλη (“παιδιά”) του εργαστηρίου ο ενθουσιασμός του όταν άκουγε για πρώτη φορά μια πρώιμη ιδέα που φαινόταν πως θα οδηγήσει σε κάποιο αποτέλεσμα. Χωρίς την πίεση του και τη συνεχή παρουσία του για οτιδήποτε και αν με απασχολούσε δεν νομίζω πως θα είχα καταφέρει να ολοκληρώσω την εργασία αυτή.

Επίσης θα ήθελα να ευχαριστήσω τα μέλη της επταμελούς επιτροπής, Καθ. Γιώργο Αλεξίου, Καθ. Κωνσταντίνο Γκούτη, Καθ. Θάνο Στουραϊτή, Καθ. Γιώργο Τσιατούχα, και τους Καθ. Χάρη Βέργο και Καθ. Βασίλη Παλιουρά. Με τους δύο τελευταίους, Χάρη και Βασίλη (ας μου επιτρέψουν τον ενικό), πέρα από τη συνεργασία που είχαμε κατά τη διάρκεια των μεταπτυχιακών μου σπουδών, μας συνδέει και μια πιο στενή φιλική σχέση η οποία με τιμά ιδιαίτερα.

Φυσικά, η πενταετής παραμονή μου στο “γραφείο” δεν θα ήταν η ίδια χωρίς τον φίλο πια Μανώλη Καλλίγερο. Νομίζω πως χωρίς τις ατέρμονες συζητήσεις που είχαμε και την παρέα που κάναμε η καθημερινή ζωή στο πανεπιστήμιο θα ήταν ιδιαίτερος βαρετή. Φυσικά δεν ξεχνώ την Ειρήνη Καρύμπαλη η οποία φρόντιζε να μας επαναφέρει όταν η κουβέντα ξέφευγε από τα λογικά πλαίσια της διάρκειας μιας συζήτησης, αν και ομολογουμένως τις περισσότερες φορές έριχνε περισσότερο λάδι στη φωτιά. Παιδιά σας ευχαριστώ πολύ και θαυμάζω το κουράγιο σας ιδιαίτερα για τις στιγμές που ακούγατε με περισσή σοβαρότητα την γκρίνια μου. Επίσης, θα ήθελα να ευχαριστήσω το Γιώργο Σιδηρόπουλο και τον Ματσαί Μπέλλο (για μας Μαθιός) για την παρέα που κάναμε και για την πολύ ευχάριστη συγκατοίκηση στο γραφείο.

Η εργασία αυτή μπορεί να μην είχε ολοκληρωθεί αν δεν είχα την τύχη να συνεργαστώ με τον Κώστα Γαλανόπουλο και το Χρήστο Μαυροκεφαλίδη. Ευχαριστώ και τους δύο για τη μεγάλη τους όρεξη για δουλειά και που ανέχτηκαν τις παραξενιές μου κατά τη διάρκεια της συνεργασίας μας. Θεωρώ πως η σχέση που αναπτύξαμε θα ευδοκιμήσει στο μέλλον και προσδοκώ να λύσουμε παρέα πολλά από τα νέα ερωτήματα που προέκυψαν κατά τη διάρκεια αυτής της ερευνητικής προσπάθειας. Ευχαριστώ επίσης το Δημήτρη Αμπελιώτη που συνέβαλλε ουσιαστικά στη στοιχειοθέτηση του κειμένου.

Ουσιαστική ήταν επίσης η οικονομική βοήθεια που μου προσέφερε το Ερευνητικό Ακαδημαϊκό Ινστιτούτο Τεχνολογίας Υπολογιστών μέσω της μεταπτυχιακής υποτροφίας στη μνήμη του Καθ. Δ. Μαρίτσα και το έργο Πυθαγόρας στο οποίο συμμετείχα.

Φυσικά όλη αυτή η συζήτηση δεν θα είχε κανένα νόημα αν δεν είχα όλα αυτά τα χρόνια την αμέριστη συμπαράσταση της οικογένειάς μου. Στον Αλέκο και στην Καίτη οφείλω το μεγαλύτερο μέρος του ευατού μου και στον τρόπο που με έμαθαν να αντιμετωπίζω τη ζωή. Σας ευχαριστώ

βέβαια περισσότερο για όλα αυτά που δεν γράφονται σε μια διδακτορική διατριβή. Εσείς ξέρετε. Τον αδερφό μου Στέφανο πρέπει να τον ευχαριστήσω για πολλά. Κυρίως όμως τον χαίρομαι γι' αυτό που είναι σαν άνθρωπος και τον ευχαριστώ για τη βοήθεια που μου προσέφερε κάθε φορά που είχα ανάγκη.

Κλείνοντας, θέλω να ευχαριστήσω τη σύντροφο μου Μαρίνα που καταφέρνει με ένα χαμόγελο να κάνει τη ζωή να μοιάζει πιο όμορφη ακόμα και όταν όλα γύρω δείχνουν μουντά. Το καλό είναι πως αυτή τη μαγική ιδιότητα έχει αρχίσει να την αποκτάει και ο μικρός Αλέκος που ήρθε πρόσφατα στην παρέα μας.

Περιεχόμενα

1	Εισαγωγή	1
1.1	Η επίδραση της τεχνολογίας	2
1.2	Υποστήριξη νέων αρχιτεκτονικών	5
1.3	Γενικές σχεδιαστικές πρακτικές	6
1.3.1	Επίπεδο μικροαρχιτεκτονικής	7
1.3.2	Λογικό επίπεδο	8
1.3.3	Επίπεδο κυκλώματος	10
1.3.4	Σχεδιαστικοί περιορισμοί	15
1.4	Οργάνωση διατριβής	16
2	Ακολουθούμενη ροή σχεδιασμού	19
2.1	Μέτρηση της απόδοσης των κυκλωμάτων	20
2.2	Γεωμετρικός προγραμματισμός	23
2.3	Μοντέλα καθυστέρησης	25
2.3.1	Η επίδραση των γραμμών διασύνδεσης	31
2.4	Μοντελοποίηση ενέργειας και εμβαδού	32
2.5	Αναγωγή σε πρόβλημα γεωμετρικού προγραμματισμού	34
2.6	Σύνοψη της διαδικασίας βελτιστοποίησης	35
3	Αθροιστές παράλληλου προθέματος	39
3.1	Αθροιστές παράλληλου προθέματος - Βασικές αρχές	40
3.2	Τα κρατούμενα του Ling	45
3.2.1	Υπολογισμός του αθροίσματος	51
3.3	Δομές παράλληλου προθέματος και επιλογής κρατουμένου	52
3.4	Πειραματικά αποτελέσματα	55
4	Αναδιοργάνωση δεδομένων	59
4.1	Εισαγωγή	59
4.2	Εντολές αναδιοργάνωσης δεδομένων	61
4.3	Η εντολή ομαδοποίησης GRP	64
4.4	Προηγούμενες υλοποιήσεις	66
4.5	Βασική ιδέα	70
4.5.1	Έλεγχος διατήρησης σημαντικότητας	71
4.5.2	Η εκτέλεση της GRP μέσω επαυξημένων δικτύων ταξινόμησης	72
4.6	Επαυξημένο bitonic δίκτυο ταξινόμησης	72

4.6.1 Η σχεδίαση του υποδικτύου EBS	74
4.6.2 Αλλαγή φοράς ταξινόμησης	77
4.7 Επαυξημένο merge δίκτυο ταξινόμησης	77
4.7.1 Η σχεδίαση του υποδικτύου EMS	78
4.8 Επαυξημένο γραμμικό δίκτυο αντιμετάθεσης	81
4.8.1 Μια ταχύτερη εκδοχή	81
4.9 Πρακτικές περιπτώσεις	85
4.10 Πειραματικά αποτελέσματα	87
5 Κυκλώματα μονάδων κινητής υποδιαστολής	93
5.1 Πρόσθεση αριθμών κινητής υποδιαστολής	94
6 Αθροιστές μεγεθών	97
6.1 Προηγούμενες υλοποιήσεις	98
6.2 Βασική θεωρία και επεκτάσεις	99
6.3 Σχεδιαστική μεθοδολογία	101
6.3.1 Μείωση σχεδιαστικών επιλογών	103
6.3.2 Διαδικασία σχεδίασης	105
6.4 Υιοθέτηση του κρατούμενου του Ling	106
6.5 Αραιές δομές	108
6.6 Πειραματικά αποτελέσματα	109
7 Μονάδα Κανονικοποίησης	113
7.1 Μέτρηση προπορευόμενων μηδενικών - Σύντομη ανασκόπηση	115
7.2 Προτεινόμενες μονάδες μέτρησης προπορευόμενων μηδενικών	118
7.2.1 Η οργάνωση του νέου κυκλώματος μέτρησης	121
7.3 Πρόβλεψη της θέσης του προπορευόμενου ψηφίου	124
7.3.1 Οι προτεινόμενες μονάδες πρόβλεψης	127
7.3.2 Αντιμετώπιση του λάθους πρόβλεψης	128
7.4 Βελτιστοποίηση της ενέργειας και της καθυστέρησης	130
8 Σύνοψη και συμπεράσματα	137

Εισαγωγή

Το θέμα της διδακτορικής μου διατριβής εστιάζεται στην ανάλυση και τη σχεδίαση νέων κυκλωμάτων επεξεργασίας δεδομένων, τα οποία αποτελούν τις βασικές δομικές μονάδες κάθε επεξεργαστή. Με τη χρήση του όρου μονάδες επεξεργασίας δεδομένων, αναφερόμαστε στα κυκλώματα εκείνα τα οποία υλοποιούν βασικές αριθμητικές πράξεις πάνω σε δεδομένα σταθερής και κινητής υποδιαστολής, αλλά και στα κυκλώματα τα οποία αναλαμβάνουν την αναδιοργάνωση των δεδομένων αυτών για την επιτάχυνση του υπολογισμού. Η σχεδίαση των κυκλωμάτων αυτών εξελίσσεται συνεχώς εξαιτίας της προόδου της αρχιτεκτονικής των επεξεργαστών και της ωρίμανσης των τεχνικών υλοποίησης τους σε επίπεδο κυκλώματος. Η ανάγκη για μεγαλύτερη απόδοση οδήγησε στην αύξηση της συχνότητας λειτουργίας των μικροεπεξεργαστών και στην παράλληλη εκτέλεση όσο το δυνατόν περισσότερων εντολών. Κατ' επέκταση αποτέλεσμα αυτής της προόδου είναι η ραγδαία αύξηση της πολυπλοκότητας των κυκλωμάτων, η οποία αναπόφευκτα γεννάει μια πληθώρα άλλων προβλημάτων, με κύριο αυτό της αυξημένης κατανάλωσης ισχύος [70], [66]. Παράλληλα, οι νέες τεχνολογίες υλοποίησης σε πυρίτιο δε συμβάλλουν πλέον στην εξομάλυνση του προβλήματος και απαιτούν την εύρεση νέων σχεδιαστικών λύσεων ακόμα και για γνωστά κυκλώματα που βρίσκονται ήδη σε ευρεία χρήση [118].

Επιπλέον, νέες εφαρμογές έρχονται να προστεθούν στον φόρτο εργασίας των μικροεπεξεργαστών. Η αποδοτική επεξεργασία εικόνων, βίντεο, ήχου και γραφικών απαιτεί νέες προσεγγίσεις τόσο σε αρχιτεκτονικό επίπεδο όσο και σε επίπεδο υλοποίησης. Νέες εντολές προστίθενται στους επεξεργαστές και νέες εξειδικευμένες μονάδες επεξεργασίας δεδομένων σχεδιάζονται, ώστε να επιταχύνουν αποτελεσματικά την εκτέλεση των εφαρμογών αυτών [34], [6], [101].

Είναι εύλογο λοιπόν, ότι τόσο οι πολύ μικρές τεχνολογίες κατασκευής όσο και οι νέες αρχιτεκτονικές, γεννούν νέες σχεδιαστικές προκλήσεις για τις μονάδες επεξεργασίας δεδομένων και απαιτούν πρακτικές και βιώσιμες λύσεις. Στη συνέχεια, θα αναλύσουμε εν συντομία τις

δυσκολίες που πηγάζουν τόσο από τις νέες τεχνολογίες κατασκευής όσο και από τις νέες εφαρμογές και τον τρόπο που αυτές επηρεάζουν τη σχεδίαση των μονάδων επεξεργασίας δεδομένων.

1.1 Η επίδραση της τεχνολογίας

Η βελτίωση των χαρακτηριστικών των τεχνολογιών κατασκευής και η σμίκρυνση των μεγεθών των τρανζίστορ (transistor scaling) είναι αυτή που οδηγούσε επί τρεις δεκαετίες την εξέλιξη όλων των ολοκληρωμένων κυκλωμάτων και επέτρεψε τη σχεδίαση αποδοτικών μικροεπεξεργαστών που λειτουργούσαν σε συχνότητες της τάξης του GHz. Οι ιδιότητες των πολύ μικρών τεχνολογιών κατασκευής (VDSM, Very Deep Sub Micron) με ελάχιστα επιτρεπόμενα μεγέθη τρανζίστορ μικρότερα των 90nm, έχουν σαν αποτέλεσμα, από τη μια πλευρά σχεδιαστικές τεχνικές οι οποίες κυριαρχούσαν για χρόνια, όπως η δυναμική λογική, να εφαρμόζονται πλέον με δυσκολία και μόνο σε ορισμένες περιπτώσεις, ενώ από την άλλη οι ιδιότητες αυτές γεννούν νέα προβλήματα τα οποία απαιτούν δραστικές λύσεις, ώστε τα νέα κυκλώματα να είναι βιώσιμα και αποτελεσματικά.

Αυτό που παραδοσιακά χαρακτηρίζει τις μονάδες επεξεργασίας δεδομένων είναι η ταχύτητα τους. Η υψηλή ταχύτητα επιτυγχάνονταν μέσα από δύο βασικές οδούς. Αφ' ενός οι σχεδιαστές χρησιμοποιούσαν στο έπακρο τις δυνατότητες της διαθέσιμης τεχνολογίας και αφ' ετέρου εφαρμόζαν έξυπνες τεχνικές οι οποίες συνέβαλαν στη μείωση της καθυστέρησης των κυκλωμάτων [24]. Οι τεχνικές αυτές μέχρι και σήμερα συνοψίζονται σε τέσσερις σχεδιαστικές προσεγγίσεις, τις οποίες θα περιγράψουμε στη συνέχεια αναλύοντας τα πλεονεκτήματα και τα μειονεκτήματά τους [1]. Οι δύο πρώτες δημιουργούσαν από παλιά πολλά επιμέρους προβλήματα, τα οποία ήταν όμως ανεκτά χάριν της υψηλής απόδοσης που προσέφεραν¹, ενώ η χρήση τους στο μέλλον πρέπει να επαναπροσδιοριστεί. Οι δύο επόμενες είναι βέβαιο πως θα χρησιμοποιηθούν για πολλά χρόνια ακόμη προσαρμοσμένες φυσικά στους περιορισμούς των μελλοντικών τεχνολογιών.

Την πρώτη σχεδιαστική προσέγγιση χαρακτηρίζει η σχεδόν αποκλειστική χρήση της δυναμικής CMOS λογικής για την υλοποίηση των κυκλωμάτων αντί της στατικής CMOS² [119], [56]. Η τακτική αυτή έχει σαν αποτέλεσμα τη μείωση της καθυστέρησης του κρίσιμου μονοπατιού από 30% έως 50% σε σχέση με τις στατικές υλοποιήσεις με άμεσο αποτέλεσμα την περίπου ίδια αύξηση της κατανάλωσης ισχύος του κυκλώματος. Η αύξηση της κατανάλωσης ισχύος είναι απόρροια του πολύ μεγάλου αριθμού μεταβάσεων που εμφανίζονται στους εσωτερικούς κόμβους του κυκλώματος, της αναγκαστικής χρήσης του σήματος ρολογιού για την “ενορχήστρωση” της δυναμικής λειτουργίας των πυλών και των αυξημένων ρευμάτων διαρροής εξαιτίας των παράλληλων διασυνδέσεων των τρανζίστορ. Άμεση συνέπεια της αυξημένης κατανάλωσης ισχύος και της σχεδόν 100% χρησιμοποίησης των κυκλωμάτων αυτών σε κάθε κύκλο ρολογιού του επεξεργαστή, είναι η πολύ μεγάλη άνοδος της τοπικής θερμοκρασίας (thermal hotspot) [170]. Η αύξηση της θερμοκρασίας γεννά με τη σειρά της πολλά προβλήματα για την αξιόπιστη λειτουργία του κυκλώματος και τη φθορά του στο χρόνο. Η χρήση της

¹ Δεν πρέπει να ξεχνάμε πως η ταχύτητα λειτουργίας των επεξεργαστών αποτέλεσε επί χρόνια τον κύριο τρόπο διαφήμισης τους και το κύριο μέσο αύξησης των πωλήσεων τους ιδίως στην αγορά των προσωπικών υπολογιστών.

² Πέρα από τις μονάδες επεξεργασίας δεδομένων αυτή την πρακτική ακολουθεί εδώ και χρόνια η σχεδίαση των αρχείων καταχωρητών (register files) των μικροεπεξεργαστών, τα οποία υποστηρίζουν πολλές παράλληλες πόρτες ανάγνωσης και εγγραφής [73].

δυναμικής λογικής στο μέλλον τίθεται υπο μεγάλη αμφισβήτηση. Πέραν της αυξημένης κατανάλωσης ισχύος, η χρήση της δυναμικής λογικής απαιτεί επιπλέον μεγάλη προσπάθεια ώστε να επιβεβαιωθεί η ορθότητα του σχεδιασμού στις διάφορες συνθήκες λειτουργίας του κυκλώματος. Στις μέρες μας η ευαισθησία της δυναμικής λογικής γίνεται μεγαλύτερη εξαιτίας των αυξημένων τυχαίων μεταβολών των ιδιοτήτων των τρανζίστορ που οφείλονται στην κατασκευή των ολοκληρωμένων (process variations). Πολλές είναι οι λύσεις που έχουν προταθεί με σκοπό την αντιμετώπιση αυτών των προβλημάτων, μεταξύ των οποίων η τεχνική του Limited Switch Dynamic Logic (LSDL) [11] είναι από τις πιο ενδιαφέρουσες. Στη γενική περίπτωση θεωρούμε πως η στατική λογική CMOS θα χρησιμοποιείται για τη σχεδίαση της πλειονότητας των ψηφιακών κυκλωμάτων ενώ η δυναμική λογική θα εφαρμόζεται μόνο σε περιπτώσεις επεξεργαστών που θα αποζητούν πολύ υψηλές ταχύτητες λειτουργίας και το όφελος σε ταχύτητα θα ισοσκελίζει τα μειονεκτήματα που ήδη αναφέραμε. Αυτή η πρακτική ήδη άρχισε να εφαρμόζεται και στις μέρες μας, όπου η δυναμική λογική χρησιμοποιείται για να αντικαταστήσει τοπικά αργές στατικές πύλες ενώ συνδέεται αμέσως με το υπόλοιπο στατικό κύκλωμα μέσω μονάδων μετατροπής της δυναμικής λειτουργίας σε στατική (dynamic to static interface) [155].

Η δεύτερη σχεδιαστική προσέγγιση αφορούσε μέχρι τώρα στη σχεδίαση των πυλών χρησιμοποιώντας τρανζίστορ μεγάλου εύρους (transistor width W) σε συνδιασμό με το ελάχιστο δυνατό μήκος καναλιού³ (transistor length L) και με την ελάχιστη δυνατή τάση κατωφλίου V_{th} με σκοπό τη μεγιστοποίηση της ταχύτητας [118]. Η τεχνική αυτή έχει σαν αποτέλεσμα οι πύλες να παράγουν στην έξοδο τους το μέγιστο δυνατό ρεύμα φόρτισης ή αποφόρτισης των χωρητικοτήτων, επιταχύνοντας τον υπολογισμό στον μέγιστο δυνατό βαθμό. Πιο συγκεκριμένα, το ρεύμα I το οποίο μπορεί να διατρέξει ένα τρανζίστορ κατά τη διάρκεια μιας μετάβασης της εισόδου περιγράφεται χονδρικά από τη σχέση

$$I = k \frac{W}{L} (V_{dd} - V_{th})^a$$

όπου οι παράμετροι k και a καθορίζονται μετά από μετρήσεις ενώ η τιμή της τάσης τροφοδοσίας V_{dd} θεωρούμε ότι καθορίζεται από τη διαθέσιμη τεχνολογία [138]. Επιλέγοντας σωστά τις τιμές των παραμέτρων (W , L) και διαλέγοντας τρανζίστορ χαμηλού κατωφλίου μπορούμε να επιταχύνουμε τη λειτουργία αφού ο χρόνος που απαιτείται για την φόρτιση ή την αποφόρτιση μιας χωρητικότητας C για μια διαφορά τάσης ΔV είναι ανάλογος του $\frac{C\Delta V}{I}$. Παράλληλα, με αυτόν τον τρόπο οι παρασιτικές χωρητικότητες των μεγάλων γραμμών διασύνδεσης αντιμετωπίζονται αποδοτικά, μειώνοντας έτσι την αρνητική τους επίδραση στην ταχύτητα του κυκλώματος [64]. Φυσικά, αυτός ο τρόπος σχεδίασης οδηγεί σε σημαντική αύξηση της κατανάλωσης ισχύος του κυκλώματος τόσο της δυναμικής όσο και της στατικής που προέρχεται από τα ρεύματα διαρροής (leakage current). Ιδίως η χρήση τρανζίστορ χαμηλού κατωφλίου επιφέρει εκθετική αύξηση των ρευμάτων διαρροής επηρεάζοντας σημαντικά την κατανάλωση ισχύος του κυκλώματος ακόμα και όταν αυτό είναι ανενεργό [120]. Ο σχεδιαστής για ένα δεδομένο κύκλωμα έχει μεγάλη ελευθερία να επιλέξει τις τιμές του εύρους των τρανζίστορ ενώ είναι σχετικά περιορισμένος όσον αφορά το μήκος του καναλιού των τρανζίστορ, την τάση

³Στις μέρες μας σε πολλές περιπτώσεις χρησιμοποιούνται τρανζίστορ με 10% μεγαλύτερο από το ελάχιστο μήκος καναλιού, με σκοπό τη μείωση των ρευμάτων διαρροής [137]. Η πρακτική αυτή εφαρμόζεται όμως ελάχιστα στις μονάδες επεξεργασίας δεδομένων οι οποίες λειτουργούν κάτω από αυστηρούς χρονικούς περιορισμούς (time critical).

κατωφλίου τους και την τάση τροφοδοσίας. Για παράδειγμα, συνήθως ο σχεδιαστής έχει στη διάθεση του τρανζίστορ με δύο ή τρία διαφορετικά κατώφλια τα οποία συμβολίζονται ως High-Vt, Standard-Vt και Low-Vt αντίστοιχα, ενώ για να δημιουργήσει ενδιάμεσες τιμές του κατωφλίου απαιτούνται επιπλέον κυκλώματα πόλωσης του υποστρώματος (substrate biasing) και τεχνολογίες με ανεξάρτητα πηγάδια (wells). Επίσης, η διαμοίραση διαφορετικών τάσεων τροφοδοσίας σε γειτονικές πύλες εμπεριέχει πολλές δυσκολίες και είναι δύσκολο να εφαρμοστεί, όταν ο αριθμός των διακριτών τιμών της τάσης τροφοδοσίας ξεπερνάει τις 2 [159], [146], [89], ενώ ταυτόχρονα απαιτεί επιπλέον σχεδιαστική προσπάθεια στα κυκλώματα μετατροπής της τάσης [76]. Στο μέλλον οι τιμές των παραμέτρων αυτών πρέπει να καθοριστούν με τέτοιο τρόπο ώστε να ικανοποιούν ταυτόχρονα πολλούς άλλους σχεδιαστικούς στόχους πέρα από τη μικρότερη δυνατή καθυστέρηση [118], [70], όπως την ελαχιστοποίηση της κατανάλωσης ισχύος και τη μείωση της επίδρασης των τυχαίων μεταβολών που εμφανίζονται μετά την κατασκευή. Ο τρόπος επιλογής των κατάλληλων τιμών είναι πλέον πολύ σύνθετος και ο σχεδιαστής είναι αναγκασμένος να βασιστεί σε αυτόματα εργαλεία βελτιστοποίησης [15], [181], [25].

Η επόμενη προσέγγιση αφορά στην πλήρη αξιοποίηση της ομοιομορφίας των μονάδων επεξεργασίας δεδομένων κατά την τελική χωροθέτηση τους στο ολοκληρωμένο [121]. Είναι γνωστό πως για τις μονάδες επεξεργασίας δεδομένων εφαρμόζεται μια ειδική τεχνική χωροθέτησης των πυλών του κυκλώματος, που καλείται datapath layout με αποτέλεσμα το κύκλωμα να χωρίζεται σε τόσες στήλες ή γραμμές (bit-slices) όσες και τα δυαδικά ψηφία των δεδομένων. Οι πύλες της ίδιας στήλης συνδέονται με κάθετες γραμμές διασύνδεσης μικρού μήκους ενώ οι πύλες που ενώνουν ενδιάμεσα αποτελέσματα μεταξύ διαφορετικών στηλών ενώνονται με οριζόντιες γραμμές διασύνδεσης. Με αυτόν τον τρόπο οι γραμμές διασύνδεσης έχουν μια ομοιόμορφη δομή και επιβαρύνουν το λιγότερο δυνατό την καθυστέρηση του κυκλώματος, ενώ ταυτόχρονα μειώνεται σημαντικά το απαιτούμενο εμβαδόν χωροθέτησης όλου του κυκλώματος. Η τεχνική αυτή δεν χρησιμοποιείται μεμονωμένα για μια μονάδα επεξεργασίας δεδομένων του μικροεπεξεργαστή, αλλά εφαρμόζεται ενιαία σε όλα τα κυκλώματα που συμμετέχουν στην αντίστοιχη μονάδα εκτέλεσης (Integer unit, floating-point unit, media processing unit). Βασική προϋπόθεση αυτής της τεχνικής είναι η χωροθέτηση να πραγματοποιηθεί είτε από τον ίδιο το σχεδιαστή είτε από κάποιο εξειδικευμένο datapath layout εργαλείο. Γενικά εργαλεία χωροθέτησης δεν είναι σε θέση να πετύχουν το επιθυμητό αποτέλεσμα, γιατί αγνοούν πλήρως την εγγενή ομοιομορφία των κυκλωμάτων [26]. Και σ' αυτήν την περίπτωση οι νέες τεχνολογίες κατασκευής αναμένεται να αλλάξουν λίγο τα δεδομένα [7]. Αφ' ενός η κοντινή τοποθέτηση γειτονικών καλωδίων δημιουργεί μεγάλη αύξηση της χωρητικότητας που οι πύλες χρειάζεται να οδηγήσουν ενώ τα σήματα αλλοιώνονται στο χρόνο εξαιτίας του crosstalk [64], [165]. Αφ' ετέρου, οι μεγάλες διακυμάνσεις που εμφανίζουν οι ιδιότητες των τρανζίστορ λόγω της κατασκευής επιβάλλουν συγκεκριμένους κανόνες στην χωροθέτηση των πυλών [128], [130], όπως για παράδειγμα ο ενιαίος προσανατολισμός της πολυσιλικόνης της πύλης (single poly direction) σε όλα τα τρανζίστορ της ίδιας περιοχής, ο οποίος ήδη εφαρμόζεται στα κυκλώματα στατικής RAM [90].

Εν τέλει, η τέταρτη σχεδιαστική πρακτική αναφέρεται στο διαχωρισμό του υπολογισμού σε επιμέρους pipeline στάδια. Η τεχνική του pipeline είναι πολύ αποτελεσματική για την αύξηση της ταχύτητας λειτουργίας των κυκλωμάτων αλλά και της επικάλυψης των διαφόρων σταδίων του υπολογισμού. Ιδιαίτερα στην περίπτωση των μονάδων επεξεργασίας δεδομένων τα στάδια του pipeline χωρίζονται με απλά latches ή με latches παλμού (pulsed ή glitch latch) ώστε να μειωθεί

η χρονική επιβάρυνση που εισάγεται εξαιτίας του χρόνου setup και hold των ακολουθιακών στοιχείων μνήμης [119], [131]. Επίσης, είναι κοινή πρακτική να εισάγεται μέσα στη λογική του στοιχείου μνήμης και η δυνατότητα του υπολογισμού βασικών λογικών πράξεων μειώνοντας έτσι περισσότερο την καθυστέρηση (latches with embedded logic) [82]. Οι τεχνικές αυτές επιφέρουν σημαντική βελτίωση στην καθυστέρηση του κυκλώματος και επίσης αντιμετωπίζουν αποτελεσματικά τις επιπλέον καθυστερήσεις που εισάγονται από τις αναπόφευκτες χρονικές διαφορές μεταξύ των ακμών του ρολογιού σε διαφορετικά σημεία του κυκλώματος (clock skew). Αυτές οι κυκλωματικές τεχνικές δεν υποστηρίζονται εύκολα από εμπορικά εργαλεία αυτόματου σχεδιασμού ούτε από τις προσχεδιασμένες βιβλιοθήκες βασικών πυλών, τα οποία “προτιμούν” να λειτουργούν με στοιχεία μνήμης με μεγαλύτερη χρονική επιβάρυνση αλλά και μεγαλύτερη ασφάλεια στη χρήση τους, όπως flip-flops ή master-slave latches και ο σχεδιαστής πρέπει στην χειρότερη περίπτωση να τις εισάγει προσωπικά σε επίπεδο τρανζίστορ [26]. Αν και η τεχνική του pipeline αποτελεί ένα από τα πιο ισχυρά εργαλεία που ο σχεδιαστής έχει στη διάθεση του, η υπερβολική χρήση της μπορεί να οδηγήσει σε αρνητικά αποτελέσματα όσον αφορά την απόδοση του κυκλώματος, κυρίως σε σχέση με την κατανάλωση ισχύος, αλλά και την απόδοση του επεξεργαστή στο σύνολο του [61], [149], [151], [150]. Δεν πρέπει να ξεχνάμε πως ένα flip-flop καταλαμβάνει περίπου 6 φορές περισσότερο χώρο απ’ ό,τι μία πύλη NAND της ίδιας οδηγητικής ικανότητας καταναλώντας αντίστοιχα τουλάχιστον 2.5 φορές περισσότερη ισχύ (Τα δεδομένα αυτά προέρχονται από μια εμπορική προσχεδιασμένη βιβλιοθήκη των 130nm και αφορούν πλήρως στατικές υλοποιήσεις των στοιχείων αποθήκευσης). Για το λόγο αυτό πρέπει η εισαγωγή των σταδίων του pipeline να γίνει με ιδιαίτερη προσοχή, ώστε να μην καταλήξουμε να πληρώνουμε όλη τη διαθέσιμη ισχύ για την αποθήκευση της πληροφορίας παρά για την επεξεργασία της.

1.2 Υποστήριξη νέων αρχιτεκτονικών

Το σύνολο εντολών των μικροεπεξεργαστών επαυξάνεται συνεχώς με σκοπό να υποστηρίξει με τον καλύτερο δυνατό τρόπο την εκτέλεση νέων απαιτητικότερων εφαρμογών που προέρχονται κυρίως από τον χώρο των πολυμέσων. Οι νέες εντολές απαιτούν είτε τη σχεδίαση νέων μονάδων επεξεργασίας δεδομένων είτε την αναπροσαρμογή των κυκλωμάτων που ήδη υπάρχουν με σκοπό την αποτελεσματική εκτέλεση τους. Χαρακτηριστικό παράδειγμα της προσθήκης νέων εντολών και των αντίστοιχων μονάδων επεξεργασίας δεδομένων στους επεξεργαστές γενικού σκοπού αποτελούν οι εντολές και οι μονάδες επεξεργασίας πολυμέσων (multimedia instructions και media processing units). Ως πιο πρόσφατα παραδείγματα μπορούμε να αναφέρουμε τη νέα ομάδα εντολών SSE4 της Intel [135] και την αρχιτεκτονική NEON που υλοποιούν οι επεξεργαστές της ARM.

Οι μονάδες επεξεργασίας πολυμέσων αποθηκεύουν και επεξεργάζονται λέξεις μεγάλου εύρους, έως και 128 δυαδικών ψηφίων, οι οποίες δεν αναπαριστούν ένα δεδομένο αλλά αποτελούνται από ανεξάρτητα δεδομένα πολύ μικρότερου εύρους των 8 ή των 16 δυαδικών ψηφίων, τα οποία καλούνται υπολέξεις (subwords). Η επεξεργασία εκτελείται παράλληλα σε όλες τις υπολέξεις ακολουθώντας την πρακτική του SIMD (Single Instruction Multiple Data). Η λογική αυτή προσομοιάζει στη λειτουργία των διανυσματικών επεξεργαστών [4], μόνο που σ’ αυτήν την περίπτωση τα διανύσματα αποτελούνται από λίγα στοιχεία τα οποία αποθηκεύονται στους ήδη υπάρχοντες καταχωρητές του επεξεργαστή. Για το λόγο αυτό, η ομαδοποίηση των

πράξεων με τη χρήση εντολών SIMD και η εκμετάλλευση αυτού του είδους της παραλληλίας μεταξύ των δεδομένων καλείται συχνά και διανυσματική εκτέλεση μεταξύ καταχωρητών (Intra-Register Vectorization) [12]. Πέρα από την υλοποίηση βασικών αριθμητικών πράξεων, οι μονάδες επεξεργασίας πολυμέσων υποστηρίζουν πολλές εντολές αναδιάταξης των υπολέξεων, οι οποίες συμβάλλουν αποτελεσματικά στην επιτάχυνση της εφαρμογής στο σύνολο της [136]. Η σχεδίαση των κυκλωμάτων που υλοποιούν τις αναδιατάξεις των υπολέξεων δεν είναι απλή εξαιτίας κυρίως του μεγάλου αριθμού των καλωδίων που απαιτεί η οργάνωση τους [107]. Ακόμη σε πολλές περιπτώσεις ο ορισμός νέων εντολών αναδιάταξης δε συνοδεύεται από κάποιο προφανή τρόπο υλοποίησης της λειτουργίας. Έτσι, ο σχεδιαστής πρέπει να εφεύρει νέα κυκλώματα τα οποία να έχουν μικρή καθυστέρηση και λογική κατανάλωση ισχύος. Μια εισαγωγική παρουσίαση των νέων εντολών και του τρόπου με τον οποίο χρησιμοποιούνται συμπεριλαμβάνεται στο κεφάλαιο 4.

Πέρα από τους επεξεργαστές γενικού σκοπού δεν πρέπει να ξεχνάμε και τους εξειδικευμένους επεξεργαστές πολυμέσων (multimedia processors) ή γραφικών (graphics processors) στους οποίους οι μονάδες επεξεργασίας δεδομένων αποτελούν ένα πολύ μεγαλύτερο μέρος του σχεδιασμού [156]. Το ίδιο συμβαίνει και στους επεξεργαστές ψηφιακού σήματος (DSP, Digital Signal Processors) οι οποίοι στην πλειοψηφία τους ακολουθούν τη φιλοσοφία παράλληλης εκτέλεσης VLIW (Very Long Instruction Word) [52]. Όλοι οι επεξεργαστές αυτής της κατηγορίας βασίζονται τη λειτουργία τους σε πολλές παράλληλες μονάδες επεξεργασίας δεδομένων (40 για τον πιο νέο επεξεργαστή της οικογένειας Trimedia TM3270 [161]) στις οποίες εκτελούνται παράλληλα εντολές των οποίων οι εξαρτήσεις έχουν λυθεί στη φάση της μεταγλώττισης. Ένα από τα βασικά προβλήματα που αντιμετωπίζει ο σχεδιαστής τέτοιων εξειδικευμένων επεξεργαστών είναι από τη μια να βρεί εκείνες τις λειτουργίες των αλγόριθμων, οι οποίες θα ήταν πιο αποδοτικό να υλοποιηθούν σε υλικό μέσω νέων συγκεκριμένων εντολών, ενώ ταυτόχρονα η αρχιτεκτονική να παραμείνει αρκετά γενική, ώστε να μπορεί να επαναχρησιμοποιηθεί και σε άλλου είδους εφαρμογές, χωρίς φυσικά η υλοποίηση να αγγίζει την πολυπλοκότητα των επεξεργαστών γενικού σκοπού.

Άλλωστε, πολύ πρόσφατες μελέτες δείχνουν πως η αρχιτεκτονική πρακτική των superscalar επεξεργαστών με πολλά στάδια pipeline οι οποίοι προσκομίζουν πολλές εντολές ταυτόχρονα και ανακαλύπτουν σε υλικό την παραλληλία ανάμεσα τους, δεν οδηγεί πλέον σε αποτελεσματικές λύσεις στις πολύ μικρές τεχνολογίες υλοποίησης κυρίως εξαιτίας της επιβάρυνσης σε κατανάλωση ισχύος [151]. Αυτό που προβλέπετε [5], είναι ότι οι αρχιτεκτονικές στο μέλλον θα προσομοιάζουν περισσότερο με παράλληλα συστήματα, οι βασικές μονάδες των οποίων θα υιοθετούν απλές αρχιτεκτονικές in-order issue με λίγα pipeline στάδια (5 - 9) και θα έχουν μια μεγάλη ποικιλία από μονάδες επεξεργασίας δεδομένων, όπως μονάδες κινητής υποδιαστολής και μονάδες που θα επιτρέπουν τη διανυσματική επεξεργασία των δεδομένων. Παράλληλα, θα υποστηρίζονται απευθείας σε υλικό υπό τη μορφή βοηθητικών επεξεργαστών (co-processors) πολλοί αλγόριθμοι με μεγάλες υπολογιστικές απαιτήσεις, όπως για παράδειγμα οι αλγόριθμοι κρυπτογραφίας [13].

1.3 Γενικές σχεδιαστικές πρακτικές

Η αποτελεσματικότητα (efficiency) ενός κυκλώματος εξαρτάται από πολλούς παράγοντες. Πολλές είναι και οι μετρικές οι οποίες μας επιτρέπουν να καθορίσουμε ποσοτικά την

αποτελεσματικότητα του. Παραδοσιακά η μέτρηση της αποτελεσματικότητας ενός κυκλώματος γίνεται με βάση την καθυστέρηση του, την κατανάλωση ισχύος και το εμβαδόν που καταλαμβάνει στο ολοκληρωμένο. Παρόλα αυτά, ο ορισμός της αποτελεσματικότητας ενός κυκλώματος δεν θα είχε νόημα αν πρώτα δεν καθορίζαμε με σαφήνεια τις εφαρμογές στις οποίες το ζητούμενο κύκλωμα θα χρησιμοποιηθεί. Για παράδειγμα, στους μικροεπεξεργαστές υψηλών επιδόσεων η ταχύτητα ενός κυκλώματος αποτελεί τον πρωταρχικό σχεδιαστικό στόχο, ενώ σε φορητές συσκευές χωρίς υψηλές υπολογιστικές απαιτήσεις η μείωση της κατανάλωσης ισχύος θα απασχολούσε περισσότερο τους σχεδιαστές. Στις μέρες μας τέτοιες σαφείς διαχωριστικές γραμμές κατά το σχεδιασμό έπαψαν πλέον να υπάρχουν. Το ζητούμενο σε όλα τα υπο-σχεδίαση κυκλώματα είναι η μείωση της καθυστέρησης του κυκλώματος χωρίς να υπερβαίνουμε ένα ανώτερο όριο κατανάλωσης ισχύος. Η ανώτατη τιμή της κατανάλωσης ισχύος που επιτρέπεται σε κάθε υπο-μονάδα είναι ένα ποσοστό της συνολικής κατανάλωσης ισχύος του συστήματος και μπορεί να είναι διαφορετική ανάλογα με την εφαρμογή που εκτελείται. Το ποσοστό κατανάλωσης ισχύος που ανατίθεται σε κάθε υπο-μονάδα (ή που ανεχόμαστε από αυτή) καθορίζεται από μια πληθώρα άλλων σχεδιαστικών παραγόντων, όπως για παράδειγμα την αξιόπιστη λειτουργία του δικτύου διαμοίρασης της τάσης τροφοδοσίας [169].

Ο σχεδιαστής έχει ταυτόχρονα στη διάθεση του μια ποικιλία από τεχνικές που θα του επιτρέψουν να πετύχει το στόχο του, αλλά και ένα σύνολο από περιορισμούς, οι οποίοι οφείλονται συνήθως στην τεχνολογία ή τη ροή σχεδιασμού και του περιορίζουν σημαντικά το χώρο των σχεδιαστικών επιλογών [26]. Οι τεχνικές αυτές συνήθως εφαρμόζονται ιεραρχικά ξεκινώντας από το επίπεδο της μικροαρχιτεκτονικής και φτάνοντας στις τελικές βελτιστοποιήσεις σε επίπεδο κυκλώματος [109]. Στη συνέχεια, θα αναλύσουμε εν συντομία τις επιλογές που έχει στη διάθεση του ο σχεδιαστής και θα περιγράψουμε τον τρόπο με τον οποίο εφαρμόζονται, στοχεύοντας στη βελτίωση της αποτελεσματικότητας του κυκλώματος.

1.3.1 Επίπεδο μικροαρχιτεκτονικής

Στην κατηγορία αυτή ανήκουν όλες εκείνες οι τεχνικές οι οποίες προσπαθούν να αντιμετωπίσουν τον αλγόριθμο που θέλουμε να εκτελέσουμε σε υλικό στο σύνολο του. Στη περίπτωση αυτή, το πρόβλημα αντιμετωπίζεται αφαιρετικά και σημαντικές λεπτομέρειες της υλοποίησης παραβλέπονται εν μέρει, ώστε να μειωθεί ο χώρος λύσεων και να αναδειχθούν εκείνες οι παράμετροι που επηρεάζουν περισσότερο την απόδοση του κυκλώματος στο σύνολο του. Μερικές από τις πιο ευρέως χρησιμοποιούμενες τεχνικές που ανήκουν σ' αυτήν την κατηγορία είναι η εφαρμογή του pipeline που ήδη αναφέραμε, η παράλληλη επεξεργασία και η τεχνική του μοιράσματος (folding) ή του απλώματος (unfolding) των μονάδων που χρησιμοποιούμε για την εκτέλεση του αλγορίθμου [110], [23].

Οι τεχνικές αυτές—με εξαίρεση αυτή του pipeline—βρίσκουν ελάχιστες εφαρμογές στη σχεδίαση των μονάδων επεξεργασίας δεδομένων των επεξεργαστών. Ο κύριος λόγος είναι ότι οι μικροαρχιτεκτονικές τεχνικές λειτουργούν αποτελεσματικά σε επαναληπτικούς αλγορίθμους, όπου η επεξεργασία ολοκληρώνεται σε πολλούς κύκλους. Κάτι τέτοιο δε συμβαίνει στους επεξεργαστές. Σ' αυτή την περίπτωση, οι μονάδες επεξεργασίας δεδομένων καλούνται να εκτελέσουν απλές εντολές, είτε αριθμητικών πράξεων είτε αναδιοργάνωσης δεδομένων, με τη βοήθεια των οποίων η εκτέλεση του αλγορίθμου ολοκληρώνεται σε επίπεδο λογισμικού. Ελάχιστες είναι οι εξαιρέσεις των πράξεων οι οποίες εκτελούνται επαναληπτικά στους επεξεργαστές γενικού σκοπού, όπως για

παράδειγμα η διαίρεση αριθμών κινητής υποδιαστολής.

1.3.2 Λογικό επίπεδο

Στην κατηγορία αυτή ανήκουν όλες εκείνες οι αλγοριθμικές τεχνικές, οι οποίες προσπαθούν να βελτιώσουν διάφορες παραμέτρους του κυκλώματος όπως καθυστέρηση, κατανάλωση ισχύος, εμβαδόν χωροθέτησης, εφευρίσκοντας ένα νέο τρόπο υλοποίησης της λειτουργίας που θέλουμε να εκτελεί το κύκλωμα μας. Ένα κλασσικό παράδειγμα βελτιστοποίησης σε λογικό επίπεδο είναι οι μονάδες πρόβλεψης κρατουμένου, τα δένδρα υπολογισμού του αθροίσματος πολλών όρων όπως το δένδρο που προτάθηκε από τον Wallace ή τον Dadda ή ο αλγόριθμος του Booth για τη μείωση των μερικών γινομένων στην περίπτωση του πολλαπλασιασμού [129].

Φυσικά, οποιαδήποτε βελτιστοποίηση, και αν προταθεί, σε λογικό επίπεδο πρέπει οπωσδήποτε να λαμβάνει υπόψη την υλοποίηση σε επίπεδο κυκλώματος που θα ακολουθήσει αλλά και την τεχνολογία υλοποίησης. Για παράδειγμα, ένας νέος αλγόριθμος, ο οποίος θα μειώνει μεν σημαντικά το πλήθος των πυλών που απαιτούνται για την εκτέλεση μιας λειτουργίας αλλά θα αυξάνει δραστικά το πλήθος και το μήκος των εσωτερικών γραμμών διασύνδεσης, είναι απίθανο να υιοθετηθεί στις σημερινές τεχνολογίες. Αυτό οφείλεται στο γεγονός ότι η χωρητικότητα και η αντίσταση των γραμμών διασύνδεσης επιφέρουν σημαντική επιβάρυνση στην καθυστέρηση του κυκλώματος, ενώ η ανάγκη για ομοιόμορφους (regular) σχεδιασμούς είναι πλέον επιτακτική. Στην κατεύθυνση αυτή της γρήγορης εκτίμησης της απόδοσης του κυκλώματος [80], [8], κυρίως όσον αφορά την καθυστέρηση του, πριν αυτό υλοποιηθεί σε επίπεδο τρανζίστορ, έχουν προταθεί πολλά μοντέλα με κυριότερο αυτό του Logical Effort [153]. Περισσότερες λεπτομέρειες για το μοντέλο αυτό θα παρουσιαστούν στο κεφάλαιο 2.

Είναι κοινή πρακτική σήμερα να αναλαμβάνει τις βελτιστοποιήσεις σε λογικό επίπεδο ένα αυτόματο εργαλείο σύνθεσης του σχεδιασμού και αντιστοίχισης της λειτουργίας του σε βασικές πύλες (synthesis και technology mapping). Παρόλα αυτά, είναι πολύ δύσκολο να βρεθούν αποδοτικές υλοποιήσεις σε επίπεδο πυλών για αριθμητικά κυκλώματα ή κυκλώματα με μια εγγενή ομοιομορφία, ξεκινώντας από μια αφαιρετική περιγραφή της λειτουργίας τους. Για να δείξουμε τη δυσκολία αυτή θα χρησιμοποιήσουμε ως παράδειγμα το κύκλωμα του σχήματος 1.1. Το κύκλωμα που απεικονίζεται μετράει τα μηδενικά που βρίσκονται πριν από τον πιο σημαντικό άσσο μιας λέξης των 16 δυαδικών ψηφίων. Αν για παράδειγμα η λέξη εισόδου είναι η 0001101001001100 τότε στην έξοδο του κυκλώματος θα εμφανιστεί η τιμή 0|0011(3₁₀) ενώ για τη λέξη 0000000000000000 η αντίστοιχη έξοδος είναι 1|0000(0₁₀). Το πιο σημαντικό ψηφίο δηλώνει χωριστά την περίπτωση που τα δυαδικά ψηφία της εισόδου είναι όλα μηδενικά. Το κύκλωμα αυτό έχει πολλές εφαρμογές με κυριότερη τη σχεδίαση μονάδων κινητής υποδιαστολής. Η λειτουργία του κυκλώματος περιγράφηκε σε Verilog, η οποία φαίνεται επίσης στο σχήμα 1.1, ενώ το αντίστοιχο κύκλωμα προέκυψε μετά από τη σύνθεση και τη βελτιστοποίηση του στο εργαλείο Synopsys. Αντίθετα, το κύκλωμα το οποίο προέκυψε αναλυτικά, εφαρμόζοντας μια πληθώρα βελτιστοποιήσεων σε λογικό επίπεδο, οι οποίες αναλύονται στο κεφάλαιο 7, φαίνεται στο σχήμα 1.2. Η διαφορά στην ομοιομορφία των δύο κυκλωμάτων και στη δομή των μονοπατιών είναι προφανής και πάντα υπέρ της αναλυτικής προσέγγισης. Στην δεύτερη περίπτωση, απαιτούνται ουσιαστικά μόνο δύο είδη πυλών για την ολοκλήρωση του σχεδιασμού, οι οποίες είναι συνδεδεμένες με απλό και επαλαμβανόμενο τρόπο. Για το λόγο αυτό, ακόμη και εμπορικά εργαλεία με μεγάλες δυνατότητες εύρεσης αποδοτικών


```

// 16 bit leading zero counter
module LZC_16bit(A, Z);
input  [15:0] A; // Input bits
output [ 4:0] Z; // Leading Zero Representation

reg  [ 4:0] Z;

always @(A) // When input changes
begin
    casez (A)
        16'b0000000000000000 : Z <= 5'b10000; // All-zero flag set
        16'b0000000000000001 : Z <= 5'b01111; // 15 Leading Zeros
        16'b000000000000001? : Z <= 5'b01110; // 14 Leading Zeros
        16'b000000000000001?? : Z <= 5'b01101; // 13 Leading Zeros
        16'b000000000000001??? : Z <= 5'b01100; // 12 Leading Zeros
        16'b00000000000001???? : Z <= 5'b01011; // 11 Leading Zeros
        16'b00000000000001????? : Z <= 5'b01010; // 10 Leading Zeros
        16'b0000000001??????? : Z <= 5'b01001; // 9 Leading Zeros
        16'b000000001???????? : Z <= 5'b01000; // 8 Leading Zeros
        16'b00000001????????? : Z <= 5'b00111; // 7 Leading Zeros
        16'b0000001?????????? : Z <= 5'b00110; // 6 Leading Zeros
        16'b000001??????????? : Z <= 5'b00101; // 5 Leading Zeros
        16'b00001???????????? : Z <= 5'b00100; // 4 Leading Zeros
        16'b0001????????????? : Z <= 5'b00011; // 3 Leading Zeros
        16'b001?????????????? : Z <= 5'b00010; // 2 Leading Zeros
        16'b01??????????????? : Z <= 5'b00001; // 1 Leading Zeros
        16'b1???????????????? : Z <= 5'b00000; // 0 Leading Zeros
    endcase
end
endmodule

```

design: LZC_16bit	designer: Giorgos Dimitrakopoulos	date: 1/27/2007
technology: ucb1301803_1a_120v	library: CE1D	sheet: 1 of 1

Σχήμα 1.1: Αυτόματη σύνθεση από μια περιγραφή υψηλού επιπέδου ενός κυκλώματος μέτρησης των προπορευόμενων μηδενικών των 16 δυαδικών ψηφίων.

Σχήμα 1.2: Κυκλώμα μέτρησης των προπορευόμενων μηδενικών των 16 δυαδικών ψηφίων σχεδιασμένο με τη βοήθεια μιας νέας αναλυτικής προσέγγισης.

υλοποιήσεων, όταν αντιμετωπίζουν κυκλώματα βασικών αριθμητικών πράξεων, τα οποία έχουν περιγραφεί από το σχεδιαστή σε αφαιρετικό επίπεδο, αρκούνται στη χρήση προσχεδιασμένων κυκλωμάτων που ικανοποιούν τις ζητούμενες ιδιότητες σε ταχύτητα και κατανάλωση ισχύος. Χαρακτηριστικό παράδειγμα είναι η βιβλιοθήκη DesignWare της Synopsys και το εξειδικευμένο εργαλείο σύνθεσης μονάδων επεξεργασίας δεδομένων Module Compiler της ίδιας εταιρείας.

1.3.3 Επίπεδο κυκλώματος

Οι σχεδιαστικές πρακτικές σε επίπεδο κυκλώματος είναι άμεσα συνυφασμένες με αυτές που εφαρμόζονται σε λογικό επίπεδο. Σ' αυτή τη φάση, ο σχεδιαστής έχει να επιλέξει διάφορες άλλες παραμέτρους που επηρεάζουν σημαντικά την αποτελεσματικότητα του κυκλώματος. Οι παράμετροι αυτοί είναι η λογική οικογένεια των πυλών και το μέγεθος τους, η τάση τροφοδοσίας και η τάση κατωφλίου των τρανζίστορ και η εισαγωγή επιπλέον ενισχυτικών στοιχείων (απλοί αντιστροφείς, ή buffers) για την αντιμετώπιση των προβλημάτων που προκύπτουν κυρίως εξαιτίας της χωροθέτησης του κυκλώματος και των γραμμών διασύνδεσης. Οι βέλτιστες τιμές των παραμέτρων αυτών δεν είναι εύκολο να βρεθούν και σε πολλές περιπτώσεις, η αλλαγή μιας από αυτές με σκοπό την επίτευξη ενός σχεδιαστικού στόχου, απαιτεί τον επαναπροσδιορισμό και των υπολοίπων μεταβλητών που επηρεάζονται από την αλλαγή αυτή. Το ζητούμενο σε κάθε περίπτωση είναι να εντοπίσουμε εκείνη την εκδοχή του κυκλώματος που επιτρέπει την καλύτερη δυνατή συμπεριφορά ως προς την ενέργεια και την καθυστέρηση.

Ο μόνος αξιόπιστος τρόπος για να εξερευνήσουμε το διαθέσιμο χώρο λύσεων και την

Σχήμα 1.3: Ο χώρος λύσεων ως προς την ενέργεια και την καθυστέρηση.

επίδραση των σχεδιαστικών τεχνικών που εφαρμόσαμε, είναι να παράγουμε για το κύκλωμα μας τις βέλτιστες καμπύλες της ενέργειας ως προς την καθυστέρηση [180], [125]. Σύμφωνα με αυτή τη λογική, όλες οι δυνατές εναλλακτικές εκδοχές που έχουμε για το υπο-εξέταση κύκλωμα μπορούν να τοποθετηθούν πάνω σε ένα διάγραμμα με άξονες την καθυστέρηση και την ενέργεια, όπως αυτό που φαίνεται στο σχήμα 1.3. Η καμπύλη του διαγράμματος αναπαριστά τη βέλτιστη καμπύλη της ενέργειας ως προς την καθυστέρηση και όλες οι λύσεις που μπορούμε να πετύχουμε είναι είτε σημεία της καμπύλης αυτής είτε βρίσκονται πάνω από αυτήν. Στη συνέχεια θα περιγράψουμε τον τρόπο με τον οποίο μπορούμε να δημιουργήσουμε την καμπύλη αυτή, αλλά και πως μπορούμε να την αξιοποιήσουμε ώστε να σχεδιάσουμε αποδοτικότερα κυκλώματα.

Κάθε δυνατή εκδοχή του κυκλώματος που σχεδιάζουμε, η οποία προκύπτει είτε μέσω σχεδιαστικών τεχνικών είτε μέσω βελτιστοποίησης κάποιων παραμέτρων του κυκλώματος, όπως το μέγεθος των τρανζίστορ, αναπαρίσταται με ένα σημείο στο χώρο ενέργεια-καθυστέρηση. Για παράδειγμα, έστω ότι το σημείο 1 του σχήματος 1.3 αναπαριστά έναν αρχικό σχεδιασμό. Η πρώτη βελτιστοποίηση που εκτελούμε είναι να αφαιρέσουμε όσο το δυνατόν την επιπλέον ισχύ που καταναλώνει το κύκλωμα μας χωρίς να επηρεάσουμε την ταχύτητα του. Τότε οδηγούμαστε σε μία λύση, η οποία αντιστοιχεί στο σημείο 2 του σχήματος 1.3. Το σημείο 2 είναι ένα βέλτιστο σημείο ως προς την ενέργεια και την καθυστέρηση επειδή δεν είναι δυνατόν να μειώσουμε την ενέργεια του περισσότερο χωρίς να αυξήσουμε την καθυστέρησή του. Ακόμη, αν, βάσει των περιορισμών που μας έχουν τεθεί, το σημείο 2 ξεπερνά τη μέγιστη επιτρεπτή ενέργεια $E_{\max}$ ανά λειτουργία που μπορούμε να χρησιμοποιήσουμε, τότε μπορούμε να μειώσουμε την ενέργεια που απαιτεί το κύκλωμα μας αυξάνοντας όμως αναγκαστικά την καθυστέρησή του (σημείο 3). Παρόλα αυτά, εφόσον κινούμαστε στη βέλτιστη καμπύλη της ενέργειας ως προς την καθυστέρηση μπορούμε να πούμε ότι στο σημείο 3 το κύκλωμα μας πετυχαίνει την ελάχιστη δυνατή ταχύτητα σεβόμενο τον περιορισμό της διαθέσιμης ενέργειας. Τα σημεία 2 και 3 του διαγράμματος είναι κατά μία έννοια και τα δύο βέλιστα σημεία, εφόσον καταναλώνουν την ελάχιστη δυνατή ισχύ πετυχαίνοντας τη μικρότερη δυνατή ταχύτητα και το αντίθετο. Τα κυκλώματα τα οποία αντιστοιχούν σε σημεία που βρίσκονται πάνω από την καμπύλη του σχήματος 1.3, όπως αυτό που αντιστοιχεί στο σημείο 1, είναι μη βέλιστα επειδή πάντα μπορεί

Σχήμα 1.4: Διακριτά σημεία της βέλτιστης καμπύλης ως προς την ενέργεια και την καθυστέρηση.

να βρεθεί ένα κύκλωμα το οποίο να πετυχαίνει μικρότερη καθυστέρηση για την ίδια ενέργεια ή μικρότερη ενέργεια για την ίδια καθυστέρηση. Γενικά, τα κυκλώματα τα οποία δεν είναι βέλτιστα ως προς την καθυστέρηση και την ενέργεια ζοδεύουν άδικα χρήσιμους πόρους του σχεδιασμού και πρέπει να αποφεύγονται.

Για ένα κύκλωμα με δεδομένη τοπολογία⁴ η βέλτιστη καμπύλη ως προς την ενέργεια και την καθυστέρηση έχει δυο καλώς ορισμένα ακραία σημεία, όπως φαίνεται στο διάγραμμα του σχήματος 1.4. Το σημείο 1 αναπαριστά το ταχύτερο δυνατό κύκλωμα που μπορούμε να σχεδιάσουμε για μια δεδομένη τοπολογία. Αυτό είναι το σημείο που παίρνουμε αν εφαρμόσουμε για παράδειγμα τη μέθοδο του Logical Effort στο κύκλωμα μας. Το σημείο 2 αναπαριστά το κύκλωμα με τις μικρότερες απαιτήσεις σε ενέργεια για τη δεδομένη τοπολογία. Στις περισσότερες των περιπτώσεων μπορούμε να βρούμε το κύκλωμα αυτό χρησιμοποιώντας τα ελάχιστα δυνατά μεγέθη τρανζίστορ. Σε κάποιες περιπτώσεις αυτό δεν είναι πάντα ο σωστός τρόπος γιατί με τα ελάχιστα μεγέθη τρανζίστορ μειώνεται σημαντικά η οδηγητική ικανότητα των πυλών με αποτέλεσμα τα σήματα να παρουσιάζουν μεγάλους χρόνους ανόδου και καθόδου οδηγώντας έτσι στην αύξηση του ρεύματος βραχυκυκλώματος και συνεπώς στην αύξηση της κατανάλωσης ισχύος. Αν αυτά τα σημεία προέκυπταν μέσα από κάποιας μορφής βελτιστοποίηση τότε το σημείο 1 αντιστοιχεί στην ελαχιστοποίηση της καθυστέρησης D χωρίς άλλους περιορισμούς ενώ το σημείο 2 αντιστοιχεί στην ελαχιστοποίηση μόνο της ενέργειας E . Τα ενδιάμεσα σημεία τα οποία σημειώνονται με τον αριθμό 3 στην καμπύλη του σχήματος 1.4 ανήκουν σε βέλτιστα σημεία τα οποία προκύπτουν μετά την ελαχιστοποίηση σχέσεων της μορφής $E^a D^b$. Η πιο γνωστή από αυτές τις μετρικές είναι το γινόμενο της καθυστέρησης και της ενέργειας που είναι γνωστό ως EDP (Energy Delay Product) [71]. Έτσι, από τη στιγμή που έχουμε στη διάθεση μας τη βέλτιστη καμπύλη της ενέργειας ως προς την καθυστέρηση, διαθέτουμε μια πλήρη περιγραφή ολόκληρου του χώρου λύσεων του κυκλώματος μας. Αυτό που πρέπει να προσέξουμε, είναι, ότι οι βέλτιστες καμπύλες μετακινούνται ανάλογα με τις τιμές της χωρητικότητας εξόδου και εισόδου του κυκλώματος [32] (σχήμα 1.5). Τοποθετώντας τις

⁴Ως τοπολογία ενός κυκλώματος ορίζουμε το είδος των πυλών με τις οποίες είναι κατασκευασμένο, τη λογική οικογένεια στην οποία ανήκουν (στατική ή δυναμική CMOS, τρανζίστορ περάσματος (pass gates)) και τις διασυνδέσεις μεταξύ τους.

Σχήμα 1.5: Η συμπεριφορά των βέλτιστων καμπυλών ανάλογα με τη διαθέσιμη χωρητικότητα εισόδου του κυκλώματος.

Σχήμα 1.6: Ο καθορισμός της ελάχιστης ενέργειας ενός κυκλώματος με σταθερή καθυστέρηση μεταβάλλοντας τη διαθέσιμη χωρητικότητα εισόδου της δεύτερης υπο-μονάδας.

καμπύλες δύο κυκλωμάτων που το ένα οδηγεί το άλλο (η χωρητικότητα εξόδου του πρώτου αντιστοιχεί στην χωρητικότητα εισόδου του δεύτερου) στο ίδιο διάγραμμα, όπως αυτό του σχήματος 1.6, μπορούμε να παρατηρήσουμε τις χωρητικότητες εκείνες, οι οποίες δίνουν τη βέλτιστη ενέργεια για μια δεδομένη καθυστέρηση για το συνολικό κύκλωμα.

Οι βέλτιστες καμπύλες της ενέργειας ως προς την καθυστέρηση προσφέρουν τον καλύτερο δυνατό τρόπο για να συγκρίνουμε δύο διαφορετικές αρχιτεκτονικές ή τοπολογίες κυκλωμάτων οι οποίες υλοποιούν την ίδια λειτουργία. Σε επίπεδο συστήματος, αυτό που μας απασχολεί, είναι

Σχήμα 1.7: Η επιλογή της κατάλληλης αρχιτεκτονικής ή τοπολογίας με τη βοήθεια των βέλτιστων καμπυλών ως προς την ενέργεια και την καθυστέρηση.

το κύκλωμα που θα χρησιμοποιήσουμε για μια λειτουργία, να προσφέρει την ελάχιστη δυνατή καθυστέρηση για ένα συγκεκριμένο όριο της κατανάλωσης ισχύος ή να απαιτεί την ελάχιστη δυνατή ενέργεια για μια δεδομένη περίοδο του ρολογιού. Το διάγραμμα του σχήματος 1.7 παρουσιάζει τον τρόπο με τον οποίο μπορούμε να χρησιμοποιήσουμε τις βέλτιστες καμπύλες ως προς την ενέργεια και την καθυστέρηση δύο διαφορετικών αρχιτεκτονικών για την ίδια λειτουργία, με σκοπό να επιλέξουμε την πιο συμφέρουσα. Αν στοχεύουμε σε καθυστερήσεις μικρότερες της D_0 η αρχιτεκτονική A είναι η καλύτερη γιατί πετυχαίνει το στόχο που θέτουμε απαιτώντας λιγότερη ενέργεια σε σχέση με την B. Για τον ίδιο λόγο αν μας επιτρέπεται η καθυστέρηση του κυκλώματος να είναι μεγαλύτερη από D_0 τότε η αρχιτεκτονική B είναι η καλύτερη επιλογή.

Κατά τη διάρκεια της βελτιστοποίησης ως προς την ενέργεια και την καθυστέρηση οι τιμές πολλών σχεδιαστικών παραμέτρων (μεταβλητών του προβλήματος) μπορούν να συνυπολογιστούν. Για παράδειγμα, τα μεγέθη των πυλών και οι τιμές του κατωφλίου των τρανζίστορ μπορούν να αποτελέσουν μεταβλητές του προβλήματος βελτιστοποίησης. Αντίθετα κάποιες άλλες παράμετροι όπως το συνολικό εμβαδόν του κυκλώματος και το εύρος κάθε στήλης bit-slice στο layout τίθενται πριν τη σχεδίαση εξαιτίας άλλων περιορισμών σε επίπεδο αρχιτεκτονικής ή για άλλους πρακτικούς λόγους που αφορούν το ολοκληρωμένο στο σύνολο του. Η επιλογή των παραμέτρων που θα συμμετέχουν στην βελτιστοποίηση επηρεάζουν σημαντικά το τελικό αποτέλεσμα, όπως θα δείξουμε στο σχήμα 1.8.

Ξεκινώντας από έναν αρχικό σχεδιασμό, μπορούμε να πάρουμε διαφορετικές καμπύλες ενέργειας καθυστέρησης αν προσπαθήσουμε να καθορίσουμε τις τιμές διαφόρων μεταβλητών ξεχωριστά. Στο σχήμα 1.8 οι μεταβλητές M1 και M2 μπορεί να είναι για παράδειγμα η τάση τροφοδοσίας ή η τάση κατωφλίου των τρανζίστορ, η οποία ρυθμίζεται με την κατάλληλη πόλωση του υποστρώματος (body bias). Καθορίζοντας τις τιμές των μεταβλητών ξεχωριστά παράγουμε δύο συγκεκριμένες καμπύλες της ενέργειας ως προς την καθυστέρηση. Από την άλλη μεριά, αν βελτιστοποιήσουμε απο κοινού αυτές τις δύο μεταβλητές παίρνουμε καλύτερα

Σχήμα 1.8: Η επίδραση των μεταβλητών που συμμετέχουν στην βελτιστοποίηση.

κυκλώματα σύμφωνα με την λογική της απόδοσης στο χώρο ενέργεια προς καθυστέρηση. Αν συμπεριλάβουμε και άλλες μεταβλητές στη βελτιστοποίηση, όπως για παράδειγμα τα μεγέθη των τρανζίστορ, η ενέργεια και η καθυστέρηση του υπο-εξέτασης κυκλώματος μπορεί να μειωθεί περαιτέρω. Στη γενική περίπτωση, όσο περισσότερες οι μεταβλητές του προβλήματος που συνεκτιμούνται κατά τη διάρκεια της βελτιστοποίησης τόσο καλύτερο το αποτέλεσμα. Αν δεν συμπεριλάβουμε μια μεταβλητή στη βελτιστοποίηση, είναι σαν να τη θέτουμε σε μια αυθαίρετη σταθερή τιμή, μειώνοντας έτσι σημαντικά το χώρο λύσεων που η μέθοδος βελτιστοποίησης μπορεί να διερευνήσει ψάχνοντας την ελάχιστη τιμή της στοχευόμενης συνάρτησης (objective function). Κάποιες μεταβλητές του προβλήματος βελτιστοποίησης του σχεδιασμού είναι συνεχείς (μπορούν να τεθούν σε οποιαδήποτε τιμή) όπως η τάση τροφοδοσίας, τα μεγέθη των πυλών σε έναν πλήρως ελεύθερο σχεδιασμό (full custom design), ενώ κάποιες άλλες είναι από τη φύση τους διακριτές, όπως ο βαθμός παράλληλης επεξεργασίας ή το πλήθος των σταδίων pipeline. Αν και η ταυτόχρονη βελτιστοποίηση όλων των μεταβλητών που επηρεάζουν την απόδοση του κυκλώματος είναι η πιο επιθυμητή διαδικασία, παρόλα αυτά κάτι τέτοιο είναι αδύνατο να συμβεί στις περισσότερες των περιπτώσεων. Για το λόγο αυτό ακολουθείται μια ιεραρχική προσέγγιση όπου αποφάσεις, όπως αυτή του pipeline, παίρνονται σε επίπεδο μικροαρχιτεκτονικής ενώ η τοπολογία του κυκλώματος και τα μεγέθη των πυλών καθορίζονται στο λογικό και στο επίπεδο κυκλώματος αντίστοιχα [109].

1.3.4 Σχεδιαστικοί περιορισμοί

Η εφαρμογή όλων των τεχνικών που μόλις αναφέραμε πρέπει να λαμβάνει υπόψιν και έναν άλλο πολύ σημαντικό παράγοντα που επηρεάζει την αποτελεσματικότητα ενός κυκλώματος και δεν είναι άλλος από την επιβεβλημένη ροή σχεδιασμού (design flow). Η αποτελεσματικότητα των κυκλωμάτων επηρεάζεται σημαντικά από τον τρόπο με τον οποίο σχεδιάζονται, τον χρόνο που διαθέτει ο σχεδιαστής, και φυσικά από τις γνώσεις και την εμπειρία του.

Οι σχεδιαστικές τεχνικές τις οποίες αναφέραμε μπορούν να εφαρμοστούν πλήρως μόνο στην περίπτωση του απόλυτα ελεύθερου σχεδιασμού. Σ' αυτήν την περίπτωση ο σχεδιαστής

έχει την ευθύνη να διαλέξει την αρχιτεκτονική, να ολοκληρώσει το σχεδιασμό σε λογικό επίπεδο, να επιλέξει τις πύλες και τα μεγέθη τους που τον βοηθούν να επιτύχει την ζητούμενη ταχύτητα και κατανάλωση ισχύος και να πραγματοποιήσει τη χωροθέτηση των τρανζίστορ και των ενώσεων τους όπως αυτός επιθυμεί. Η τακτική αυτή βρίσκει πλέον εφαρμογή μόνο στη σχεδίαση μικροεπεξεργαστών υψηλών επιδόσεων, στη σχεδίαση μνημών και αναλογικών κυκλωμάτων όπου όλος ο σχεδιαστικός χρόνος δαπανάται για την ποιοτικότερη βελτίωση των χαρακτηριστικών των κυκλωμάτων. Στις περισσότερες όμως των περιπτώσεων, συμπεριλαμβανομένων και των επεξεργαστών που σχεδιάζονται για ενσωματωμένα συστήματα (embedded systems) [160], ο σχεδιαστής είναι υποχρεωμένος να ολοκληρώσει την εργασία του σε ένα πολύ περιορισμένο χρονικό διάστημα χρησιμοποιώντας αυτοματοποιημένα εργαλεία σχεδιασμού και προ-σχεδιασμένες βιβλιοθήκες βασικών πυλών. Σ' αυτήν την περίπτωση οι βελτιστοποιήσεις σε επίπεδο κυκλώματος είναι αρμοδιότητα του εργαλείου σχεδιασμού και ο σχεδιαστής μπορεί να επέρμει περιγράφοντας είτε μια νέα αρχιτεκτονική για το ίδιο κύκλωμα είτε μια νέα υλοποίηση σε λογικό επίπεδο.

Η χρήση ενός αυτοματοποιημένου εργαλείου επιφέρει προφανώς κάποια επιβάρυνση στην απόδοση του κυκλώματος κυρίως για δύο λόγους [31], [27]. Πρώτον, τα μεγέθη των πυλών που έχει στη διάθεση του ο σχεδιαστής είναι περιορισμένα, εξαιτίας της προσχεδιασμένης βιβλιοθήκης, με αποτέλεσμα να είναι δύσκολο να επιτευχθεί ο βέλτιστος, ως προς την ταχύτητα, σχεδιασμός. Στην καλύτερη περίπτωση δεν απαιτούνται πολλά είδη διαφορετικών πυλών για την υλοποίηση του σχεδιασμού αρκεί το καθένα από αυτά να είναι διαθέσιμο σε όσα το δυνατόν περισσότερα μεγέθη. Δεύτερον, εφόσον η χωροθέτηση των κυκλωμάτων πραγματοποιείται με αυτόματο τρόπο, είναι πολύ δύσκολο να επιτευχθούν προς το παρόν πολύ αποδοτικοί σχεδιασμοί αφού η όποια ομοιομορφία διαθέτει το κύκλωμα καταστρέφεται. Φυσικά, στην περίπτωση ενός πολύ μεγάλου σχεδιασμού η μικρότερη απόδοση είναι ανεκτή, επειδή η χρήση αυτοματοποιημένων εργαλείων μειώνει δραστικά το χρόνο σχεδιασμού και συμβάλλει, ώστε το τελικό προϊόν να βρεθεί πιο γρήγορα στην αγορά.

Τέλος, υπάρχει και η κατηγορία σχεδιασμού σε επαναπρογραμματιζόμενες συσκευές όπως αυτές των FPGAs (Field Programmable Gate Arrays) όπου οι κανόνες σχεδιασμού είναι εντελώς διαφορετικοί [88]. Σε αυτή την περίπτωση ο σχεδιαστής μπορεί να περιγράψει μόνο μια καινούρια μικρο-αρχιτεκτονική για το κύκλωμα του, η οποία φροντίζει να εκμεταλλεύεται όσο το δυνατό καλύτερα τη δομή των επαναπρογραμματιζόμενων κελιών. Χαρακτηριστικό παράδειγμα είναι ότι η τεχνική του pipeline μπορεί πολύ εύκολα να εφαρμοστεί χωρίς να ανησυχούμε για το κόστος των ακολουθιακών στοιχείων μνήμης εφόσον υπάρχει ένα τέτοιο διαθέσιμο σε κάθε επαναπρογραμματιζόμενο κελί. Και σ' αυτήν την περίπτωση η αντιστοίχιση της λογικής των μονάδων επεξεργασίας δεδομένων στα επαναπρογραμματιζόμενα κελιά δεν είναι ικανοποιητική [172]. Πλέον τα συστήματα αυτά εμπεριέχουν ανεξάρτητες μονάδες υπολογισμού βασικών πράξεων ALUs, MAC (Mutlply and ACcumulate), οι οποίες είναι σχεδιασμένες ανεξάρτητα σε επίπεδο τρανζίστορ. Οι μονάδες αυτές δεν είναι επαναπρογραμματιζόμενες απλά χρησιμοποιούνται στο ζητούμενο σχεδιασμό ανάλογα με τις υπολογιστικές απαιτήσεις και αν αυτό ζητηθεί από το σχεδιαστή [171].

1.4 Οργάνωση διατριβής

Στο κεφάλαιο αυτό περιγράψαμε συνοπτικά τα προβλήματα που αντιμετωπίζει η σχεδίαση

των μονάδων επεξεργασίας δεδομένων και πως αυτά πρέπει να αντιμετωπιστούν, ώστε να επιτευχθούν αποδοτικότεροι σχεδιασμοί ως προς την ενέργεια και την καθυστέρηση.

Η βάση όλων των νέων λύσεων που προτείνονται στα πλαίσια της παρούσας διδακτορικής διατριβής είναι η εύρεση ενός εγγενώς απλούστερου αλγορίθμου, ο οποίος θα επιτρέπει την αποτελεσματική υλοποίηση των αντίστοιχων κυκλωμάτων ανεξάρτητα από την τεχνολογία που θα χρησιμοποιηθεί και από τους επιπλέον περιορισμούς που τυχόν θα επιβληθούν στο μέλλον κατά την κατασκευή των κυκλωμάτων αυτών. Για να μπορέσουμε να εξερευνήσουμε σε βάθος τις ιδιότητες των κυκλωμάτων που εξετάσαμε, υλοποιήσαμε ένα αυτόματο εργαλείο βελτιστοποίησης της ενέργειας και της καθυστέρησής τους. Το εργαλείο αυτό επιλέγει κατάλληλα τα μεγέθη των τρανζίστορ με σκοπό την ικανοποίηση των περιορισμών που θέτουμε. Έτσι, πριν αναφερθούμε στα νέα κυκλώματα που σχεδιάσαμε, στο κεφάλαιο 2 αναλύουμε τις βασικές αρχές λειτουργίας της βελτιστοποίησης που εφαρμόσαμε και τον τρόπο με τον οποίο αυτή χρησιμοποιήθηκε για την παραγωγή των τελικών συγκριτικών αποτελεσμάτων.

Στα επόμενα κεφάλαια περιγράφουμε αναλυτικά τη δομή και τη λειτουργία των νέων κυκλωμάτων καθώς και τα συγκριτικά αποτελέσματα που συλλέξαμε με σκοπό τη σύγκριση της απόδοσής τους με τις ήδη γνωστές υλοποιήσεις από τη διεθνή βιβλιογραφία. Στο κεφάλαιο 3 παρουσιάζονται οι νέοι αθροιστές παράλληλου προθέματος, οι οποίοι βασίζουν τη λειτουργία τους στα κρατούμενα που προτάθηκαν από τον Ling και αποτελούν απλοποιημένες μορφές των κλασικών σχέσεων πρόβλεψης κρατουμένου [43]. Το νέο κύκλωμα, που προτείνουμε, αποτελεί ουσιαστικά τη γενίκευση των σχέσεων αυτών, επιτρέποντας την υλοποίησή τους με απλοποιημένες δομές παράλληλου προθέματος, με αποτέλεσμα τη μείωση τόσο της καθυστέρησης όσο και της απαιτούμενης ενέργειας.

Τα κυκλώματα που περιγράφονται στο κεφάλαιο 4 αναφέρονται σε μονάδες αναδιάταξης των δεδομένων που είναι αποθηκευμένα στους καταχωρητές του επεξεργαστή. Η αποδοτική αναδιάταξη των δεδομένων καταλήγει να είναι σε πολλούς αλγορίθμους (κρυπτογραφία, ψηφιακή επεξεργασία σήματος, πολυμέσα) τόσο αναγκαία όσο και η γρήγορη υλοποίηση των βασικών αριθμητικών πράξεων, αλλά και η ταχεία μεταφορά των δεδομένων από και προς τη μνήμη. Η προσπάθειά μας εστιάστηκε στην αποδοτική υλοποίηση μιας γενικής εντολής αναδιάταξης δεδομένων, στοχεύοντας σε όσο το δυνατόν ταχύτερες υλοποιήσεις. Όλες οι εκδοχές που προτείναμε [38], [37], [42], στηρίζονται σε μια επέκταση των δικτύων ταξινόμησης, η οποία μας επιτρέπει να παρέχουμε λύσεις που είναι σημαντικά πιο αποδοτικές σε σχέση με τις ήδη υπάρχουσες.

Επίσης, στο κεφάλαιο 6 περιγράφουμε νέα κυκλώματα τα οποία χρησιμοποιούνται σε υλοποιήσεις αθροιστών αριθμητικής κινητής υποδιαστολής. Αρχικά, παρουσιάζουμε τη σχεδίαση απλοποιημένων μονάδων πρόσθεσης για αριθμούς συμπληρώματος ως προς ένα [40], αναπαράσταση η οποία κυρίως χρησιμοποιείται για την υποστήριξη της πρόσθεσης των μεγεθών των αριθμών κινητής υποδιαστολής. Παράλληλα, αναλύουμε πως καταφέραμε να ενώσουμε για πρώτη φορά τις απλοποιημένες σχέσεις κρατουμένου του Ling με την πρόσθεση αριθμών που ακολουθούν την αναπαράσταση συμπληρώματος ως προς ένα [39]. Επιπρόσθετα, στο κεφάλαιο 7 περιγράφεται ένας νέος απλός τρόπος για την υλοποίηση της πρόβλεψης και της μέτρησης των προπορευόμενων μηδενικών που εμφανίζονται στα αποτελέσματα των πράξεων κινητής υποδιαστολής [36]. Δείχνουμε, ότι με τη χρήση των νέων κυκλωμάτων, η κανονικοποίηση του αποτελέσματος μπορεί να πραγματοποιηθεί σε λιγότερο χρόνο και με σημαντικά μικρότερη

ενέργεια.

Τέλος, με το κεφάλαιο 8 ολοκληρώνουμε τη διδακτορική διατριβή σχολιάζοντας τα νέα ερευνητικά ερωτήματα που προέκυψαν μετά την ολοκλήρωση της παρούσας ερευνητικής προσπάθειας και τις κατευθύνσεις στις οποίες θα στραφούμε για τη περαιτέρω βελτιστοποίηση των μονάδων επεξεργασίας δεδομένων των μικροεπεξεργαστών υψηλών επιδόσεων.

Ακολουθούμενη ροή σχεδιασμού

Οι βελτιστοποιήσεις που παρουσιάζονται στα επόμενα κεφάλαια της παρούσας διδακτορικής διατριβής εστιάζονται στο λογικό επίπεδο και στις υλοποιήσεις σε επίπεδο κυκλώματος. Η ανάλυση βασίζεται σε υλοποιήσεις σε επίπεδο τρανζίστορ, που είναι η κύρια μέχρι τώρα σχεδιαστική πρακτική όλων των μικροεπεξεργαστών υψηλών επιδόσεων. Παρόλα αυτά, τα αποτελέσματα που προκύπτουν, όπως διαπιστώσαμε με επιπλέον πειράματα, είναι εφικτά ακόμη και σε πλήρως αυτοματοποιημένες υλοποιήσεις. Φυσικά, σ' αυτήν την περίπτωση πρέπει να λάβουμε υπόψη την επιβάρυνση που εισάγει η σχεδίαση μέσω αυτοματοποιημένων εργαλείων (synthesis, auto place-and-route) και η αναγκαστική χρήση των προσχεδιασμένων βιβλιοθηκών βασικών πυλών (standard cells), τόσο σε καθυστέρηση όσο και σε ενέργεια. Η μελέτη που πραγματοποιήσαμε στοχεύει στην ταχύτερη δυνατή υλοποίηση των κυκλωμάτων αυτών, αλλά και στην εξεύρεση λύσεων με σχετικά μεγαλύτερη καθυστέρηση, με στόχο τη μείωση της απαιτούμενης ενέργειας. Για το σκοπό αυτό, υλοποιήθηκε ένα εργαλείο αυτόματης εύρεσης των μεγεθών των πυλών του κυκλώματος, ανάλογα με τους χρονικούς περιορισμούς που θέτουμε. Το εργαλείο αυτό λύνει ένα μαθηματικό πρόβλημα βελτιστοποίησης σύμφωνα με τις αρχές του γεωμετρικού προγραμματισμού. Η επιλογή αυτή πραγματοποιήθηκε γιατί μέχρι και σήμερα δεν υπάρχει διαθέσιμο εμπορικό εργαλείο που να επιτρέπει, πέρα από την ανάλυση του σχεδιασμού, τη βελτιστοποίηση των κυκλωμάτων σε επίπεδο τρανζίστορ. Έτσι, μας επιτρέπεται να πειραματιστούμε με διαφορετικές τοπολογίες του ίδιου κυκλώματος, να εξάγουμε απαραίτητα συμπεράσματα και να απορρίψουμε μη αποδοτικές λύσεις.

Η λειτουργία του εργαλείου που υλοποιήσαμε μαζί με την πειραματική μεθοδολογία που ακολουθήσαμε για την εξερεύνηση του χώρου λύσεων ως προς την ενέργεια και την καθυστέρηση, θα αναλυθούν στη συνέχεια του κεφαλαίου. Επίσης, παρουσιάζονται όλες εκείνες οι λεπτομέρειες που απαιτούνται για να γίνουν καλύτερα κατανοητές οι δυνατότητες των τεχνολογιών υλοποίησης, στις οποίες πραγματοποιήσαμε τα πειράματά μας. Μέρος της

ανάλυσης που ακολουθεί δεν αποτελεί τμήμα της πρωτότυπης δουλειάς που πραγματοποιήθηκε στα πλαίσια της διδακτορικής διατριβής. Οι μέθοδοι που χρησιμοποιούμε έχουν προταθεί από άλλους ερευνητές [28], [15]. Ωστόσο, η παρουσίαση αυτή χρησιμεύει ώστε να γίνει καλύτερα κατανοητή η ακολουθούμενη ροή σχεδιασμού και ο τρόπος εξαγωγής των πειραματικών αποτελεσμάτων.

2.1 Μέτρηση της απόδοσης των κυκλωμάτων

Το πρώτο βήμα για την ανάλυση της συμπεριφοράς των κυκλωμάτων στο χώρο ενέργεια καθυστέρηση είναι να αποφασίσουμε τις μονάδες μέτρησης για αυτές τις δύο ποσότητες και τον τρόπο υπολογισμού τους.

Είναι προφανές πως ο καλύτερος δυνατός τρόπος για να μετρήσουμε την απόδοση ενός κυκλώματος είναι να το κατασκευάσουμε και να μετρήσουμε με ακρίβεια τις παραμέτρους που μας απασχολούν. Παρόλα αυτά, στην περίπτωση που θέλουμε να δοκιμάσουμε διαφορετικές εκδοχές του ίδιου κυκλώματος αναφερόμενοι σε πολύ μικρές τεχνολογίες κατασκευής κάτι τέτοιο είναι ανέφικτο, κυρίως από πλευράς κόστους. Αυτό λοιπόν που μπορούμε να κάνουμε είναι να εξομοιώσουμε τα κυκλώματα που μας απασχολούν στοχεύοντας στη μεγαλύτερη δυνατή ακρίβεια των αποτελεσμάτων. Το μόνο εργαλείο που μας επιτρέπει να προσεγγίσουμε όσο το δυνατόν περισσότερο τις πραγματικές συνθήκες λειτουργίας του κυκλώματος είναι το εργαλείο SPICE οδηγούμενο από τα μοντέλα των τρανζίστορ που χρησιμοποιούμε και από τα αντίστοιχα χαρακτηριστικά των γραμμών διασύνδεσης (αντίσταση και χωρητικότητα). Χρησιμοποιώντας το SPICE είμαστε σίγουροι για τα αποτελέσματα των μετρήσεων μας αλλά πληρώνουμε την ακρίβεια αυτή με τους υπερβολικά αυξημένους χρόνους εξομοιώσεων. Παρόλο το κόστος σε χρόνο, επιλέξαμε τη μοντελοποίηση όλων των υπο-εξέταση κυκλωμάτων και την εξομοίωση τους σε SPICE στοχεύοντας σε όσο το δυνατόν πιο ρεαλιστικά αποτελέσματα. Ταυτόχρονα, σε πολλές περιπτώσεις χρησιμοποιήσαμε και εργαλεία σχεδίασης των εταιριών Synopsys και Cadence με σκοπό να εξετάσουμε τη συμπεριφορά των κυκλωμάτων όταν αυτά σχεδιαστούν με αυτόματο τρόπο. Σ' αυτές τις περιπτώσεις η διαδικασία ολοκληρώνεται σύντομα αφού τα εργαλεία εφαρμόζουν αποτελεσματικούς αλγορίθμους υπολογισμού της καθυστέρησης και της κατανάλωσης ισχύος σε επίπεδο λογικών πυλών χρησιμοποιώντας αφαιρετικά μοντέλα για τις ιδιότητες των πυλών.

Η καθυστέρηση ενός ψηφιακού κυκλώματος μετρείται συνήθως σε απόλυτες χρονικές μονάδες όπως ns ή ps ή στις αντίστοιχες τιμές MHz ή GHz που προκύπτουν μεταφράζοντας την περίοδο του κύκλου ρολογιού στην αντίστοιχη τιμή της συχνότητας. Ωστόσο, τα αποτελέσματα μας θα έχουν μεγαλύτερη αξία αν τα εκφράσουμε με ένα τρόπο ο οποίος θα είναι από τη μια ρεαλιστικός και από την άλλη όσο το δυνατόν ανεξάρτητος από τις λεπτομέρειες της τεχνολογίας κατασκευής και τις συνθήκες λειτουργίας του κυκλώματος. Έτσι, θα ήμασταν σε θέση να προβλέψουμε με σχετική σιγουριά την απόδοση του ίδιου κυκλώματος όταν κατασκευαστεί σε μια νέα τεχνολογία. Έχει καταδειχτεί από πολλούς ερευνητές πως η καθυστέρηση ενός κυκλώματος καλό είναι να εκφράζεται σε αντίστοιχες μονάδες FO4 (Fanout-of-4), οι οποίες αντιπροσωπεύουν την καθυστέρηση ενός αντιστροφέα που οδηγεί στην έξοδο του 4 αντιστροφείς του ίδιου μεγέθους (Σχήμα 2.1) [69], [64]. Συνεπώς, όταν θέλουμε να εκφράσουμε την καθυστέρηση ενός κυκλώματος αντί να αποτυπώσουμε την πραγματική απόλυτη τιμή της, κανονικοποιούμε την καθυστέρηση που μετρήσαμε στην αντίστοιχη τιμή

Σχήμα 2.1: Το κύκλωμα που χρησιμοποιείται για τη μέτρηση της τιμής του FO4 για μια δεδομένη τεχνολογία υλοποίησης.

του FO4 της τεχνολογίας που χρησιμοποιούμε. Για παράδειγμα, αν ένα κύκλωμα το οποίο σχεδιάστηκε σε μια τεχνολογία 180nm, στην οποία η τιμή του FO4 ισούται με 90ps, έχει καθυστέρηση ίση με 990ps θεωρούμε πως η καθυστέρηση του είναι ίση με 11FO4.

Μ' αυτόν το τρόπο μπορούμε να προβλέψουμε την ταχύτητα του ίδιου κυκλώματος όταν αυτό κατασκευαστεί σε μια νέα τεχνολογία γνωρίζοντας απλά την αντίστοιχη τιμή του FO4 για την τεχνολογία αυτή. Αυτό είναι εφικτό γιατί η τιμή του FO4 για μια δεδομένη τεχνολογία κατασκευής παρουσιάζει ελάχιστες μεταβολές στις μικρές αλλαγές της τάσης τροφοδοσίας (5-15%) και της θερμοκρασίας λειτουργίας του κυκλώματος. Επομένως, είμαστε σε θέση να εκφράσουμε την καθυστέρηση του κυκλώματος με έναν τρόπο που είναι όσο το δυνατόν ανεξάρτητος από τις λεπτομέρειες της τεχνολογίας. Στη γενική περίπτωση για δύο διαφορετικές τεχνολογίες του ίδιου μεγέθους οι τιμές του FO4 δεν πρέπει να διαφέρουν πολύ. Αυτό δεν ισχύει πάντα, ιδίως όταν οι εταιρείες προχωρούν σε πολύ επιθετικές βελτιστοποιήσεις της τεχνολογίας κατασκευής πετυχαίνοντας 30-50% μικρότερες τιμές της καθυστέρησης FO4. Για παράδειγμα η τεχνολογία 180nm της Intel [54] έχει τιμή FO4 τα 50ps ενώ η τυπική τιμή αυτού του μεγέθους είναι τα 90ps¹.

Οι σχεδιασμοί που θα παρουσιάσουμε στη συνέχεια του κειμένου πραγματοποιήθηκαν στην τεχνολογία της UMC με τυπικό μήκος καναλιού τα 130nm τα δεδομένα της οποίας μας έγιναν διαθέσιμα μέσω του Europractice. Για να αποκτήσουμε λοιπόν μια αίσθηση των δυνατοτήτων της τεχνολογίας αυτής παρουσιάζουμε στα διαγράμματα του σχήματος 2.2 τις τιμές του FO4 για τρεις εκδοχές των συνθηκών κατασκευής (TT - Typical, FF - Fast, SS - Slow) μεταβάλλοντας την τάση τροφοδοσίας και τη θερμοκρασία λειτουργίας του κυκλώματος. Οι τιμές αυτές προέκυψαν μετά από εξομοιώσεις σε SPICE του κυκλώματος του σχήματος 2.1 θεωρώντας πως οι αντιστροφείς είναι υλοποιημένοι με τρανζίστορ τυπικών επιδόσεων $V_{th} = 305mV$, όπου το σχετικό μέγεθος μεταξύ των PMOS και NMOS τρανζίστορ είναι 2 προς 1. Σημειώνουμε πως οι τιμές του FO4 είναι διαφορετικές αν μετρούμε την καθυστέρηση ανόδου ή καθόδου του αντιστροφέα. Πάντως, ο μέσος όρος των δύο περιπτώσεων αρκεί για την εξαγωγή της τιμής

¹Αυτό που πρέπει να απασχολεί το σχεδιαστή είναι η δημιουργία ενός κυκλώματος που θα έχει μειωμένη καθυστέρηση επειδή υιοθετεί μια απλούστερη και πιο αποδοτική κυκλωματική τοπολογία και όχι επειδή η τεχνολογία που έχει στη διάθεση του είναι γρηγορότερη.

Σχήμα 2.2: Οι τιμές του FO4 σε ps για διάφορες τιμές (α) της τάσης τροφοδοσίας και (β) της θερμοκρασίας λειτουργίας κυκλώματος συνεκτιμώντας την επίδραση τριών διαφορετικών συνθηκών κατασκευής.

του FO4. Στην τυπική περίπτωση κατασκευής και χρησιμοποιώντας ως τάση τροφοδοσίας τα 1.2V, η οποία είναι επιλεγμένη από τον κατασκευαστή, η τιμή του FO4 ορίζεται στα 62ps. Η τιμή αυτή αναμένεται να είναι χαμηλότερη για τα πιο γρήγορα τρανζίστορ της ίδιας τεχνολογίας (Low-Vth) ενώ θα αυξηθεί με τη χρήση των πιο αργών τρανζίστορ με μεγαλύτερη τιμή κατωφλίου (High-Vth) που στοχεύουν στη μείωση των ρευμάτων διαρροής. Η αλλαγή της θερμοκρασίας δεν επηρεάζει την τιμή του FO4 περισσότερο από 5%, όπως παρατηρούμε από τα σχήματα, ενώ το ίδιο συμβαίνει όταν μεταβάλλουμε την τάση τροφοδοσίας στην περιοχή τιμών 0.9-1.3V ($\pm 15\%$ της τυπικής τιμής της τάσης τροφοδοσίας). Σε γενικές γραμμές η τιμή του FO4 σε ps για μια τεχνολογία μπορεί να υπολογιστεί χρησιμοποιώντας εμπειρικές εξισώσεις [64], [26]. Δεν πρέπει όμως να ξεχνάμε ότι οι προσεγγίσεις αυτές χρησιμεύουν απλά για να αποκτήσουμε μια πρώτη αίσθηση της τεχνολογίας που χρησιμοποιούμε, ενώ οι ακριβείς τιμές πρέπει να υπολογιστούν είτε μέσω εξομοιώσεων σε SPICE είτε μέσω μετρήσεων σε κατασκευασμένα κυκλώματα.

Πέρα από την καθυστέρηση του κυκλώματος, ενδιαφερόμαστε επίσης για την κατανάλωση ισχύος των κυκλωμάτων μας, η οποία μπορεί να μετρηθεί ή να εκτιμηθεί με διάφορους τρόπους. Για τις εφαρμογές των μικροεπεξεργαστών υψηλών επιδόσεων μας ενδιαφέρει κυρίως η μέγιστη τιμή της κατανάλωσης ισχύος του κυκλώματος, αφού αυτή καθορίζει το κόστος απαγωγής της θερμότητας και του πακεταρίσματος του ολοκληρωμένου. Αντίθετα, στις φορητές συσκευές όπου μας ενδιαφέρει η επιμήκυνση της διάρκειας ζωής της μπαταρίας, η μέση τιμή κατανάλωσης ισχύος πρέπει να είναι ο στόχος του σχεδιαστή. Μπορούμε από τη μέση κατανάλωση ισχύος να υπολογίσουμε τη μέγιστη και αντίστροφα αν λάβουμε υπόψη στους υπολογισμούς μας τον κύκλο ρολογιού, την κατανάλωση εξαιτίας των ρευμάτων διαρροής και τις μεταβάσεις στις εσωτερικές γραμμές του κυκλώματος. Για να διατηρήσουμε την απλότητα και τη γενικότητα στις μετρήσεις μας χρησιμοποιούμε ως μετρική της κατανάλωσης ισχύος τη μέση ενέργεια που απαιτείται ανά λειτουργία του κυκλώματος. Έτσι, για παράδειγμα, γνωρίζοντας ότι το κύκλωμα μας καταναλώνει 2pJ ανά λειτουργία μπορούμε εύκολα να υπολογίσουμε ότι η κατανάλωση ισχύος του είναι ίση με 1nW όταν αυτό υποθετικά λειτουργεί στα 500MHz. Οι τιμές της ενέργειας ανά λειτουργία, οι οποίες θα παρουσιαστούν στα επόμενα κεφάλαια, προέκυψαν και αυτές μετά από μετρήσεις σε Spice. Σ' αυτήν την περίπτωση τα κυκλώματα οδηγούνται σε κάθε κύκλο από ένα

τυχαίο διάνυσμα εισόδου κατά τη διάρκεια του οποίου καταγράφουμε τη συνολική ενέργεια που μεταφέρθηκε. Αυτό το πετυχαίνουμε, ολοκληρώνοντας σε αυτό το χρονικό διάστημα την τιμή του ρεύματος που διέρρευσε μέσω της πηγής τροφοδοσίας του κυκλώματος. Η μέση ενέργεια ανά λειτουργία προκύπτει από το μέσο όρο των τιμών που συλλέξαμε για όλα τα διανύσματα.

2.2 Γεωμετρικός προγραμματισμός

Πολλοί αλγόριθμοι έχουν προταθεί τα τελευταία χρόνια για τη βελτιστοποίηση των κυκλωμάτων και τον καθορισμό των παραμέτρων που καθορίζουν την απόδοση τους [109], [25]. Στην πλειοψηφία τους οι αλγόριθμοι αυτοί βασίζονται σε κάποιο ευρυστικό τρόπο ψαξίματος των μεγεθών των πυλών και δεν εξασφαλίζουν τη βέλτιστη λύση παρά μόνο για πολύ μικρά κυκλώματα. Ωστόσο, ο χρόνος εκτέλεσης τους είναι πολύ μικρός και αποτελούν αξιόπιστες επιλογές για πολύ μεγάλα κυκλώματα. Η βασική αρχή λειτουργίας των μεθόδων αυτών είναι ότι υπολογίζουν σε κάθε βήμα του αλγορίθμου την ευαισθησία της αλλαγής κάποιας από τις παραμέτρους ως προς το στοχευόμενο αποτέλεσμα σε καθυστέρηση και ενέργεια $\frac{\partial E/\partial x}{\partial D/\partial x}$ και επιλέγουν με ευρυστικό τρόπο εκείνες τις λύσεις οι οποίες προσφέρουν το πιο σημαντικό όφελος συνολικά για το κύκλωμα [94], [152], [32].

Μια άλλη πρακτική η οποία κερδίζει συνεχώς έδαφος είναι η πραγματοποίηση της βελτιστοποίησης του κυκλώματος μέσω μαθηματικού προγραμματισμού [28], [15], [25]. Στην περίπτωση αυτή κάθε χαρακτηριστικό του κυκλώματος και των πυλών του πρέπει να εκφραστεί σαν μια συνάρτηση των μεταβλητών του προβλήματος. Η διαδικασία αυτή της αναγκαστικής μαθηματικής μοντελοποίησης μπορεί να αποκλίνει από τις πραγματικές τιμές των χαρακτηριστικών αυτών. Παρόλα αυτά, ακόμη και λάθη της τάξης του 10% είναι ανεκτά εφόσον το τελικό αποτέλεσμα που προκύπτει είναι καλύτερο από τις λύσεις οι οποίες δεν εξασφαλίζουν τη βέλτιστη λύση. Αυτή την προσέγγιση επιλέξαμε και εμείς λύνοντας το πρόβλημα επιλογής των μεγεθών των πυλών του κυκλώματος με στόχο τη μείωση της απαιτούμενης ενέργειας για μια δεδομένη τιμή της καθυστέρησης με τη βοήθεια του γεωμετρικού προγραμματισμού. Οι βασικές αρχές αυτής της διαδικασίας και το εργαλείο που υλοποιήσαμε θα περιγραφούν στη συνέχεια.

Ένα τυπικό μη γραμμικό πρόβλημα βελτιστοποίησης έχει την εξής μορφή [14]:

$$\begin{array}{ll} \text{Ελαχιστοποίηση} & f(x) \\ \text{σύμφωνα με} & g_i(x) \leq 1, \quad i = 1, \dots, m \\ & h_i(x) = 1, \quad i = 1, \dots, k \end{array}$$

Το γενικό πρόβλημα βελτιστοποίησης αναζητά εκείνες τις τιμές των ελεύθερων μεταβλητών οι οποίες ελαχιστοποιούν την τιμή της στοχευόμενης συνάρτησης ικανοποιώντας ταυτόχρονα τους περιορισμούς ανισότητας g_i και ισότητας h_i . Κάθε σημείο στο χώρο των μεταβλητών x που ικανοποιεί τους περιορισμούς καλείται δυνατό σημείο (feasible point), το σύνολο των οποίων ορίζει το σύνολο των δυνατών λύσεων (feasible set). Φυσικά, η λύση στο πρόβλημα βελτιστοποίησης πρέπει να ανήκει στο σύνολο αυτό. Αν το σύνολο αυτό είναι κενό τότε το πρόβλημα βελτιστοποίησης δεν έχει λύση (infeasible). Ένα γενικού τύπου πρόβλημα βελτιστοποίησης είναι πολύ δύσκολο να λυθεί αν δεν κάνουμε καμία παραδοχή για τις ιδιότητες της στοχευόμενης συνάρτησης και των περιορισμών. Αν και υπάρχουν αρκετοί αλγόριθμοι οι οποίοι μπορούν να αντιμετωπίσουν γενικές συναρτήσεις με σχετική επιτυχία είναι στην

Σχήμα 2.3: Η διαφορά μεταξύ κυρτών και μη κυρτών συναρτήσεων.

πλειονότητα τους αργοί για μεγάλα προβλήματα και μπορεί να παγιδευτούν σε τοπικά ελάχιστα. Στο σχήμα 2.3(a) φαίνεται πως ο αλγόριθμος βελτιστοποίησης μπορεί να εγκλωβιστεί σε ένα τοπικό ελάχιστο συγκλίνοντας σε μια λάθος τιμή. Αν η αρχική τιμή που δίνεται στον ελαχιστοποιητή είναι το σημείο 1 τότε η τελική λύση στην οποία θα συγκλίνει ο αλγόριθμος είναι η ζητούμενη. Αν η αρχική πρόβλεψη όμως είναι το σημείο 2, τότε είναι πολύ πιθανό ο αλγόριθμος βελτιστοποίησης να σταματήσει στο σημείο B, το οποίο, αν και τοπικό ελάχιστο, δεν είναι το επιθυμητό ολικό ελάχιστο της συνάρτησης.

Μια εναλλακτική προσέγγιση είναι να υλοποιήσουμε αλγορίθμους βελτιστοποίησης για συγκεκριμένες περιπτώσεις προβλημάτων εκμεταλλευόμενοι τις ιδιότητες των περιπτώσεων αυτών με τον καλύτερο δυνατό τρόπο. Μια κατηγορία τέτοιων προβλημάτων είναι τα κυρτά προβλήματα βελτιστοποίησης (convex optimization problems) [14]. Σε αυτά τα προβλήματα τόσο η στοχευόμενη συνάρτηση όσο και το σύνολο των δυνατών λύσεων πρέπει να είναι κυρτό. Μια κυρτή συνάρτηση έχει την ιδιότητα ότι μια γραμμή η οποία θα ενώνει δύο σημεία της θα βρίσκεται πάντα πάνω από αυτήν και αναπαρίσταται γραφικά στο διάγραμμα του σχήματος 2.3(β). Με άλλα λόγια, μια κυρτή συνάρτηση ικανοποιεί πάντα τη συνθήκη $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$ για κάθε x, y που ανήκουν στο πεδίο ορισμού της f και $0 \leq \lambda \leq 1$. Η βασική ιδιότητα των κυρτών συναρτήσεων είναι ότι κάθε τοπικό ελάχιστο είναι ταυτόχρονα και ολικό. Για να μπορεί ο χώρος λύσεων να είναι κυρτός πρέπει όλες οι συναρτήσεις περιορισμού που τον ορίζουν να είναι και αυτές κυρτές. Τότε κάθε γραμμή που ενώνει δύο σημεία του κυρτού χώρου βρίσκεται και αυτή μέσα στο χώρο.

Βέβαια πολλά από τα προβλήματα που θέλουμε να λύσουμε δεν είναι εξαρχής εκφρασμένα διαμέσου κυρτών συναρτήσεων. Αυτό το εμπόδιο μπορεί σε πολλές περιπτώσεις να ξεπεραστεί με την εμφάνιση βοηθητικών μεταβλητών και με την εφαρμογή μαθηματικών μετασχηματισμών. Αυτό ακριβώς συμβαίνει και στην περίπτωση του γεωμετρικού προγραμματισμού, του οποίου η βασική δομή, αν και ταιριάζει στη μοντελοποίηση της καθυστέρησης και της ενέργειας των κυκλωμάτων, δεν οδηγεί σε κυρτά προβλήματα βελτιστοποίησης [51], [81]. Παρά το γεγονός αυτό, η μετατροπή τους μπορεί πολύ εύκολα να προκύψει μετά από μια πολύ απλή αλλαγή μεταβλητών [14], [139].

Τα προβλήματα γεωμετρικού προγραμματισμού ακολουθούν και αυτά τη γενική μορφή ενός

προβλήματος βελτιστοποίησης. Μόνο που σε αυτή την περίπτωση [14], τόσο η στοχευόμενη συνάρτηση f όσο και οι περιορισμοί g_i και h_i πρέπει να είναι ποζυώνυμα (posynomials). Ένα ποζυώνυμο n μεταβλητών x_i έχει την εξής μορφή:

$$f(x) = \sum_{k=1}^K c_k x_1^{a_{1k}} x_2^{a_{2k}} \dots x_n^{a_{nk}}$$

Οι συντελεστές c_k και a_{ij} είναι σταθερές ενώ τόσο τα c_k όσο και οι μεταβλητές x_i κάθε όρου πρέπει να είναι θετικοί αριθμοί. Για παράδειγμα η συνάρτηση $3x^2yz^{-12} + 2z\sqrt{x/y}$ είναι ένα ποζυώνυμο. Επίσης, οι όροι που αποτελούν το άθροισμα κάθε ποζυωνύμου ονομάζονται μονόνομα (monomial) όπως το $3x^2yz^{-12}$ του παραδείγματος. Η γρήγορη επίλυση τέτοιων προβλημάτων επιτυγχάνεται με τη μετατροπή τους σε κυρτά προβλήματα βελτιστοποίησης τα οποία μπορούν να λυθούν πολύ αποδοτικά. Η μετατροπή πραγματοποιείται με μια απλή αλλαγή μεταβλητής όπου κάθε x_i γράφεται ως e^{z_i} . Αυτό λοιπόν που μας απασχολεί είναι ο τρόπος που θα διαμορφώσουμε το πρόβλημα της βελτιστοποίησης των κυκλωμάτων ως προς την ενέργεια και την καθυστέρηση, ώστε να αποκτήσει τη μορφή που απαιτεί ο γεωμετρικός προγραμματισμός. Η μοντελοποίηση του προβλήματος που θέλουμε να λύσουμε θα είναι εφικτή, μόνο αν εξασφαλίσουμε ότι, όλες οι συναρτήσεις που εμφανίζονται τόσο για την στοχευόμενη συνάρτηση όσο και για συναρτήσεις περιορισμών, μπορούν να αναπαρασταθούν σαν ποζυώνυμα.

2.3 Μοντέλα καθυστέρησης

Για να μπορέσουμε να χρησιμοποιήσουμε τον γεωμετρικό προγραμματισμό αποσκοπώντας στη βελτιστοποίηση των κυκλωμάτων μας, χρειαζόμαστε την μαθηματική μοντελοποίηση της καθυστέρησης, της ενέργειας και του εμβαδού της κάθε πύλης, συναρτήσει του μεγέθους των τρανζίστορ που την αποτελούν. Για την καθυστέρηση, η πιο απλή, αλλά παρόλα αυτά πολύ χρήσιμη τεχνική, στηρίζεται στη μοντελοποίηση της με τη χρήση ισοδύναμων δικτύων αντιστάσεων και πυκνωτών, γνωστά και ως δίκτυα RC [65]. Θα περιγράψουμε τη διαδικασία της μοντελοποίησης της καθυστέρησης των πυλών συναρτήσει του μεγέθους των τρανζίστορ, ξεκινώντας από την καθυστέρηση του αντιστροφέα και στη συνέχεια θα τη γενικεύσουμε και στις υπόλοιπες λογικές πύλες.

Στο σχήμα 2.4 φαίνεται η CMOS υλοποίηση του αντιστροφέα μαζί με το αντίστοιχο δίκτυο RC που χρησιμοποιείται για τον υπολογισμό της καθυστέρησης ανόδου της εξόδου, όπου η χωρητικότητα εξόδου C_L και η παρασιτική χωρητικότητα του αντιστροφέα C_{par} φορτίζονται μέσω της ισοδύναμης αντίστασης R_{drive} . Στην πραγματικότητα η τιμή της αντίστασης R_{drive} και της χωρητικότητας C_{par} εξαρτώνται από τη γεωμετρία των τρανζίστορ, την τάση που εφαρμόζεται στην είσοδο, την τάση τροφοδοσίας, την κινητικότητα του τρανζίστορ, και από την τάση κατωφλίου. Πάντως, για τον υπολογισμό της καθυστέρησης, θεωρούμε ότι το P τρανζίστορ συμπεριφέρεται σαν μια αντίσταση R_{drive} η τιμή της οποία εξαρτάται μόνο από τη γεωμετρία του και υπολογίζεται μέσω της σχέσης:

$$R_{drive} = R_{sq} \frac{L}{W}$$

Η τιμή του R_{sq} , η οποία κυμαίνεται στα 51 ΚΩ·μm, έχει υπολογιστεί μετά από εξομοιώσεις και

Σχήμα 2.4: Ο αντιστροφέας και η μοντελοποίηση με δίκτυα RC της καθυστέρησης ανόδου.

αναφέρεται στη στοιχειώδη αντίσταση του P τρανζίστορ. Αντίστοιχα για το N τρανζίστορ η τιμή του R_{sq} προκύπτει ότι είναι ίση με $17 \text{ K}\Omega \cdot \mu\text{m}^2$. Επίσης, για την παρασιτική χωρητικότητα C_{par} θεωρούμε ότι η τιμή της είναι σταθερή στο χρόνο και εξαρτάται μόνο από τη χωρητικότητα των περιοχών drain και source των τρανζίστορ που συνδέονται στην έξοδο. Οι χωρητικότητες αυτές καθορίζονται κυρίως από το εύρος των τρανζίστορ. Επομένως, για να εκφράσουμε τη συσχέτιση της χωρητικότητας αυτής με τα μεγέθη των τρανζίστορ γράφουμε ότι $C_{par} = \gamma C_{in}$. Η τιμή του C_{in} που αναπαριστά την χωρητικότητα εισόδου της πύλης, είναι άμεσα εξαρτώμενη από το μέγεθος των τρανζίστορ και ισούται με

$$C_{in} = C_g (W_p + W_n)$$

όπου με C_g συμβολίζουμε την ισοδύναμη χωρητικότητα της πύλης ανά μm . Για τον υπολογισμό της καθυστέρησης, η τιμή του $C_g/\mu\text{m}$ είναι σχεδόν σταθερή για τις περισσότερες τεχνολογίες κατασκευής κυμαίνεται μεταξύ 1.5 και $2 \text{ fF}/\mu\text{m}$, ενώ για την τεχνολογία των 130 nm που χρησιμοποιούμε προέκυψε ότι είναι ίση με $1.35 \text{ fF}/\mu\text{m}$.

Ο λόγος της στοιχειώδους αντίστασης για τα P και N τρανζίστορ είναι ίσος με $51/17 = 3$. Συνεπώς, αν φροντίσουμε το μέγεθος του P τρανζίστορ να είναι 3 φορές μεγαλύτερο από αυτό του N, διατηρώντας το ίδιο μήκος του καναλιού για τα δύο τρανζίστορ, θα πετύχουμε ίσες αντιστάσεις και για τα δύο, με αποτέλεσμα οι χρόνοι ανόδου και καθόδου να είναι ίσοι. Κάτι τέτοιο όμως, θα αύξανε πολύ την χωρητικότητα εισόδου του αντιστροφέα και την παρασιτική του χωρητικότητα, επηρεάζοντας αρνητικά την καθυστέρησή του. Για το λόγο αυτό, είναι κοινή πρακτική να θεωρούμε ότι το εύρος των τρανζίστορ P και N ακολουθεί το λόγο 2:1, $W_p = 2W_n = 2W$. Η μεταβλητή W θεωρούμε πως είναι παράμετρος του προβλήματος δίνοντας αντίστοιχα ότι $C_{in} = 3WC_g$. Με αυτήν την επιλογή, αν και η καθυστέρηση για τις μεταβάσεις ανόδου και καθόδου θα είναι διαφορετική, εξασφαλίζουμε ότι η διαφορά αυτή θα κινείται σε λογικά πλαίσια. Για το λόγο αυτό, στην περίπτωση του αντιστροφέα είναι αρκετό να εξετάσουμε την καθυστέρηση

²Οι τιμές αυτές έχουν νόημα αποκλειστικά για τον υπολογισμό της καθυστέρησης και δεν πρέπει να χρησιμοποιούνται σε άλλους υπολογισμούς.

Σχήμα 2.5: Η πύλη NOR και τα αντίστοιχα δίκτυα RC για τον υπολογισμό της καθυστέρησης ανόδου και καθόδου διαμέσου της εισόδου A της πύλης.

της μιας από τις δύο μεταβάσεις. Η καθυστέρηση λοιπόν του αντιστροφέα όπως προκύπτει από το μοντέλο του Elmore [47] για το δίκτυο RC είναι ίση με

$$D = R_{drive} (C_L + C_{par})$$

Αν διαιρέσουμε και πολλαπλασιάσουμε ταυτόχρονα τη σχέση αυτή με C_{in} προκύπτει ότι

$$D = R_{drive} C_{in} \left(\frac{C_L}{C_{in}} + \frac{C_{par}}{C_{in}} \right) = R_{drive} C_{in} \left(\frac{C_L}{C_{in}} + \gamma \right)$$

Μπορούμε εύκολα να διαπιστώσουμε ότι ο όρος $R_{drive} C_{in} = 3 L R_{sq} C_g$, είναι ανεξάρτητος του εύρους των τρανζίστορ και εξαρτάται μόνο από τα δεδομένα της τεχνολογίας και την τοπολογία της πύλης. Στη συνέχεια η τιμή $3 L R_{sq} C_g$ θα καλείται τ_{INV} .

Μπορούμε να ακολουθήσουμε την ίδια διαδικασία για τον υπολογισμό της καθυστέρησης ανόδου και καθόδου της πύλης NOR που φαίνεται στο σχήμα 2.5. Στο ίδιο σχήμα απεικονίζονται τα ισοδύναμα δίκτυα RC για τον υπολογισμό της καθυστέρησης ανόδου και καθόδου αντίστοιχα, θεωρώντας πως οι μεταβάσεις στην έξοδο της πύλης προκαλούνται αποκλειστικά από την αλλαγή της κατάστασης της εισόδου A της πύλης. Για την περίπτωση μιας μετάβασης από το Vdd στο 0 (falling transition) η διαδικασία είναι ακριβώς ίδια με αυτή του αντιστροφέα. Σε αυτή την περίπτωση θεωρούμε ότι για την παρασιτική χωρητικότητα στην έξοδο $C_{pA} = \gamma_1 C_{in}$, ενώ η χωρητικότητα εισόδου υπολογίζεται σε $C_{in} = 5 W C_g$. Έτσι προκύπτει ότι

$$D_F = R_{drive(n)} C_{in} \left(\frac{C_L}{C_{in}} + \gamma_1 \right) = 5 L R_{sq} C_g \left(\frac{C_L}{C_{in}} + \gamma_{NOR-Fall} \right)$$

Αντίστοιχα για το δεύτερο δίκτυο RC, που μοντελοποιεί την μετάβαση από 0 σε Vdd μπορούμε

να υπολογίσουμε ότι η καθυστέρηση είναι ίση με $R_{drive(p)} C_{pB} + 2R_{drive(p)}(C_{pA} + C_L)$, όπου $C_{pB} = \gamma_2 C_{in}$. Ουσιαστικά η τιμή της παρασιτικής χωρητικότητας C_{pB} εξαρτάται μόνο από τη γεωμετρία και τη φυσική σχεδίαση (layout) των P τρανζίστορ και δεν είναι άμεσα εξαρτώμενη από την χωρητικότητα εισόδου. Παρόλα αυτά, αν θεωρήσουμε ότι τα μεγέθη των τρανζίστορ έχουν επιλεγεί κατάλληλα ώστε να επιτυγχάνουν αντίσταση ίση με αυτή του 2:1 αντιστροφέα, τότε μπορούμε με ασφάλεια να θεωρήσουμε ότι η τιμή του C_{pB} είναι ανάλογη της χωρητικότητας εισόδου. Έτσι, για την περίπτωση της καθυστέρησης ανόδου προκύπτει ότι

$$D_R = R_{drive} C_{in} \left(\frac{C_L}{C_{in}} + 2\gamma_1 + \gamma_2 \right) = 5 L R_{sq} C_g \left(\frac{C_L}{C_{in}} + \gamma_{NOR-Rise} \right)$$

Τα βασικά συμπεράσματα από αυτή τη διαδικασία είναι ότι το κομμάτι της καθυστέρησης της πύλης NOR που οφείλεται στη χωρητικότητα εξόδου C_L είναι σταθερό και ανεξάρτητο από ποια μετάβαση εξετάζουμε. Αυτό που διαφοροποιεί την καθυστέρηση της πύλης είναι το κομμάτι εκείνο, το οποίο είναι ανεξάρτητο από την χωρητικότητα εξόδου και εξαρτάται αποκλειστικά από τη διασύνδεση των τρανζίστορ της πύλης και τις εσωτερικές παρασιτικές χωρητικότητες. Για την πύλη NOR ο στοιχειώδης χρόνος $\tau_{NOR} = 5 L R_{sq} C_g$ είναι κατά 5/3 φορές μεγαλύτερος από αυτόν του αντίστοιχου αντιστροφέα ($\tau_{NOR}/\tau_{INV} = 5/3$) ανεξάρτητα από την τεχνολογία υλοποίησης. Αντίστοιχα, για μια πύλη NAND προκύπτει ότι $\tau_{NAND} = 4 L R_{sq} C_g$ δείχνοντας ότι η πύλη αυτή είναι κατά 4/3 φορές πιο αργή από τον αντιστροφέα, τουλάχιστον όσον αφορά στο κομμάτι της καθυστέρησης που εξαρτάται από την χωρητικότητα εξόδου. Αυτή είναι και η βασική παρατήρηση πάνω στην οποία βασίστηκε η πολύ χρήσιμη θεωρία του Logical Effort [153]. Οι εμπνευστές της θεωρίας αυτής φρόντισαν να κανονικοποιήσουν την καθυστέρηση οποιασδήποτε πύλης με αυτή του βασικού αντιστροφέα συμψηφίζοντας έτσι την επίδραση της τεχνολογίας στα μοντέλα της καθυστέρησης των πυλών μόνο σε ένα πολλαπλασιαστικό παράγοντα (τ) ο οποίος μπορεί να εξαληφθεί αρχικά από τους υπολογισμούς και να επανεμφανισθεί μόνο στο τέλος.

Σύμφωνα με τη θεωρία του Logical Effort η καθυστέρηση της πύλης ή καλύτερα κάθε εισόδου της πύλης μπορεί να εκφραστεί ως εξής³:

$$D = \tau(g h + p)$$

Οι παράμετροι που καθορίζουν την τιμή της καθυστέρησης ορίζονται ως εξής:

- Η βασική σταθερά χρόνου τ είναι η καθυστέρηση του τυπικού αντιστροφέα που θεωρούμε για τη σχεδίαση μας όταν αυτός δεν οδηγεί κάποιο φορτίο. Ο τυπικός αντιστροφέας δεν είναι ανάγκη να έχει τη βασική αναλογία 2:1 για τα τρανζίστορ P και N. Η αναλογία αυτή πάντως προτιμάται παραδοσιακά.
- Η ηλεκτρική προσπάθεια (electrical effort) h είναι ο λόγος της χωρητικότητας εξόδου προς την χωρητικότητα εισόδου της πύλης $h = C_L/C_{in}$.
- Η λογική προσπάθεια g (logical effort). Η τιμή αυτή περιγράφει πόσο πιο δύσκολα μπορεί μια πύλη με δεδομένο μέγεθος να οδηγήσει μια χωρητικότητα εξόδου σε σχέση με τον τυπικό

³Για περισσότερη ακρίβεια θα ήταν καλύτερα να διαχωρίσουμε την καθυστέρηση της πύλης ανάλογα με την κατεύθυνση της μετάβασης, άνοδος ή κάθοδος, στην περίπτωση που δεν είναι πλήρως συμμετρικές. Τότε για κάθε είσοδο θα χρειαζόμασταν δύο τιμές για το λογικό φόρτο g και την παρασιτική καθυστέρηση p .

Σχήμα 2.6: Οι τιμές της καθυστέρησης (α) ενός αντιστροφέα και (β) της εσωτερικής εισόδου της πύλης NAND για διάφορες τιμές του C_{out}/C_{in} σε τυπικές συνθήκες κατασκευής με θερμοκρασία 70° και τάση τροφοδοσίας 1.2V.

αντιστροφέα με την ίδια χωρητικότητα εισόδου. Η τιμή αυτή μπορεί να υπολογιστεί ως

$$g = \frac{R_{gate} C_{in, gate}}{R_{typ-inv} C_{typ-inv}}$$

- Η παρασιτική καθυστέρηση p . Η μεταβλητή αυτή συνοψίζει την καθυστέρηση της πύλης που οφείλεται στις εσωτερικές της χωρητικότητες και στη δομή της και είναι ανεξάρτητη από την χωρητικότητα που οδηγεί η πύλη στην έξοδο της.

Μπορούμε εύκολα να διαπιστώσουμε πως αυτό το μοντέλο της καθυστέρησης είναι ένα ποζυώνυμο συναρτήσεως της χωρητικότητας εισόδου και εξόδου και έτσι μπορεί να χρησιμοποιηθεί στη βελτιστοποίηση με βάση το γεωμετρικό προγραμματισμό. Οι τιμές των g και p αν και μπορούν να υπολογιστούν αναλυτικά, όπως δείξαμε, είναι καλύτερο να υπολογιστούν μέσω εξομοιώσεων. Το μοντέλο αυτό για τιμές της ηλεκτρικής προσπάθειας οι οποίες κυμαίνονται μεταξύ 1 και 10 και δεν αφορούν υπερβολικά μεγάλες χωρητικότητες⁴ δίνει αποτελέσματα με επαρκή ακρίβεια. Αυτό επιβεβαιώνεται από το διάγραμμα του σχήματος 2.6 που αναπαριστά την καθυστέρηση ενός αντιστροφέα και μιας πύλης NAND, οι οποίες σχεδιάστηκαν στην διαθέσιμη τεχνολογία των 130nm, συναρτήσεως της ηλεκτρικής προσπάθειας. Στην πειραματική διαδικασία που ακολουθήσαμε, μοντελοποιούμε την καθυστέρηση της πύλης χωριστά για την κάθε είσοδο της. Η λογική προσπάθεια και η παρασιτική καθυστέρηση που υπολογίσαμε μετά από εξομοιώσεις για την κάθε είσοδο των βασικών πυλών και που χρησιμοποιούμε στους σχεδιασμούς μας φαίνονται στον πίνακα 2.1. Στον πίνακα αυτό οι εισοδοί των πυλών είναι ταξινομημένες λεξικογραφικά και αυτές που είναι τοποθετημένες πιο κοντά στην έξοδο της πύλης δηλώνονται με το γράμμα A.

Συνήθως το γινόμενο $f = gh$ καλείται προσπάθεια του κάθε επιπέδου (stage effort) και έχει δείχθει πως σε καλά σχεδιασμένα κυκλώματα μπορούμε με σχετική ασφάλεια να

⁴Αν και η ηλεκτρική προσπάθεια στην περίπτωση του 20/10fF είναι ίση με την περίπτωση του 300/150fF η καθυστέρηση των αντίστοιχων πυλών μπορεί να διαφέρει σημαντικά.

Πίνακας 2.1: Η λογική προσπάθεια και η παρασιτική χωρητικότητα των πυλών που χρησιμοποιήσαμε στα πειράματά μας για την τεχνολογία των 130nm. Η βασική σταθερά χρόνου τ καθορίστηκε από ένα αντιστροφέα 2:1 και ισούται με 12.2ps.

Gate	Input	Logical Effort	Parasitic Delay
INV	A	1	1.02
NAND	A	1.21	1.92
	B	1.22	1.48
NOR	A	1.60	2.48
	B	1.53	1.79
OAI	A	1.75	3.27
	B	1.69	2.62
	C	1.23	2.55
AOI	A	1.76	3.47
	B	1.79	3.11
	C	1.54	2.41
TGMUX	In	1.74	3.80
	Sel	1.72	2.76
XOR	A	1.36	5.13
	B	1.73	3.82

θεωρήσουμε πως η τιμή του κυμαίνεται μεταξύ 3 και 6. Θεωρούμε πως σε ένα τέτοιο κύκλωμα η οδήγηση πολύ μεγάλων χωρητικοτήτων στηρίζεται πέρα από τη λογική του κυκλώματος στην προσθήκη επιπλέον ενισχυτικών στοιχείων που είναι συνήθως απλοί αντιστροφείς. Οι επιπλέον αντιστροφείς από τη μια προσφέρουν το επιπλέον ρεύμα που απαιτείται για να οδηγηθούν οι μεγάλες χωρητικότητες και από την άλλη μειώνουν την χωρητικότητα που “βλέπουν” στην έξοδο τους οι λογικές πύλες του κυκλώματος. Αυτό έχει ως αποτέλεσμα να μειώνεται η καθυστέρηση των πυλών αυτών και να αποκτούμε μεγαλύτερη ελευθερία στον καθορισμό του μεγέθους τους ώστε να πετύχουμε αποδοτικότερες λύσεις από πλευράς ενέργειας. Αν και φαίνεται κατ’ αρχήν παράξενο πως μπορούμε να θεωρήσουμε την τιμή f σταθερή για οποιοδήποτε κύκλωμα, ωστόσο κάτι τέτοιο έχει μια πολύ καλά ορισμένη φυσική σημασία. Αν θεωρήσουμε για μια πύλη μια δεδομένη χωρητικότητα εξόδου C_L , τότε η τιμή του f είναι ίση με $g \frac{C_L}{C_{in}}$. Για να διατηρήσουμε την τιμή του λόγου αυτού σταθερή, αναγκάζουμε τις σύνθετες πύλες με μεγαλύτερη λογική προσπάθεια να έχουν μικρότερη χωρητικότητα εισόδου σε σχέση με τις πιο απλές πύλες (μικρότερη λογική προσπάθεια). Σ’ αυτές αναθέτουμε μεγαλύτερη χωρητικότητα εφόσον μπορούν να μεταφέρουν αποτελεσματικότερα το απαραίτητο φορτίο στην έξοδο τους εξαιτίας της μικρότερης λογικής προσπάθειας. Με αυτό τον τρόπο καταφέρνουμε να εξισορροπήσουμε καλύτερα και την κατανάλωση ισχύος. Η προσέγγιση αυτή βοηθά διαισθητικά το σχεδιαστή να επιλέξει κατάλληλα μεγέθη για τις πύλες του κυκλώματός του, ενώ οι βέλτιστες τιμές της προσπάθειας του κάθε σταδίου καθορίζονται μόνο μετά από την εκτέλεση ολόκληρης της βελτιστοποίησης. Βέβαια, η πράξη δείχνει ότι ακόμη και μια παρέκκλιση της τάξης του 10% από τη βέλτιστη τιμή της προσπάθειας του κάθε σταδίου επηρεάζει ελάχιστα τη συνολική καθυστέρηση του μονοπατιού.

Το μοντέλο αυτό της καθυστέρησης έχει βέβαια και κάποιους περιορισμούς. Πιο

Σχήμα 2.7: Η επίδραση των γραμμών διασύνδεσης στην καθυστέρηση της πύλης. (α) Η αναπαράσταση των φυσικών χωρητικότητων και της αντίστασης και (β) το αντίστοιχο π δίκτυο RC που χρησιμοποιείται για τη μοντελοποίηση της καθυστέρησης.

συγκεκριμένα, δεν συμπεριλαμβάνει το χρόνο ανόδου ή καθόδου του σήματος της εισόδου κατά τη διάρκεια της μετάβασης (slew) και τη σχετική χρονική διαφορά στην άφιξη των σημάτων στις εισόδους των πυλών. Πάντως, στη γενική περίπτωση ενός καλά σχεδιασμένου κυκλώματος όπως αυτό ορίστηκε στην προηγούμενη παράγραφο, οι παράμετροι αυτοί είναι περίπου ίδιες για όλες τις πύλες του κυκλώματος και το απλό μοντέλο της καθυστέρησης είναι επαρκές για να καθορίσει τη βελτιστοποίηση των σχεδιασμών.

2.3.1 Η επίδραση των γραμμών διασύνδεσης

Η επίδραση των γραμμών διασύνδεσης (ή καλωδίων) μοντελοποιείται μέσω της χωρητικότητας και της αντίστασης τους. Πολύ συνοπτικά η χωρητικότητα ενός καλωδίου όπως φαίνεται και από το σχήμα 2.7 έχει δύο βασικούς συντελεστές. Ο ένας είναι η χωρητικότητα C_{area} , η οποία οφείλεται στο μήκος και το εύρος του καλωδίου και αναφέρεται είτε στη χωρητικότητα προς τη γη είτε προς τις γραμμές διασύνδεσης, που περνούν πάνω ή κάτω από το υπο-εξέταση καλώδιο. Ο δεύτερος αναφέρεται στην αμοιβαία χωρητικότητα C_C με τις γειτονικές γραμμές διασύνδεσης η οποία εξαρτάται από την απόσταση των γειτονικών γραμμών και το εμβαδόν της περιοχής επικάλυψης. Η χωρητικότητα αυτή σε περίπτωση που δύο γειτονικές γραμμές κάνουν μετάβαση προς την αντίθετη κατεύθυνση συνεισφέρει διπλάσια στην καθυστέρηση της πύλης. Εφόσον το ύψος της γραμμής είναι σταθερό και καθορίζεται από την κατασκευή, ο σχεδιαστής μπορεί μόνο να επέμβει είτε αυξάνοντας την απόσταση των γειτονικών γραμμών είτε μειώνοντας το μήκος της γειτνίασης τους.

Για τις μονάδες επεξεργασίας δεδομένων που εξετάσαμε, κάναμε κάποιες υποθέσεις για τη γεωμετρία των γραμμών διασύνδεσης και τη μεταξύ τους χωροθέτηση. Για όλα τα κυκλώματα θεωρούμε ότι το εύρος (πάχος) των γραμμών είναι το ελάχιστο που ορίζεται από την τεχνολογία.

Έτσι, έχουμε τη μέγιστη αντίσταση R_W και την ελάχιστη τιμή για το C_{area} , ενώ οι τιμές των παραμέτρων αυτών καθορίζονται πλέον αποκλειστικά από το μήκος της γραμμής διασύνδεσης. Επίσης για να διατηρήσουμε την πυκνή δομή που ορίζει το datapath layout θεωρούμε πως η απόσταση μεταξύ δύο γειτονικών γραμμών είναι επίσης η ελάχιστη δυνατή. Έτσι μεγαλώνουμε την τιμή του C_C , του οποίου η τιμή, μετά από αυτήν την παραδοχή, καθορίζεται μόνο από το μήκος της γειτνίασης των καλωδίων. Εφόσον το πάχος και η απόσταση των γειτονικών γραμμών είναι σταθερά για όλα τα κυκλώματα, τότε σε όλες τις περιπτώσεις η χωρητικότητα ενός καλωδίου είναι αποκλειστικά συνάρτηση του μήκους του και ισούται στην χειρότερη περίπτωση (θεωρώντας αντίθετες μεταβάσεις για γειτονικές γραμμές) με $C_W = (C_{area} + 2C_C) \times Length = C_{wire/\mu m} \times Length$. Το ίδιο ισχύει και για την αντίσταση του καλωδίου η οποία υπολογίζεται ως $R_W = R_{wire/\mu m} \times Length$. Η θεώρηση αυτή ταιριάζει πολύ στην περίπτωση των μονάδων επεξεργασίας δεδομένων με πυκνές διασυνδέσεις και αναπαριστά με μεγάλη ακρίβεια την πραγματικότητα [121]. Για την τεχνολογία που χρησιμοποιούμε οι τιμές που καθορίζουν τις παραμέτρους των γραμμών διασύνδεσης είναι ίσες με $C_{wire/\mu m} = 0.31 fF$ και $R_{wire/\mu m} = 0.34 \Omega$.

Για τον υπολογισμό της καθυστέρησης μιας πύλης η οποία οδηγεί ένα καλώδιο μπορούμε να χρησιμοποιήσουμε το μοντέλο π του σχήματος 2.7 το οποίο οδηγεί σε σχετικά ακριβή αποτελέσματα για μικρού και μεσαίου μεγέθους γραμμές διασύνδεσης όπως αυτές που συναντούμε εσωτερικά στις μονάδες επεξεργασίας δεδομένων [3]. Σ' αυτές τις περιπτώσεις το μήκος των καλωδίων δεν ξεπερνάει, ακόμα και σε ακραία σενάρια, τα 1.6K metal-1 tracks⁵ θεωρώντας μήκος δεδομένων τα 64 δυαδικά ψηφία και μέγεθος στήλης τα 25 metal-1 tracks. Έτσι, η καθυστέρηση μιας πύλης στην έξοδο της οποίας υπάρχει ένα καλώδιο υπολογίζεται ως εξής:

$$D = \tau \left(g \frac{C_W + C_L}{C_{in}} + p \right) + R_W \left(\frac{C_W}{2} + C_L \right)$$

Η σχέση αυτή δεν δημιουργεί κανένα πρόβλημα στη διαδικασία βελτιστοποίησης μέσω του γεωμετρικού προγραμματισμού εφόσον η μαθηματική εξίσωση που περιγράφει τη συνεισφορά των γραμμών διασύνδεσης στην ταχύτητα της πύλης παραμένει ένα ποζώνυμο⁶.

2.4 Μοντελοποίηση ενέργειας και εμβαδού

Έχοντας περιγράψει συνοπτικά τον τρόπο με τον οποίο η καθυστέρηση των πυλών αναπαρίσται σαν ποζώνυμο του μεγέθους των τρανζίστορ που την αποτελούν, θα αναφερθούμε τώρα στη μοντελοποίηση των βασικών μετρικών που ενδιαφερόμαστε να ελαχιστοποιήσουμε. Οι μετρικές αυτές δεν είναι άλλες από την συνολική ενέργεια που απαιτεί ένα κύκλωμα για κάθε λειτουργία του και το εμβαδόν που καταλαμβάνει στο ολοκληρωμένο.

Η ισχύς που καταναλώνει ένα κύκλωμα αποτελείται από δύο βασικά μέρη. Τη δυναμική κατανάλωση και τη στατική. Η δυναμική κατανάλωση οφείλεται κυρίως στη φόρτιση και στην εκφόρτιση των χωρητικοτήτων του κυκλώματος και αποτελεί τουλάχιστον για τις μονάδες επεξεργασίας δεδομένων την κύρια πηγή κατανάλωσης ισχύος. Επίσης, ένα μέρος της

⁵Η απόσταση metal-1 track ορίζεται ως το ελάχιστο εύρος ενός μετάλλου του πρώτου επιπέδου αθροίζοντας την ελάχιστη απόσταση από ένα γειτονικό καλώδιο.

⁶Για τέτοιου είδους καλώδια θεωρούμε ότι τα g και p είναι ανεξάρτητα της αντίστασης των καλωδίων. Κάτι τέτοιο δεν ισχύει στη γενική περίπτωση [33], όμως δεν πρέπει να μας ανησυχεί, γιατί στις μονάδες επεξεργασίας δεδομένων που εξετάζουμε η αντίσταση οδήγησης είναι πάντα πολύ μεγαλύτερη σε σχέση με αυτή του καλωδίου.

δυναμικής κατανάλωσης ισχύος οφείλεται στο ρεύμα βραχυκυκλώματος που διαρρέει το κύκλωμα κατά τη διάρκεια κάθε μετάβασης της εισόδου. Η ισχύς εξαιτίας του βραχυκυκλώματος για καλά σχεδιασμένα κυκλώματα, όπως αυτά ορίστηκαν στην ενότητα 2.3, είναι συνήθως το 10% της συνολικής δυναμικής κατανάλωσης ισχύος [164]. Θεωρούμε αυτό το ποσοστό συνεισφοράς σταθερό και δεν το μοντελοποιούμε συναρτήσει του μεγέθους των πυλών. Η ενέργεια E_{sw} που μεταφέρεται από μια πύλη για την φόρτιση των χωρητικοτήτων της εξόδου της είναι ίση με

$$E_{sw} = \alpha (C_p + C_L) V_{dd}^2$$

αν θεωρήσουμε πως η έξοδος κάθε πύλης κάνει μια πλήρη μετάβαση από 0 σε Vdd και αντίστροφα. Το α αναπαριστά το πλήθος των μεταβάσεων που καταναλώνουν ισχύ ενώ με C_p και C_L αναπαριστούμε την παρασιτική χωρητικότητα της κάθε πύλης και τη χωρητικότητα εξόδου της αντίστοιχα. Οι μεταβάσεις στους εσωτερικούς κόμβους του κυκλώματος αυξάνονται κατά 15-20% σε υλοποιήσεις στατικής CMOS, εξαιτίας των ξαφνικών μεταβατικών παλμών (glitches) που εμφανίζονται στις γραμμές του κυκλώματος [147].

Η στατική κατανάλωση ισχύος προέρχεται από τις διάφορες μορφές των ρευμάτων διαρροής και είναι υπεύθυνη για το 10-30% της συνολικής κατανάλωσης ισχύος των σημερινών ολοκληρωμένων κυκλωμάτων κατά τη διάρκεια της λειτουργίας τους [120], [137]. Το ποσοστό αυτό της στατικής κατανάλωσης ισχύος αυξάνει σημαντικά στην περίπτωση που το κύκλωμα παραμένει ανενεργό. Στις σημερινές τεχνολογίες CMOS το ρεύμα υποκατωφλίου είναι ο κυριότερος παράγοντας των ρευμάτων διαρροής. Το ρεύμα υποκατωφλίου

$$I_{sub} = I_0 \frac{W}{L} e^{\frac{V_{gs}-V_{th}}{nv_T}} \left(1 - e^{\frac{-V_{ds}}{v_T}} \right)$$

είναι εκθετικά εξαρτώμενο από την τάση τροφοδοσίας και την τάση κατωφλίου των τρανζίστορ, ενώ εξαρτάται επίσης σημαντικά από το μήκος του καναλιού το οποίο σε πολύ μικρές τεχνολογίες κατασκευής επηρεάζει και την τάση κατωφλίου των τρανζίστορ [65]. Το ρεύμα διαρροής που προκύπτει από τη διαπέραση ρεύματος μέσω της πύλης του τρανζίστορ, έχει αρχίσει να γίνεται σημαντικό όσο μικραίνει το πάχος του οξειδίου της πύλης εξαιτίας της συρρίκνωσης των τρανζίστορ [95]. Σημειώνουμε πως τα ρεύματα διαρροής παρουσιάζουν για μια πύλη μεγάλες διαφορές στην τιμή τους ανάλογα με τις τιμές της εισόδου.

Η συνολική ενέργεια του κυκλώματος υπολογίζεται από την παρακάτω σχέση

$$E = \sum_{i \in \text{allnodes}} a_i C_i V_{dd}^2 + T_{clk} \sum_{j \in \text{allgates}} (W_j I_{leak} V_{dd} + a_j C_{pj} V_{dd}^2).$$

Η σχέση αυτή διαχωρίζει την ενέργεια που μεταφέρεται μέσω της πύλης και που οφείλεται στην χωρητικότητα της εξόδου της, απ' αυτήν που οφείλεται στη εσωτερική δομή της και αποτελείται από μία στατική και μία δυναμική συνιστώσα. Με I_{leak} συμβολίζουμε τη μέση τιμή του ρεύματος διαρροής μιας πύλης ανά μm το οποίο υπολογίζεται μετά από εξομοιώσεις, ενώ με W_j συμβολίζουμε το μέγεθος της πύλης. Ο κύκλος ρολογιού T_{clk} είναι σταθερή τιμή και δηλώνεται εξαρχής. Όλες οι χωρητικότητες που εμφανίζονται μπορούν είτε να εκφραστούν συναρτήσει του μεγέθους των αντίστοιχων πυλών είτε είναι σταθερές χωρητικότητες καλωδίου. Σε όλες τις περιπτώσεις, η σχέση που περιγράφει την συνολική ενέργεια ενός κυκλώματος είναι

Σχήμα 2.8: Παράδειγμα του αναδρομικού σχηματισμού των περιορισμών του προβλήματος βελτιστοποίησης.

ένα ποζώνυμο του μεγέθους των πυλών που το αποτελούν και μπορεί να χρησιμοποιηθεί στην περίπτωση του γεωμετρικού προγραμματισμού.

Το εμβαδόν του κυκλώματος A μπορεί απλά να υπολογιστεί σαν το άθροισμα του εμβαδού των πυλών του κυκλώματος,

$$A = \sum_{j \in \text{allgates}} k_j W_j$$

όπου το k_i συμβολίζει το εμβαδόν μιας πύλης όταν είναι μεγέθους 1. Η σχέση αυτή μπορεί να χρησιμοποιηθεί στο πρόβλημα του γεωμετρικού προγραμματισμού γιατί είναι ένα απλό ποζώνυμο των μεγεθών των πυλών. Στην πραγματικότητα ο τελικός χώρος που καταλαμβάνει το κύκλωμα είναι κάπως μεγαλύτερος εξαιτίας της επιβάρυνσης που εισάγει η χωροθέτηση των γραμμών διασύνδεσης και οι αναγκαστικές αποστάσεις που πρέπει να τηρηθούν μεταξύ τους.

2.5 Αναγωγή σε πρόβλημα γεωμετρικού προγραμματισμού

Έχοντας τα απλά ή τα γενικευμένα ποζωνυμικά μοντέλα για κάθε πύλη του κυκλώματος μπορούμε εύκολα να περιγράψουμε το πρόβλημα ελαχιστοποίησης σύμφωνα με τους εκάστοτε περιορισμούς κάνοντας ένα απλό πέρασμα στο γράφο που περιγράφει το κύκλωμα. Θα δείξουμε εν συντομία τον τρόπο κατασκευής της συνάρτησης που θέλουμε να ελαχιστοποιήσουμε και την αναδρομική γέννηση των περιορισμών μέσα από το παράδειγμα που απεικονίζεται στο σχήμα 2.8.

Για το παράδειγμά μας θεωρούμε πως θέλουμε να ελαχιστοποιήσουμε την κατανάλωση ισχύος του κυκλώματος επιτυγχάνοντας ταυτόχρονα μια επιδιωκόμενη καθυστέρηση. Για να διαμορφώσουμε το πρόβλημα γεωμετρικού προγραμματισμού εμφανίζουμε τις μεταβλητές AT_i που συμβολίζουν τους χρόνους άφιξης των σημάτων (arrival times) στους κόμβους του κυκλώματος, ενώ με d_{ij} συμβολίζεται η καθυστέρηση από την είσοδο i στην έξοδο j της κάθε πύλης. Πλέον οι μεταβλητές του προβλήματος μας είναι οι χωρητικότητες εισόδου κάθε πύλης C_i και οι χρόνοι άφιξης των κόμβων AT_i . Σ' αυτήν την περίπτωση το πρόβλημα τίθεται ως εξής:

Ελαχιστοποίησε	E_{total}
σύμφωνα με	$AT_7 \leq D_{target}$
	$AT_7 = \max\{AT_5 + d_{57}, AT_6 + d_{67}\}$
	$AT_4 = \max\{AT_1 + d_{14}, AT_2 + d_{24}\}$
	$AT_5 \leq AT_3 + d_{35}$
	$AT_6 \leq AT_4 + d_{46}$
	$\{C_1, C_2, C_3\} \leq C_{in \max}$

Σχήμα 2.9: Η συνολική ροή σχεδιασμού για την βελτιστοποίηση των κυκλωμάτων στην περίπτωση της πλήρους ελεύθερης σχεδίασης.

Η εμφάνιση των συναρτήσεων επιλογής του μεγαλύτερου δεν πρέπει να μας ανησυχεί γιατί μπορούν εύκολα να εξαφανιστούν με την εμφάνιση επιπλέον βοηθητικών μεταβλητών [16]. Για παράδειγμα το $c = \max\{a, b\}$ μπορεί να αναπαρασταθεί με δύο περιορισμούς ανισότητας και με τη βοήθεια της βοηθητικής μεταβλητής z ως εξής: $c = z$, $a \leq z$ και $b \leq z$. Εφόσον οι συναρτήσεις που περιγράφουν τις καθυστερήσεις d_{ij} είναι ποζώνυμα τότε όλες οι συναρτήσεις περιορισμού που προέκυψαν είναι και αυτές ποζώνυμα, οπότε το πρόβλημα είναι καλώς ορισμένο ως πρόβλημα γεωμετρικού προγραμματισμού. Αυτό φαίνεται αν γράψουμε τη σχέση που περιγράφει τον περιορισμό $AT_4 + d_{46} = AT_4 + \tau \left(g_{inv} \frac{C_6}{C_4} + p_{inv} \right)$ ο οποίος είναι ένα ποζώνυμα των μεταβλητών $\{AT_4, C_6, C_4\}$ και αποτελείται από τρία μονόμια.

Στη γενική περίπτωση το πρόβλημα είναι αρκετά πιο σύνθετο εφόσον αντιμετωπίζει χωριστά τους χρόνους ανόδου και καθόδου και την κλίση slew των σημάτων. Στην περίπτωση αυτή για κάθε κόμβο εξόδου μιας πύλης θα χρειαζόμασταν πολλές επιπλέον μεταβλητές. Έτσι, μπορούμε να πούμε πως ακόμα και για σχετικά μικρά κυκλώματα το σύνολο των μεταβλητών και των περιορισμών αυξάνει εκθετικά κάνοντας πολύ δύσκολη την αντιμετώπιση πολύ μεγάλων προβλημάτων από το γεωμετρικό προγραμματισμό. Στην κατεύθυνση της βελτίωσης του χρόνου εκτέλεσης κινούνται πολλές μέθοδοι οι οποίες προσπαθούν να μειώσουν τόσο το πλήθος των μεταβλητών όσο και των περιορισμών που απαιτούνται [166].

2.6 Σύνοψη της διαδικασίας βελτιστοποίησης

Μετά τη συνοπτική περιγραφή των μοντέλων που χρησιμοποιούμε για την καθυστέρηση και την ενέργεια αλλά και τον τρόπο με τον οποίο ανάγουμε την επιλογή των μεγεθών των πυλών σε πρόβλημα γεωμετρικού προγραμματισμού, θα αναφερθούμε στη δομή του εργαλείου που υλοποιήσαμε για την αυτόματη επιλογή των μεγεθών των τρανζίστορ, τον τρόπο που το χρησιμοποιούμε και τους περιορισμούς του. Η βασική ροή βελτιστοποίησης του σχεδιασμού

που ακολουθούμε φαίνεται στο σχήμα 2.9.

Αρχικά, δημιουργούμε την περιγραφή του κυκλώματος μας, την οποία εξομοιώνουμε για να υπολογίσουμε το πλήθος των μεταβάσεων στους εσωτερικούς κόμβους του κυκλώματος. Η διαδικασία αυτή δεν χρειάζεται να είναι ακριβής και θα μπορούσε να απαλειφθεί, εκτός αν το ζητούμενο κύκλωμα βασίζει τη συμπεριφορά του στο γεγονός ότι ανακατανέμει τις εσωτερικές μεταβάσεις σε διαφορετικά σημεία του κυκλώματος. Στη συνέχεια κάνουμε μια πρόχειρη εκτίμηση της χωροθέτησης των πυλών σύμφωνα με την αρχή του datapath layout ώστε να εκτιμήσουμε το μήκος των καλωδίων. Ουσιαστικά μας ενδιαφέρει το πλήθος των στηλών που θα διατρέξει ένα καλώδιο. Για τις κοντινές συνδέσεις το μήκος των καλωδίων είναι πολύ μικρό και μπορεί πολύ εύκολα να εκτιμηθεί. Η διαδικασία αυτή για τις μονάδες επεξεργασίας δεδομένων μπορεί να πραγματοποιηθεί σχετικά εύκολα και είναι αρκετά ακριβής. Φυσικά για γενικού τύπου κυκλώματα κάτι τέτοιο είναι ανέφικτο. Έχοντας αυτήν την πληροφορία και τα μοντέλα καθυστέρησης και ενέργειας μπορούμε να ξεκινήσουμε τη διαδικασία βελτιστοποίησης και των μετρήσεων. Αυτό πραγματοποιείται σε τρεις φάσεις.

- ▶ Η πρώτη λαμβάνει όλη την πληροφορία εισόδου όπως φαίνεται στο σχήμα 2.9 και τη συνδιάζει ώστε να σχηματίσει ένα γεωμετρικό πρόγραμμα όπως αυτό που δείξαμε στην προηγούμενη ενότητα. Η στοχευόμενη συνάρτηση μπορεί να είναι είτε η κατανάλωση ισχύος είτε η καθυστέρηση του κυκλώματος. Η επιλογή είναι στην ευχέρεια του χρήστη. Επίσης, καθορίζουμε τη χωρητικότητα που θα οδηγεί το κύκλωμα μας και τη μέγιστη επιτρεπόμενη χωρητικότητα εισόδου. Με τον ίδιο τρόπο θα μπορούσαμε να φράξουμε και το μέγιστο ή το ελάχιστο μέγεθος των πυλών εσωτερικά στο κύκλωμα. Φυσικά, δεν υπάρχει η δυνατότητα αλλαγής του κυκλώματος σε επίπεδο λογικών πυλών και η τοποθέτηση επιπλέον σταδίων ενίσχυσης γίνεται με το χέρι, ιδίως σε περιπτώσεις όπου το μέγεθος των πυλών, όπως προκύπτει μετά μια πρώτη βελτιστοποίηση, ξεφεύγει των λογικών ορίων (προσπάθεια σταδίου f μεγαλύτερη από 7).
- ▶ Στη δεύτερη φάση το γεωμετρικό πρόβλημα λύνεται και παράγει στην έξοδο του μια Spice περιγραφή του κυκλώματος, στο οποίο πλέον το μέγεθος κάθε πύλης έχει καθοριστεί ώστε να ικανοποιεί τους περιορισμούς που θέσαμε. Για τη λύση του γεωμετρικού προγράμματος χρησιμοποιήσαμε τον επιλυτή GGPLab ο οποίος είναι διαθέσιμος στο διαδίκτυο [116].
- ▶ Στο τέλος, μένει η εξομοίωση των κυκλωμάτων σε Spice οδηγούμενα από τα ίδια διανύσματα που χρησιμοποιήσαμε στην αρχή για τον υπολογισμό των μεταβάσεων. Για να παράγουμε την καμπύλη ενέργειας ως προς την καθυστέρηση του κυκλώματος εκτελούμε τη διαδικασία βελτιστοποίησης επαναληπτικά βάζοντας κάθε φορά μεγαλύτερη τιμή για τη στοχευόμενη καθυστέρηση. Η ελάχιστη δυνατή καθυστέρηση του κυκλώματος καθορίζεται από μια πρώτη βελτιστοποίηση στην οποία δεν φράσουμε ούτε την ενέργεια ούτε το εμβαδόν του κυκλώματος. Ο υπολογισμός του κρίσιμου μονοπατιού και η εύρεση του διανύσματος που το ενεργοποιεί γίνεται από τον χρήστη και επιτυγχάνεται μετά από λίγες προσπάθειες με τη μελέτη της δομής του κυκλώματος. Η αναζήτηση δεν είναι διαισθητική αλλά βασίζεται στις τιμές που προέκυψαν για τις μεταβλητές του χρόνου άφιξης των σημάτων στους εσωτερικούς κόμβους του κυκλώματος μετά από τη βελτιστοποίηση. Η διαδικασία αυτή θα μπορούσε να είχε αυτοματοποιηθεί, όμως κάτι τέτοιο δεν έχει γίνει προς το παρόν.

Η διαδικασία της βελτιστοποίησης και των μετρήσεων καταλήγει να είναι για πιο μεγάλα κυκ-

λώματα ιδιαίτερα χρονοβόρα. Χονδρικά η παραγωγή μιας καμπύλης ενέργειας καθυστέρησης μπορεί να απαιτεί έως και 3 μέρες για πολύ μεγάλα κυκλώματα συμπεριλαμβανομένου του χρόνου των βελτιστοποιήσεων και των μετρήσεων. Παρόλα αυτά, είναι ο καλύτερος δυνατός τρόπος που έχουμε στη διάθεση μας μέχρι στιγμής για την εξερεύνηση του χώρου λύσεων και τον πειραματισμό με διάφορες τοπολογίες των κυκλωμάτων που εξετάσαμε.

Πέρα από την επιλογή των μεγεθών των πυλών, για μια δεδομένη χωρητικότητα εισόδου και εξόδου του κυκλώματος, στην βελτιστοποίηση των προτεινόμενων μονάδων επεξεργασίας δεδομένων θα μπορούσαν να συμμετέχουν και άλλες μεταβλητές που επηρεάζουν την απόδοση τους όπως η τάση τροφοδοσίας και η τιμή της τάσης κατωφλίου των τρανζίστορ. Αυτό δεν πραγματοποιήθηκε για δύο λόγους κυρίως. Πρώτον, όπως έχουμε ήδη εξηγήσει, τα κυκλώματα που εξετάζουμε στοχεύουν σε υλοποιήσεις μικροεπεξεργαστών με πολύ μικρούς κύκλους ρολογιού (11-18F04). Συνεπώς, η μείωση της τάσης τροφοδοσίας από τα $1.2V(\pm 10\%)$ της τεχνολογίας μας σε μια μικρότερη βέλτιστη ως προς την κατανάλωση ισχύος τιμή θα επέφερε ένα σοβαρό κόστος στην καθυστέρηση του κυκλώματος. Η εφαρμογή της μειωμένης τάσης, χωρίς απώλειες σε καθυστέρηση, θα μπορούσε να συμβεί σε κυκλώματα τα οποία υποστηρίζουν διπλές τάσεις τροφοδοσίας [89], [146], [159] στα οποία μειώνουμε την κατανάλωση ισχύος μόνο στα μη κρίσιμα μονοπάτια. Στα υπο-εξέταση κυκλώματα κάτι τέτοιο δύσκολα μπορεί να εφαρμοστεί καθώς, λόγω δομής, η πλειοψηφία των μονοπατιών των κυκλωμάτων είναι εξίσου κρίσιμα. Ένας άλλος τρόπος να ισοσκελίσουμε την απώλεια σε καθυστέρηση είναι να επιτρέψουμε την αύξηση της μέγιστης δυνατής χωρητικότητας εισόδου του κυκλώματος. Μ' αυτόν τον τρόπο θα αυξήσουμε το μέγεθος των εσωτερικών πυλών βελτιώνοντας την οδηγητική τους ικανότητα. Κάτι τέτοιο όμως δεν μπορεί να συμβεί μιας και η αύξηση των μεγεθών των πυλών της εισόδου θα καθυστερούσε αναπόφευκτα τα προηγούμενα κυκλώματα. Η βελτιστοποίηση αυτή θα είχε νόημα αν εξετάζαμε ταυτόχρονα ένα πολύ μεγάλο μέρος του συνολικού κυκλώματος οπότε θα είχαμε την ευχέρεια να βρούμε τη βέλτιστη τιμή για τη χωρητικότητα εισόδου για κάθε μία από τις υπο-μονάδες που σχεδιάζουμε [32]. Δεύτερον, όσον αφορά στην τάση κατωφλίου, η τεχνολογία που έχουμε στη διάθεση μας, επιτρέπει την επιλογή τριών διαφορετικών κατωφλίων για τα τρανζίστορ. Οποιαδήποτε άλλη ενδιάμεση τιμή μπορεί να παραχθεί μόνο μέσω επιπλέον κυκλωματικών τεχνικών που θα πολώνουν κατάλληλα το υπόστρωμα. Κάτι τέτοιο απαιτεί μεγάλη σχεδιαστική προσπάθεια στις πολύ μικρές τεχνολογίες κατασκευής και σπάνια εφαρμόζεται σε μονάδες επεξεργασίας δεδομένων. Έτσι, αν πρέπει να επιλέξουμε μεταξύ των διαθέσιμων τριών τιμών το πρόβλημα χάνει τη συνέχειά του και δεν μπορεί να λυθεί. Γνωρίζουμε ακόμα ότι η απόδοση είναι άμεσα εξαρτημένη από την τοπολογία του κυκλώματος και από τα μεγέθη των πυλών. Η επιτάχυνση ή η επιβράδυνση συγκεκριμένων μονοπατιών μέσω της αλλαγής του κατωφλίου θα μπορούσε να συμβεί σε όλες τις υπο-σύγκριση λύσεις με τον ίδιο τρόπο όποτε δε θα άλλαζε τις σχετικές διαφορές μεταξύ τους.

Κλείνοντας, θα ήταν σκόπιμο να πούμε, πως αυτό που κύρια μας ενδιαφέρει είναι η σχεδίαση κυκλωμάτων τα οποία θα πετυχαίνουν καλύτερα αποτελέσματα σε καθυστέρηση και σε ενέργεια σε σχέση με προηγούμενες υλοποιήσεις, εξαιτίας της απλής τοπολογίας τους διατηρώντας έτσι τα ποιοτικά χαρακτηριστικά τους ανεξάρτητα από την τεχνολογία στην οποία θα κατασκευαστούν. Μικρές παρεκκλίσεις από την πραγματική βέλτιστη τιμή, λόγω των περιορισμών των εργαλείων, ελάχιστα θα επηρεάσουν τόσο το τελικό αποτέλεσμα όσο και τα οφέλη που θα προσφέρουν έναντι των ήδη γνωστών λύσεων.

Αθροιστές παράλληλου προθέματος

Η εκτέλεση της δυαδικής πρόσθεσης είναι ένα από τα πιο θεμελιώδη προβλήματα οποιουδήποτε υπολογιστικού συστήματος. Οι αθροιστές χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών το οποίο εκτείνεται από μικρά κυκλώματα ειδικού σκοπού μέχρι τους μικροεπεξεργαστές υψηλών επιδόσεων. Είναι ευρέως γνωστή η χρήση τους στις αριθμητικές/λογικές μονάδες (Arithmetic Logic Unit, ALU) και στα κυκλώματα υπολογισμού των διευθύνσεων (Address Generation Unit, AGU). Επίσης αποτελούν βασικό μέρος του σχεδιασμού άλλων αριθμητικών πράξεων, όπως του πολλαπλασιασμού και της διαίρεσης τόσο ακεραίων αριθμών όσο και αριθμών κινητής υποδιαστολής. Οι σημερινοί μικροεπεξεργαστές περιλαμβάνουν συνήθως πολλά κυκλώματα αθροιστών. Χαρακτηριστικά αναφέρουμε ότι στο νέο επεξεργαστή CELL της κοινοπραξίας (SONY-IBM-Toshiba) υπάρχουν περισσότερα από 100 κυκλώματα αθροιστών των 32 δυαδικών ψηφίων τα οποία λειτουργούν σε πολύ υψηλές ταχύτητες της τάξης των 10FO4 [107], [115]. Αντίθετα, στη γενική περίπτωση το εμβαδόν που καταλαμβάνουν στο ολοκληρωμένο δεν είναι παρά ένα μικρό ποσοστό του συνόλου. Επίσης, λόγω της πολύ ομοιόμορφης δομής τους και των πολλών νέων αλγορίθμων που έχουν προταθεί για την υλοποίησή τους τα τελευταία χρόνια μπορούν με μικρή σχεδιαστική προσπάθεια να πετύχουν πολύ υψηλές ταχύτητες λειτουργίας. Έτσι, θα αναρωτιόταν κανείς, ποιος ο λόγος να ασχοληθούμε με τη σχεδίαση τους μετά από τόσα χρόνια εντατικής έρευνας και τόσα ήδη γνωστά κυκλώματα με πολύ καλές ιδιότητες.

Η απάντηση στο ερώτημα αυτό είναι πολύπλευρη. Πρώτον, οι αθροιστές χρησιμοποιούνται σχεδόν σε κάθε κύκλο του επεξεργαστή οδηγώντας τους εσωτερικούς κόμβους του κυκλώματος σε μεγάλο αριθμό από μεταβάσεις. Αυτό έχει σαν αποτέλεσμα την αυξημένη κατανάλωση ισχύος, η οποία γίνεται χειρότερη, αν λάβουμε υπόψη μας την συνήθως πολύ υψηλή συχνότητα λειτουργίας των κυκλωμάτων αυτών. Η διαδικασία αυτή έχει παρατηρηθεί εδώ και καιρό πως οδηγεί σε πολύ μεγάλη αύξηση της τοπικής θερμοκρασίας του ολοκληρωμένου και στη δημιουργία θερμών πυρήνων (thermal hotspot) γύρω από τα κυκλώματα των αθροιστών, οι

οποίοι θέτουν σε κίνδυνο την αξιοπίστη λειτουργία του. Δεύτερον, πολλοί από τους υπάρχοντες αλγόριθμους δυαδικής πρόσθεσης οδηγούν σε κυκλώματα τα οποία πετυχαίνουν την καλύτερη δυνατή απόδοση μόνο όταν σχεδιαστούν σε συνδυασμό με επιθετικές τεχνικές σε επίπεδο κυκλώματος όπως αυτή της δυναμικής λογικής. Οι αλγόριθμοι αυτοί είναι αβέβαιο αν θα μπορούν να χρησιμοποιηθούν στο μέλλον, γιατί, από τη μια οι πολύ μικρές διαστάσεις των υλικών και από την άλλη η αβεβαιότητα για τις ιδιότητες τους, που εισάγει η διαδικασία κατασκευής τους, κάνουν την επιβεβαίωση και την εξασφάλιση της ορθής λειτουργίας των δυναμικών κυκλωμάτων από δύσκολη έως ανέφικτη. Επίσης, η βελτίωση των ήδη γνωστών κυκλωμάτων ή ακόμα καλύτερα η δημιουργία μιας νέας τοπολογίας αθροιστών, η οποία θα παραμένει αποδοτική σε όλο το φάσμα των σχεδιαστικών μετρικών όπως καθυστέρηση, εμβαδόν και ενέργεια, θα επηρέαζε θετικά μια πληθώρα άλλων κυκλωμάτων, στα οποία οι αθροιστές αποτελούν αναπόσπαστο μέρος του σχεδιασμού. Για παράδειγμα, η καθυστέρηση του τελικού αθροιστή σε έναν πολλαπλασιαστή αντιστοιχεί πάνω από το 30% της συνολικής καθυστέρησης του κυκλώματος, το οποίο μεγαλώνει όσο αυξάνεται το μήκος λέξης της εισόδου.

Ο σχεδιαστής έχει στη διάθεση του μια πληθώρα από διαφορετικούς αλγόριθμους για την υλοποίηση των αθροιστών [169]. Οι πιο πολλές τεχνικές βασίζονται στη διάδοση του κρατουμένου από τις λιγότερο σημαντικές θέσεις του αθροιστή προς τις περισσότερες σημαντικές. Σ' αυτήν την κατηγορία ανήκουν οι ripple-carry αθροιστές, οι αθροιστές πρόβλεψης κρατουμένου (carry lookahead) και η γενίκευσή τους, οι αθροιστές παράλληλου προθέματος (parallel prefix) με τις διάφορες παραλλαγές τους. Επίσης, μπορούν να ενταχθούν στην ίδια κατηγορία, αν και εκ πρώτης όψεως ακολουθούν μια εντελώς διαφορετική στρατηγική, οι αθροιστές παράκαμψης κρατουμένου (carry skip) και οι αθροιστές υπό-συνθήκη αθροίσματος (conditional sum). Αυτή η παρατήρηση παρουσιάστηκε στις εργασίες [20] και [103] αντίστοιχα, γενικεύοντας έτσι τη θεωρία λειτουργίας των κυκλωμάτων αυτών. Μια συγκεντρωτική παρουσίαση των αλγορίθμων αυτών και των χαρακτηριστικών τους, –λαμβάνοντας υπόψη και το κόστος της υλοποίησης τους– πραγματοποιήθηκε από τον Zimmerman στις εργασίες [178] και [179]. Επίσης, μια περισσότερο μαθηματική αντιμετώπιση των κυκλωμάτων μπορεί να βρει ο ενδιαφερόμενος αναγνώστης σε κλασσικά βιβλία αριθμητικής υπολογιστών [86], [129], [48].

Στη συνέχεια, θα αναφερθούμε στους αθροιστές παράλληλου προθέματος, οι οποίοι αποτελούν την πιο ευρέως χρησιμοποιούμενη τεχνική υλοποίησης γρήγορων αθροιστών κυρίως σε κυκλώματα μικροεπεξεργαστών [112], [49], [111]. Πιο συγκεκριμένα, θα αναλύσουμε τη θεωρία σχεδίασής τους, τις διαφορετικές εναλλακτικές τοπολογίες κυκλωμάτων που επιτρέπει η τεχνική του παράλληλου προθέματος [84], [10] και τον τρόπο που αυτές επηρεάζονται από τις πολύ μικρές τεχνολογίες κατασκευής [74], [87]. Στη συνέχεια θα παρουσιάσουμε τους προτεινόμενους αθροιστές, οι οποίοι καταφέρνουν να ενώσουν με έναν αποτελεσματικό τρόπο την εγγενή απλότητα των κρατουμένων του Ling [104] και τα δίκτυα παράλληλου προθέματος. Θα περιγράψουμε αναλυτικά τη νέα μεθοδολογία σχεδιασμού και τον τρόπο με τον οποίο προέκυψε, ενώ θα ολοκληρώσουμε την περιγραφή με τα πειραματικά αποτελέσματα που συλλέξαμε για την ενέργεια και την καθυστέρηση των κυκλωμάτων.

3.1 Αθροιστές παράλληλου προθέματος - Βασικές αρχές

Ας θεωρήσουμε ότι θέλουμε να υπολογίσουμε το άθροισμα $S = S_{n-1}S_{n-2} \dots S_0$ δύο αριθμών των n δυαδικών ψηφίων $A = A_{n-1}A_{n-2} \dots A_0$ και $B = B_{n-1}B_{n-2} \dots B_0$. Στη γενική περίπτωση

το i -οστό δυαδικό ψηφίο του αθροίσματος S_i μπορεί να υπολογιστεί σύμφωνα με την παρακάτω επαναληπτική διαδικασία η οποία προκύπτει από την πιο απλή μορφή πρόσθεσης, αυτή του αθροιστή ripple carry.

$$\begin{aligned} S_i &= A_i \oplus B_i \oplus C_{i-1} \\ C_i &= A_i \cdot B_i + (A_i + B_i) \cdot C_{i-1} \end{aligned} \quad (3.1)$$

Σ' αυτήν την αναδρομική σχέση συμβολίζουμε με C_i το κρατούμενο που παράγεται από τη θέση i , ενώ το C_{-1} ισούται με το κρατούμενο εισόδου, το οποίο θα θεωρήσουμε προς το παρόν ότι είναι ίσο με μηδέν. Παρατηρώντας τη σχέση (3.1) προκύπτει πως ο υπολογισμός του αθροίσματος περιλαμβάνει τρία διαφορετικά στάδια. Στο πρώτο στάδιο, μπορούμε να υπολογίσουμε παράλληλα για όλα τα δυαδικά ψηφία της εισόδου τους όρους γέννησης και διάδοσης κρατουμένου G_i , P_i και τον όρο του μερικού αθροίσματος T_i σύμφωνα με τις παρακάτω σχέσεις:

$$G_i = A_i \cdot B_i \quad P_i = A_i + B_i \quad T_i = A_i \oplus B_i$$

Στο δεύτερο στάδιο, το οποίο καθορίζει ουσιαστικά την ταχύτητα του υπολογισμού, πρέπει να υπολογίσουμε αναδρομικά τα κρατούμενα C_i για όλες τις θέσεις των δυαδικών ψηφίων σύμφωνα με τη γραμμική αναδρομή

$$C_i = G_i + P_i \cdot C_{i-1} \quad (3.2)$$

Στο τελευταίο στάδιο, αφού έχουμε υπολογίσει τα κρατούμενα που προέκυψαν κατά την πρόσθεση, είμαστε σε θέση να παράγουμε, παράλληλα για όλες τις θέσεις, τα δυαδικά ψηφία του αθροίσματος σύμφωνα με τη σχέση

$$S_i = T_i \oplus C_{i-1}$$

Ο κύριος στόχος των αθροιστών πρόβλεψης κρατουμένου ή των αθροιστών παράλληλου προθέματος είναι να επιταχύνουν τον υπολογισμό των κρατουμένων. Στην ουσία, η αντιμετώπιση του προβλήματος υπολογισμού των κρατουμένων σαν πρόβλημα παράλληλου προθέματος αποτελεί τη γενίκευση της τεχνικής πρόβλεψης κρατουμένου και τη μόνη τεχνική που χρησιμοποιείται στις μέρες μας για την υλοποίηση υψηλής ταχύτητας αθροιστών στους σύγχρονους μικροεπεξεργαστές.

Ένα κύκλωμα παράλληλου προθέματος δέχεται n εισόδους $X_1, X_2, \dots, X_n$ και υπολογίζει παράλληλα n εξόδους $Y_1, Y_2, \dots, Y_n$ χρησιμοποιώντας έναν τελεστή $\odot$, ο οποίος διατηρεί την ιδιότητα της αντιμεταθετικότητας (associativity property). Στην περίπτωση της απλής γραμμικής αναδρομής $Y_i = Y_{i-1} \odot X_i$ οι έξοδοι Y_i μπορούν να υπολογιστούν απευθείας από τις εισόδους X_i σύμφωνα με τις παρακάτω σχέσεις:

$$\begin{aligned} Y_1 &= X_1 \\ Y_2 &= X_1 \odot X_2 \\ Y_3 &= X_1 \odot X_2 \odot X_3 \\ &\dots \\ Y_n &= X_1 \odot X_2 \odot \dots \odot X_n \end{aligned}$$

Σχήμα 3.1: Τοπολογίες δένδρων παράλληλου προθέματος.

Η χρήση των τεχνικών παράλληλου προθέματος είναι πολύ ελκυστική σε υλοποιήσεις VLSI. Αυτό συμβαίνει γιατί ο υπολογισμός δεν μοιράζεται σε n χωριστά κυκλώματα τα οποία δέχονται ένα μέρος της εισόδου X και υπολογίζουν ανεξάρτητα την έξοδο Y , αλλά χρησιμοποιείται ένα ενιαίο και ομοιόμορφο κύκλωμα, στο οποίο τα ενδιάμεσα αποτελέσματα που παράγονται μπορούν να επαναχρησιμοποιηθούν για τον υπολογισμό των τιμών όλων των εξόδων. Η ιδιότητα αυτή των δένδρων παράλληλου προθέματος οδήγησε σε μια πληθώρα διαφορετικών τοπολογιών οι οποίες μπορούν να χρησιμοποιηθούν για τον υπολογισμό της εξόδου [91], [85], [17], [58], [84], [175], [177], [59], [92]. Στο σχήμα 3.1 παρουσιάζονται 4 διαφορετικές τοπολογίες, οι οποίες υπολογίζουν την έξοδο Y με διαφορετικό τρόπο. Αρχικά, μπορούμε εύκολα να παρατηρήσουμε πως η κάθε τοπολογία απαιτεί διαφορετικό αριθμό από τελεστές. Επίσης, τα εσωτερικά καλώδια του κυκλώματος διαμοιράζονται με διαφορετικό τρόπο, ενώ το πλήθος των επιπέδων υπολογισμού μπορεί να διαφέρει. Στην καλύτερη περίπτωση, και θεωρώντας ότι ο τελεστής $\odot$ δέχεται το πολύ δύο εισόδους, οι εξοδοί Y_i μπορούν να υπολογιστούν σε $\lceil \log_2 n \rceil$ επίπεδα του δένδρου παράλληλου προθέματος.

Η μαθηματική αντιμετώπιση του προβλήματος κατασκευής ενός δένδρου παράλληλου προθέματος n εισόδων θεωρεί πως, όταν το πλήθος των χρησιμοποιημένων τελεστών AR και το πλήθος των επιπέδων LV ακολουθεί τη σχέση (3.3), τότε η τοπολογία αυτή είναι βέλτιστη ως προς το χώρο και την καθυστέρηση [92].

$$AR + LV \geq 2n + 2 \quad (3.3)$$

Πολλές είναι οι λύσεις που προσπαθούν να πετύχουν τη δημιουργία δένδρων παράλληλου προθέματος που θα ικανοποιούν τη συνθήκη (3.3) [175]. Στην ουσία, η αναζήτηση τέτοιων λύσεων έχει νόημα όταν το μήκος της εισόδου δεν είναι δύναμη του δύο και όταν μπορούμε

Σχήμα 3.2: Παράδειγμα μιας τοπολογίας δένδρου παράλληλου προθέματος των 16 δυαδικών ψηφίων όπου οι τελεστές συνδυάζουν σε κάθε επίπεδο περισσότερους από δύο όρους.

να θυσιάσουμε κάτι από την καθυστέρηση του κυκλώματος με σκοπό να μειώσουμε το πλήθος των τελεστών. Στην πράξη κάτι τέτοιο δεν έχει νόημα, γιατί απλά η μείωση του αριθμού των τελεστών που συμμετέχουν στο δένδρο παράλληλου προθέματος δεν συνεπάγεται αναγκαστικά τη μείωση του εμβαδού χωροθέτησης του κύκλωματος, ούτε της κατανάλωσης ισχύος. Αυτό συμβαίνει γιατί από τη στιγμή που ο οποιοσδήποτε τελεστής $\odot$ υλοποιηθεί σε υλικό, το μέγεθος των πυλών που τον υλοποιούν είναι άμεσα εξαρτώμενο από το είδος τους, από τη λογική οικογένεια στην οποία ανήκουν και από τη χωρητικότητα που καλούνται να οδηγήσουν [21]. Χαρακτηριστικό παράδειγμα είναι η περίπτωση των δένδρων που προτάθηκαν από τους Ladner και Fischer, τα οποία, αν και απαιτούν το μικρότερο αριθμό από τελεστές, δεν οδηγούν πάντα στις αποδοτικότερες υλοποιήσεις, εξαιτίας του μεγάλου φορτίου εξόδου συγκεκριμένων γραμμών του κυκλώματος. Το ερώτημα για το ποια από τις τοπολογίες οδηγεί στην αποδοτικότερη υλοποίηση, δεν μπορεί να απαντηθεί, αν πρώτα δεν πραγματοποιηθεί μια συνολική βελτιστοποίηση του κυκλώματος στοχεύοντας επακριβώς στις συνθήκες κάτω από τις οποίες θα χρησιμοποιηθεί (ζητούμενη καθυστέρηση, μέγιστη δυνατή χωρητικότητα εισόδου, χωρητικότητα εξόδου, διαθέσιμες πύλες, κ.ο.κ.) [105], [180].

Ανάλογα με τη μορφή του τελεστή $\odot$ και την ευκολία με την οποία μπορεί να υλοποιηθεί σε υλικό, μπορούμε να σχεδιάσουμε δένδρα παράλληλου προθέματος τα οποία συνδυάζουν σε κάθε επίπεδο περισσότερα από ένα ενδιάμεσα αποτελέσματα (high radix). Τελεστές όπως η λογική πράξη AND ή OR είναι υποψήφιοι για τέτοιες υλοποιήσεις αφού η καθυστέρηση μιας τέτοιας πύλης των δύο εισόδων δε διαφέρει σημαντικά από την καθυστέρηση της ίδιας πύλης των τριών εισόδων. Ένα δένδρο παράλληλου προθέματος των 16 εισόδων που αποτελείται από τελεστές τεσσάρων εισόδων φαίνεται στο σχήμα 3.2. Τέτοια δένδρα χρησιμοποιούνται ευρέως στους σύγχρονους μικροεπεξεργαστές ιδιαίτερα στην περίπτωση όπου τα κυκλώματα είναι υλοποιημένα σε δυναμική CMOS.

Ο υπολογισμός του κρατουμένου μετασχηματίστηκε σε πρόβλημα παράλληλου προθέματος από τους Brent και Kung στην εργασία [17]. Για τον υπολογισμό του κρατουμένου, στη θέση του X_i θεωρούμε το ζεύγος γέννησης και διάδοσης κρατουμένου (G_i, P_i) και τον τελεστή $\circ$, ο οποίος ορίζεται ως εξής:

$$(G_x, P_x) \circ (G_y, P_y) = (G_x + P_x \cdot G_y, P_x \cdot P_y) \quad (3.4)$$

Σχήμα 3.3: Η υλοποίηση του τελεστή παράλληλου προθέματος που χρησιμοποιείται για τον υπολογισμό των κρατουμένων.

Έτσι, για κάθε θέση i πρέπει να συνδυάσουμε με τον τελεστή $\circ$ όλους τους όρους (G_i, P_i) που έχουν προϋπολογιστεί από τη θέση 0 έως και τη θέση i . Με άλλα λόγια, κάθε έξοδος Y_i είναι ισοδύναμη με ένα ζεύγος τιμών που συμβολίζει πλέον ομάδες γέννησης και διάδοσης κρατουμένου και αναπαρίσται ως $(G_{i:0}, P_{i:0})$. Κάθε τέτοιο ζεύγος τιμών υπολογίζεται ως εξής:

$$(G_{i:0}, P_{i:0}) = (G_i, P_i) \circ (G_{i-1}, P_{i-1}) \circ \dots \circ (G_1, P_1) \circ (G_0, P_0) \quad (3.5)$$

Η τελική τιμή του κρατουμένου C_i ισούται με την τιμή του $G_{i:0}$, ενώ το $P_{i:0}$ δεν είναι ανάγκη να υπολογιστεί μέχρι τέλους, αφού δεν συμμετέχει στον τελικό υπολογισμό του αθροίσματος. Οι τοπολογίες των δένδρων παράλληλου προθέματος που παρουσιάστηκαν στο σχήμα 3.1 μπορούν απευθείας να χρησιμοποιηθούν για τον παράλληλο υπολογισμό των κρατουμένων. Η μόνη διαφορά είναι πως κάθε γραμμή διασύνδεσης που παρουσιάζεται στα σχήματα αυτά, στην περίπτωση του υπολογισμού κρατουμένου αναφέρεται σε δύο καλώδια, τα οποία μεταφέρουν αντίστοιχα τις τιμές των ομάδων γέννησης και διάδοσης $(G_{k:j}, P_{k:j})$ που έχουν δημιουργηθεί μεταξύ δύο οποιωνδήποτε θέσεων k και j , με $k > j$. Η υλοποίηση του τελεστή $\circ$ που χρησιμοποιείται για τον υπολογισμό των κρατουμένων φαίνεται στο σχήμα 3.3, τόσο σε επίπεδο πυλών όσο και σε επίπεδο τρανζίστορ, θεωρώντας πύλες στατικής CMOS. Στην περίπτωση που ο τελεστής $\circ$ συνδυάζει σε κάθε επίπεδο περισσότερους από δύο όρους (G_x, P_x) ακολουθώντας τοπολογίες όπως αυτή του σχήματος 3.2, τότε οι υλοποιήσεις σε επίπεδο τρανζίστορ αφορούν σχεδόν αποκλειστικά κυκλώματα δυναμικής λογικής. Ο ενδιαφερόμενος αναγνώστης μπορεί να βρει αποδοτικές μορφές των κυκλωμάτων που έχουν προταθεί για αυτούς τους σχεδιασμούς

στις εργασίες [49], [1], [10], [79].

Κλείνοντας την περιγραφή του υπολογισμού του κρατουμένου σαν ένα πρόβλημα παράλληλου προθέματος, θα πρέπει να αναφερθούμε στην απορροφητική ιδιότητα του τελεστή $\circ$. Στην απλή περίπτωση μπορούμε εύκολα να διαπιστώσουμε ότι $(G_i, P_i) \circ (G_i, P_i) = (G_i, P_i)$. Η γενικευμένη μορφή αυτής της ιδιότητας αποδείχθηκε από τους Lynch και Swartzlander στην εργασία [106] και μας επιτρέπει να παράγουμε ομάδες γέννησης και διάδοσης κρατουμένου ακόμα και αν κάποια από τα ενδιαμέσζα ζεύγη (G_i, P_i) , που υπάρχουν στις προϋπολογισμένες ομάδες επικαλύπτονται. Για παράδειγμα, είναι εφικτό να υπολογίσουμε τον όρο $(G_{6:1}, P_{6:1})$ από τους $(G_{6:3}, P_{6:3})$ και $(G_{4:1}, P_{4:1})$ χωρίς να ανησυχούμε για την ταυτόχρονη παρουσία του όρου $(G_{4:3}, P_{4:3})$ και στις δύο σχέσεις. Έτσι, στη γενική της μορφή η απορροφητική ιδιότητα ορίζεται ως εξής: Ο όρος $(G_{i:j}, P_{i:j})$ μπορεί να προκύψει από τους όρους $(G_{i:k}, P_{i:k})$ και $(G_{m:j}, P_{m:j})$, όπου $i > m \geq k > j$, οι οποίοι μερικώς επικαλύπτονται αφού $(G_{i:j}, P_{i:j}) = (G_{i:k}, P_{i:k}) \circ (G_{m:j}, P_{m:j})$. Η απορροφητική ιδιότητα είναι ιδιαίτερα χρήσιμη στη δημιουργία δένδρων παράλληλου προθέματος με μήκος που δεν είναι δύναμη του δύο και επιτρέπει τη διαμοίραση κοινών όρων μεταξύ των κρατουμένων ώστε να μειωθεί αποτελεσματικά το πλήθος των τελεστών, χωρίς να αυξηθεί ταυτόχρονα το φορτίο στις εσωτερικές γραμμές του κυκλώματος.

3.2 Τα κρατούμενα του Ling

Ο Ling στην εργασία [104] πρότεινε τη διάδοση μιας απλοποιημένης μορφής κρατουμένου που συμβολίζουμε με H_i αντί του παραδοσιακού κρατουμένου C_i . Το κρατούμενο που πρότεινε ο Ling ονομάζεται συνήθως ψευδό-κρατούμενο και προκύπτει συνδυάζοντας το κρατούμενο που παράγεται από την i -οστή θέση C_i με το κρατούμενο που εισέρχεται στην ίδια θέση C_{i-1} .

$$H_i = C_i + C_{i-1} \quad (3.6)$$

Αντικαθιστώντας την τιμή του C_i από τη σχέση (3.2) προκύπτει ότι

$$H_i = G_i + C_{i-1} \quad (3.7)$$

Από τη σχέση αυτή μπορούμε να συμπεράνουμε πως το ψευδο-κρατούμενο H_i αποτελεί μια απλοποιημένη μορφή του C_i (σχέση (3.2)) και πως μπορεί να υπολογιστεί ταχύτερα. Το συμπέρασμα αυτό φαίνεται καλύτερα αν γράψουμε τις πλήρως ανοιγμένες σχέσεις για τα C_3 και H_3 αντίστοιχα.

$$\begin{aligned} C_3 &= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 \\ H_3 &= G_3 + G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0. \end{aligned}$$

Αν θεωρήσουμε ότι οι σχέσεις θα υλοποιηθούν με λογικές πύλες δύο εισόδων, τότε για τον υπολογισμό του C_3 απαιτούνται τουλάχιστον 4 λογικά επίπεδα ενώ για το H_3 αρκούν μόνο 3. Η βασική ιδέα του Ling γενικεύτηκε από τον Doran στην εργασία [44] δείχνοντας ότι υπάρχουν 32 ισοδύναμες σχέσεις κρατουμένου οι οποίες μπορούν να χρησιμοποιηθούν για τον υπολογισμό του αθροίσματος αλλά μόνο 4 από αυτές διατηρούν την απλότητα των σχέσεων του Ling. Πολλές είναι οι εργασίες, που έχουν παρουσιαστεί στη διεθνή βιβλιογραφία, και

προσπαθούν να εκμεταλλευτούν την απλή δομή των κρατούμενων του Ling ώστε να μειώσουν την καθυστέρηση των κυκλωμάτων [117], [78], [132], [55], [167], [168]. Στις περισσότερες των περιπτώσεων, τα κυκλώματα που προτείνονται, αναφέρονται σε κατά περίπτωση (ad hoc) δυναμικές υλοποιήσεις και δεν γενικεύονται εύκολα σε μια γενικού τύπου δομή, η οποία θα οδηγεί σε αποδοτικά κυκλώματα ανεξάρτητα της λογικής οικογένειας και της τεχνολογίας κατασκευής.

Για να μπορέσουμε να υπολογίσουμε το H_i με ένα δένδρο παράλληλου προθέματος πρέπει αρχικά να βρούμε μια αναδρομική σχέση η οποία να συνδέει το H_i με το προηγούμενο ψευδο-κρατούμενο H_{i-1} [163]. Γνωρίζουμε ότι $G_i \cdot P_i = G_i$ όποτε η σχέση που περιγράφει το κρατούμενο C_i μπορεί να γραφεί ως εξής: $C_i = G_i + P_i \cdot C_{i-1} = P_i \cdot (G_i + C_{i-1})$. Μπορούμε εύκολα να παρατηρήσουμε πως σύμφωνα με την (3.7) ο όρος μέσα στην παρένθεση ισούται με H_i . Έτσι προκύπτει ότι $C_i = P_i \cdot H_i$. Αντικαθιστώντας τη νέα σχέση που προέκυψε πίσω στην εξίσωση (3.7) προκύπτει η ζητούμενη αναδρομική σχέση.

$$H_i = G_i + P_{i-1} \cdot H_{i-1} \quad (3.8)$$

Ο υπολογισμός των H_i απευθείας από τα G_i και P_{i-1} δεν επιφέρει καμία επιτάχυνση στον υπολογισμό και η εγγενώς απλοποιημένη μορφή των κρατούμενων του Ling πηγαίνει χαμένη. Αυτό μπορεί να γίνει εύκολα αντιληπτό αν γράψουμε την τιμή του H_3 με τη χρήση του τελεστή $\circ$ όπως και στην περίπτωση της κλασσικής μορφής των αθροιστών παράλληλου προθέματος.

$$H_3 \leftrightarrow (G_3, P_2) \circ (G_2, P_1) \circ (G_1, P_0) \circ (G_0, P_{-1})$$

Όπως βλέπουμε, ο υπολογισμός του H_3 απαιτεί και πάλι 4 όρους της μορφής (G_i, P_{i-1}) και δε διαφέρει καθόλου σε πολυπλοκότητα συγκρινόμενος με τη σχέση παράλληλου προθέματος που υπολογίζει το C_3 στην κλασσική περίπτωση.

Για το λόγο αυτό, προτείνουμε μια διαφορετική μορφοποίηση της αναδρομικής σχέσης, η οποία όπως θα φανεί στη συνέχεια, προσφέρει αποδοτικότερες υλοποιήσεις και αξιοποιεί με τον καλύτερο δυνατό τρόπο την απλοποιημένη μορφή των Ling κρατούμενων. Αυτό που κάνουμε είναι να ανοίξουμε τη σχέση (3.8) μία φορά και να συλλέξουμε νέους όρους για το στάδιο της προεπεξεργασίας του αθροιστή. Αυτό συμβαίνει ως εξής:

$$\begin{aligned} H_i &= G_i + P_{i-1} \cdot (G_{i-1} + P_{i-2} \cdot H_{i-2}) \\ &= G_i + G_{i-1} + P_{i-1} \cdot P_{i-2} \cdot H_{i-2} \end{aligned}$$

Εμφανίζοντας δύο νέους όρους G_i^* και P_i^*

$$G_i^* = G_i + G_{i-1} \quad P_i^* = P_i \cdot P_{i-1}, \quad (3.9)$$

όπου στη γενική περίπτωση $G_{-1} = C_{in}$ (κρατούμενο εισόδου) και $P_{-1} = 1$, μπορούμε να φέρουμε τη σχέση που περιγράφει τα κρατούμενα του Ling σε μια νέα μορφή η οποία δίνεται στη σχέση (3.10).

$$H_i = G_i^* + P_{i-1}^* \cdot H_{i-2} \quad (3.10)$$

Σύμφωνα με την (3.10) μπορούμε να παράγουμε κάθε ψευδο-κρατούμενο H_i χρησιμοποιώντας

Σχήμα 3.4: Δύο δένδρα παράλληλου προθέματος, τα οποία υπολογίζουν τα ψευδοκρατούμενα στην περίπτωση των 8 δυαδικών ψηφίων ξεκινώντας από τα νέα ψευδο-ζεύγη γέννησης και διάδοσης κρατουμένου (G_i^*, P_{i-1}^*) .

μόνο $i/2$ όρους της μορφής (G_i^*, P_{i-1}^*) . Αυτό έχει ως αποτέλεσμα, αφ' ενός ο ελάχιστος αριθμός επιπέδων που απαιτείται για τον υπολογισμό κάθε H_i να είναι μικρότερος, δηλαδή $\lceil \log_2 n/2 \rceil$, και αφ' ετέρου, εφόσον η αναδρομή συνδέει το H_i μόνο με το H_{i-2} , μπορούμε να υπολογίσουμε τα ψευδο-κρατούμενα στις άρτιες και στις περιττές θέσεις ανεξάρτητα. Αυτό γίνεται πιο εύκολα κατανοητό αν γράψουμε με τη βοήθεια του τελεστή $\circ$ όλα τα ψευδο-κρατούμενα που προκύπτουν από τη σχέση (3.10) για την περίπτωση ενός αθροιστή των 8 δυαδικών ψηφίων.

$$\begin{aligned}
 H_0 &\leftrightarrow (G_0^*, P_{-1}^*) \\
 H_2 &\leftrightarrow (G_2^*, P_1^*) \circ (G_0^*, P_{-1}^*) \\
 H_4 &\leftrightarrow (G_4^*, P_3^*) \circ (G_2^*, P_1^*) \circ (G_0^*, P_{-1}^*) \\
 H_6 &\leftrightarrow (G_6^*, P_5^*) \circ (G_4^*, P_3^*) \circ (G_2^*, P_1^*) \circ (G_0^*, P_{-1}^*) \\
 H_1 &\leftrightarrow (G_1^*, P_0^*) \\
 H_3 &\leftrightarrow (G_3^*, P_2^*) \circ (G_1^*, P_0^*) \\
 H_5 &\leftrightarrow (G_5^*, P_4^*) \circ (G_3^*, P_2^*) \circ (G_1^*, P_0^*) \\
 H_7 &\leftrightarrow (G_7^*, P_6^*) \circ (G_5^*, P_4^*) \circ (G_3^*, P_2^*) \circ (G_1^*, P_0^*) .
 \end{aligned}$$

Δύο δένδρα παράλληλου προθέματος που υλοποιούν τις παραπάνω εξισώσεις φαίνονται στο σχήμα 3.4. Εύκολα μπορούμε να παρατηρήσουμε πως χρειαζόμαστε ένα επίπεδο παράλληλου προθέματος λιγότερο σε σχέση με τα παραδοσιακά κρατούμενα. Η γενική δομή των προτεινόμενων αθροιστών συνοψίζεται στο σχήμα 3.5, ενώ αντίστοιχα η υλοποίηση κάποιων από τα δένδρα παράλληλου προθέματος με το ελάχιστο λογικό βάθος που μπορούν να χρησιμοποιηθούν στην περίπτωση αθροιστών των 16 δυαδικών ψηφίων φαίνονται στο σχήμα 3.6. Αυτό που μένει να μελετήσουμε είναι αν το πιο σύνθετο στάδιο προεπεξεργασίας τελικά συμβάλλει ή όχι στην επιτάχυνση του υπολογισμού.

Οι νέοι όροι (G_i^*, P_{i-1}^*) μπορούν να υπολογιστούν σε τεχνολογία CMOS με δύο τρόπους. Οι δύο δυνατές υλοποιήσεις φαίνονται στο σχήμα 3.7. Στη μια περίπτωση, μία μόνο σύνθετη

Σχήμα 3.5: Η γενική δομή των προτεινόμενων αθροιστών παράλληλου προθέματος που βασίζονται στον υπολογισμό των κρατουμένων του Ling.

πύλη αρκεί για τον υπολογισμό τους, ενώ στη δεύτερη περίπτωση απαιτούνται δύο επίπεδα από απλούστερες πύλες. Οι δύο υλοποιήσεις, όπως διαπιστώσαμε από τα πειράματά μας, είναι σχεδόν ισοδύναμες τόσο σε καθυστέρηση όσο και σε ενέργεια, με τη δεύτερη να είναι ελαφρά πιο αποδοτική, επειδή απαιτεί μικρότερο μέγεθος πυλών από την πρώτη για την ίδια μέγιστη επιτρεπόμενη χωρητικότητα εισόδου. Σε γενικές γραμμές, στην περίπτωση των παραδοσιακών αθροιστών το κρίσιμο μονοπάτι για τον υπολογισμό των κρατουμένων C_i διασχίζει $\log_2 n$ επίπεδα παράλληλου προθέματος, που αποτελούνται από AOI ή OAI λογικές πύλες, και ένα επίπεδο προεπεξεργασίας που αποτελείται από μία NAND ή NOR πύλη, ανάλογα για τον αν πρόκειται για το σήμα G ή το σήμα P . Αντίστοιχα, για τον υπολογισμό των ψευδο-κρατουμένων, στην περίπτωση των προτεινόμενων αθροιστών, χρειάζονται $\log_2 n - 1$ πύλες AOI ή OAI του δένδρου παράλληλου προθέματος, οι οποίες οδηγούνται από τη μία ή τις δύο πύλες του σχήματος 3.7, ανάλογα με την υλοποίηση που επιλέχθηκε για τον υπολογισμό των (G_i^*, P_{i-1}^*) .

Ουσιαστικά, τα τελευταία $\log_2 n - 1$ στάδια του δένδρου παράλληλου προθέματος είναι ίδια και για τις δύο μορφές αθροιστών. Επομένως, ο υπολογισμός των ψευδο-κρατουμένων θα ήταν πιο γρήγορος αν τα κυκλώματα του σχήματος 3.7 ήταν πιο γρήγορα από μια NAND και μια OAI πύλη, που υπάρχουν στα δύο πρώτα επίπεδα του κρίσιμου μονοπατιού των παραδοσιακών αθροιστών. Στην περίπτωση που χρησιμοποιήσουμε μία σύνθετη πύλη, μειώνουμε κατά μία πύλη το κρίσιμο μονοπάτι του αθροιστή σε σχέση με τις παραδοσιακές υλοποιήσεις. Η πύλη που χρησιμοποιούμε μπορεί να είναι κάπως πιο αργή από μια απλή NAND, είναι όμως πάντα πιο γρήγορη από τις 2 πύλες στη σειρά NAND και OAI που χρειάζονται στην περίπτωση των παραδοσιακών αθροιστών. Στην περίπτωση που χρησιμοποιήσουμε δύο πύλες στη σειρά, τότε το συνολικό πλήθος σταδίων και στα δύο κυκλώματα είναι το ίδιο. Για τους Ling αθροιστές 1 NAND/NOR + 1 NAND/NOR + $\log_2 n - 1$ OAI/AOI ενώ για τους παραδοσιακούς αθροιστές 1 NAND/NOR + $\log_2 n$ OAI/AOI. Επομένως, η διαφορά στην ταχύτητα μεταξύ των δύο κυκλωμάτων εξαρτάται από την διαφορά της καθυστέρησης για το ίδιο φορτίο εξόδου μεταξύ μιας πύλης NOR ή NAND και μιας πύλης OAI/AOI που εμφανίζεται στους παραδοσιακούς

Σχήμα 3.6: Τα προτεινόμενα δένδρα παράλληλου προθέματος στην περίπτωση των 16 δυαδικών ψηφίων.

Σχήμα 3.7: Οι δύο διαφορετικοί τρόποι υπολογισμού των νέων ψευδο-ζευγών γέννησης και διάδοσης κρατούμενου (G_i^*, P_{i-1}^*). (α) Σε ένα επίπεδο λογικής με τη χρήση σύνθετων πυλών και (β) σε δύο επίπεδα λογικής.

αθροιστές. Από τον πίνακα 2.1 είναι εύκολο να διαπιστώσουμε πως οι NOR ή NAND πύλες είναι ταχύτερες περίπου της OAI/AOI κατά 30% σε συνθήκες χαμηλής ηλεκτρικής προσπάθειας και αυξάνει περισσότερο από 50% για μεγάλη ηλεκτρική προσπάθεια. Συνεπώς, ανεξάρτητα από την υλοποίηση που θα διαλέξουμε για τα ψευδο-κρατούμενα οδηγούμαστε πάντα σε πιο γρήγορα κυκλώματα. Το ακριβές κέρδος θα ποσοτικοποιηθεί ύστερα από τις πειραματικές μετρήσεις που παρουσιάζουμε στην ενότητα 3.4.

Στη γενική περίπτωση που το μήκος των δεδομένων εισόδου δεν είναι δύναμη του δύο, η σχεδίαση αποδοτικών δένδρων παράλληλου προθέματος βασίζεται στην απορροφητική ιδιότητα (idempotency property). Σκοπός μας είναι να διερευνήσουμε αν η ιδιότητα αυτή συνεχίζει να ισχύει και για τα ψευδο-κρατούμενα του Ling. Αρχικά, ορίζουμε ότι με $(G_{i:j}^*, P_{i:j}^*)$ συμβολίζουμε τις ομάδες που προέκυψαν συνδυάζοντας διαδοχικούς όρους της μορφής (G_k^*, P_{k-1}^*) .

$$(G_{i:j}^*, P_{i:j}^*) = (G_i^*, P_{i-1}^*) \circ (G_{i-2}^*, P_{i-3}^*) \circ \dots \circ (G_j^*, P_{j-1}^*),$$

Η απορροφητική ιδιότητα στην περίπτωση των ψευδο-κρατούμενων του Ling μπορεί να οριστεί ως εξής: Αν i, m, k και j είναι όλοι είτε περιττοί είτε άρτιοι ακέραιοι αριθμοί και $i > m \geq k > j$, τότε ισχύει πως $(G_{i:j}^*, P_{i:j}^*) = (G_{i:k}^*, P_{i:k}^*) \circ (G_{m:j}^*, P_{m:j}^*)$. Αυτό μπορούμε να το αποδείξουμε ως

εξής: Εφόσον $m \geq k$ και $k > j$ τότε μπορούμε να γράψουμε ότι

$$(G_{i:k}^*, P_{i:k}^*) \circ (G_{m:j}^*, P_{m:j}^*) = [(G_{i:m-2}^*, P_{i:m-2}^*) \circ (G_{m:k}^*, P_{m:k}^*)] \circ [(G_{m:k}^*, P_{m:k}^*) \circ (G_{k-2:j}^*, P_{k-2:j}^*)]$$

Αντικαθιστώντας τον ορισμό του τελεστή $\circ$ μεταξύ των δύο $G_{m:k}^*$ προκύπτει ότι

$$(G_{i:k}^*, P_{i:k}^*) \circ (G_{m:j}^*, P_{m:j}^*) = (G_{i:m-2}^*, P_{i:m-2}^*) \circ (G_{m:k}^* + P_{m:k}^* \cdot G_{m:k}^*, P_{m:k}^* \cdot P_{m:k}^*) \circ (G_{k-2:j}^*, P_{k-2:j}^*)$$

το οποίο είναι ισοδύναμο με

$$(G_{i:k}^*, P_{i:k}^*) \circ (G_{m:j}^*, P_{m:j}^*) = (G_{i:m-2}^*, P_{i:m-2}^*) \circ (G_{m:k}^*, P_{m:k}^*) \circ (G_{k-2:j}^*, P_{k-2:j}^*)$$

Μαζεύοντας διαδοχικά τους όρους από τα αριστερά προς τα δεξιά προκύπτει ότι

$$(G_{i:k}^*, P_{i:k}^*) \circ (G_{m:j}^*, P_{m:j}^*) = (G_{i:k}^*, P_{i:k}^*) \circ (G_{k-2:j}^*, P_{k-2:j}^*) = (G_{i:j}^*, P_{i:j}^*)$$

που είναι και το ζητούμενο.

3.2.1 Υπολογισμός του αθροίσματος

Ο υπολογισμός των ψευδο-κρατουμένων του Ling αντί των παραδοσιακών κρατουμένων περιπλέκει κάπως τον υπολογισμό των δυαδικών ψηφίων του αθροίσματος. Για το i -οστό δυαδικό ψηφίο του αθροίσματος S_i χρειαζόμαστε το κρατούμενο C_{i-1} το οποίο δεν έχουμε πλέον στη διάθεση μας ($S_i = T_i \oplus C_{i-1}$). Αυτό που έχουμε όμως υπολογίσει είναι το H_{i-1} το οποίο μπορεί να δώσει το ζητούμενο άθροισμα σύμφωνα με την παρακάτω σχέση

$$S_i = T_i \oplus (P_{i-1} \cdot H_{i-1}) \quad (3.11)$$

Αν υπολογίσουμε το άθροισμα υλοποιώντας απευθείας την προηγούμενη εξίσωση, τότε θα χρειαζόμασταν δύο τουλάχιστον επιπλέον πύλες στο κρίσιμο μονοπάτι. Η μία θα παρήγαγε τον όρο $P_{i-1} \cdot H_{i-1}$ και η επόμενη θα έκανε το λογικό XOR αυτού του όρου με το T_i . Κάτι τέτοιο δεν μας συμφέρει καθώς θα ακύρωνε το κέρδος που πετύχαμε σε καθυστέρηση αναμορφώνοντας τον υπολογισμό των κρατουμένων του Ling. Για το λόγο αυτό, καλούμαστε να αναδιατάξουμε τον υπολογισμό των ψηφίων S_i , με σκοπό να μειώσουμε την καθυστέρηση του μονοπατιού από το H_{i-1} , που υπολογίζεται αργότερα χρονικά, προς την έξοδο. Αυτό μπορεί να συμβεί εφαρμόζοντας την τεχνική επιλογής κρατουμένου. Σ' αυτήν την περίπτωση η σχέση (3.11) γράφεται ως εξής:

$$S_i = \overline{H_{i-1}} \cdot T_i + H_{i-1} \cdot (T_i \oplus P_{i-1}), \quad (3.12)$$

Το άθροισμα ισούται είτε με T_i είτε με $T_i \oplus P_{i-1}$, ανάλογα με την τιμή του ψευδο-κρατουμένου H_{i-1} . Αυτό υλοποιείται με έναν πολυπλέκτη 2-σε-1, ο οποίος ελέγχεται από το ψευδο-κρατούμενο H_{i-1} . Και οι δύο επιλογές μπορούν να υπολογιστούν ταχύτερα από το H_{i-1} και έτσι το κρίσιμο μονοπάτι περνάει από το σήμα επιλογής του πολυπλέκτη προς την έξοδο μειώνοντας έτσι κατά μία πύλη την καθυστέρηση του κρίσιμου μονοπατιού. Εφόσον η καθυστέρηση ενός πολυπλέκτη διαφέρει ελάχιστα από αυτήν της πύλης XOR, η οποία χρησιμοποιείται ούτως ή άλλως για τον υπολογισμό του αθροίσματος από τα παραδοσιακά κρατούμενα,

Σχήμα 3.8: Ο γρήγορος υπολογισμός του αθροίσματος στην περίπτωση των προτεινόμενων αθροιστών.

μπορούμε εύκολα να θεωρήσουμε πως μετά από αυτή την προσαρμογή ο υπολογισμός των Ling κρατούμενων δεν επιβαρύνει χρονικά τον υπολογισμό του αθροίσματος. Η υιοθέτηση των κρατούμενων του Ling, όπως φαίνεται στο σχήμα 3.8, απλά αυξάνει το απαιτούμενο υλικό,

3.3 Δομές παράλληλου προθέματος και επιλογής κρατουμένου

Στην ενότητα αυτή θα παρουσιάσουμε τον τρόπο με τον οποίο μπορούμε να σχεδιάσουμε υβριδικές δομές παράλληλου προθέματος και επιλογής κρατουμένου, οι οποίες χρησιμοποιούν τα ψευδο-κρατούμενα του Ling. Οι υβριδικές αυτές δομές βρίσκουν ευρεία χρήση στους σύγχρονους μικροεπεξεργαστές γιατί πετυχαίνουν σχεδόν την ίδια ταχύτητα με τις κλασσικές δομές παράλληλου προθέματος, έχοντας όμως σημαντικά μειωμένο αριθμό καλωδίων και κατανάλωση ισχύος.

Η κεντρική ιδέα κατασκευής των δομών αυτών είναι αρκετά απλή. Αντί να παράγουμε τα κρατούμενα σε κάθε θέση του δένδρου (n στο σύνολο τους) δημιουργούμε μόνο ορισμένα από αυτά. Έτσι καταφέρνουμε να μειώσουμε σημαντικά την πολυπλοκότητα του δένδρου παράλληλου προθέματος. Πώς όμως θα υπολογίσουμε το άθροισμα σε όλες τις θέσεις του αθροιστή, αφού πλέον δεν έχουμε στη διάθεση μας τα κρατούμενα αλλά μόνο ένα μέρος από αυτά; Την απάντηση στο ερώτημα αυτό θα τη δώσουμε μέσα από ένα παράδειγμα. Έστω, λοιπόν, ότι θέλουμε να σχεδιάσουμε έναν αθροιστή, στον οποίο παράγουμε ένα κρατούμενο κάθε τέσσερις θέσεις του δένδρου παράλληλου προθέματος. Με άλλα λόγια, υπολογίζονται μόνο τα κρατούμενα C_{4k-1} με $k = 0, 1, 2, \dots$. Για τις υπόλοιπες θέσεις δεν έχουμε κανένα κρατούμενο για να υπολογίσουμε το ζητούμενο άθροισμα. Για να μπορέσουμε να ολοκληρώσουμε το σχεδιασμό, φροντίζουμε να προ-υπολογίσουμε για αυτές τις θέσεις δύο διαφορετικές τιμές του τοπικού αθροίσματος $S_{4k+3}S_{4k+2}S_{4k+1}S_{4k}$ ανάλογα με την τιμή του C_{4k-1} . Η τιμή που υπολογίζουμε θεωρώντας ότι το $C_{4k-1} = 0$ αναπαρίσταται ως S_j^0 , ενώ αντίθετα η περίπτωση που $C_{4k-1} = 1$ συμβολίζεται με S_j^1 . Ουσιαστικά το κρατούμενο C_{4k-1} λειτουργεί ως κρατούμενο εισόδου για τους διπλούς αθροιστές των 4 δυαδικών ψηφίων, οι οποίοι υλοποιούνται σαν

Σχήμα 3.9: Η υβριδική δομή παράλληλου προθέματος και επιλογής κρατουμένου των 32 δυαδικών ψηφίων στην περίπτωση των παραδοσιακών κρατουμένων.

Σχήμα 3.10: Η υλοποίηση της μονάδας επιλογής κρατουμένου, η οποία εκτείνεται στα 4 δυαδικά ψηφία.

απλοί αθροιστές ripple-carry, αφού δε συμμετέχουν στο κρίσιμο μονοπάτι του κυκλώματος. Έτσι, έχοντας αυτές τις τιμές διαθέσιμες πριν υπολογιστούν όλα τα κρατούμενα C_{4k-1} από το αραιό δένδρο παράλληλου προθέματος, αυτό που μένει είναι να προσθέσουμε ένα επίπεδο από πολυπλέκτες, οι οποίοι θα ελέγχονται από τις τιμές των C_{4k-1} και θα διαλέγουν μία από τις δύο ήδη υπολογισμένες τιμές του τοπικού αθροίσματος S_j^0 ή S_j^1 . Ένα παράδειγμα αυτής της οργάνωσης φαίνεται στο σχήμα 3.9 για την περίπτωση των κλασσικών αθροιστών παράλληλου προθέματος των 32 δυαδικών ψηφίων. Αντίστοιχα, στο σχήμα 3.10 φαίνεται η υλοποίηση της μονάδας επιλογής κρατουμένου που υπολογίζει τα αθροίσματα των θέσεων 24-27.

Η σχεδίαση υβριδικών δομών παράλληλου προθέματος και επιλογής κρατουμένου στην περίπτωση των αθροιστών που υπολογίζουν τα ψευδο-κρατούμενα του Ling απαιτεί κάποιες μικρές αλλαγές στη σχεδίαση του κελιού επιλογής κρατουμένου (carry select block). Πρώτιστα, οι αλλαγές αυτές είναι αναγκαίες, γιατί, στην περίπτωση αυτή, η επιλογή του τελικού

Σχήμα 3.11: Η υβριδική δομή παράλληλου προθέματος και επιλογής κρατουμένου στην περίπτωση των προτεινόμενων αθροιστών που υιοθετούν τα κρατούμενα του Ling.

αθροίσματος πρέπει να γίνει με βάση τα ψευδο-κρατούμενα H_i . Έτσι, απλές δομές, όπως οι αθροιστές ripple-carry που χρησιμοποιούνται στην κλασσική μορφή των υβριδικών αθροιστών, δεν μπορούν απευθείας να χρησιμοποιηθούν. Επίσης, η τοπολογία του τελικού κυκλώματος πρέπει να είναι διαφορετική, επειδή τα ψευδο-κρατούμενα από τις άρτιες και τις περιττές θέσεις του δένδρου υπολογίζονται χωριστά, ενώ θέλουμε να χρησιμοποιήσουμε όσο το δυνατόν περισσότερο τα ήδη υπολογισμένα G_i^* και P_i^* .

Το αντίστοιχο υβριδικό κύκλωμα των 32 δυαδικών ψηφίων, το οποίο προέκυψε από την προτεινόμενη μεθοδολογία σχεδιασμού, φαίνεται στο σχήμα 3.11. Τα ψευδο-κρατούμενα H_{4k} και H_{4k-1} υπολογίζονται στις άρτιες και στις περιττές θέσεις του δένδρου και χρησιμοποιούνται για την επιλογή των δυαδικών ψηφίων του αθροίσματος, τα οποία έχουν προϋπολογιστεί από την αναθεωρημένη μονάδα επιλογής κρατουμένου Modified Carry-Select Adders (MCSA) των 4ων δυαδικών ψηφίων. Για να εξηγήσουμε τη λειτουργία του MCSA θα πάρουμε σαν παράδειγμα την περίπτωση του S_{28} , η τιμή του οποίου πρέπει να επιλεγεί από το H_{23} που έχουμε στη διάθεση μας. Γνωρίζουμε ότι $S_{28} = T_{28} \oplus P_{27} \cdot H_{27}$. Επομένως, για να συνδέσουμε την τιμή του S_{28} με την τιμή του H_{23} πρέπει να ανοίξουμε τη σχέση (3.10) για το H_{27} τρεις φορές. Μετά από αυτή τη μετατροπή προκύπτει ότι

$$H_{27} = G_{27}^* + P_{26}^* \cdot G_{25}^* + P_{26}^* \cdot P_{24}^* \cdot H_{23}.$$

Τώρα, ανάλογα με την τιμή του H_{23} μπορούμε να υπολογίσουμε τις δύο υπο-συνθήκη τιμές για το άθροισμα στη θέση 28, S_{28}^0 και S_{28}^1 αντίστοιχα. Οι σχέσεις που προκύπτουν αντικαθιστώντας την τιμή του H_{27} στη βασική σχέση του αθροίσματος και θέτοντας το H_{23} στις τιμές 0 και 1

Σχήμα 3.12: Ένα παράδειγμα της αλλαγμένη δομής επιλογής κρατούμενου (M-CSA) που χρησιμοποιείται στην περίπτωση των προτεινόμενων Ling αθροιστών.

αντίστοιχα είναι οι παρακάτω.

$$\begin{aligned}
 S_{28}^0 &= T_{28} \oplus [P_{27} \cdot (G_{27}^* + P_{26}^* \cdot G_{25}^*)] \\
 &= \overline{(G_{27}^* + P_{26}^* \cdot G_{25}^*)} \cdot T_{28} + (G_{27}^* + P_{26}^* \cdot G_{25}^*) \cdot (P_{27} \oplus T_{28}), \\
 S_{28}^1 &= T_{28} \oplus [P_{27} \cdot (G_{27}^* + P_{26}^* \cdot (G_{25}^* + P_{24}^*))] \\
 &= \overline{(G_{27}^* + P_{26}^* \cdot (G_{25}^* + P_{24}^*))} \cdot T_{28} + (G_{27}^* + P_{26}^* \cdot (G_{25}^* + P_{24}^*)) \cdot (P_{27} \oplus T_{28}).
 \end{aligned}$$

Μετά από αυτόν το μετασχηματισμό μπορούμε να υπολογίσουμε την τιμή του S_{28}^0 χρησιμοποιώντας ένα πολυπλέκτη ο οποίος θα επιλέγει μεταξύ του T_{28} και του $(P_{27} \oplus T_{28})$ ανάλογα με την τιμή του $(G_{27}^* + P_{26}^* \cdot G_{25}^*)$. Το ίδιο μπορεί να συμβεί και για την περίπτωση του S_{28}^1 όπου η τελική τιμή του επιλέγεται βάση της τιμής του σήματος $(G_{27}^* + P_{26}^* \cdot (G_{25}^* + P_{24}^*))$. Εφόσον έχουμε στη διάθεση μας τις τιμές των S_{28}^0 και S_{28}^1 , το τελικό άθροισμα επιλέγεται από την τιμή του H_{23} όταν αυτό υπολογιστεί από το προτεινόμενο δένδρο παράλληλου προθέματος. Η υλοποίηση του νέου κελιού επιλογής κρατούμενου που βασίζεται στα ψευδο-κρατούμενα του Ling φαίνεται στο σχήμα 3.12.

3.4 Πειραματικά αποτελέσματα

Για να μπορέσουμε να συγκρίνουμε τους προτεινόμενους αθροιστές σε σχέση με τις ήδη υπάρχουσες τοπολογίες παράλληλου προθέματος πραγματοποιήσαμε δύο ομάδες πειραμάτων. Στη μια περίπτωση ακολουθήσαμε μια αυτοματοποιημένη ροή σχεδιασμού. Ο σχεδιασμός προέκυψε μετά από την αντιστοίχισή του σε μια προσχεδιασμένη βιβλιοθήκη βασικών πυλών των 180nm, ενώ οι μετρήσεις πραγματοποιήθηκαν με τα κατάλληλα εργαλεία σε επίπεδο λογικών πυλών. Στην άλλη περίπτωση, ακολουθήσαμε έναν πλήρως ελεύθερο τρόπο σχεδίασης, όπου τα μεγέθη των τρανζίστορ καθορίστηκαν σύμφωνα με το εργαλείου γεωμετρικού προγραμματισμού που υλοποιήσαμε, ενώ οι μετρήσεις της ενέργειας και της καθυστέρησης πραγματοποιήθηκαν σε Spice για τη διαθέσιμη τεχνολογία υλοποίησης των 130nm.

Οι αθροιστές που σχεδιάσαμε ακολουθούν τις τοπολογίες παράλληλου προθέματος που

Πίνακας 3.1: Το εμβαδόν και η καθυστέρηση των παραδοσιακών και των προτεινόμενων δομών χρησιμοποιώντας τα δένδρα παράλληλου προθέματος Ladner-Fischer και Kogge-Stone στα 0.18μm.

n	Ladner-Fischer					Kogge-Stone				
	Area (μm^2)		Delay (FO4)			Area (μm^2)		Delay (FO4)		
	Norm.	Prop.	Norm.	Prop.	Saving	Norm.	Prop.	Norm.	Prop.	Saving
16	4336	4928	10.8	9.3	13.8%	5716	5755	10.3	8.83	14%
32	10250	11038	13.1	11.3	13.7%	13404	13561	12.6	11	13%
64	26808	28132	16.3	14.1	13.4%	33205	33904	14.8	13.3	10%

προτάθηκαν από τους Ladner-Fischer [91] και Kogge-Stone [85] αντίστοιχα. Οι δύο αυτές τοπολογίες αποτελούν τα δύο άκρα του συνόλου των δένδρων παράλληλου προθέματος, τα οποία πρότεινε ο Knowles στην εργασία [84]. Η πρώτη περίπτωση απαιτεί το μικρότερο αριθμό από τελεστές οι έξοδοι των οποίων πρέπει όμως να υποστηρίξουν τις μεγάλες χωρητικότητες των εσωτερικών γραμμών του κυκλώματος. Στη δεύτερη περίπτωση χρησιμοποιούνται οι περισσότεροι τελεστές, οι πύλες των οποίων όμως έχουν μικρά μεγέθη, μιας και οδηγούν την ελάχιστη δυνατή χωρητικότητα δηλαδή δύο πύλες του επόμενου επιπέδου. Οι υπόλοιπες τοπολογίες δένδρων παράλληλου προθέματος δεν εξετάζονται, γιατί θεωρούμε πως θα δίνουν αποτελέσματα που θα βρίσκονται μεταξύ αυτών των δύο λύσεων όσον αφορά στην ενέργεια και στην καθυστέρηση.

Η αυτόματη διαδικασία σχεδιασμού περιελάμβανε τα εξής βήματα. Αρχικά, κάθε αθροιστής περιγράφηκε σε Verilog HDL και αντιστοιχήθηκε στις πύλες της τεχνολογίας [158] μεγέθους 180nm με τυπική τάση τροφοδοσίας τα 1.8V χρησιμοποιώντας το εργαλείο Synopsys Design Compiler v.2003.06. Η τεχνολογία αυτή έχει υποστεί πολλές βελτιώσεις και είναι από τη φύση της πολύ γρήγορη. Ουσιαστικά η τιμή του FO4 = 60ps για την τεχνολογία αυτή είναι μικρότερη από ότι στην τεχνολογία των 130nm που χρησιμοποιούμε στην περίπτωση του πλήρως ελεύθερου σχεδιασμού. Κάθε σχεδιασμός βελτιστοποιήθηκε επαναληπτικά με σκοπό τη μείωση της ταχύτητας. Έπειτα, οι τοπολογίες οι οποίες προέκυψαν μαζί με το σύνολο περιορισμών, όπως για παράδειγμα η στοχευόμενη ταχύτητα, δόθηκαν στο εργαλείο Cadence Silicon Ensemble v.5.3, το οποίο ανέλαβε την αυτόματη χωροθέτηση των πυλών και τη σύνδεση τους. Όλοι οι σχεδιασμοί που μελετήσαμε πέρασαν από την ίδια διαδικασία διατηρώντας σταθερούς σε όλες τις περιπτώσεις τους σχεδιαστικούς περιορισμούς, όπως το φορτίο εξόδου (100fF), τις πληροφορίες αρχικοποίησης της χωροθέτησης και την θέση των ακροδεκτών εισόδου και εξόδου. Η τελική καθυστέρηση κάθε κυκλώματος μετρήθηκε με τη βοήθεια του εργαλείου PrimeTime της Synopsys, στο οποίο, πέρα από το σχεδιασμό, δόθηκαν και οι τιμές όλων των παρασιτικών χωρητικοτήτων και αντιστάσεων των καλωδίων που προέκυψαν από τη χωροθέτηση του κυκλώματος.

Τα αποτελέσματα που πήραμε όσον αφορά στην καθυστέρηση και στο απαιτούμενο εμβαδόν κάθε κυκλώματος για διαφορετικές τιμές του μήκους λέξης της εισόδου, φαίνονται στον πίνακα 3.1. Από τα αποτελέσματα διαπιστώνουμε πως οι προτεινόμενοι αθροιστές έχουν σε όλες τις περιπτώσεις τη μικρότερη καθυστέρηση. Το κέρδος σε ταχύτητα είναι μεγαλύτερο στην περίπτωση των αθροιστών που ακολουθούν τη δομή παράλληλου προθέματος του Ladner-Fischer. Τότε η μέγιστη χωρητικότητα που οδηγούν οι πύλες του κυκλώματος, μειώνεται

Πίνακας 3.2: Το εμβαδόν και η καθυστέρηση των υπο-εξέταση υβριδικών δομών των 64 δυαδικών ψηφίων όπου η επιλογή κρατουμένου εκτείνεται στα 4 δυαδικά ψηφία.

64-bit hybrid adders	Area (μm^2)	Delay (FO4)
Ladner-Fischer Normal	21228	15.1
Ladner-Fischer Proposed	23654	13.8
Kogge-Stone Normal	26754	14.8
Kogge-Stone Proposed	28385	13.5

Σχήμα 3.13: Οι καμπύλες ενέργειας ως προς την καθυστέρηση που προέκυψαν μετά τη βελτιστοποίηση και τη μέτρηση των κυκλωμάτων στην τεχνολογία 130nm θεωρώντας $C_{in}/C_{out} = 25/100fF$. Οι δομές παράλληλου προθέματος που χρησιμοποιήθηκαν είναι (α) η Kogge-Stone και (β) η Ladner-Fischer.

στο μισό σε σχέση με τους απλούς αθροιστές, ενώ οι πύλες που εμφανίζονται στο κρίσιμο μονοπάτι είναι απλούστερες. Η αξία της απλοποίησης αυτής των πυλών φαίνεται καλύτερα στην περίπτωση της Kogge-Stone δομής παράλληλου προθέματος, όπου τόσο οι κλασικοί όσο και οι προτεινόμενοι αθροιστές έχουν το ίδιο φορτίο στις εσωτερικές γραμμές του κυκλώματος. Σ' αυτές τις περιπτώσεις οι δομές που προτείνουμε είναι και πάλι πάντα ταχύτερες. Το μέσο κέρδος σε ταχύτητα που επιτυγχάνουμε για όλες τις περιπτώσεις ισούται περίπου με 13%.

Επίσης, χρησιμοποιώντας την αυτοματοποιημένη διαδικασία σχεδιασμού συγκρίναμε τις κλασσικές υβριδικές δομές παράλληλου προθέματος και επιλογής κρατουμένου με αυτές που προέκυψαν από την προτεινόμενη προσέγγιση. Η σύγκριση αφορά αθροιστές των 64 δυαδικών ψηφίων όπου η επιλογή κρατουμένου εκτείνεται σε 4 θέσεις. Τα αποτελέσματα που συλλέξαμε για την μικρότερη δυνατή καθυστέρηση των υπο-εξέταση κυκλωμάτων φαίνονται στο Πίνακα 3.2. Όπως προκύπτει οι προτεινόμενες δομές είναι κατά μέσο όρο 9% πιο γρήγορες, ενώ καταλαμβάνουν σ' αυτή την καθυστέρηση περισσότερο χώρο στο ολοκληρωμένο περίπου με το ίδιο ποσοστό.

Τέλος, ελέγξαμε την απόδοση των προτεινόμενων δομών στο χώρο ενέργεια καθυστέρηση χρησιμοποιώντας μια πλήρως ελεύθερη διαδικασία σχεδιασμού υποβοηθούμενοι από το εργαλείο επιλογής των μεγεθών των πυλών που δημιουργήσαμε. Οι μετρήσεις που πήραμε για την περίπτωση των αθροιστών των 64 δυαδικών ψηφίων που ακολουθούν τη δομή του Ladner-

Fischer και του Kogge-Stone δένδρου παράλληλου προθέματος αντίστοιχα, φαίνονται στο σχήμα 3.13. Υπενθυμίζουμε πως τα αποτελέσματα αυτά αφορούν την τεχνολογία υλοποίησης των 130nm, στην οποία η τιμή του FO4 στις τυπικές συνθήκες λειτουργίας είναι ίση με 62ps. Αυτή τη φορά τα αποτελέσματα έχουν μεγαλύτερη ακρίβεια εφόσον οι μετρήσεις πραγματοποιήθηκαν στο Spice. Η γενική συμπεριφορά των αποτελεσμάτων είναι ίδια με αυτή που φάνηκε στην περίπτωση του αυτοματοποιημένου σχεδιασμού. Επίσης, μπορούμε ξεκάθαρα να διαπιστώσουμε πως η αυτόματη διαδικασία σχεδιασμού και η χρησιμοποίηση της βιβλιοθήκης προσχεδιασμένων πυλών επιβαρύνει σημαντικά την καθυστέρηση του κυκλώματος. Στο βέλτιστο σημείο όσον αφορά την καθυστέρηση οι δύο λύσεις μπορεί να διαφέρουν έως και 4FO4. Η διαφορά αυτή οφείλεται κυρίως στην μικρή ποικιλία μεγεθών των πυλών της προσχεδιασμένης βιβλιοθήκης, που έχει ως αποτέλεσμα το εργαλείο να μην μπορεί να επιλέξει τα βέλτιστα μεγέθη πυλών που ταιριάζουν σε κάθε μονοπάτι. Φυσικά, αυτό το πρόβλημα δεν υπάρχει στην περίπτωση της βελτιστοποίησης με τη χρήση του γεωμετρικού προγραμματισμού, όπου το μέγεθος των τρανζίστορ μπορεί να καθοριστεί στην κατάλληλη τιμή ικανοποιώντας τους περιορισμούς που τίθενται, εκτός ελαχίστων εξαιρέσεων. Συνοψίζοντας τα αποτελέσματα που φαίνονται στα διαγράμματα του σχήματος 3.13, προκύπτει ότι οι προτεινόμενοι αθροιστές είναι πιο γρήγοροι κατά 10% κατά μέσο όρο, χωρίς να έχουν σημαντικές διαφορές στην κατανάλωση ενέργειας με τους παραδοσιακούς αθροιστές. Για τα δένδρα Kogge-Stone οι προτεινόμενοι σχεδιασμοί προσφέρουν πάντα τους πιο αποδοτικούς σχεδιασμούς, ενώ στην περίπτωση των δένδρων Ladner-Fischer οι παραδοσιακοί αθροιστές, αν και με μεγαλύτερη καθυστέρηση, απαιτούν λιγότερη ενέργεια για την υλοποίηση τους.

Αναδιοργάνωση δεδομένων

4.1 Εισαγωγή

Οι εφαρμογές πολυμέσων, όπως επεξεργασία εικόνας και βίντεο, τρισδιάστατα γραφικά, επεξεργασία ήχου και κειμένου, αποτελούν πλέον αναπόσπαστο κομμάτι του φόρτου εργασίας των επεξεργαστών γενικού σκοπού [34]. Αυτό εξηγεί τη διαρκή προσθήκη νέων εντολών στα υπάρχοντα σύνολα εντολών των επεξεργαστών με σκοπό την αποτελεσματικότερη εκτέλεση τους [101]. Οι αλλαγές σε αρχιτεκτονικό επίπεδο αποσκοπούν στη βελτίωση της απόδοσης εκμεταλλευόμενες τα ιδιαίτερα χαρακτηριστικά των εφαρμογών πολυμέσων. Ένα από τα κύρια χαρακτηριστικά τους είναι η χρησιμοποίηση δεδομένων χαμηλής ακρίβειας (όπως ένα ή δύο bytes) για την αναπαράσταση της πληροφορίας. Η ελάχιστη συσχέτιση ανάμεσα στα δεδομένα αυτά επιτρέπει την παράλληλη επεξεργασία τους [29], [148]. Έτσι είναι εφικτό να ομαδοποιήσουμε μέσα σε μία λέξη του επεξεργαστή περισσότερα από ένα δεδομένα τα οποία αποθηκεύονται στον ίδιο καταχωρητή και μπορούν να επεξεργαστούν παράλληλα με τη χρήση μίας μόνο εντολής. Τα δεδομένα που τοποθετούνται παράλληλα σε μια λέξη του επεξεργαστή καλούνται υπολέξεις. Για παράδειγμα σε μια λέξη των 64 δυαδικών ψηφίων μπορούν να αποθηκευτούν 8 υπολέξεις του ενός byte, ή 4 υπολέξεις των 2 bytes είτε 2 υπολέξεις των 4 bytes. Η λογική αυτή της αποθήκευσης και της εκτέλεσης των εντολών προσομοιάζει τους διανυσματικούς επεξεργαστές και ονομάζεται SIMD (Single Instruction Multiple Data).

Χαρακτηριστικά παραδείγματα επαύξησης του συνόλου εντολών των εμπορικών επεξεργαστών με στόχο την καλύτερη υποστήριξη των πολυμεσικών εφαρμογών είναι οι ομάδες εντολών MAX-2 [97], MMX [114], VIS [113], SSE [157] (μαζί με τις μεταγενέστερες εκδόσεις SSE 2-4), AltiVec [35] και 3DNOW! [123]. Επίσης, από ακαδημαϊκής πλευράς, αξίζει να αναφέρουμε το σύνολο εντολών PLX [98]. Ευρεία χρήση παρόμοιων εντολών που ακολουθούν τη

Σχήμα 4.1: Παραδείγματα εντολών πολυμέσων που ακολουθούν την φιλοσοφία του SIMD. Η λέξη του επεξεργαστή είναι των 64 δυαδικών ψηφίων στην οποία αποθηκεύονται 8 υπολέξεις του 1 byte.

φιλοσοφία SIMD συναντούμε, πέρα από τους επεξεργαστές γενικού σκοπού, σε εξειδικευμένους επεξεργαστές ψηφιακής επεξεργασίας σήματος και τηλεπικοινωνιακών συστημάτων, οι οποίοι αναλαμβάνουν όλη την επεξεργασία του σήματος βασικής ζώνης (baseband processors) [102] καθώς και σε επεξεργαστές ενσωματωμένων συστημάτων με έμφαση την επεξεργασία ήχου, γραφικών και βίντεο [161].

Οι SIMD εντολές που εμφανίζονται στα νέα σύνολα εντολών μπορούν να χωριστούν σε τέσσερις βασικές κατηγορίες:

- Στην πρώτη ανήκουν εκείνες οι εντολές που εκτελούν βασικές αριθμητικές πράξεις, όπως πρόσθεση και πολλαπλασιασμός, πάνω στις υπολέξεις οι οποίες μπορούν να αναπαριστούν ακέραια δεδομένα ή αριθμούς κινητής υποδιαστολής. Επίσης, συνήθως υποστηρίζονται οι πράξεις MIN και MAX οι οποίες επιστρέφουν τη μικρότερη ή τη μεγαλύτερη από τις δύο υπολέξεις και χρησιμοποιούνται ευρέως στην ψηφιακή επεξεργασία εικόνας και βίντεο αλλά και για την επιστροφή της κορεσμένης τιμής ενός ακεραίου αποτελέσματος (saturation). Ένα παράδειγμα της παράλληλης πρόσθεσης σε υπολέξεις του ενός byte φαίνεται στο σχήμα 4.1(a).
- Στη δεύτερη κατηγορία διακρίνουμε τις εντολές παράλληλης σύγκρισης των υπολέξεων δύο καταχωρητών. Σε αυτήν την περίπτωση, οι υπολέξεις συγκρίνονται ανεξάρτητα χρησιμοποιώντας σε όλες τον ίδιο τελεστή σύγκρισης (>, <, ==). Για κάθε μία σύγκριση

παράγεται μία σημαία που δηλώνει αν η συνθήκη ικανοποιείται ή όχι. Όλες οι σημαίες αποθηκεύονται στο λιγότερο σημαντικό κομμάτι μιας νέας λέξης εξόδου. Λόγου χάρη, στο σχήμα 4.1(β) η τιμή κάθε σημαίας παίρνει την τιμή 1 στην περίπτωση που η συνθήκη της σύγκρισης ικανοποιείται για το συγκεκριμένο ζεύγος υπολέξεων.

- Στην τρίτη κατηγορία ανήκουν οι εντολές οι οποίες προσπαθούν να συνενώσουν την πληροφορία που είναι αποθηκευμένη σε όλες τις υπολέξεις ενός καταχωρητή του επεξεργαστή. Οι εντολές αυτές είναι γνωστές ως πράξεις μείωσης (reductions) και συνήθως υπολογίζουν το άθροισμα των τιμών των υπολέξεων σε μία μεταβλητή εξόδου. Επίσης, σ' αυτήν την κατηγορία ανήκουν και οι πράξεις εύρεσης της μικρότερης ή της μεγαλύτερης υπολέξης από αυτές που είναι αποθηκευμένες σε έναν καταχωρητή. Ένα παράδειγμα υπολογισμού της μεγαλύτερης υπολέξης που είναι αποθηκευμένη σε μια λέξη του επεξεργαστή φαίνεται στο σχήμα 4.1(γ).
- Στην τελευταία κατηγορία ανήκουν οι εντολές αναδιοργάνωσης των υπολέξεων που περιέχονται σε έναν ή σε περισσότερους καταχωρητές. Οι εντολές αυτές και η υλοποίηση τους είναι το κύριο θέμα αυτού του κεφαλαίου. Η χρησιμότητα και η λειτουργία τους θα αναλυθούν στις επόμενες παραγράφους ενώ μια εκδοχή τους απεικονίζεται στο σχήμα 4.1(δ).

4.2 Εντολές αναδιοργάνωσης δεδομένων

Οι εντολές αναδιοργάνωσης των δεδομένων χρησιμοποιούνται κυρίως για δύο λόγους. Ο προφανής λόγος είναι για να υποστηρίξουν τις απαραίτητες αντιμεταθέσεις δεδομένων που επιβάλλει ο ίδιος ο αλγόριθμος τον οποίο προσπαθούμε να εκτελέσουμε. Για παράδειγμα, είναι γνωστό ότι οι αλγόριθμοι κρυπτογραφίας στηρίζουν τη λειτουργία τους στην αναδιοργάνωση κομματιών του μηνύματος με σκοπό να “μπερδέψουν” ή να “διαχύσουν” την πληροφορία (confuse ή diffuse) ώστε αυτή να μπορεί να γίνει πιο δύσκολα αντιληπτή από μη εγκεκριμένους χρήστες [99]. Επίσης, μια πληθώρα αλγορίθμων ψηφιακής επεξεργασίας σήματος ή πράξεων γραμμικής άλγεβρας, όπως για παράδειγμα ο FFT και η αντιμετάθεση πινάκων, απαιτούν συγκεκριμένες μορφές αντιμετάθεσης των δεδομένων της εισόδου, τα οποία είναι αποθηκευμένα σε χωριστές υπολέξεις των καταχωρητών, με σκοπό να επιταχύνουν τον υπολογισμό [12].

Ο δεύτερος λόγος που επιβάλλει τη χρήση εντολών αντιμετάθεσης των δεδομένων είναι για να βοηθήσει τις βασικές λειτουργίες που υποστηρίζουν οι εντολές SIMD. Χωρίς την απαραίτητη αναδιοργάνωση των δεδομένων η παράλληλη εκτέλεση που παρέχουν οι εντολές SIMD χάνει σε πολλές περιπτώσεις την αξία της και ελάχιστα συμβάλλει στην επιτάχυνση του υπολογισμού [136], [122]. Θα ήταν καλύτερο να εξηγήσουμε τη λειτουργία αυτή χρησιμοποιώντας σαν παράδειγμα τον κώδικα του σχήματος 4.2. Τα διανύσματα a , b και c έχουν μέγεθος 8 bytes και θεωρούμε πως οι τιμές τους μπορούν να αποθηκευτούν, σε μια υποτιθέμενη αρχιτεκτονική, σε δύο λέξεις των 32 δυαδικών ψηφίων σαν υπολέξεις του ενός byte. Για να μεταφέρουμε τα στοιχεία του b από τη μνήμη ώστε να τα επεξεργαστούμε απαιτούνται δύο εντολές `load` ενώ για τα στοιχεία του a απαιτείται μόνο μία εντολή `load` εφόσον τα στοιχεία $a[0]$, $a[1]$, $a[2]$ και $a[3]$ που συμμετέχουν στον πυρήνα του βρόχου χωρούν σε μία λέξη του επεξεργαστή. Για τον ίδιο λόγο μόνο μια `store` εντολή αρκεί για να τοποθετήσει στη μνήμη τις τιμές του $c[0:3]$ που παράγονται από το βρόχο του σχήματος 4.2. Από το σχήμα 4.2 μπορούμε εύκολα να διαπιστώσουμε ότι οι απαιτούμενες πράξεις $a[0]+b[0]$, $a[1]+b[2]$, $a[2]+b[4]$, $a[3]+b[6]$ δεν μπορούν να πραγματοποιηθούν με μία SIMD


```

unsigned char a[8], b[8], c[8];
int i;
for (i=0; i < N; i++) {
    c[i] = a[i] + b[2*i];
}

```

```

LOAD R0, M[a];      // R0 <- a[0:3]
LOAD R1, M[b];      // R1 <- b[0:3]
LOAD R2, M[b+4];    // R2 <- b[4:7]
MIX.EVEN R3,R1,R2;  // Collect in R3 <- B[0],B[2],B[4],B[6]
VEC.ADD R4,R0,R3;   // Parallel Bytes Add R4 <- R0 + R3
STORE R4, M[c];     // c[0:3] <- R4

```

(α)

(β)

Σχήμα 4.2: Ένα παράδειγμα εκτέλεσης ενός προγράμματος με εντολές SIMD στο οποίο φαίνεται η χρησιμότητα των εντολών αναδιοργάνωσης δεδομένων. (α) Το αρχικό πρόγραμμα και η αντιστοίχισή του σε assembly εντολές και (β) η συμβολική εκτέλεση του.

εντολή παράλληλης πρόσθεσης των 4 υπολέξεων γιατί τα απαιτούμενα στοιχεία των a και b δεν είναι μεταξύ τους ευθυγραμμισμένα. Την ευθυγράμμιση των δεδομένων με σκοπό την καλύτερη δυνατή χρήση των εντολών SIMD μπορεί να αναλάβει μία από τις εντολές αναδιάταξης όπως η `mix.even`, η οποία επιλέγει τις υπολέξεις που βρίσκονται στις άρτιες θέσεις δύο καταχωρητών και τις μαζεύει σε ένα νέο καταχωρητή. Η εκτέλεση του βρόγχου του παραδείγματος με τη χρήση των εντολών αντιμετάθεσης και παράλληλης πρόσθεσης φαίνεται στο σχήμα 4.2. Η λειτουργία της εντολής `vec.add` που εμφανίζεται στο σχήμα 4.2 είναι ίδια με αυτή που δείξαμε στο σχήμα 4.1(α). Μ' αυτόν τον τρόπο καταφέραμε να μειώσουμε σημαντικά το χρόνο εκτέλεσης του βρόγχου αλλά και το πλήθος των εντολών που απαιτούνται για την υλοποίηση του.

Αν το προηγούμενο παράδειγμα αφορούσε διανυσματικούς επεξεργαστές τα πράγματα θα ήταν απλούστερα. Σε αυτήν την περίπτωση οι διανυσματικές πράξεις εκτελούνται επαναληπτικά στα στοιχεία του κάθε διανύσματος ενώ υποστηρίζονται διανυσματικές εντολές `vector_load` και `vector_store` οι οποίες μπορούν να φορτώσουν μη διαδοχικά δεδομένα από τη μνήμη (strided access) [4]. Στο παράδειγμα μας το διάνυσμα b μπορεί να φορτωθεί από τη μνήμη με `stride 2`. Για το λόγο αυτό, παρόλο που η μετατροπή ακολουθιακών προγραμμάτων σε διανυσματική μορφή είναι μια ώριμη περιοχή έρευνας, η υιοθέτηση των SIMD εντολών από τους σύγχρονους μικροεπεξεργαστές γεννά καινούρια προβλήματα τόσο στους προγραμματιστές όσο και στους συγγραφείς μεταγλωττιστών και απαιτεί νέες βελτιστοποιήσεις [12].

Τέλος, οφείλουμε να σημειώσουμε πως οι εντολές αναδιοργάνωσης χρησιμοποιούνται συχνά ώστε να επιτύχουν την απαραίτητη ευθυγράμμιση των δεδομένων κατά τη φόρτωση τους από τη μνήμη, κυρίως σε περιπτώσεις όπου η αρχιτεκτονική επιτρέπει την εκτέλεση `load` ή `store` σε διευθύνσεις μη ευθυγραμμισμένες με το μήκος της λέξης του επεξεργαστή [46]. Αναλυτικά παραδείγματα αντιμετώπισης της ευθυγράμμισης των διανυσμάτων με τη χρήση εντολών αναδιοργάνωσης δεδομένων μπορούν να βρεθούν στην εργασία [136].

Οι εντολές που έχουν προταθεί μέχρι σήμερα για την αποτελεσματική αναδιοργάνωση των δεδομένων μπορούν να χωριστούν σε δύο ομάδες ανάλογα με το αν υιοθετούν δύο βασικά χαρακτηριστικά:

- ▶ Κάποιες εντολές αντιμετάθεσης αναδιοργανώνουν τα δεδομένα ενός μόνο καταχωρητή, ενώ κάποιες άλλες συνδυάζουν τις υπολέξεις δύο καταχωρητών για να παράγουν μια νέα λέξη. Ανάλογα με το πρόγραμμα που θέλουμε να εκτελέσουμε χρησιμοποιούμε εντολές της μιας ή της άλλης κατηγορίας.
- ▶ Η αναδιάταξη που επιφέρει κάθε εντολή είτε ελέγχεται από επιπλέον δεδομένα ελέγχου, τα οποία είναι αποθηκευμένα σε έναν καταχωρητή, είτε είναι δεδομένη εξ αρχής και περιγράφεται μόνο από τον ορισμό της εντολής (opcode).

Ωστόσο, υπάρχουν και εντολές οι οποίες ανήκουν και στις δύο κατηγορίες. Οι εντολές αυτές προσπαθούν να αναδιατάξουν στοιχεία από δυο καταχωρητές και χρησιμοποιούν επιπλέον πληροφορία ελέγχου, δέχονται δηλαδή τρία έντελα όπως για παράδειγμα `perm` της Altivec [75]. Παρόλο που τέτοιες γενικές και πιο σύνθετες εντολές θα μπορούσαν να αντικαταστήσουν τη λειτουργία απλούστερων εντολών, οι σημερινοί επεξεργαστές υποστηρίζουν ταυτόχρονα μια πληθώρα από απλές και πιο σύνθετες εντολές. Οι λόγοι γι' αυτή την προσέγγιση είναι πολλοί. Στην απλούστερη περίπτωση θα θέλαμε μόνο εντολές οι οποίες δεν θα δέχονταν καμιά πληροφορία ελέγχου. Έτσι θα γλιτώναμε την επιπλέον επικοινωνία με τη μνήμη, μιας και δεν θα απαιτούσαν η φόρτωση των ψηφίων ελέγχου σε κάποιον

GRP definition

```

j=0;
for (i=0; i < N; i++) {
    if (Rc[i] == 1) {
        Rd[j]=Rs[i];
        j++;
    }
}
for (i=0; i < N; i++) {
    if (Rc[i] == 0) {
        Rd[j]=Rs[i];
        j++;
    }
}

```

Σχήμα 4.3: Ο ορισμός της GRP και ένα παράδειγμα της λειτουργίας της στα 8 δυαδικά ψηφία.

καταχωρητή. Επίσης, θα χρησιμοποιούσαμε λιγότερες πόρτες του αρχείου καταχωρητών κατά την εκτέλεση, ενώ είναι πιο πιθανό η λειτουργία που θα εκτελούσαμε να μπορούσε να υλοποιηθεί πιο εύκολα σε υλικό με μικρότερη καθυστέρηση για την εύρεση του πρώτου αποτελέσματος (latency). Η χρήση μόνο απλών εντολών αναδιοργάνωσης δεν οδηγεί αναγκαστικά σε μείωση της απόδοσης, γιατί σε πολλές εφαρμογές οι αναδιατάξεις δεδομένων που απαιτούνται είναι απλές και επαναλαμβανόμενες [136]. Πάρολα αυτά, ο εμπλουτισμός του συνόλου εντολών με μια πληθώρα εντολών αναδιοργάνωσης από διαφορετικές κατηγορίες επιτρέπει στους μεταγλωττιστές ή τους προγραμματιστές να βρουν κατά περίπτωση την αποδοτικότερη δυνατή λύση ενώ η αρχιτεκτονική διατηρεί το γενικό της χαρακτήρα και μπορεί εύκολα να προσαρμοστεί σε νέες εφαρμογές.

4.3 Η εντολή ομαδοποίησης GRP

Μία από τις εντολές αναδιοργάνωσης δεδομένων η οποία εμφανίζει μεγάλο ενδιαφέρον, είναι η εντολή ομαδοποίησης GRP (GRouP) που προτάθηκε από τον Shi και την Lee στην εργασία [142]. Η εντολή αυτή αναδιοργανώνει τα περιεχόμενα ενός μόνο καταχωρητή χρησιμοποιώντας ένα δυαδικό ψηφίο ελέγχου για κάθε ένα από τα δεδομένα εισόδου (είτε απλά δυαδικά ψηφία είτε υπολέξεις). Σύμφωνα με τον ορισμό της GRP R_D , R_S , R_C στον καταχωρητή R_S περιέχονται τα προς αναδιάταξη δεδομένα, στον καταχωρητή R_C αποθηκεύονται τα δυαδικά ψηφία ελέγχου ενώ στον καταχωρητή R_D αποθηκεύεται το αποτέλεσμα. Κατά την εκτέλεση της εντολής τα δεδομένα του καταχωρητή R_S χωρίζονται σε δύο ομάδες. Στη μία ομάδα τοποθετούνται τα δεδομένα που έχουν δυαδικό ψηφίο ελέγχου ίσο με ένα, ενώ στην άλλη τοποθετούνται τα δεδομένα με μηδενικά ψηφία ελέγχου. Παράλληλα, ο ορισμός της GRP εξασφαλίζει ότι τα δεδομένα μέσα στις δύο ανεξάρτητες ομάδες διατηρούν τη σχετική σημαντικότητά τους. Ένα παράδειγμα της εκτέλεσης της GRP φαίνεται στο σχήμα 4.3 μαζί με μια αλγοριθμική περιγραφή της λειτουργίας της.

Ο ορισμός της εντολής GRP είναι γενικός και μπορεί να προσαρμοστεί ανάλογα με την

Σχήμα 4.4: Η αντιστροφή 8 υπολέξεων με τη χρήση 3 εντολών GRP.

επιλεγμένη αρχιτεκτονική. Σε κάποιες περιπτώσεις, όπως στους αλγορίθμους κρυπτογραφίας, είναι χρήσιμο η GRP να αναδιατάσσει απλά δυαδικά ψηφία, ενώ στις πολυμεσικές εφαρμογές απαιτείται η αναδιάταξη μεγαλύτερων υπολέξεων του ενός ή δύο bytes. Στην πρώτη περίπτωση, αν θεωρήσουμε πως η λέξη αποτελείται από n δυαδικά ψηφία τότε απαιτούνται n δυαδικά ψηφία ελέγχου για να διαχωρίσουν τα δεδομένα της εισόδου σε δύο ομάδες. Αντίστοιχα, στην περίπτωση υπολέξεων των b δυαδικών ψηφίων αρκούν n/b δυαδικά ψηφία ελέγχου. Στη χειρότερη περίπτωση απαιτούνται $\log_2(n/b)$ εντολές GRP για να συνθέσουν μια οποιαδήποτε αντιμετάθεση με $b \geq 1$. Για παράδειγμα η αντιστροφή μιας λέξης των 8 δυαδικών ψηφίων χρειάζεται τρεις GRP εντολές. Η εκτέλεση της αντιστροφής αυτής φαίνεται στο σχήμα 4.4.

Οι Shi και Lee έδειξαν στις εργασίες [143], [100], [144] πως η χρήση της εντολής GRP είναι γενική και πως μπορεί να χρησιμοποιηθεί σε μια πληθώρα εφαρμογών επιταχύνοντας αποτελεσματικά την εκτέλεση τους. Πιο συγκεκριμένα, η υλοποίηση αλγορίθμων κρυπτογραφίας με τη χρήση της εντολής GRP αυξάνει την ανθεκτικότητα του αλγορίθμου σε επιθέσεις ενώ ταυτόχρονα πετυχαίνει σημαντικές επιταχύνσεις σε σχέση με υλοποιήσεις οι οποίες εκτελούν τις αναδιοργανώσεις των δεδομένων με απλές εντολές απλής ή κυκλικής ολίσθησης (shift ή rotate). Επίσης, σημαντικές είναι και οι βελτιώσεις που μπορεί να προσφέρει η GRP στον υπολογισμό των συντελεστών του DCT, αλλά και στην ταξινόμηση των στοιχείων ενός πίνακα.

Αν και η εντολή GRP προσφέρει πολλές επιλογές στους προγραμματιστές για την υλοποίηση διαφόρων εφαρμογών, το κύριο πρόβλημα που εμφανίζει έως τώρα είναι η αποδοτική υλοποίηση της σε υλικό. Οι υλοποιήσεις που προτάθηκαν μέχρι στιγμής οδηγούν σε κυκλώματα τα οποία έχουν σημαντική καθυστέρηση και δεν επιτρέπουν την ολοκλήρωση της εκτέλεσης της GRP σε ένα ή το πολύ δύο κύκλους του επεξεργαστή¹. Η πιο αποδοτική έως τώρα υλοποίηση απαιτεί τη διάσπαση του κυκλώματος σε τουλάχιστον τρία pipeline στάδια αυξάνοντας έτσι σημαντικά το χρόνο παραγωγής του πρώτου αποτελέσματος ενώ ταυτόχρονα επιβραδύνει την

¹Στους σύγχρονους μικροεπεξεργαστές ο κύκλος ρολογιού είναι της τάξης των 11-18 FO4.

Σχήμα 4.5: Η υλοποίηση της GRP χρησιμοποιώντας δύο μονάδες εξαγωγής.

εκτέλεση των πραγματικών αριθμητικών υπολογισμών οι οποίοι περιμένουν την κατάλληλη αναδιοργάνωση των δεδομένων για να ξεκινήσουν. Η προσπάθεια μας λοιπόν εστιάστηκε στην αποδοτική υλοποίηση της εντολής GRP και στη διερεύνηση των χαρακτηριστικών των διαφόρων νέων κυκλωμάτων που μπορούν να χρησιμοποιηθούν για την υλοποίησή της.

Στη συνέχεια του κεφαλαίου θα παρουσιάσουμε συνοπτικά τα κυκλώματα που έχουν μέχρι τώρα προταθεί για την υλοποίηση της εντολής και στη συνέχεια θα περιγράψουμε αναλυτικά τα κυκλώματα που προτείνουμε εμείς. Τα νέα κυκλώματα ακολουθούν μια εντελώς διαφορετική προσέγγιση για την υλοποίηση της εντολής GRP και βασίζονται σε μια αποδοτική παραλλαγή των δικτύων ταξινόμησης (sorting networks). Τα πειραματικά αποτελέσματα που συλλέξαμε οδηγούν στο συμπέρασμα ότι οι προτεινόμενες υλοποιήσεις της GRP επιτυγχάνουν μείωση της καθυστέρησης έως και 25% σε σχέση με τις προηγούμενες λύσεις καταναλώνοντας περίπου την ίδια ισχύ. Επίσης, στις πιο πρακτικές περιπτώσεις, όπου η αναδιοργάνωση εκτελείται σε υπολέξεις των 8 ή των 16 δυαδικών ψηφίων, η γραμμική μορφή των προτεινόμενων δικτύων προσφέρει αποτελεσματικές λύσεις όπου η καθυστέρηση και η κατανάλωση ισχύος των νέων κυκλωμάτων είναι σχεδόν όμοια με μιας αριθμητικής/λογικής μονάδας του ίδιου εύρους δυαδικών ψηφίων.

4.4 Προηγούμενες υλοποιήσεις

Μέχρι τώρα δύο βασικές υλοποιήσεις έχουν προταθεί για την υλοποίηση της εντολής GRP [145], [62]. Και στις δύο περιπτώσεις τα δυαδικά ψηφία των δεδομένων που έχουν διαφορετικά σήματα ελέγχου εξάγονται από την αρχική λέξη χωριστά και τοποθετούνται στην έξοδο στις σωστές θέσεις χρησιμοποιώντας δύο ανεξάρτητες μονάδες εξαγωγής. Η γενική δομή της μονάδας που υλοποιεί την εντολή GRP και για τις δύο προηγούμενες υλοποιήσεις φαίνεται στο σχήμα 4.5. Η αριστερή μονάδα εξαγωγής (extraction unit) αναλαμβάνει να μαζέψει τα δεδομένα με δυαδικά ψηφία ελέγχου ίσα με ένα στην αριστερή πλευρά της εξόδου. Αντίστοιχα, η δεξιά μονάδα εξαγωγής, η οποία θεωρεί ανεστραμμένα δυαδικά ψηφία ελέγχου, μαζεύει τα δεδομένα στη

δεξιά πλευρά της εξόδου. Τα δύο επιμέρους αποτελέσματα μπορούν να ενωθούν μέσω της λογικής πράξης OR. Για να επιτρέψουμε την ένωση μέσω της πράξης OR των δύο μερικών αποτελεσμάτων θα πρέπει, πριν δώσουμε τα αρχικά δεδομένα στις δύο μονάδες εξαγωγής, να εφαρμόσουμε σε αυτά μία μάσκα. Για την αριστερή μονάδα εξαγωγής η μάσκα είναι ίση με τα δυαδικά ψηφία ελέγχου, ενώ για την δεξιά η μάσκα αποτελείται από τα ανεστραμμένα ψηφία ελέγχου. Μετά την εφαρμογή της μάσκας τα δυαδικά ψηφία που ελέγχονται από μηδενικά ψηφία ελέγχου παίρνουν κι αυτά τιμή ίση με μηδέν. Οι δύο μονάδες εξαγωγής έχουν σχεδόν ίδια δομή. Η διαφορετική πλευρά μαζέματος των δυαδικών ψηφίων των δεδομένων επιφέρει μικρές αλλαγές στη σχεδίαση τους. Έτσι, στη συνέχεια, θα επικεντρώσουμε την περιγραφή μας στην σχεδίαση της αριστερής μονάδας εξαγωγής.

Η πρώτη υλοποίηση που προτάθηκε [145] ακολουθεί μια ιεραρχική στρατηγική μέτρησης και ολίσθησης. Αρχικά εξετάζονται ομάδες των δύο δυαδικών ψηφίων και τα δεδομένα που έχουν ψηφία ελέγχου ίσα με ένα μεταφέρονται στην αριστερή πλευρά της εξόδου. Μ' αυτόν τον τρόπο καταφέρνουμε να δημιουργήσουμε σωστά ευθυγραμμισμένες ομάδες των δύο δυαδικών ψηφίων. Στη συνέχεια, οι ομάδες αυτές πρέπει να ενωθούν ώστε να παράγουμε σωστά ευθυγραμμισμένες ομάδες των τεσσάρων δυαδικών ψηφίων. Για τον σχηματισμό των νέων ευθυγραμμισμένων ομάδων διπλάσιου μεγέθους τα δεδομένα της δεξιάς μικρότερης ομάδας, που έχουν ψηφία ελέγχου ίσα με ένα, θα μεταφερθούν αριστερά και θα τοποθετηθούν δίπλα στα δεδομένα της αριστερής ομάδας με δυαδικά ψηφία ελέγχου ίσα με ένα. Στην ουσία, τα πιο σημαντικά ψηφία της δεξιάς ομάδας αντικαθιστούν τα μηδενικά δεδομένα της αριστερής, τα οποία ελέγχονται από μηδενικά ψηφία ελέγχου και μηδενίστηκαν μετά την εφαρμογή της μάσκας. Για να δημιουργήσουμε λοιπόν την νέα ευθυγραμμισμένη ομάδα των τεσσάρων δυαδικών ψηφίων πρέπει να εκτελέσουμε δύο λειτουργίες. Από τη μια πρέπει να μετρήσουμε τα μηδενικά της αριστερής ομάδας και στη συνέχεια να ολισθήσουμε προς τ' αριστερά τη δεξιά ομάδα τόσες θέσεις όσες και το πλήθος των μηδενικών που υπολογίσαμε. Επίσης, για να γλιτώσουμε την όποια επιπλέον καθυστέρηση πρέπει ταυτόχρονα με την ολίσθηση να υπολογίσουμε το πλήθος των μηδενικών ψηφίων ελέγχου που θα έχει στο δεξί μέρος της η νέα λέξη των 4 δυαδικών ψηφίων που θα δημιουργήσουμε. Η τιμή αυτή θα χρησιμοποιηθεί για την ολίσθηση του επόμενου σταδίου, επικαλύπτοντας έτσι χρονικά τη μέτρηση με την ολίσθηση. Για να πραγματοποιηθεί αυτή η λειτουργία δεν έχουμε παρά να προσθέσουμε τα μηδενικά των ομάδων των 2 δυαδικών ψηφίων που θέλουμε να συνενώσουμε. Αυτή η λειτουργία της μέτρησης των μηδενικών και της ολίσθησης συνεχίζεται επαναληπτικά μέχρι να παράγουμε μια λέξη των n δυαδικών ψηφίων συνενώνοντας στο τελευταίο στάδιο δύο ενδιάμεσες λέξεις των $n/2$ δυαδικών ψηφίων.

Η δεύτερη υλοποίηση GRP που προτάθηκε στην εργασία [62] ακολουθεί μια γενική αρχιτεκτονική η οποία είναι γνωστή εδώ και χρόνια [92] και χρησιμοποιείται για την υλοποίηση της λειτουργίας του πακεταρίσματος (pack) το οποίο μοιάζει να είναι σχεδόν ταυτόσημο με τη λειτουργία της εντολής GRP. Αρχικά θα περιγράψουμε τη βασική δομή της αρχιτεκτονικής αυτής χρησιμοποιώντας ένα παράδειγμα όμοιο με την περίπτωση της εντολής GRP και στη συνέχεια, θα αναφέρουμε τις παραλλαγές που προτάθηκαν για την υλοποίηση της στην εργασία [62]. Η εκτέλεση του πακεταρίσματος παραδοσιακά πραγματοποιούνταν σε δύο φάσεις [92]. Αρχικά πρέπει να καθοριστεί για τα δεδομένα σε ποια έξοδο πρέπει να κατευθυνθούν. Για κάθε δεδομένο της εισόδου παράγεται ένας αριθμός ο οποίος χρησιμοποιείται ως δείκτης της κατάλληλης εξόδου. Για παράδειγμα, για την εκτέλεση της

Σχήμα 4.6: Παράδειγμα εκτέλεσης της GRP με λειτουργία όμοια με αυτή του πακεταρίσματος.

εντολής που φαίνεται στο σχήμα 4.6 πρέπει να εξάγουμε ότι το δεδομένο E που έχει δυαδικό ψηφίο ελέγχου ίσο με ένα, πρέπει να κατευθυνθεί στην έξοδο 2, ενώ το δεδομένο B πρέπει να κατευθυνθεί στην έξοδο 5. Στη συνέχεια, έχοντας για κάθε δεδομένο εισόδου την έξοδο στην οποία πρέπει να μεταφερθεί, μπορούμε να εκτελέσουμε τη δρομολόγηση με τη χρήση οποιουδήποτε κυκλώματος μεταγωγής. Στη γενική περίπτωση θεωρούμε ότι έχουμε στη διάθεση μας ένα πλήρως συνδεδεμένο κύκλωμα μεταγωγής (crossbar) που επιτρέπει την ταυτόχρονη μετάδοση των δεδομένων εισόδου προς οποιαδήποτε έξοδο επιθυμούν. Για την εκτέλεση της εντολής GRP τα πράγματα είναι απλούστερα για δύο λόγους. Πρώτον, δύο διαφορετικά δεδομένα εισόδου δεν θα ζητήσουν ποτέ να κατευθυνθούν προς την ίδια έξοδο και δεύτερον, από τον ορισμό της εντολής εξασφαλίζεται ότι σε καμία περίπτωση ένα δεδομένο εισόδου δεν θα μεταφερθεί ταυτόχρονα σε δύο διαφορετικές εξόδους.

Στη συνέχεια θα περιγράψουμε πως από τα αρχικά δυαδικά ψηφία ελέγχου προκύπτει η έξοδος στην οποία πρέπει να μεταφερθεί κάθε δεδομένο της εισόδου. Για κάθε είσοδο i υπολογίζουμε δύο τιμές (4.1) αντιμετωπίζοντας τα δυαδικά ψηφία ελέγχου $C_i \in \{0, 1\}$ σαν να είναι αριθμοί με τιμές 0 και 1 αντίστοιχα.

$$\text{Prefix}(i) = \sum_{k=1}^i C_k \quad \text{και} \quad \text{Suffix}(i) = N + 1 - \sum_{k=N}^i (1 - C_k) \quad (4.1)$$

Για το παράδειγμα του σχήματος 4.6 οι τιμές των $\text{Prefix}(i)$ και $\text{Suffix}(i)$ που υπολογίσαμε φαίνονται στις γραμμές Prefix Scan και Suffix Scan αντίστοιχα. Επεξεργαζόμενοι τα δυαδικά ψηφία ελέγχου ξεκινώντας από την αριστερή πλευρά πετυχαίνουμε να διατηρήσουμε στην έξοδο τη σχετική σημαντικότητα των δεδομένων εισόδου. Για να γίνει αυτή η διαδικασία πιο εύκολα κατανοητή δεν έχουμε παρά να παρατηρήσουμε στο σχήμα 4.6 τα δεδομένα που έχουν δυαδικά

ψηφία ελέγχου ίσα με 1. Το δεδομένο που έχει ψηφίο ελέγχου ίσο με 1 και είναι τοποθετημένο στην πιο αριστερή θέση (A) θα έχει πάντα τιμή Prefix ίση με 1, μιας και τα πιο μπροστά από αυτό δεδομένα με μηδενικά ψηφία ελέγχου, δεν μπορούν να αυξήσουν την τιμή του Prefix. Έτσι εξασφαλίζεται, πως αυτό το δεδομένο θα ζητήσει σωστά να μεταφερθεί στην πιο αριστερή θέση της εξόδου. Αντίστοιχα, για το επόμενο από αριστερά δεδομένο εισόδου με δυαδικό ψηφίο ελέγχου ίσο με 1 (E) η τιμή του Prefix, που υπολογίζεται γι' αυτό, περιγράφει το πλήθος των άσων που υπάρχουν μπροστά από αυτό και ισούται με 2. Μ' αυτόν τον τρόπο καθορίζεται ότι το E πρέπει να μεταφερθεί στην επόμενη πιο αριστερή θέση απ' το A διατηρώντας έτσι τη σχετική τους σημαντικότητα. Η ίδια πληροφορία παράγεται για τα δεδομένα που έχουν μηδενικά δυαδικά ψηφία ελέγχου μέσω της συνάρτησης Suffix ξεκινώντας τη μέτρηση από δεξιά.

Η τελική επιλογή της κατάλληλης εξόδου στην οποία ζητά να κατευθυνθεί κάθε ένα από τα δεδομένα της εισόδου πραγματοποιείται με μια απλή επιλογή. Αν το δυαδικό ψηφίο ελέγχου της i -οστής θέσης είναι 1 ($C_i = 1$) τότε η έξοδος στην οποία πρέπει να κατευθυνθεί το δεδομένο της ίδιας θέσης ισούται με την τιμή του $\text{Prefix}(i)$. Στην αντίθετη περίπτωση, όπου $C_i = 0$, η έξοδος δίνεται από την τιμή του $\text{Suffix}(i)$. Η εφαρμογή της επιλογής στις τιμές των Prefix και Suffix φαίνεται στο σχήμα 4.6 όπου ο τελικός δείκτης της εξόδου για κάθε θέση φαίνεται στη γραμμή Output Request. Έτσι εύκολα μπορούμε να παρατηρήσουμε πως αν μεταφέρουμε τα δεδομένα εισόδου στην αντίστοιχη έξοδο που δηλώνεται στη γραμμή Output Request επιτυγχάνουμε τη λειτουργία της GRP. Σ' αυτό το σημείο οφείλουμε να σημειώσουμε πως η μέθοδος αυτή διαχωρίζει τον υπολογισμό της θέσης στην οποία πρέπει να δρομολογηθούν τα δεδομένα από την ίδια τη δρομολόγηση τους. Επίσης, ο υπολογισμός των Prefix και Suffix για κάθε είσοδο μπορεί να επιταχυνθεί μέσω δένδρων παράλληλου προθέματος, μιας και οι σχέσεις που περιγράφουν τις τιμές τους δεν είναι τίποτε άλλο παρά απλές γραμμικές αναδρομές.

Η βασική αυτή αρχιτεκτονική, αν και ήδη γνωστή, προτάθηκε με κάποιες παραλλαγές από τους Hilewitz, Shi και Lee στην εργασία [62] για την υλοποίηση της εντολής GRP και της αντίθετης της (PDEP, Parallel Deposit) [63]. Οι κύριες διαφοροποιήσεις είναι δύο. Πρώτον, για τον υπολογισμό της εξόδου στην οποία θα δρομολογηθεί κάθε δεδομένο εισόδου χρησιμοποιήθηκε carry-save αριθμητική [129] με σκοπό τη μείωση της καθυστέρησης. Δεύτερον, ως δίκτυο δρομολόγησης χρησιμοποιήθηκαν δύο δίκτυα πεταλούδας (butterfly). Το ένα αναλαμβάνει τη μεταγωγή των δεδομένων που ελέγχονται από ψηφία ελέγχου ίσα με ένα και το άλλο τη μεταγωγή των δεδομένων με μηδενικά ψηφία ελέγχου. Η λειτουργία των πολυπλεκτών του δικτύου πεταλούδας καθορίζεται μέσω ενός χωριστού κυκλώματος κωδικοποίησης. Το κύκλωμα κωδικοποίησης δέχεται σαν είσοδο την carry-save αναπαράσταση της εξόδου στην οποία πρέπει να κατευθυνθεί κάθε δεδομένο της εισόδου και αναλαμβάνει τη δημιουργία των $n/2 \log_2 n$ σημάτων επιλογής των πολυπλεκτών του δικτύου πεταλούδας ώστε να επιτευχθεί η μεταγωγή. Η δρομολόγηση όλων των δεδομένων με ψηφία ελέγχου ίσα με ένα (ή μηδέν) στην αριστερή (ή στην δεξιά) πλευρά της εξόδου είναι εφικτή, γιατί οι εξόδοι, στις οποίες ζητούν να τοποθετηθούν τα αντίστοιχα δεδομένα, όπως προέκυψε από την πρώτη φάση της κωδικοποίησης (στήλη OutputRequest), είναι πάντα ταξινομημένες, διατηρώντας έτσι τη σημαντικότητα μεταξύ των δεδομένων με ίσα ψηφία ελέγχου. Σε αντίθετη περίπτωση, η δρομολόγηση μπορεί να οδηγήσει σε συγκρούσεις μην επιτρέποντας την εκτέλεση της GRP. Αν και αυτή η ήδη γνωστή ιδιότητα των δικτύων πεταλούδας χρησιμοποιήθηκε έμμεσα στις εργασίες [62], [63] δεν δηλώθηκε ποτέ ρητά.

Σχήμα 4.7: Δίκτυα ταξινόμησης των 8 δυαδικών ψηφίων ακολουθώντας την αρχή (α) της ταξινόμησης bitonic ακολουθιών (β) της σύνδεσης ακολουθιών και (γ) της εναλλαγής μεταξύ των άρτιων και των περιπών θέσεων. (δ) Η λειτουργία του στοιχείου σύγκρισης.

4.5 Βασική ιδέα

Σύμφωνα με τον ορισμό της GRP τα δεδομένα τα οποία συνδέονται με μηδενικά δυαδικά ψηφία ελέγχου τοποθετούνται στη δεξιά πλευρά της εισόδου ενώ αυτά με ψηφία ελέγχου ίσα με ένα πηγαίνουν στην αριστερή πλευρά. Η λειτουργία αυτή προσομοιάζει τη διαδικασία της ταξινόμησης για τα ψηφία ελέγχου, όπου τα μεγαλύτερα δυαδικά ψηφία (αυτά που είναι ίσα με 1) τοποθετούνται στην αριστερή πλευρά της εξόδου μπροστά από τα μικρότερα δυαδικά ψηφία ελέγχου (αυτά που είναι ίσα με 0). Η ιδιότητα αυτή μας επιτρέπει να χρησιμοποιήσουμε ήδη γνωστά δίκτυα ταξινόμησης, ώστε να υλοποιήσουμε τη λειτουργία της εντολής GRP. Αυτό που μένει, είναι να επαυξήσουμε τη λειτουργία τους ώστε ταυτόχρονα με την ταξινόμηση των δυαδικών ψηφίων ελέγχου να επιτρέπουν και την αναδιοργάνωση των δεδομένων. Σ' αυτήν την περίπτωση, το κύριο πρόβλημα, που περιπλέκει το σχεδιασμό και καλούμαστε να λύσουμε, είναι η διατήρηση της σημαντικότητας των δεδομένων που έχουν ίσα δυαδικά ψηφία ελέγχου.

Ο πιο αποδοτικός τρόπος για την ταξινόμηση δεδομένων σε υλικό προκύπτει με τη χρήση των δικτύων ταξινόμησης [30]. Για την υλοποίηση λοιπόν της βασικής ιδέας που μόλις περιγράψαμε για την εκτέλεση της εντολής GRP μελετήσαμε τα πιο γνωστά δίκτυα ταξινόμησης, το δίκτυο ταξινόμησης bitonic ακολουθιών (Bitonic Sorter), το δίκτυο ταξινόμησης μέσω της σύνδεσης ακολουθιών (Merge Sort) και το γραμμικό δίκτυο εναλλαγής (Odd-Even Transposition Network). Όποιες άλλες μορφές δικτύων ταξινόμησης έχουν παρουσιαστεί στη διεθνή βιβλιογραφία, είναι απλά παραλλαγές των τριών αυτών βασικών δομών. Τα δίκτυα που

μελετήσαμε απεικονίζονται στα σχήματα 4.7(α)-(γ). Κάθε δίκτυο αποτελείται από το ίδιο δομικό στοιχείο σύγκρισης του οποίου η λειτουργία φαίνεται στο σχήμα 4.7(δ). Το μεγαλύτερο μεταξύ των δύο δυαδικών ψηφίων που συγκρίνονται τοποθετείται στην αριστερή πλευρά της εξόδου. Η λειτουργία αυτή αναπαρίσταται με την προς τα αριστερά κατεύθυνση του βέλους.

Όλα τα δίκτυα του σχήματος 4.7 ταξινομούν την ίδια δυαδική λέξη. Η λέξη αυτή αναπαριστά τα δυαδικά ψηφία ελέγχου της εντολής GRP. Αυτό που θέλουμε να εξετάσουμε είναι, αν στην έξοδο, όπου τα δυαδικά ψηφία έχουν ταξινομηθεί, διατηρείται η σημαντικότητα μεταξύ των άσων και των μηδενικών. Το δίκτυο που καταφέρει λόγω της δομής του να διατηρήσει τη σημαντικότητα ίσων δυαδικών ψηφίων μπορεί να υλοποιήσει απευθείας τη GRP. Σ' αυτήν την περίπτωση, τα δεδομένα μπορούν απλά να ακολουθήσουν μέσα στο δίκτυο τα δυαδικά ψηφία ελέγχου και θα τοποθετηθούν στη σωστή θέση στην έξοδο. Σε αντίθετη περίπτωση, δηλαδή όταν το δίκτυο ταξινόμησης δεν διατηρεί τη σημαντικότητα των ψηφίων ελέγχου, πρέπει να προσαρμόσουμε κατάλληλα τη λειτουργία του, ώστε να επιτρέπει την εκτέλεση της GRP.

4.5.1 Έλεγχος διατήρησης σημαντικότητας

Αρχικά θα αναλύσουμε την περίπτωση του bitonic δικτύου ταξινόμησης, το οποίο παρουσιάζεται στο σχήμα 4.7(α). Το δίκτυο αυτό αποτελείται από μικρότερα υποδίκτυα που μπορούν να ταξινομήσουν μόνο bitonic ακολουθίες [9]. Μια ακολουθία καλείται bitonic όταν είναι της μορφής $11 \dots 100 \dots 01 \dots 11$ ή $00 \dots 01 \dots 110 \dots 00$. Κάθε τέτοιο υποδίκτυο, αν και δεν είναι προφανές από το σχήμα 4.7(α), είναι στην ουσία ένα δίκτυο πεταλούδας και αποτελείται από $\log_2 n$ επίπεδα στοιχείων σύγκρισης. Για να κατασκευάσουμε ένα γενικό bitonic δίκτυο ταξινόμησης πρέπει να τοποθετήσουμε στη σειρά $\log_2 n$ υποδίκτυα όπου το καθένα έχει διπλάσιο εύρος δυαδικών ψηφίων. Έτσι, ο συνολικός αριθμός σταδίων σύγκρισης που απαιτεί ένα bitonic δίκτυο ταξινόμησης είναι ίσος με $\frac{1}{2}(\log_2 n)(1 + \log_2 n)$. Τα bitonic δίκτυα ταξινόμησης δεν διατηρούν τη σχετική σημαντικότητα δύο ίσων δυαδικών ψηφίων. Αυτό φαίνεται αν παρατηρήσουμε τη διαδρομή που ακολουθούν οι δύο σημειωμένοι άσσοι στο σχήμα 4.7(α). Παρόλο που έχουν ταξινομηθεί σωστά στην έξοδο (κανένα μηδενικό δεν υπάρχει σε πιο σημαντική θέση), η διαδρομή που ακολούθησαν μέσα στο δίκτυο άλλαξε τη σχετική σημαντικότητα που είχαν αρχικά. Έτσι, αν τα αντίστοιχα δεδομένα της εισόδου με τα οποία ήταν συνδεδεμένα ακολουθούσαν την ίδια διαδρομή θα βρίσκονταν σε λάθος θέση στην έξοδο σύμφωνα με τον ορισμό της GRP. Η ανεπιθύμητη αυτή ιδιότητα των bitonic δικτύων ταξινόμησης οφείλεται στη δομή πεταλούδας των υποδικτύων. Εκεί τα δυαδικά ψηφία εξαιτίας της σύγκρισης τοποθετούνται διαδοχικά σε δύο ανεξάρτητα μέρη. Συνεπώς, όταν ένα δυαδικό ψηφίο τοποθετείται στη μια πλευρά του δικτύου, δεν μπορεί να επανακτήσει τη σχετική του σημαντικότητα σε σχέση με τα υπόλοιπα ίσα δυαδικά ψηφία τα οποία παρέμειναν στην άλλη πλευρά.

Το merge δίκτυο ταξινόμησης, που φαίνεται στο σχήμα 4.7(β), ακολουθεί μια διαφορετική στρατηγική. Και αυτό το δίκτυο αποτελείται από μικρότερα υποδίκτυα τα οποία προσπαθούν να συνενώσουν δύο ήδη ταξινομημένες ακολουθίες της μορφής $11 \dots 10 \dots 00$ και να παράγουν μια νέα ταξινομημένη ακολουθία διπλάσιου μεγέθους. Το merge δίκτυο ταξινόμησης απαιτεί τον ίδιο αριθμό επιπέδων σύγκρισης με το bitonic δίκτυο ταξινόμησης, αλλά χρησιμοποιεί λιγότερα στοιχεία σύγκρισης και έχει κάπως διαφορετική δομή. Μια πολύ σαφής περιγραφή των ιδιοτήτων των δύο δικτύων και των διαφορών τους περιγράφεται στην εργασία [68]. Επίσης

και στη περίπτωση του merge δικτύου ταξινόμησης, όπως φαίνεται στο σχήμα 4.7(β), η σχετική σημαντικότητα μεταξύ ίσων δυαδικών ψηφίων δεν διατηρείται.

Η τελευταία περίπτωση αφορά το γραμμικό δίκτυο αντιμετάθεσης (σχήμα 4.7(γ)). Σ' αυτήν την περίπτωση σε κάθε επίπεδο του δικτύου οι άσσοι μετακινούνται προς τ' αριστερά και τα μηδενικά προς τα δεξιά. Οι εισοδοί κάθε στοιχείου σύγκρισης εναλλάσσονται όταν το δυαδικό ψηφίο που βρίσκεται στην πιο σημαντική είσοδο είναι ίσο με 0 και αυτό που βρίσκεται στη λιγότερο σημαντική είσοδο είναι ίσο με 1. Εξαιτίας της δομής του δικτύου ένα δυαδικό ψηφίο δεν μπορεί να προσπεράσει ένα περισσότερο σημαντικό ψηφίο με την ίδια τιμή. Έτσι όταν τα δυαδικά ψηφία ταξινομούνται με το γραμμικό δίκτυο αντιμετάθεσης η σχετική σημαντικότητα μεταξύ ίσων δυαδικών ψηφίων διατηρείται. Για το λόγο αυτό, το γραμμικό δίκτυο μπορεί απευθείας να υλοποιήσει τη λειτουργία της εντολής GRP. Όμως, σε αυτήν την περίπτωση το πλήθος των επιπέδων σύγκρισης που απαιτείται είναι ίσο με το πλήθος των δυαδικών ψηφίων που θέλουμε να ταξινομήσουμε και είναι απαγορευτικό για τις λέξεις των σημερινών μικροεπεξεργαστών των 32 ή των 64ων δυαδικών ψηφίων. Η χρησιμότητα του δικτύου αυτού θα αναλυθεί στην ενότητα 4.9.

4.5.2 Η εκτέλεση της GRP μέσω επαυξημένων δικτύων ταξινόμησης

Όπως φάνηκε το bitonic και το merge δίκτυο ταξινόμησης δεν μπορούν να χρησιμοποιηθούν απευθείας για την εκτέλεση της εντολής GRP. Για το λόγο αυτό, η εκτέλεση της GRP μέσω αυτών των δύο δικτύων επιβάλλει την υιοθέτηση της αρχιτεκτονικής του σχήματος 4.5 και τη χρήση δύο χωριστών μονάδων εξαγωγής. Για την υλοποίηση της κάθε μονάδας εξαγωγής θα παρουσιάσουμε στη συνέχεια δύο νέες δομές που ονομάζονται *Enhanced Bitonic Sorting Network*, (EBSN) και *Enhanced Merge Sorting Network*, (EMSN) οι οποίες επιτρέπουν την ομαδοποίηση των δεδομένων εισόδου με ψηφία ελέγχου 1 στην αριστερή πλευρά της εξόδου ή στη δεξιά αντίστοιχα. Το EBSN ή αντίστοιχα το EMSN έχει δύο εισόδους των n δυαδικών ψηφίων. Τα δεδομένα εισόδου που θέλουμε να ομαδοποιηθούν στη αριστερή πλευρά της εξόδου και τα αντίστοιχα δυαδικά ψηφία ελέγχου που περιγράφουν ποια δεδομένα πρέπει να επιλεγούν. Σύμφωνα με την αρχιτεκτονική του σχήματος 4.5, τα δυαδικά ψηφία δεδομένων που ελέγχονται από μηδενικά ψηφία ελέγχου μηδενίζονται πριν εισέλθουν στη μονάδα εξαγωγής. Επειδή τα μηδενισμένα δεδομένα δε μας ενδιαφέρουν, στη συνέχεια θα τα αναπαριστούμε ως αδιάφορες τιμές x . Η δομή και η λειτουργία του EBSN και του EMSN περιγράφονται στις ενότητες 4.6 και 4.7 αντίστοιχα.

Αντίθετα, η εκτέλεση της GRP στο γραμμικό δίκτυο αντιμετάθεσης μπορεί να επιτευχθεί χωρίς μεγάλη σχεδιαστική προσπάθεια. Σε αυτήν την περίπτωση δεν απαιτούνται δύο μονάδες εξαγωγής, αλλά ένα μόνο δίκτυο αρκεί για την ολοκλήρωση της λειτουργίας της εντολής. Η βασική και η βελτιστοποιημένη ως προς την καθυστέρηση εκδοχή του επαυξημένου γραμμικού κυκλώματος, το οποίο καλείται *Enhanced Transposition sorting Network*, (ETN), παρουσιάζονται στην ενότητα 4.8.

4.6 Επαυξημένο bitonic δίκτυο ταξινόμησης

Ακολουθώντας τη ίδια δομή με το bitonic δίκτυο ταξινόμησης το EBSN αποτελείται από κατάλληλα συνδεδεμένα υποδίκτυα τα οποία ονομάζονται επαυξημένοι ταξινομητές bitonic

Σχήμα 4.8: Η δομή ενός επαυξημένου bitonic δικτύου ταξινόμησης και η τοπολογία των αντίστοιχων EBS υπο-δικτύων μεγέθους 2, 4 και 8 δυαδικών ψηφίων.

ακολουθιών *Enhanced Bitonic Sorters* (EBS). Ένα παράδειγμα ενός τέτοιου δικτύου φαίνεται στο σχήμα 4.8. Κάθε EBS δέχεται δύο λέξεις στην είσοδο του. Τα δυαδικά ψηφία ελέγχου, τα οποία πρέπει να σχηματίζουν μια bitonic ακολουθία, και τα αντίστοιχα δυαδικά ψηφία των δεδομένων. Ο σκοπός του EBS είναι να ταξινομήσει τα δυαδικά ψηφία ελέγχου και να μεταφέρει αντίστοιχα στη σωστή τους θέση τα δυαδικά ψηφία των δεδομένων. Κάθε EBS του σχήματος 4.8 χαρακτηρίζεται από έναν αριθμό ο οποίος περιγράφει το μέγεθος του, δηλαδή το εύρος των λέξεων της εισόδου του, και από ένα βέλος, η κατεύθυνση του οποίου δείχνει την μεριά προς την οποία κατευθύνονται τα δεδομένα με δυαδικά ψηφία ελέγχου ίσα με 1. Κάθε δύο γειτονικά EBS του ίδιου μεγέθους έχουν αντίθετες κατευθύνσεις συγκέντρωσης των ταξινομημένων δεδομένων. Αυτό απαιτείται, ώστε η νέα διπλάσιου μεγέθους ακολουθία που θα προκύψει από τη συνένωση των μικρότερων ταξινομημένων ακολουθιών των δυαδικών ψηφίων ελέγχου και που θα χρησιμοποιηθεί σαν είσοδος στο επόμενο επίπεδο, να είναι σε μορφή bitonic. Στη συνέχεια, θα περιγράψουμε αναλυτικά τη λειτουργία του EBS και πως καταφέρνει να μετακινήσει τα δεδομένα στις σωστές τους θέσεις χωρίς να χαλάει τη σχετική σημαντικότητα των δεδομένων με δυαδικά ψηφία ελέγχου ίσα με ένα.

Σχήμα 4.9: Η απλοποίηση των δυαδικών ψηφίων ελέγχου.

4.6.1 Η σχεδίαση του υποδικτύου EBS

Το EBS έχει την ίδια δομή με ένα δίκτυο πεταλούδας σε κάθε κόμβο του οποίου υπάρχει ένα από τα δύο βασικά δομικά στοιχεία του κυκλώματος, δηλαδή τα κελιά σύγκρισης και εναλλαγής HB και LB. Ένα EBS δίκτυο των 8 δυαδικών ψηφίων φαίνεται στο σχήμα 4.8. Το EBS καλείται ταυτόχρονα να επιτελέσει δυο λειτουργίες. Αφ' ενός πρέπει να ταξινομήσει την bitonic ακολουθία των δυαδικών ψηφίων ελέγχου και αφ' ετέρου να μεταφέρει τα αντίστοιχα δυαδικά ψηφία δεδομένων στη σωστή τους θέση στην έξοδο. Όταν δύο ψηφία ελέγχου συγκρίνονται τότε το μεγαλύτερο δίνεται από τη λογική συνάρτηση AND των δύο και παράγεται στην έξοδο του HB. Όμοια, το ελάχιστο των δύο δίνεται από τη λογική συνάρτηση OR και παράγεται στην έξοδο του LB.

Όσον αφορά στη μετακίνηση των δεδομένων με ψηφία ελέγχου ίσα με ένα στην αριστερή πλευρά της εξόδου μπορούμε να κάνουμε μια απλοποίηση στα δυαδικά ψηφία ελέγχου. Εφόσον τα ψηφία ελέγχου σχηματίζουν μια bitonic ακολουθία, μπορούμε να χρησιμοποιήσουμε μόνο ένα υποσύνολο αυτών και να επιτύχουμε ακριβώς το ίδιο αποτέλεσμα για τα δεδομένα. Θα εξηγήσουμε αυτή την απλοποίηση χρησιμοποιώντας το παράδειγμα του σχήματος 4.9. Στο σχήμα 4.9(α) φαίνονται τα δεδομένα εισόδου που θέλουμε να μετακινήσουμε μαζί με τα αντίστοιχα δυαδικά ψηφία ελέγχου. Στην περίπτωση που η EBS μονάδα μπορούσε να μετακινήσει σωστά τα δεδομένα με ψηφία ελέγχου 1 τότε το αποτέλεσμα θα ήταν ίσο με $CDExxxx$. Θα μπορούσαμε να πετύχουμε ακριβώς το ίδιο αποτέλεσμα, αν θεωρούσαμε όλα τα $n/2$ λιγότερο σημαντικά ψηφία ελέγχου ίσα με 1. Αυτό συμβαίνει, γιατί ακόμη και, αν χαλάσουμε την σημαντικότητα μεταξύ των δεδομένων x , δεν πρέπει να ανησυχούμε αφού είναι όλα ίσα με μηδέν. Η απλοποιημένη αυτή λειτουργία φαίνεται στο σχήμα 4.9(β). Έτσι, για να μετακινήσουμε τα δεδομένα χρησιμοποιούμε μόνο τα $n/2$ περισσότερα σημαντικά ψηφία ελέγχου ενώ τα υπόλοιπα θεωρούμε ότι είναι ίσα με 1. Σημειώνουμε πως η λειτουργία αυτή αφορά μόνο στη μετακίνηση των δεδομένων, ενώ η αρχική bitonic ακολουθία ταξινομείται κανονικά χωρίς να απλοποιηθεί.

Στη συνέχεια, θα περιγράψουμε τον τρόπο που τα δεδομένα μετακινούνται στην αριστερή πλευρά της εξόδου της EBS μονάδας μέσα από ένα παράδειγμα των 8 δυαδικών ψηφίων (σχήμα 4.10). Τα δεδομένα εισόδου όπως και τα ψηφία ελέγχου φαίνονται στο σχήμα 4.10(α). Τα δεδομένα A_7 , A_6 , A_5 και A_0 είναι αδιάφορες τιμές (x) αλλά κρατάμε τα ονόματά τους χάριν της σαφήνειας του παραδείγματος. Επίσης, σύμφωνα με την απλοποίηση που περιγράψαμε στην προηγούμενη παράγραφο, το δυαδικό ψηφίο ελέγχου του A_0 μπορεί με ασφάλεια να θεωρηθεί ίσο με 1. Τέλος, σημειώνουμε πως οι συνδέσεις που παρουσιάζονται στα σχήματα 4.10(β)–(δ) αναπαριστούν τις συνδέσεις του πρώτου, του δεύτερου και του τρίτου επιπέδου

Σχήμα 4.10: Ένα παράδειγμα της λειτουργίας του κυκλώματος EBS των 8 δυαδικών ψηφίων.

αντίστοιχα του EBS των 8 δυαδικών ψηφίων που φαίνεται στο σχήμα 4.8.

Θα ξεκινήσουμε το παράδειγμα μας αναλύοντας τη λειτουργία του πρώτου επιπέδου του EBS των 8 δυαδικών ψηφίων όπως φαίνεται στο σχήμα 4.10(β). Τα δυαδικά ψηφία ελέγχου που είναι ίσα με μηδέν στην αριστερή πλευρά της λέξης ελέγχου αναπαριστούν κενές θέσεις οι οποίες πρέπει να καλυφθούν από τα πιο σημαντικά ψηφία δεδομένων της δεξιάς πλευράς. Τα δεδομένα A_3 , A_2 και A_1 πρέπει να αλλάξουν θέσεις με τα A_7 , A_6 , A_5 , τα οποία έχουν ψηφία ελέγχου 0 ώστε να καλύψουν τις κενές θέσεις στην αριστερή πλευρά της λέξης δεδομένων. Αυτή η εναλλαγή επιφέρει δύο αποτελέσματα. Από τη μια τα A_3 , A_2 , και A_1 πέρασαν στη σωστή πλευρά της εξόδου, ενώ από την άλλη η σχετική σημαντικότητα τους σε σχέση με το A_4 , το οποίο έχει και αυτό ψηφίο ελέγχου 1, έχει αλλοιωθεί. Για το λόγο αυτό, στην έξοδο του πρώτου σταδίου ένα νέο σήμα ελέγχου παράγεται για κάθε δυαδικό δεδομένο, που καλείται σήμα εναλλαγής (swap signal). Όταν το swap σήμα είναι ίσο με 1 υποδηλώνεται ότι το αντίστοιχο δεδομένο εισόδου έχει αλλάξει τη θέση του σε αυτό το στάδιο του EBS. Έτσι τα swap σήματα των A_3 , A_2 , A_1 και A_7 , A_6 , A_5 τίθενται ίσα με 1. Αντίθετα, τα swap σήματα που σχετίζονται με τα δυαδικά δεδομένα A_4 και A_0 τίθενται ίσα με 0 γιατί δεν χρειάστηκε να αλλάξουν θέσεις στο πρώτο στάδιο του EBS και βρίσκονται ήδη στη σωστή πλευρά της εξόδου. Γενικά στο πρώτο στάδιο του EBS επιτρέπουμε την αλλαγή μεταξύ των δεδομένων $A_{j+n/2}$ και A_j , $0 \leq j < n/2$ όταν τα αντίστοιχα δυαδικά ψηφία ελέγχου $C_{j+n/2}$ του άνω μισού και C_j του κάτω μισού είναι ίσα με 0 και 1 αντίστοιχα. Εφόσον όμως μπορούμε να θεωρήσουμε πως όλα τα δυαδικά ψηφία ελέγχου του κάτω μισού είναι ίσα με 1 τότε μπορούμε απλά να πούμε για το πρώτο επίπεδο πως η εναλλαγή πραγματοποιείται όταν $C_{j+n/2} = 0$. Έτσι, τα swap σήματα του πρώτου επιπέδου ισούνται με $\overline{C}_{j+n/2}$. Η υλοποίηση των HB και LB του πρώτου επιπέδου του EBS φαίνεται στο σχήμα 4.11(α). Οι AND/OR πύλες χρησιμοποιούνται για την ταξινόμηση της bitonic ακολουθίας των δυαδικών

Σχήμα 4.11: Η υλοποίηση των κελιών HB και LB που χρησιμοποιούνται (α) στο πρώτο επίπεδο και (β) στα υπόλοιπα επίπεδα του EBS.

ψηφίων ελέγχου, ενώ αντίστοιχα οι πολυπλέκτες χρησιμοποιούνται για την μετακίνηση των δεδομένων και ελέγχονται από τα σήματα swar που παράγονται σ' αυτό το επίπεδο. Σ' αυτό το σημείο θα θέλαμε να σημειώσουμε πως, αν και τα HB και LB φαίνονται κοντά το ένα στο άλλο στο σχήμα 4.11(α), παρόλα αυτά μπορεί να τοποθετηθούν σε απομακρυσμένες θέσεις ακολουθώντας τη δομή πεταλούδας του δικτύου EBS.

Κάθε ένα από τα επόμενα επίπεδα του EBS χρησιμοποιεί κατάλληλα τα swar σήματα που παράγονται στο προηγούμενο επίπεδο ώστε να διορθώσει τη σχετική σημαντικότητα των δεδομένων εισόδου. Όταν τα swar σήματα δύο δυαδικών δεδομένων είναι διαφορετικά, σημαίνει ότι η σχετική τους θέση δεν είναι σωστή και πρέπει να αλλάξουν θέσεις. Για παράδειγμα στο δεύτερο επίπεδο του EBS (σχήμα 4.10(γ)) τα δεδομένα A_2, A_4 και A_6, A_0 έχουν διαφορετικά swar σήματα. Όσον αφορά στο ζευγάρι A_2, A_4 , αυτό συμβαίνει γιατί το A_2 προήλθε από το κάτω μισό της αρχικής λέξης δεδομένων και πέρασε μπροστά από το A_4 , το οποίο δεν μετακινήθηκε στο πρώτο επίπεδο. Έτσι, τα A_2, A_4 πρέπει να αλλάξουν θέσεις για να διορθωθεί η μεταξύ τους σημαντικότητα. Αντίστοιχα, για το ζευγάρι A_6, A_0 παρατηρούμε ότι το A_6 ήρθε από την αριστερή πλευρά της αρχικής λέξης και τοποθετήθηκε αριστερότερα του A_0 . Αυτό συνέβη γιατί το A_6 συγκρίθηκε στο πρώτο επίπεδο με το A_2 , το οποίο είχε μεγαλύτερο ψηφίο ελέγχου, και έτσι έχασε τη θέση του από αυτό. Αντίθετα, το A_0 συγκρίθηκε αρχικά με το A_4 με το οποίο είχε ίσο ψηφίο ελέγχου (ίσο με 1) και δεν μετακινήθηκε. Έτσι, το A_0 εφόσον είχε αρχικά μεγαλύτερο ψηφίο ελέγχου σε σχέση με το A_6 πρέπει να τοποθετηθεί μπροστά απ' αυτό στην έξοδο του

δευτέρου επιπέδου. Αντίθετα παρατηρούμε πως τα δεδομένα που έχουν ίδια swar σήματα δεν συντρέχει κάποιος λόγος να αλλάζουν θέση σ' αυτό το επίπεδο, γιατί εφόσον μετακινήθηκαν και τα δύο στο προηγούμενο επίπεδο δεν έχει αλλοιωθεί η μεταξύ τους σημαντικότητα.

Στην έξοδο του δευτέρου επιπέδου, τα δεδομένα που άλλαξαν θέση παίρνουν swar σήματα ίσα με 1. Το επόμενο επίπεδο του EBS λειτουργεί ακριβώς με τον ίδιο τρόπο εξαιτίας της συμμετρικής δομής του δικτύου πεταλούδας. Στο σχήμα 4.10(δ) παρουσιάζεται η λειτουργία του τελευταίου επιπέδου του EBS δικτύου των 8 δυαδικών ψηφίων όπου τα δεδομένα με διαφορετικά swar σήματα εναλλάσσουν τις θέσεις τους. Εφόσον τα τελευταία τέσσερα δυαδικά δεδομένα $A_0A_7A_6A_5$ είναι από την αρχή ίσα με 0 (σχήμα 4.10(α)), το πραγματικό αποτέλεσμα $A_4A_3A_2A_1xxxx$ είναι και το ζητούμενο.

Στη γενική περίπτωση σε όλα τα στάδια ενός EBS, εκτός του πρώτου, τα δεδομένα αλλάζουν θέση εφόσον έχουν διαφορετικά swar σήματα και τα νέα swar σήματα παίρνουν την τιμή 1. Έτσι μπορούμε εύκολα να συμπεράνουμε πως τα νέα swar σήματα ισούνται με το αποτέλεσμα της λογικής πράξης XOR των σημάτων swar του προηγούμενου επιπέδου. Το εσωτερικό των HB/LB που υλοποιούν τη λειτουργία που μόλις περιγράψαμε φαίνεται στο σχήμα 4.11(β). Τέλος, οφείλουμε να πούμε πως, αν και τα swar σήματα που παράγονται στα HB και LB είναι ίδια, διαλέξαμε να τα παράγουμε δύο φορές ώστε να μειώσουμε τη χωρητικότητα εξόδου κάθε κελιού και να αυξήσουμε την ομοιομορφία του κυκλώματος.

4.6.2 Αλλαγή φοράς ταξινόμησης

Η παραπάνω περιγραφή της λειτουργία του EBS περιορίστηκε στην περίπτωση που τα δεδομένα με δυαδικά ψηφία ελέγχου ίσα με 1 μαζεύονται στην αριστερή πλευρά της εξόδου. Παρόλα αυτά για την υλοποίηση της GRP χρειαζόμαστε μία ακόμη μονάδα εξαγωγής, η οποία θα συγκεντρώνει τα επιλεγμένα δεδομένα στη δεξιά πλευρά της εξόδου. Για να επιτρέψουμε τη νέα αυτή λειτουργία, απαιτούνται δύο μικρές αλλαγές στο EBS δίκτυο που μόλις περιγράψαμε. Στο πρώτο επίπεδο κάθε EBS τα σήματα που ελέγχουν τη λειτουργία των πολυπλεκτών πρέπει να αλλάζουν από $\overline{C}_{j+n/2}$ σε $\overline{C}_j$. Η αλλαγή αυτή αυτόματα αλλάζει και τη γέννηση των σημάτων swar στα επόμενα επίπεδα. Δεύτερον, σε όλα τα επίπεδα του EBS οι λογικές πύλες OR των κελιών HB αντικαθίστανται από λογικές πύλες AND, ενώ οι πύλες AND των κελιών LB μετατρέπονται σε λογικές πύλες OR.

4.7 Επαυξημένο merge δίκτυο ταξινόμησης

Η βασική δομή του EMSN των 8 δυαδικών ψηφίων, το οποίο μαζεύει τα επιλεγμένα δεδομένα προς την αριστερή πλευρά, φαίνεται στο σχήμα 4.12. Όπως και στην περίπτωση του EBSN, το επαυξημένο merge δίκτυο ταξινόμησης EMSN αποτελείται και αυτό από κατάλληλα συνδεδεμένα υποδίκτυα, τα οποία ονομάζονται επαυξημένοι ταξινομητές merge (*Enhanced Merge Sorters*, EMS). Η λειτουργία του EMS είναι ανάλογη με αυτή του EBS που περιγράφηκε στις προηγούμενες παραγράφους. Πάντως, λόγω της διαφορετικής αρχής λειτουργίας του EMS, απαιτούνται νέες αλλαγές σε επίπεδο κυκλώματος ώστε να καταφέρουμε να διατηρήσουμε τη σχετική σημαντικότητα των επιλεγμένων δεδομένων της εισόδου, παράλληλα με τη μετακίνηση τους προς την αριστερή πλευρά της εξόδου.

Σχήμα 4.12: Η δομή ενός επαυξημένου merge δικτύου ταξινόμησης και η τοπολογία των αντίστοιχων EMS υπο-δικτύων μεγέθους 2, 4 και 8 δυαδικών ψηφίων.

4.7.1 Η σχεδίαση του υποδικτύου EMS

Η δομή ενός EMS των 8 δυαδικών ψηφίων φαίνεται στο σχήμα 4.12. Οι συνδέσεις των κελιών του EMS είναι διαφορετικές από αυτές του δικτύου πεταλούδας του EBS. Στην περίπτωση του EMS τα κελιά σύγκρισης και εναλλαγής που εκτελούν τη λειτουργία του αλγορίθμου ονομάζονται HM και LM αντίστοιχα. Επίσης, όπως και στην περίπτωση του EBS μόνο τα $n/2$ περισσότερο σημαντικά δυαδικά ψηφία ελέγχου απαιτούνται ώστε να καθορίσουν τη μετακίνηση των δεδομένων, ενώ τα υπόλοιπα μπορούν να θεωρηθούν με ασφάλεια ίσα με 1. Η εξήγηση γι' αυτήν την απλοποίηση είναι όμοια με αυτή που παρουσιάστηκε στο σχήμα 4.9.

Στη συνέχεια η λειτουργία του EMS θα περιγραφεί με τη βοήθεια ενός παραδείγματος στα 8 δυαδικά ψηφία, οι διαφορετικές φάσεις του οποίου απεικονίζονται στο σχήμα 4.13. Οι συνδέσεις μεταξύ των δεδομένων που φαίνονται στα σχήματα 4.13(β)-(δ) αναπαριστούν τις συνδέσεις του EMS των 8 δυαδικών ψηφίων που φαίνεται στο σχήμα 4.12.

Στο σχήμα 4.13(α) τα πιο σημαντικά ψηφία ελέγχου που είναι ίσα με 1 υποδηλώνουν πως καμιά αλλαγή δεν πρέπει να συμβεί σ' αυτές τις θέσεις. Τα αντίστοιχα δεδομένα A_7 και A_6 είναι στη σωστή θέση και πρέπει να παραμείνουν στην ίδια θέση στην έξοδο του EMS. Αυτός

Σχήμα 4.13: Ένα παράδειγμα της λειτουργίας του κυκλώματος EMS των 8 δυαδικών ψηφίων.

ο περιορισμός ισχύει για όλα τα επίπεδα του EMS. Έτσι, όλα τα HM και LM κελιά τα οποία λαμβάνουν είσοδο από την αριστερή πλευρά του EMS (από τη θέση j με $j > n/2$) δεν πρέπει να πραγματοποιήσουν καμιά εναλλαγή όταν το αντίστοιχο δυαδικό ψηφίο ελέγχου $C_j = 1$. Ο περιορισμός αυτός θα αναφέρεται στο εξής ως Φράγμα I. Αντίθετα, οι θέσεις της αριστερής πλευράς με μηδενικά ψηφία ελέγχου είναι κενές θέσεις οι οποίες πρέπει να καλυφθούν από τα πιο σημαντικά δεδομένα της δεξιάς πλευράς που έχουν ψηφία ελέγχου ίσα με 1. Ακολουθώντας τις συνδέσεις του πρώτου επιπέδου του EMS των 8 δυαδικών ψηφίων, όπως φαίνονται στο σχήμα 4.13(β), οι μόνες αλλαγές που επιτρέπονται σ' αυτό το επίπεδο είναι μεταξύ των δυαδικών δεδομένων A_1 , A_5 και A_0 , A_4 αντίστοιχα. Μετά από την εναλλαγή των θέσεων τους συμβαίνουν δύο πράγματα. Τα A_1 και A_0 έχουν παραβιάσει τη σχετική σημαντικότητα τους σε σχέση με τα ψηφία A_3 και A_2 , που επίσης έχουν ψηφία ελέγχου ίσα με 1, ενώ παράλληλα τα A_4 και A_5 έχουν σωστά τοποθετηθεί στο τέλος της λέξης εξόδου. Στη γενική περίπτωση στο πρώτο επίπεδο του EMS η εναλλαγή μεταξύ των δεδομένων $A_{j+n/2}$ και A_j , $0 \leq j < n/2$ πραγματοποιείται μόνο όταν το δυαδικό ψηφίο ελέγχου $C_{j+n/2}$ είναι ίσο με 0. Ακολουθώντας αυτήν τη αρχή, μπορούμε να εγγυηθούμε ότι τα πιο σημαντικά δεδομένα της αριστερής πλευράς δεν θα χάσουν τη θέση τους και ότι τα άχρηστα δεδομένα της ίδιας πλευράς με μηδενικά ψηφία ελέγχου θα μεταφερθούν σωστά στη δεξιότερη πλευρά της εξόδου. Στα επόμενα επίπεδα όλα τα δυαδικά δεδομένα πρέπει να μετακινηθούν μεταξύ των ορίων που θέτουν τα ψηφία A_7A_6 για την αριστερή πλευρά και τα A_5A_4 για τη δεξιά. Δεν πρέπει να επιτρέψουμε καμιά εναλλαγή να συμβεί στις θέσεις 0 και 1 του δικτύου, εκεί όπου τοποθετήθηκαν τα A_4 και A_5 μετά το πρώτο στάδιο. Με άλλα λόγια, μετά το πρώτο στάδιο του EMS οι θέσεις οι οποίες πρέπει να μπλοκαριστούν στη

δεξιά πλευρά είναι ίσες με το πλήθος των μηδενικών ψηφίων ελέγχου της αριστερής πλευράς. Έτσι, φτάνουμε στο δεύτερο φράγμα (Φράγμα II) που περιορίζει τις εναλλαγές στο δίκτυο EMS και το οποίο ορίζεται ως εξής: Τα HM και LM κελιά όλων των επιπέδων μετά το πρώτο τα οποία δέχονται είσοδο από τη θέση j με $j < n/2$, (ανήκει στη δεξιά πλευρά του EMS), δεν επιτρέπουν καμία εναλλαγή όταν το δυαδικό ψηφίο ελέγχου της εισόδου $C_{j+n/2}$ είναι ίσο με 0.

Στη συνέχεια, όπως και στην περίπτωση του bitonic δικτύου ταξινόμησης στα δεδομένα που αλλάζουν θέση στο πρώτο επίπεδο ανατίθεται η τιμή του swap σήματος να είναι ίση με 1 (σχήμα 4.13(β)). Σ' αυτό το σημείο πρέπει να σημειώσουμε πως η δημιουργία των σημάτων swap είναι αναγκαία μόνο στις θέσεις όπου υπάρχουν HM και LM κελιά στην ίδια θέση στο επόμενο επίπεδο. Για τις υπόλοιπες θέσεις τα σήματα swap θεωρούνται αυτόματα ίσα με 0. Για παράδειγμα, εξαιτίας της δομής του EMS των 8 δυαδικών ψηφίων οι θέσεις 7, 6, 1, και 0 δέχονται απευθείας ένα swap σήμα που είναι ίσο με μηδέν ακόμα και αν έχει πραγματοποιηθεί κάποια εναλλαγή στις θέσεις αυτές.

Η λειτουργία των επομένων επιπέδων καθορίζεται από τα σήματα swap ακριβώς όπως και στην περίπτωση του EBS δικτύου. Παράλληλα όμως για το EMS οι περιορισμοί που θέτουν τα φράγματα I και II πρέπει να λαμβάνονται υπόψιν. Με άλλα λόγια, μια εναλλαγή επιτρέπεται στα κελιά HM και LM, όταν τα σήματα swap των δεδομένων είναι διαφορετικά και τα φράγματα που αφορούν τις αντίστοιχες θέσεις του δικτύου είναι και τα δύο ίσα με 0. Το φράγμα B_j που συνδέεται με τη θέση j του δικτύου ισούται με C_j όταν $j > n/2$ (Αριστερή πλευρά) και με $B_j = \overline{C}_{j+n/2}$ όταν $j < n/2$ (Δεξιά πλευρά). Καλό είναι να διαχωρίσουμε σ' αυτό το σημείο πως τα σήματα swap είναι άμεσα συνδεδεμένα με τα δεδομένα που μετακινούνται μέσα στο δίκτυο, ενώ τα φράγματα περιγράφουν μια ιδιότητα συγκεκριμένων θέσεων του δικτύου και παραμένουν αμετάβλητα για όλα τα επίπεδα σύγκρισης.

Έχοντας περιγράψει το γενικό κανόνα λειτουργία του EMS μπορούμε να επιστρέψουμε στο παράδειγμα μας ξεκινώντας από το σχήμα 4.13(γ). Τα δεδομένα A_3 και A_2 αλλάζουν θέσεις με τα A_1 και A_0 αντίστοιχα αφού έχουν διαφορετικά σήματα swap και δεν περιορίζονται από κανένα φράγμα. Αντίστοιχα, στο τελευταίο επίπεδο του EMS των 8 δυαδικών ψηφίων (σχήμα 4.13(δ)) τα δυαδικά δεδομένα A_2 και A_1 δεν μετακινούνται γιατί έχουν ίσα σήματα swap που υποδηλώνουν ότι είναι τοποθετημένα στη σωστή θέση της εξόδου. Επίσης, τα A_6 και A_3 αν και έχουν διαφορετικά swap σήματα δεν μετακινούνται, γιατί το φράγμα που αφορά τη θέση 6 του δικτύου είναι ίσο με 1 (φράγμα I, $B_6 = C_6 = 1$). Το ίδιο ισχύει και για τα A_0 και A_5 , η εναλλαγή των οποίων απαγορεύεται, αφού το φράγμα II, που αφορά τη θέση 1, είναι ενεργοποιημένο ($B_1 = \overline{C}_5 = 1$).

Οι συνδέσεις και η λειτουργία του πρώτου επιπέδου του EMS είναι ίδιες με αυτές του EBS. Έτσι τα κελιά HM και LM του πρώτου επιπέδου υλοποιούνται ακριβώς με τον ίδιο τρόπο όπως και στην περίπτωση του EBS και απεικονίζονται στο σχήμα 4.11(α). Για τα υπόλοιπα επίπεδα του EMS η υλοποίηση των κελιών HM και LM φαίνεται στο σχήμα 4.14. Σημειώνουμε πως οι λογικές πύλες AND και OR που χρησιμοποιούνται για να ταξινομήσουν τα δυαδικά ψηφία ελέγχου, ώστε αυτά να χρησιμοποιηθούν στα επόμενα στάδια του EMSN, δεν μπλοκάρονται από κανένα φράγμα και δεν παρεμβάλλονται στη μετακίνηση των δεδομένων πέρα από το πρώτο στάδιο. Τέλος, για την υποστήριξη της ταξινόμησης προς την αντίθετη κατεύθυνση, η αλλαγή που περιγράφεται στην παράγραφο 4.6.2 μπορεί να εφαρμοστεί και στην περίπτωση του EMS χωρίς καμία επιπλέον αλλαγή.

Σχήμα 4.14: Η υλοποίηση των κελιών HM και LM στην περίπτωση της μονάδας EMS που χρησιμοποιούνται σε όλα τα στάδια του δικτύου εκτός του πρώτου.

4.8 Επαυξημένο γραμμικό δίκτυο αντιμετάθεσης

Η σχεδίαση του γραμμικού δικτύου εναλλαγής άρτιων-περιττών στηλών είναι η πιο απλή, αφού η δομή και η λειτουργία του ταιριάζουν καλύτερα στην υλοποίηση της εντολής GRP. Στην περίπτωση αυτή δεν απαιτούνται δύο χωριστές μονάδες εξαγωγής, επειδή η σχετική σημαντικότητα μεταξύ των άσσων και των μηδενικών διατηρείται χρησιμοποιώντας ένα μόνο δίκτυο.

Η νέα μονάδα αντιμετάθεσης που υλοποιεί τη GRP των 8 δυαδικών ψηφίων φαίνεται στο σχήμα 4.15. Τα κελιά σύγκρισης και εναλλαγής δέχονται δύο ζεύγη από δεδομένα και ψηφία ελέγχου. Στην έξοδο H του κελιού εμφανίζεται το μεγαλύτερο από τα δύο ψηφία ελέγχου μαζί με το αντίστοιχο δυαδικό ψηφίο δεδομένων. Το αντίθετο ισχύει για την L έξοδο του κελιού. Η υλοποίηση του κελιού εναλλαγής φαίνεται επίσης στο σχήμα 4.15. Σε κάθε επίπεδο του δικτύου τα δυαδικά ψηφία ελέγχου ταξινομούνται χρησιμοποιώντας τις πύλες OR για την επιλογή του μεγαλύτερου ψηφίου και τις πύλες AND για την επιλογή του μικρότερου από τα υπο-σύγκριση ψηφία. Κάθε φορά που το αριστερό ψηφίο ελέγχου C_A είναι ίσο με 0 και το δεξί C_B είναι ίσο με 1 τότε τα δεδομένα D_A και D_B εναλλάσσονται. Σε οποιαδήποτε άλλη περίπτωση τα ψηφία ελέγχου είναι σε σωστή σειρά και τα αντίστοιχα δεδομένα απαγορεύεται να μετακινηθούν. Η εναλλαγή των δεδομένων καθορίζεται από το σήμα SW το οποίο ισούται με

$$SW = \overline{C_A} \cdot C_B. \quad (4.2)$$

Τα κελιά σύγκρισης και εναλλαγής του τελευταίου επιπέδου είναι απλούστερα από τα υπόλοιπα, εφόσον τα τελικά ταξινομημένα ψηφία ελέγχου δε χρειάζονται στην έξοδο. Ένα παράδειγμα της λειτουργίας του αλγορίθμου που παρουσιάζει τόσο τη μετακίνηση των δεδομένων όσο και τη διαδοχική ταξινόμηση των ψηφίων ελέγχου φαίνεται στο σχήμα 4.16.

4.8.1 Μια ταχύτερη εκδοχή

Στην πρώτη εκδοχή των κελιών σύγκρισης και εναλλαγής το σήμα SW που κατευθύνει την

Σχήμα 4.15: Η δομή του επαυξημένου γραμμικού δικτύου ταξινόμησης των 8 δυαδικών ψηφίων και η υλοποίηση του βασικού κελιού σύγκρισης και εναλλαγής που χρησιμοποιείται για την υλοποίησή του.

εναλλαγή υπολογίζεται μέσω μιας πύλης AND η οποία είχε τη μία είσοδο της ανεστραμμένη. Η λειτουργία του κελιού αυτού θα ήταν σαφέστατα πιο γρήγορη αν μπορούσε η εναλλαγή να καθοριστεί απευθείας από τα σήματα ελέγχου C_A και C_B και αν μπορούσαμε να αντικαταστήσουμε τους πολυπλέκτες από απλούστερες πύλες. Μια τέτοια απλοποίηση μπορεί να συμβεί αν συνδιάσουμε την αρχιτεκτονική των δύο μονάδων εξαγωγής του σχήματος 4.5 και την δομή του επαυξημένου γραμμικού δικτύου ταξινόμησης.

Στην περίπτωση αυτή, τα κελιά σύγκρισης και εναλλαγής θα δέχονταν τέσσερις εισόδους, ενώ τα σήματα ελέγχου θα έμεναν τα ίδια. Σύμφωνα με την αρχιτεκτονική των δύο μονάδων εξαγωγής, τα δυαδικά ψηφία των δεδομένων πρέπει να γίνουν masked με τα αντίστοιχα δυαδικά ψηφία ελέγχου πριν δοθούν στα κυκλώματα που αναλαμβάνουν τη μεταφορά τους στα δύο άκρα της λέξης εξόδου. Εξαιτίας της εφαρμογής της μάσκας, κάθε δυαδικό ψηφίο δεδομένων D που ελέγχεται από το σήμα ελέγχου C είναι ίσο με $D \cdot C$ για την αριστερή μονάδα εξαγωγής και ίσο με $D \cdot \overline{C}$ για τη δεξιά μονάδα εξαγωγής. Γνωρίζουμε ότι το ETN μπορεί απευθείας να υλοποιήσει τη λειτουργία είτε της αριστερής είτε της δεξιάς μονάδας εξαγωγής, αφού ήδη μπορεί να επιτελέσει και τις δύο λειτουργίες ταυτόχρονα. Αν θεωρήσουμε ότι αναλαμβάνει την κάθε λειτουργία χωριστά, τότε μπορούμε να υλοποιήσουμε κατά πολύ το κύκλωμα.

Θα θεωρήσουμε αρχικά την περίπτωση της αριστερής μονάδας εξαγωγής. Γι' αυτήν την περίπτωση συμβολίζουμε τις εισόδους δεδομένων των κελιών σύγκρισης και εναλλαγής ως $D_{A \rightarrow L}$ (αριστερή είσοδος) και $D_{B \rightarrow L}$ (δεξιά είσοδος). Τα δεδομένα που θα εμφανιστούν στην H και την L έξοδο αντίστοιχα, συμβολίζονται ως $D_{H \rightarrow L}$ και $D_{L \rightarrow L}$. Ακολουθώντας τη λογική της πολύπλεξης

Σχήμα 4.16: Ένα παράδειγμα της παράλληλης σύγκρισης των δυαδικών ψηφίων ελέγχου και της μετακίνησης των αντίστοιχων δεδομένων κατά την εκτέλεση της GRP με τη χρήση του επαυξημένου γραμμικού δικτύου ταξινόμησης.

κάθε κελιού (Σχήμα 4.15) και τη σχέση ενεργοποίησης της εναλλαγής (4.2), τα δεδομένα στην έξοδο δίνονται από τις σχέσεις που ακολουθούν.

$$D_{H \rightarrow L} = D_{A \rightarrow L} \cdot \overline{(\overline{C_A} \cdot C_B)} + D_{B \rightarrow L} \cdot \overline{C_A} \cdot C_B \quad (4.3)$$

$$D_{L \rightarrow L} = D_{A \rightarrow L} \cdot \overline{C_A} \cdot C_B + D_{B \rightarrow L} \cdot \overline{(\overline{C_A} \cdot C_B)} \quad (4.4)$$

Εφόσον τα δυαδικά ψηφία δεδομένων της αριστερής μονάδας εξαγωγής $D_{A \rightarrow L}$ και $D_{B \rightarrow L}$ έχουν γίνει ήδη masked με τα δυαδικά ψηφία ελέγχου C_A και C_B αντίστοιχα, τότε οι σχέσεις (4.3) και (4.4) που περιγράφουν την H και την L έξοδο του κελιού σύγκρισης και εναλλαγής μπορούν να απλοποιηθούν ως εξής:

$$D_{H \rightarrow L} = D_{A \rightarrow L} + D_{B \rightarrow L} \cdot \overline{C_A} \quad (4.5)$$

$$D_{L \rightarrow L} = D_{B \rightarrow L} \cdot C_A \quad (4.6)$$

Στη δεξιά μονάδα εξαγωγής, εξαιτίας της αντίθετης κατεύθυνσης ταξινόμησης, η λειτουργία των εξόδων H και L εναλλάσσεται και τα δυαδικά ψηφία ελέγχου είναι ανεστραμμένα. Έτσι, αν και πάλι με το δείκτη A συμβολίζουμε την αριστερή είσοδο και με B τη δεξιά είσοδο του κελιού σύγκρισης και εναλλαγής, τότε η εναλλαγή των δεδομένων συμβαίνει στη δεξιά μονάδα εξαγωγής στην περίπτωση που $(\overline{C_A}, \overline{C_B}) = (1, 0)$, μιας και τα δεδομένα κινούνται σταδιακά στη δεξιά πλευρά της εξόδου. Αυτή η λειτουργία είναι η ίδια με τη διαδικασία εναλλαγής της αριστερής μονάδας εξαγωγής, όπου η εναλλαγή πραγματοποιείται όταν $(C_A, C_B) = (0, 1)$, όπως ήδη περιγράφηκε από την εξίσωση (4.2). Αυτή η παρατήρηση μας επιτρέπει να μοιράσουμε τα σήματα ελέγχου εναλλαγής μεταξύ των δύο μονάδων εξαγωγής (αριστερής και δεξιάς). Έτσι,

Σχήμα 4.17: Η υλοποίηση του κελιού σύγκρισης και εναλλαγής που χρησιμοποιεί η ταχύτερη εκδοχή του επαυξημένου γραμμικού δικτύου αντιμετάθεσης.

βάσει αυτού του συμπεράσματος και της σχέσης (4.2) τα δεδομένα $D_{A \rightarrow R}$ και $D_{B \rightarrow R}$ επιλέγονται να κατευθυνθούν στις H και L εξόδους των κελιών της δεξιάς μονάδας εξαγωγής, σύμφωνα με τις σχέσεις που ακολουθούν.

$$D_{H \rightarrow R} = D_{B \rightarrow R} \cdot (\overline{C_A} \cdot C_B) + D_{A \rightarrow R} \cdot \overline{C_A} \cdot C_B \quad (4.7)$$

$$D_{L \rightarrow R} = D_{B \rightarrow R} \cdot \overline{C_A} \cdot C_B + D_{A \rightarrow R} \cdot (\overline{C_A} \cdot C_B) \quad (4.8)$$

Στην περίπτωση της δεξιάς μονάδας εξαγωγής τα δεδομένα $D_{A \rightarrow R}$ και $D_{B \rightarrow R}$ έχουν αρχικά καλυφθεί από τα ανεστραμμένα δυαδικά ψηφία ελέγχου $\overline{C_A}$ και $\overline{C_B}$. Λαμβάνοντας υπ' όψιν αυτή τη λειτουργία, οι σχέσεις που περιγράφουν τις τιμές στις εξόδους του κελιού σύγκρισης και εναλλαγής μπορούν να γραφούν ως εξής:

$$D_{H \rightarrow R} = D_{B \rightarrow R} + D_{A \rightarrow R} \cdot C_B \quad (4.9)$$

$$D_{L \rightarrow R} = D_{A \rightarrow R} \cdot \overline{C_B} \quad (4.10)$$

Όπως φαίνεται από τις σχέσεις που περιγράφουν τις εξόδους H και L για τα κελιά σύγκρισης και εναλλαγής και των δύο μονάδων εξαγωγής, μπορούμε να επιτύχουμε τη μεταφορά των δεδομένων απευθείας από τα σήματα ελέγχου C_A και C_B χωρίς την ανάγκη δημιουργίας επιπλέον σημάτων ελέγχου όπως αυτό του SW . Επιπλέον, οι πύλες AND και OR που χρησιμοποιούνται για να ταξινομήσουν τα δυαδικά ψηφία ελέγχου ώστε να καθορίσουν τις εναλλαγές στα επόμενα επίπεδα είναι κοινές μεταξύ των δύο δικτύων εξαγωγής. Για τους λόγους αυτούς και εξαιτίας της ομοιομορφίας του κυκλώματος, μπορούμε να τοποθετήσουμε τη λογική των δύο ανεξάρτητων δικτύων εξαγωγής στο ίδιο κελί σύγκρισης και εναλλαγής. Η νέα υλοποίηση του κελιού αυτού φαίνεται στο σχήμα 4.17. Τέλος, υπενθυμίζουμε ότι για να ολοκληρώσουμε την εκτέλεση της εντολής GRP οι τέσσερις έξοδοι των κελιών σύγκρισης και εναλλαγής ενώνονται ανά δύο μέσω της λογικής πράξης OR και παράγουν τα δύο δυαδικά ψηφία της εξόδου.

Μπορεί να γίνει εύκολα αντιληπτό πως η υλοποίηση της GRP για λέξεις των 32 ή των 64 δυαδικών ψηφίων με τη χρήση του ETN είναι μη συμφέρουσα και οδηγεί σε κυκλώματα με μη αποδεκτή καθυστέρηση. Παρόλα αυτά το ETN αποτελεί μια ιδιαίτερα ελκυστική λύση όταν οι ελάχιστες ποσότητες που μεταφέρονται από το δίκτυο είναι μεγαλύτερες του ενός δυαδικού ψηφίου, π.χ., 1 byte. Η παραλλαγή αυτής της εντολής GRP θα αναλυθεί στην επόμενη παράγραφο.

4.9 Πρακτικές περιπτώσεις

Στις περισσότερες περιπτώσεις στους σύγχρονους μικροεπεξεργαστές τα δεδομένα που χρειάζεται να αντιμετωπιστούν είναι ποσότητες μεγαλύτερες του ενός δυαδικού ψηφίου. Συνήθως η ελάχιστη μεταφερομένη ποσότητα είναι του ενός ή των δύο bytes. Οι προτεινόμενες μονάδες εκτέλεσης της GRP μπορούν κατευθείαν να υποστηρίξουν υπολέξεις των περισσότερων δυαδικών ψηφίων. Αυτό μπορεί απλά να συμβεί αν τα δυαδικά ψηφία που ανήκουν στην ίδια υπολέξη ελέγχονται από τα ίδια ψηφία ελέγχου. Μ' αυτόν τον τρόπο όλα τα δυαδικά ψηφία της υπολέξης θα μεταφερθούν στην έξοδο σαν μια ενιαία ποσότητα. Στην ενότητα αυτή θα μελετήσουμε πως οι προτεινόμενες μονάδες αντιμετάθεσης μπορούν να απλοποιηθούν όταν σχεδιάζονται για την αποκλειστική αντιμετάθεση υπολέξεων μεγαλύτερων του ενός δυαδικού ψηφίου.

Στη συνέχεια, μια GRP μονάδα των n δυαδικών ψηφίων η οποία υποστηρίζει υπολέξεις των k δυαδικών ψηφίων, θα ονομάζεται ως $(n/k) \times k$ GRP μονάδα. Για παράδειγμα, η 8×8 GRP μονάδα αναφέρεται σε μια 64 δυαδικών ψηφίων μονάδα αντιμετάθεσης όπου το ελάχιστο μέγεθος της κάθε υπολέξης είναι 8 δυαδικά ψηφία. Στην περίπτωση αυτή ο αριθμός των διακριτών υπολέξεων είναι ίσος με n/k . Έτσι, μόνο n/k δυαδικά ψηφία ελέγχου απαιτούνται για να διαχωριστούν οι υπολέξεις σε δύο ομάδες στην έξοδο σύμφωνα με τον ορισμό της GRP. Επίσης, σε κάθε πράξη σύγκρισης και εναλλαγής οι υπολέξεις των k δυαδικών ψηφίων εναλλάσσονται παράλληλα. Αυτό σημαίνει πως οι 2-σε-1 πολυπλέκτες των κελιών σύγκρισης και εναλλαγής αντικαθίστανται από k πολυπλέκτες οι οποίοι ελέγχονται από το ίδιο σήμα επιλογής. Αντίστοιχα, ο μικρότερος αριθμός από διακριτές υπολέξεις οδηγεί σε μείωση του αριθμού των επιπέδων σύγκρισης του κάθε δικτύου ταξινόμησης. Πιο συγκεκριμένα, για τα bitonic και merge δίκτυα ταξινόμησης ο αριθμός των επιπέδων είναι ίσος με $\frac{1}{2}(\log_2(\frac{n}{k}))(1 + \log_2(\frac{n}{k}))$, ενώ για το γραμμικό δίκτυο αντιμετάθεσης n/k επίπεδα απαιτούνται για την ολοκλήρωση μιας GRP λειτουργίας. Τέλος, οφείλουμε να σημειώσουμε πως μια $(n/k) \times k$ GRP μονάδα μπορεί απευθείας να υποστηρίξει και μεγαλύτερες υπολέξεις, των οποίων το μέγεθος είναι πολλαπλάσιο του k .

Ανάλογα με την τιμή του k , το γραμμικό δίκτυο αντιμετάθεσης μπορεί να είναι η πιο συμφέρουσα λύση για την υλοποίηση της GRP μονάδας που είναι προσανατολισμένη στις υπολέξεις. Λόγου χάρη, η σχεδίαση ενός γραμμικού δικτύου αντιμετάθεσης μπορεί να είναι απαγορευτική στην περίπτωση που οι υπολέξεις έχουν μέγεθος ένα ή δύο δυαδικά ψηφία ($k = 1, 2$), εξαιτίας του μεγάλου αριθμού επιπέδων σύγκρισης που απαιτούνται. Όμως, στις πιο πρακτικές περιπτώσεις, όπως για παράδειγμα μιας 8×8 GRP μονάδας, ο αριθμός των επιπέδων που απαιτούνται από όλα τα δίκτυα ταξινόμησης που εξετάσαμε είναι σχεδόν ο ίδιος. Στην περίπτωση αυτή, τα γραμμικά δίκτυα έχουν δύο σημαντικά πλεονεκτήματα έναντι των δικτύων λογαριθμικού βάθους. Πρώτον, μόνο ένα δίκτυο ταξινόμησης απαιτείται για την αντιμετάθεση των δεδομένων, με αποτέλεσμα το κόστος σε υλικό που απαιτείται για την υλοποίηση της GRP

Σχήμα 4.18: Η δομή μιας υβριδικής 16×4 μονάδας εκτέλεσης της εντολής GRP.

να είναι περίπου το μισό σε σχέση με τις άλλες υλοποιήσεις. Ταυτόχρονα, το γραμμικό δίκτυο αντιμετάθεσης προσφέρει κοντινές συνδέσεις καλωδίων, στοιχείο που επιφέρει τη μείωση τόσο του αριθμού των απαιτούμενων ενισχυτών για την οδήγηση των γραμμών διασύνδεσης όσο και του μεγέθους των προπορευόμενων πυλών.

Παρόλα αυτά, για μικρότερες υπολέξεις τα οφέλη του ETN γρήγορα χάνονται εξαιτίας της απότομης αύξησης της καθυστέρησης των κυκλωμάτων. Και σε αυτές τις περιπτώσεις μπορούμε να σχεδιάσουμε αποτελεσματικά κυκλώματα αντιμετάθεσης αν συνδιάσουμε σε μια νέα υβριδική δομή την ταχύτητα του merge ή bitonic δικτύου ταξινόμησης και το μικρό εμβαδόν χωροθέτησης του γραμμικού δικτύου αντιμετάθεσης. Για παράδειγμα, μια 16×4 GRP μονάδα μπορεί να σχεδιαστεί χρησιμοποιώντας δύο 8×4 GRP μονάδες βασισμένες στο ETN και ένα EMS, το οποίο θα αναλάβει την συνένωση 16 υπολέξεων των 4 δυαδικών ψηφίων η καθεμία. Η υβριδική αυτή δομή φαίνεται στο σχήμα 4.18. Τα δύο ETN ταξινομούν παράλληλα τα δύο ζεύγη των 8 υπολέξεων της εισόδου, ενώ στη συνέχεια δύο EMS μονάδες αναλαμβάνουν τη συνένωση των δύο επιμέρους αποτελεσμάτων. Η μια από αυτές είναι υπεύθυνη για να ενώσει τις υπολέξεις με ψηφία ελέγχου ίσα με ένα και η άλλη τις υπολέξεις που ελέγχονται από μηδενικά ψηφία ελέγχου. Οι υπολέξεις πριν δοθούν στις αντίστοιχες EMS μονάδες γίνονται masked με τα ψηφία ελέγχου. Στην έξοδο τα δύο αποτελέσματα ομαδοποιούνται τελικά σε ένα αποτέλεσμα με τη χρήση της λογικής πράξης OR. Η υβριδική 16×4 GRP δομή που παρουσιάσαμε αναμένεται να είναι πιο γρήγορη από αυτή που θα βασίζεται αποκλειστικά στο ETN, ενώ θα, απαιτεί σημαντικά λιγότερο εμβαδόν από το αντίστοιχο κύκλωμα που σχεδιάστηκε αποκλειστικά με βάση το EMSN. Ποιά από τις προτεινόμενες δομές είναι η πιο συμφέρουσα για την υλοποίηση της εντολής GRP ανάλογα με το μέγεθος της υπολέξης που υποστηρίζει, θα αναζητηθεί μέσα από τα πειραματικά αποτελέσματα που παρουσιάζονται στην επόμενη ενότητα.

Σχήμα 4.19: Η ενέργεια και η καθυστέρηση των 64×1 μονάδων εκτέλεσης GRP, οι οποίες υλοποιήθηκαν με το επαυξημένο bitonic και merge δίκτυο ταξινόμησης αντίστοιχα. Η βελτιστοποιήσεις και οι μετρήσεις πραγματοποιήθηκαν για διαφορετικές τιμές της χωρητικότητας εισόδου και εξόδου του κυκλώματος C_{in}/C_{out} . (α) $C_{in}/C_{out} = 25/100\text{fF}$ και (β) $C_{in}/C_{out} = 5/25\text{fF}$.

4.10 Πειραματικά αποτελέσματα

Αρχικά μελετήσαμε τη συμπεριφορά των κυκλωμάτων στο χώρο ενέργεια-καθυστέρηση των πιο αποδοτικών 64×1 GRP μονάδων. Η καθυστέρηση και η απαιτούμενη ενέργεια των GRP μονάδων που σχεδιάστηκαν με βάση τα EBSN και EMSN φαίνονται στο σχήμα 4.19. Τα κυκλώματα βελτιστοποιήθηκαν και μετρήθηκαν θεωρώντας δύο περιπτώσεις για τις χωρητικότητες εισόδου και εξόδου. Στην πρώτη περίπτωση οι εξοδοί του κυκλώματος είναι αναγκασμένες να οδηγήσουν μια χωρητικότητα των 100fF και η μέγιστη επιτρεπόμενη χωρητικότητα εισόδου είναι ίση με 25fF. Στη δεύτερη περίπτωση, η οποία αναφέρεται σε λιγότερο απαιτητικά μεγέθη χωρητικοτήτων, θεωρήσαμε πως οι χωρητικότητες εισόδου/εξόδου είναι ίσες με 5/25fF αντίστοιχα. Οι μετρήσεις της ενέργειας ανά λειτουργία που απαιτούν τα προτεινόμενα κυκλώματα πραγματοποιήθηκαν μετά την εξομοίωση των κυκλωμάτων σε Spice, όπου τα τυχαία διανύσματα εισόδου που χρησιμοποιήθηκαν προκάλεσαν περίπου το 30% των κόμβων του κυκλώματος να αλλάξει κατάσταση.

Όπως μπορούμε να παρατηρήσουμε από τα διαγράμματα του σχήματος 4.19, και στις δύο περιπτώσεις οι GRP μονάδες που υιοθετούν την αρχιτεκτονική του merge δικτύου ταξινόμησης είναι ταχύτερες και απαιτούν για όλες τις στοχευόμενες καθυστερήσεις λιγότερη ενέργεια ανά λειτουργία. Το αποτέλεσμα αυτό οφείλεται στο μικρότερο αριθμό στοιχείων σύγκρισης και εναλλαγής που απαιτεί το δίκτυο EMSN έναντι του EBSN. Το γεγονός αυτό οδηγεί επίσης στη μείωση της χωρητικότητας που κάθε κελί είναι αναγκασμένο να οδηγήσει και μειώνει τη συνεισφορά των καλωδίων τόσο στην καθυστέρηση όσο και στην κατανάλωση ισχύος. Από τα διαγράμματα του σχήματος 4.19 προκύπτει πως η ενέργεια που απαιτείται από τα EMSN κυκλώματα είναι από 10 έως 75% μικρότερη σε σχέση με τα EBSN για την ίδια καθυστέρηση.

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στο σχήμα 4.19, η καθυστέρηση της GRP μονάδας βασισμένη στο EMSN μπορεί να κυμανθεί μεταξύ 27 και 35 FO4, ανάλογα με τα

Σχήμα 4.20: Η καμπύλες της ενέργειας ως προς την καθυστέρηση για τις 64×1 μονάδες εκτέλεσης της GRP που υλοποιήθηκαν με βάση το EMSN για διάφορες συνθήκες της τεχνολογίας κατασκευής. (α) $C_{in}/C_{out} = 25/100\text{fF}$ και (β) $C_{in}/C_{out} = 5/25\text{fF}$.

μεγέθη των πυλών που επιλέγονται. Αντίστοιχα, ακολουθώντας την ανάλυση που παρουσιάστηκε στο [62] η πιο αποδοτική έως τώρα υλοποίηση της GRP επιτυγχάνει ελάχιστη καθυστέρηση ίση με 38 FO4. Η καθυστέρηση αυτή υπολογίστηκε θεωρώντας ίση χωρητικότητα εισόδου και εξόδου για το κύκλωμα, όπου για μεγάλες χωρητικότητες εξόδου η συνθήκη αυτή καθίσταται μη ρεαλιστική και επιτυγχάνεται μόνο με την προσθήκη επιπλέον σταδίων αντιστροφών, οι οποίοι αυξάνουν περίπου την καθυστέρηση κατά 1 FO4 ο καθένας. Έτσι, προκύπτει πως η καλύτερη από τις προτεινόμενες 64×1 GRP μονάδες είναι ταχύτερη σε σχέση με τις προηγούμενες λύσεις επιτυγχάνοντας μείωση της καθυστέρησης έως και 28%. Η μείωση αυτή της καθυστέρησης ισχύει ανεξάρτητα από τις συνθήκες λειτουργίας του κυκλώματος. Αυτό φαίνεται από τα διαγράμματα του σχήματος 4.20 όπου παρουσιάζεται η συμπεριφορά του EMSN κυκλώματος κάτω από τις TT, FF και SS καταστάσεις της χρησιμοποιούμενης τεχνολογίας κατασκευής (process corners) θεωρώντας ως τάση τροφοδοσίας τα 1.2V. Σημειώνουμε ότι η καθυστέρηση των κυκλωμάτων είναι κανονικοποιημένη ως προς την αντίστοιχη τιμή του FO4 για την κάθε κατάσταση λειτουργίας σύμφωνα με τις τιμές που δίνονται στο σχήμα 2.2. Οι προτεινόμενες λύσεις αν και πιο γρήγορες από τις προηγούμενες υλοποιήσεις [145], [62] χρειάζεται και πάλι να διασπαστούν σε δύο στάδια pipeline ώστε να επιτύχουν την ταχύτητα των σύγχρονων μικροεπεξεργαστών.

Στις εργασίες [145] και [62] δε δίνεται καμιά πληροφορία όσον αφορά στην απαιτούμενη ενέργεια των προηγούμενων υλοποιήσεων της GRP. Παρόλα αυτά, αναφέρεται ότι οι λύσεις που προτάθηκαν έχουν πολυπλοκότητα αντίστοιχη με 19.7K και 68.6K πύλες NAND των δύο εισόδων αντίστοιχα, χωρίς να αναφέρεται η ενέργεια ή το εμβαδόν της ισοδύναμης πύλης NAND. Στη χρησιμοποιούμενη τεχνολογία μια πύλη NAND με ελάχιστα μεγέθη τρανζίστορ απαιτεί περίπου 3fJ όταν οδηγεί 4 πύλες NAND ίδιου μεγέθους, ενώ αντίστοιχα, μια πύλη NAND με τέσσερις φορές μεγαλύτερο μέγεθος² 11.2fJ ανα μετάβαση. Λαμβάνοντας υπόψη και τις δύο περιπτώσεις

²Σημειώνουμε πως πύλες ελάχιστου μεγέθους χρησιμοποιούνται σπάνια σε πραγματικά κυκλώματα που λειτουργούν σε υψηλές ταχύτητες, γιατί δεν προσφέρουν την αναγκαία οδηγική ικανότητα και μπορεί να

για την πύλη NAND και θεωρώντας για τις προηγούμενες λύσεις την ίδια πιθανότητα μεταβάσεων με τα προτεινόμενα κυκλώματα, η απαίτηση σε ενέργεια των παλαιότερων υλοποιήσεων μπορεί να εκτιμηθεί ως $0.3 \times (3 \text{ ή } 11.2 \text{ fJ}) \times 19.7 \text{ K} = 17.7 \text{ pJ ή } 66.2 \text{ pJ}$. Οφείλουμε να τονίσουμε ότι στους αριθμούς αυτούς δε συμπεριλαμβάνεται η ενέργεια που μεταφέρεται για να (απο)φορτίσει τις χωρητικότητες των καλωδίων, μιας και στις εργασίες [145] και [62] δε δίνεται η απαιτούμενη πληροφορία. Παρόλα αυτά είναι βάσιμο να θεωρήσουμε πως συμπεριλαμβανομένων των καλωδίων η ενέργεια αναμένεται να αυξηθεί σημαντικά και να είναι πιο κοντά στη μεγαλύτερη από τις δύο εκτιμήσεις που υπολογίσαμε.

Όπως προκύπτει από τα δεδομένα του σχήματος 4.20(β), η προτεινόμενη GRP μονάδα που σχεδιάστηκε με βάση το EMSN, απαιτεί στην καλύτερη περίπτωση 57pJ όταν η αντίστοιχη καθυστέρηση του κυκλώματος είναι 35FO4. Μπορούμε να επιτύχουμε μικρότερη κατανάλωση ενέργειας για το κύκλωμα αν μειώσουμε τη μέγιστη επιτρεπόμενη χωρητικότητα εισόδου. Αυτό θα επιφέρει την αύξηση της καθυστέρησης του κυκλώματος. Όμως, η συνολική επιβάρυνση στην καθυστέρηση αναμένεται να είναι λιγότερη, αφού η μικρότερη χωρητικότητα εισόδου επιταχύνει αποτελεσματικά τα κυκλώματα τα οποία συνδέονται στην είσοδο της GRP μονάδας. Πραγματοποιήσαμε δύο πειράματα. Στην περίπτωση όπου η χωρητικότητα εξόδου είναι 100fF μειώσαμε τη χωρητικότητα εισόδου στα 5fF στοχεύοντας καθυστέρηση ίση με 38FO4, όμοια δηλαδή με την ταχύτητα που επιτυγχάνει η καλύτερη προηγούμενη λύση. Το ίδιο κάναμε και για την περίπτωση που η χωρητικότητα εξόδου είναι 25fF, όπου η απαιτούμενη χωρητικότητα εισόδου μειώθηκε στα 2fF. Για τις δύο περιπτώσεις, η κατανάλωση ενέργειας που μετρήσαμε ήταν ίση με 39 και 48pJ αντίστοιχα. Τα αποτελέσματα αυτά μας επιτρέπουν να πούμε με σιγουριά πως, ακόμη και αν δε λάβουμε υπόψη τις χωρητικότητες των καλωδίων, η ενέργεια που απαιτούν τα προτεινόμενα κυκλώματα είναι μικρότερη ή στη χειρότερη περίπτωση περίπου ίδια με αυτήν των προηγούμενων υλοποιήσεων.

Στη συνέχεια μελετήσαμε την καθυστέρηση και την ενέργεια της πιο πρακτικής—από πλευράς εφαρμογών—εκδοχής της GRP (δομή 8×8), σχεδιάζοντας την σύμφωνα με τις τρεις προτεινόμενες δομές. Οι μετρήσεις που πήραμε για την τυπική κατάσταση της τεχνολογίας κατασκευής φαίνονται στο σχήμα 4.21. Για τις μετρήσεις θεωρήσαμε πως οι χωρητικότητες εισόδου και εξόδου είναι ίσες με 25 και 100fF αντίστοιχα. Η GRP μονάδα που σχεδιάστηκε με βάση το merge δίκτυο ταξινόμησης είναι η ταχύτερη λύση και επιτυγχάνει 11FO4 ελάχιστη καθυστέρηση. Η ταχύτητα αυτή είναι περίπου ίδια με αυτήν που επιτυγχάνουν οι ταχύτεροι αθροιστές που παρουσιάστηκαν στο κεφάλαιο 2. Φυσικά, λόγω της διαφορετικής πολυπλοκότητας μεταξύ των κυκλωμάτων αντιμετάθεσης και των παράλληλων αθροιστών, η ενέργεια που καταναλώνει η ταχύτερη 8×8 GRP μονάδα είναι σαφέστατα μεγαλύτερη³. Για μεγαλύτερες καθυστερήσεις (μεγαλύτερες από 12.3FO4) η GRP μονάδα που ακολουθεί τη δομή του γραμμικού δικτύου αντιμετάθεσης, με βάση τη βελτιστοποιημένη εκδοχή που παρουσιάσαμε στην ενότητα 3.8.1, καταναλώνει τη λιγότερη ενέργεια και πρέπει να προτιμηθεί για την υλοποίηση της 8×8 εκδοχής της εντολής GRP, σε περιπτώσεις που η κατανάλωση ενέργειας είναι ο κυριότερος σχεδιαστικός στόχος. Στη γενική περίπτωση πάντως θεωρούμε πως αυτή η μορφή του γραμμικού δικτύου είναι ο καλύτερος σχεδιασμός για την 8×8 GRP μιας και το 1FO4 λιγότερο που πετυχαίνει το EMSN δεν αξίζει το διπλάσιο κόστος σε ενέργεια. Για ακόμη

αλλοιώνουν τη συμπεριφορά στο χρόνο των σημάτων που επεξεργάζονται.

³Η ενέργεια που καταναλώνει ένας αθροιστής 64ων δυαδικών ψηφίων κάτω από τις ίδιες συνθήκες φορτίου είναι περίπου ίση με 24pJ.

Σχήμα 4.21: Οι καμπύλες της ενέργειας και της καθυστέρησης για την υλοποίηση της 8×8 εκδοχής της εντολής GRP όταν αυτή υλοποιηθεί με βάση τα προτεινόμενα επαυξημένα δίκτυα ταξινόμησης.

Σχήμα 4.22: Οι καμπύλες της ενέργειας και της καθυστέρησης για την υλοποίηση της 16×4 εκδοχής της εντολής GRP.

μια φορά, τα κυκλώματα που προκύπτουν με την υιοθέτηση του bitonic δικτύου ταξινόμησης δεν οδηγούν σχεδιαστικά σε συμφέρουσες λύσεις, αφού απαιτούν περισσότερη ενέργεια τόσο από το merge όσο και από το γραμμικό δίκτυο ταξινόμησης, συγκρινόμενα στην ίδια ταχύτητα λειτουργίας. Το ίδιο συμβαίνει και για την απλή εκδοχή του γραμμικού δικτύου αντιμετάθεσης (σχήμα 4.15).

Επίσης, συμπεριλάβαμε στις μετρήσεις μας εκδοχές της προσανατολισμένης σε υπολέξεις GRP μονάδας όπου το ελάχιστο υποστηριζόμενο μέγεθος υπολέξης είναι τα 4 δυαδικά ψηφία.

Το σχήμα 4.22 παρουσιάζει την καθυστέρηση και την ενέργεια όλων των υπο-εξέταση 16×4 GRP μονάδων. Τα αποτελέσματα που πήραμε παρουσιάζουν την ίδια συμπεριφορά με αυτά που πήραμε έως τώρα για τις άλλες εκδοχές της GRP. Σύμφωνα με την ταχύτητα που πετυχαίνει το EMSN, μπορούμε να πούμε ότι μπορεί άνετα να υποστηρίξει τη μετακίνηση 16 ομάδων των 4 δυαδικών ψηφίων σε έναν κύκλο ρολογιού ενός γρήγορου επεξεργαστή χωρίς να απαιτεί το διαχωρισμό του κυκλώματος σε δύο επιμέρους pipeline στάδια. Επίσης, το γραμμικό ETN είναι το αργότερο, εξαιτίας του μεγάλου αριθμού πυλών που εμφανίζονται στο κρίσιμο μονοπάτι. Τέλος, μια αποδοτική μέση λύση, όταν η στοχευόμενη ταχύτητα είναι μεγαλύτερη των 19FO4, είναι η προτεινόμενη υβριδική δομή.

Κυκλώματα μονάδων κινητής υποδιαστολής

Το κεφάλαιο αυτό αποτελεί μια εμβόλιμη εισαγωγή για τη λειτουργία και την οργάνωση των μονάδων κινητής υποδιαστολής, ώστε να γίνει καλύτερα κατανοητός ο τρόπος που χρησιμοποιούνται τα νέα κυκλώματα που θα παρουσιάσουμε στα κεφάλαια 6 και 7.

Οι μονάδες υλοποίησης βασικών πράξεων αριθμών κινητής υποδιαστολής (floating point unit) μονής ή διπλής ακρίβειας αποτελούν ένα βασικό κομμάτι όλων των σύγχρονων μικροεπεξεργαστών και καταλαμβάνουν, εξαιτίας της μεγάλης τους πολυπλοκότητας, ένα σημαντικό κομμάτι του εμβαδού του ολοκληρωμένου. Σε κάθε περίπτωση, για να πετύχουμε έναν συνολικά αποδοτικό σχεδιασμό πρέπει τόσο η καθυστέρηση όσο και η ενέργεια τους να βελτιστοποιηθούν με τον καλύτερο δυνατό τρόπο. Πολλά είναι τα κυκλώματα που έχουν εμφανιστεί στη διεθνή βιβλιογραφία για την υλοποίηση κυκλωμάτων αθροιστών, πολλαπλασιαστών ή διαιρετών αριθμών κινητής υποδιαστολής. Στις μέρες μας, κυρίως εξαιτίας της ανάγκης αποτελεσματικής εκτέλεσης εφαρμογών με γραφικά, η βασική πράξη που εκτελούν οι μονάδες αριθμών κινητής υποδιαστολής είναι ο ταυτόχρονος υπολογισμός ενός γινομένου και ενός αθροίσματος τριών αριθμών $A + B \cdot C$ (Multiply and Add, MAD ή με τη μορφή του Multiply and Accumulate, MAC) [162], [125]. Σε κάθε περίπτωση κάθε κύκλωμα αυτής της μορφής είναι αναγκασμένο κατά τη διάρκεια του υπολογισμού να εκτελέσει μια πράξη πρόσθεσης αριθμών κινητής υποδιαστολής.

Εξαιτίας της μορφής των αριθμών κινητής υποδιαστολής, η πρόσθεση τους είναι μια σύνθετη διαδικασία η οποία αποτελείται από πολλά επιμέρους στάδια. Πολλές είναι οι ερευνητικές προσπάθειες που έχουν γίνει τόσο σε αλγοριθμικό επίπεδο όσο και σε επίπεδο κυκλώματος με σκοπό τη μείωση της καθυστέρησης των κυκλωμάτων πρόσθεσης αριθμών κινητής υποδιαστολής. Παρόλα αυτά, για να καταφέρει η αντίστοιχη μονάδα να λειτουργήσει στις υψηλές συχνότητες λειτουργίας του επεξεργαστή είναι αναγκαίο ο υπολογισμός να χωριστεί σε πολλά επιμέρους pipeline σταδια και να υιοθετηθούν σε πολλές περιπτώσεις επιθετικές κυκλωματικές τεχνικές, όπως αυτή της δυναμικής λογικής [124]. Μια αρκετά

αποδοτική προσέγγιση παρουσιάζεται στην εργασία [141], μαζί με μια αναλυτική περιγραφή των αλγορίθμων που υιοθετούν τα τελευταία χρόνια οι μονάδες κινητής υποδιαστολής των εμπορικών επεξεργαστών. Στα επόμενα κεφάλαια, θα παρουσιάσουμε νέα κυκλώματα τα οποία μπορούν να χρησιμοποιηθούν σε κάποια από τα πιο κρίσιμα στάδια, από πλευράς καθυστέρησης, του κυκλώματος πρόσθεσης αριθμών κινητής υποδιαστολής. Πριν περιγράψουμε τη λειτουργία των προτεινόμενων κυκλωμάτων, θα κάνουμε μια σύντομη αναφορά στην πιο απλοϊκή μορφή του αλγορίθμου πρόσθεσης αριθμών κινητής υποδιαστολής.

5.1 Πρόσθεση αριθμών κινητής υποδιαστολής

Κάθε αριθμός κινητής υποδιαστολής σύμφωνα με το πρότυπο IEEE-754 [2] αναπαρίσταται από τρεις παραμέτρους (s, f, e) , οι οποίοι είναι το πρόσημο του s , το μέγεθος του f (significant) και ο εκθέτης e (exponent). Οι παράμετροι αυτοί συνδέονται μέσω της σχέσης $(-1)^s \times 1.f \times 2^e$, η οποία υπολογίζει την τελική τιμή του αριθμού αυτού. Εφόσον στην αναπαράσταση αυτή το μέγεθος και το πρόσημο του αριθμού κωδικοποιούνται χωριστά, μπορούμε να πούμε, πως η αναπαράσταση κινητής υποδιαστολής ακολουθεί την αναπαράσταση προσημασμένου μεγέθους (sign magnitude). Επίσης, πρέπει να προσέξουμε πως σ' έναν αριθμό κινητής υποδιαστολής το μέγεθος του ακολουθεί τη μορφή $1.f$, δηλαδή ο αριθμός είναι πάντα κανονικοποιημένος (normalized). Λόγω αυτής της σύμβασης, πρέπει το πιο σημαντικό ψηφίο του μεγέθους (με βάρος 2^0) να είναι ίσο με 1. Εφόσον αυτό ισχύει για όλους του αριθμούς κινητής υποδιαστολής, το πιο σημαντικό 1 παραλείπεται από την αναπαράσταση (μόνο η τιμή του f αποθηκεύεται), ενώ συμμετέχει κανονικά στις πράξεις. Σε οποιαδήποτε άλλη περίπτωση ο αριθμός θεωρείται μη κανονικοποιημένος (de-normalized) και αντιμετωπίζεται χωριστά.

Ο πιο απλοϊκός αλγόριθμος πρόσθεσης ή αφαίρεσης μεταξύ δύο αριθμών κινητής υποδιαστολής (S_A, F_A, E_A) και (S_B, F_B, E_B) αποτελείται από τα παρακάτω βήματα [48]:

1. ΑΦΑΙΡΕΣΗ ΕΚΘΕΤΩΝ: Αφαίρεσε τους εκθέτες των δυο αριθμών $z = E_A - E_B$.
2. ΕΥΘΥΓΡΑΜΜΙΣΗ ΜΕΓΕΘΩΝ: Ολίσθησε z θέσεις προς τα δεξιά το μέγεθος του αριθμού με τον μικρότερο εκθέτη. Επέλεξε ως προσωρινό εκθέτη του αποτελέσματος το μεγαλύτερο από τους εκθέτες των δύο αριθμών $E_T = \max\{E_A, E_B\}$.
3. ΕΚΤΕΛΕΣΗ ΠΡΑΞΗΣ ΚΑΙ ΠΡΟΣΗΜΟ ΑΠΟΤΕΛΕΣΜΑΤΟΣ: Η πραγματική λειτουργία που πρέπει να εκτελέσουμε εξαιτίας της αναπαράστασης προσημασμένου μεγέθους καθορίζεται από τη λειτουργία της εντολής κινητής υποδιαστολής (`fpadd`, `fpsub`) και των προσήμων των αριθμών.

Εντολή	Πρόσημο	Πραγματική πράξη
+	Ίδια	Πρόσθεση
+	Διαφορετικά	Αφαίρεση
-	Ίδια	Πρόσθεση
-	Διαφορετικά	Αφαίρεση

Το πρόσημο και το μέγεθος του τελικού αποτελέσματος καθορίζεται από την πραγματική πράξη και τη διαφορά των μεγεθών της εισόδου στην περίπτωση που η πραγματική πράξη είναι αφαίρεση.

4. ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΟΣ: Αν το αποτέλεσμα δεν είναι κανονικοποιημένο επανέφερε το στην κανονικοποιημένη του μορφή.
- Αν η πραγματική πράξη είναι πρόσθεση, τότε πρέπει να ολισθήσουμε το αποτέλεσμα μια θέση στα δεξιά και να αυξήσουμε αντίστοιχα κατά ένα τον προσωρινό εκθέτη E_T .
 - Αντίθετα, αν η πραγματική πράξη είναι αφαίρεση, τότε το μη κανονικοποιημένο αποτέλεσμα έχει τη μορφή 0.00...01XXX και για την κανονικοποίηση του απαιτείται μια ολίσθηση προς τ' αριστερά και η αντίστοιχη ενημέρωση της τιμής του εκθέτη. Σ' αυτήν την περίπτωση, το πλήθος των θέσεων της ολίσθησης δεν είναι σταθερό και εξαρτάται από το πλήθος των μηδενικών που υπάρχουν μπροστά από τον πιο σημαντικό άσσο του μη-κανονικοποιημένου αποτελέσματος.
5. ΣΤΡΟΓΓΥΛΟΠΟΙΗΣΗ: Στρογγυλοποίησε το αποτέλεσμα ανάλογα με τη μορφή της στρογγυλοποίησης που ζητήθηκε, προς την κοντινότερη αναπαράσταση (round to nearest), προς το μηδέν (round to zero) ή προς το άπειρο (round to infinity). Η πράξη αυτή μπορεί να απαιτεί μια επιπλέον πρόσθεση. Αν η στρογγυλοποίηση, λόγω υπερχειλίσσης, καταστρέψει την κανονικοποιημένη μορφή του αποτελέσματος, τότε ολίσθησε το αποτέλεσμα μία θέση προς τα δεξιά και αύξησε αντίστοιχα την προσωρινή τιμή του εκθέτη.
6. ΕΞΑΙΡΕΤΙΚΕΣ ΚΑΤΑΣΤΑΣΕΙΣ: Καθόρισε την αναπαράσταση των ειδικών περιπτώσεων όπως όταν το αποτέλεσμα της πράξης είναι ίσο με μηδέν ή η τιμή του εκθέτη υπερχειλίζει.

Τα βήματα αυτά εξαρτώνται στενά το ένα από το άλλο και δύσκολα μπορούν να εκτελεστούν παράλληλα αν δεν προσθέσουμε επιπλέον υλικό για την εκτέλεση τους. Η χρονική επικάλυψη της εκτέλεσης κάποιων βημάτων μπορεί να επιτευχθεί κάνοντας κάποιες υποθέσεις για το τελικό αποτέλεσμα. Πιο χαρακτηριστικό παράδειγμα αυτής της λειτουργίας, είναι οι αρχιτεκτονικές διπλού μονοπατιού (dual path floating point units) [141], οι οποίες διασπούν τον υπολογισμό σε δύο ανεξάρτητα μονοπάτια, τα οποία παρέχουν υλοποιήσεις με μικρότερη καθυστέρηση σε σχέση με την απευθείας υλοποίηση σε υλικό των βημάτων που περιγράψαμε. Για παράδειγμα, μπορούμε να διασπάσουμε τον υπολογισμό στο κοντινό και στο μακρινό μονοπάτι (near/far path). Το κοντινό μονοπάτι υποστηρίζει την περίπτωση που η τελική πραγματική πράξη είναι αφαίρεση και η διαφορά μεταξύ των εκθετών των αριθμών είναι $\{-1, 0, 1\}$. Εξαιτίας της συγκεκριμένης διαφοράς των εκθετών το βήμα 2 του υπολογισμού δεν απαιτείται, αφού η ευθυγράμμιση των μεγεθών δεν θα ξεπερνά ποτέ τη μία θέση. Όμως, όσον αφορά στο στάδιο της κανονικοποίησης των μεγεθών, το κοντινό μονοπάτι είναι αναγκασμένο να υποστηρίζει την πιο γενική λειτουργία. Αντίθετα, το μακρινό μονοπάτι υποστηρίζει όλες τις άλλες περιπτώσεις. Τότε, η ευθυγράμμιση πρέπει να υποστηριχθεί στην πιο γενική της μορφή, γιατί δεν κάνουμε καμιά εκτίμηση για την διαφορά των εκθετών, ενώ αντίστοιχα η τυχόν κανονικοποίηση που θα χρειαστεί δεν ξεπερνά ποτέ τη μία θέση προς τα δεξιά, μειώνοντας έτσι την ανάγκη για μια ολόκληρη μονάδα κανονικοποίησης. Κατά τη φάση της εκτέλεσης τα δύο μονοπάτια λειτουργούν παράλληλα, ενώ ξεκινούν τον υπολογισμό χωρίς να περιμένουν τον καθορισμό της διαφοράς των εκθετών. Έτσι, στο ένα θα προκύψει το σωστό αποτέλεσμα, ενώ στο άλλο μια λανθασμένη του μορφή. Η επιλογή του σωστού αποτελέσματος πραγματοποιείται ανάλογα με την πράξη που ζητήθηκε και το διάστημα στο οποίο ανήκει η διαφορά των εκθετών των αριθμών.

Μια άλλη προσέγγιση για τη μείωση της καθυστέρησης της πρόσθεσης αριθμών κινητής υποδιαστολής, προσπαθεί να συνενώσει τη φάση της στρογγυλοποίησης με αυτή της εκτέλεσης της πράξης, που πραγματοποιείται στο βήμα 3, επαυξάνοντας τη λειτουργία των αθροιστών [22]. Η τεχνική αυτή έχει βρει μικρή ανταπόκριση από τη βιομηχανία, γιατί αυξάνει αρκετά την καθυστέρηση του αθροιστή μεγεθών που χρησιμοποιείται στο βήμα 3 του βασικού αλγορίθμου.

Η τελική καθυστέρηση όλων των αλγορίθμων που έχουν προταθεί καθορίζεται κυρίως από την καθυστέρηση των βημάτων 2 και 3. Για τα κυκλώματα που υλοποιούν τα βήματα αυτά απαιτείται προσεκτικός σχεδιασμός, ώστε να επιτύχουν χαμηλή καθυστέρηση χωρίς να αυξάνουν την κατανάλωση ισχύος. Τα προτεινόμενα κυκλώματα, τα οποία θα αναλύσουμε στα επόμενα δύο κεφάλαια, χρησιμοποιούνται για την αποτελεσματικότερη υλοποίηση σε υλικό των βημάτων αυτών. Στο κεφάλαιο 6 θα περιγράψουμε νέες δομές υλοποίησης της πρόσθεσης των μεγεθών και τα οφέλη που παρέχουν τόσο σε καθυστέρηση όσο και στη μείωση του εμβαδού χωροθέτησης του κυκλώματος. Οι αθροιστές αυτοί βασίζουν τη λειτουργία τους στην αναπαράσταση συμπληρώματος ως προς ένα, συμβάλλοντας έτσι αφενός στη μείωση της καθυστέρησης του υπολογισμού και αφετέρου στη γρήγορη επαναφορά του τελικού αποτελέσματος στην αναπαράσταση προσημασμένου μεγέθους, που είναι και η ζητούμενη. Αντίστοιχα, στο κεφάλαιο 7 θα αναλύσουμε τα προβλήματα που γεννάει η κανονικοποίηση των αποτελεσμάτων και πώς αυτά αντιμετωπίζονται αποτελεσματικά από τα νέα κυκλώματα που προτείνουμε. Οι προτεινόμενες λύσεις συμβάλλουν αποτελεσματικά στη μείωση τόσο της καθυστέρησης όσο και της κατανάλωσης ισχύος της μονάδας κανονικοποίησης στο σύνολό της.

Αθροιστές μεγεθών

Η αναπαράσταση προσημασμένου μεγέθους που επιβάλλει το πρότυπο IEEE-754 για τους αριθμούς κινητής υποδιαστολής, αναγκάζει το αποτέλεσμα της πρόσθεσης των μεγεθών να είναι πάντα θετικό. Ακόμα και στην περίπτωση που το τελικό αποτέλεσμα είναι αρνητικό $X < 0$, πρέπει να φροντίσουμε να αναπαρασταθεί ως $-|X|$, όπου η πληροφορία του προσήμου περιλαμβάνεται σε ένα επιπλέον δυαδικό ψηφίο, ενώ για την τιμή του μεγέθους υπολογίζουμε την απόλυτη τιμή του. Το πρόβλημα ουσιαστικά προκύπτει μόνο στην περίπτωση που η πραγματική πράξη είναι αφαίρεση, όπου δηλαδή τα πρόσημα των αριθμών της εισόδου είναι διαφορετικά. Στην περίπτωση που η πραγματική πράξη είναι πρόσθεση, τότε δεν έχουμε πρόβλημα, γιατί το αποτέλεσμα που θα προκύψει μετά την πρόσθεση θα είναι πάντα θετικό και το πρόσημό του μπορεί να υπολογιστεί ανεξάρτητα από την πράξη. Έτσι, η πράξη που δυσκολεύει τον υπολογισμό του αθροίσματος των μεγεθών των αριθμών κινητής υποδιαστολής είναι η $|A - B|$, με $A, B > 0$.

Για τον υπολογισμό του $|A - B|$ έχουμε δύο επιλογές. Αν διαλέξουμε να εκτελέσουμε την πράξη θεωρώντας την αναπαράσταση ως προς 2 για τα μεγέθη A και B , τότε πριν την πρόσθεση πρέπει να αντιστρέψουμε τα δυαδικά ψηφία του B και να προσθέσουμε σ' αυτά την τιμή 1. Επίσης, μετά την εκτέλεση της πρόσθεσης, αν το αποτέλεσμα είναι αρνητικό θα πρέπει να το επαναφέρουμε από την αναπαράσταση ως προς 2 και πάλι στην αναπαράσταση προσημασμένου μεγέθους αντιμετωπίζοντας το πρόσημο χωριστά. Η μετατροπή αυτή θα απαιτούσε την υπο-συνθήκη αντιστροφή των δυαδικών ψηφίων του αποτελέσματος και την αύξηση της ανεστραμμένης λέξης κατά 1. Η τελευταία αυτή μετατροπή δεν μπορεί να συγχωνευτεί μέσα στο κύκλωμα που θα εκτελούσε την πράξη αυξάνοντας έτσι σημαντικά την καθυστέρησή του. Για το λόγο αυτό, η αναπαράσταση συμπληρώματος ως προς 2 σπάνια χρησιμοποιείται για την υλοποίηση της πρόσθεσης των μεγεθών στην περίπτωση των μονάδων κινητής υποδιαστολής. Αντ' αυτής είναι πολύ βολικό να χρησιμοποιήσουμε την αναπαράσταση

συμπληρώματος ως προς ένα [141]. Σ' αυτήν την περίπτωση το $-B$ προκύπτει απλά με την αντιστροφή των δυαδικών ψηφίων του B και η απόλυτη τιμή $|A - B|$ που θέλουμε να υπολογίσουμε παίρνει τη μορφή

$$|A - B| = \begin{cases} A + \overline{B} + 1 & \text{αν } A \geq B \\ \overline{A + \overline{B}} & \text{αν } A < B \end{cases}$$

Έτσι, μετά την αντιστροφή των δυαδικών ψηφίων του B , στην περίπτωση που η πραγματική πράξη είναι αφαίρεση, χρειάζεται να υπολογίσουμε δύο αποτελέσματα. Το $A + \overline{B} + 1$ στην περίπτωση που το αποτέλεσμα είναι θετικό $A \geq B$ και το $\overline{A + \overline{B}}$ όταν το αποτέλεσμα είναι αρνητικό $A < B$. Το πρόσημο του αποτελέσματος δηλώνεται πολύ απλά από την τιμή του κρατουμένου εξόδου C_{out} . Σε πολλές υλοποιήσεις παράγονται και τα δύο αποτελέσματα $A + \overline{B} + 1$ και $A + \overline{B}$ όπου όταν $C_{out} = 1$ επιλέγεται το πρώτο ενώ όταν $C_{out} = 0$ επιλέγεται το δεύτερο και ταυτόχρονα αντιστρέφεται. Στην ουσία την επιλογή αυτή την καθορίζει πλήρως το κρατούμενο εξόδου του αθροιστή που υπολογίζει το πιο απλό άθροισμα δηλαδή το $A + \overline{B}$. Στην περίπτωση που το αποτέλεσμα είναι αρνητικό η αντιστροφή δεν είναι αναγκαστικό να πραγματοποιηθεί σε αυτό το στάδιο του υπολογισμού αλλά μπορεί να καθυστερήσει μέχρι τη φάση της στρογγυλοποίησης. Αυτή η βελτιστοποίηση αφορά ταυτόχρονα και στη σχεδίαση της μονάδας κανονικοποίησης και πρέπει να αντιμετωπιστεί από κοινού. Το μόνο “δύσκολο” κομμάτι του αθροιστή μεγεθών παραμένει η υπο-συνθήκη αύξηση ή όχι του αθροίσματος $A + \overline{B}$ ανάλογα με την τιμή του C_{out} που πηγάζει από τη χρήση της αναπαράστασης συμπληρώματος ως προς ένα. Είναι γνωστό πως η πράξη αυτή είναι ισοδύναμη με τη λειτουργία του αθροιστή υπολοίπου ως προς το διαιρέτη $2^n - 1$. Έτσι, στο κεφάλαιο αυτό, θα περιγράψουμε τη δομή και τη λειτουργία αυτών των αθροιστών υπολοίπου και θα αναλύσουμε τις νέες δομές παράλληλου προθέματος που προέκυψαν από την ερευνητική μας προσπάθεια γι' αυτήν την κατηγορία αθροιστών.

6.1 Προηγούμενες υλοποιήσεις

Στη συνέχεια, για λόγους συντομίας, θα χρησιμοποιούμε τον όρο αθροιστές υπολοίπου για να αναφερόμαστε στους αθροιστές υπολοίπου ως προς το διαιρέτη $2^n - 1$ ή αντίστοιχα στους αθροιστές που χρησιμοποιούν την αναπαράσταση του συμπληρώματος ως προς ένα. Πολλές είναι οι προτάσεις που έχουν παρουσιαστεί στη διεθνή βιβλιογραφία για την υλοποίηση των αθροιστών υπολοίπου. Ο Efstathiou et al. στην εργασία [45] πρότεινε μονάδες πρόβλεψης κρατουμένου ενός ή δύο επιπέδων. Παρόλα αυτά, για να πετύχουμε μικρότερες καθυστερήσεις απαιτούνται αθροιστές οι οποίοι θα χρησιμοποιούν τεχνικές παράλληλου προθέματος. Τέτοιες μονάδες έχουν παρουσιαστεί στις εργασίες [176], [77], [19], [10], [41]. Τα κυκλώματα που προτάθηκαν στις εργασίες αυτές έχουν διαφορετικά ποιοτικά χαρακτηριστικά. Οι αθροιστές των [176] και [19] χρησιμοποιούν κλασικούς αθροιστές παράλληλου προθέματος ενώ το κρατούμενο εξόδου επανατροφοδοτείται σε ένα επιπλέον επίπεδο από τελεστές πριν τον υπολογισμό του τελικού αθροίσματος.

Η τεχνική αυτή, αφ' ενός αυξάνει το πλήθος των επιπέδων που απαιτούνται για τον υπολογισμό των κρατουμένων, ενώ από την άλλη αυξάνει σημαντικά τη χωρητικότητα της γραμμής του κρατουμένου εξόδου, εφόσον πρέπει να διαμοιραστεί σε όλες τις στήλες του

Σχήμα 6.1: Η μονάδα υπολογισμού των κρατουμένων για έναν αθροιστή υπολοίπου των 8 δυαδικών ψηφίων σύμφωνα με τη δομή που προτάθηκε από τον (α) Zimmerman (176) και τους Kalampoukas et al. (77).

αθροιστή υπολοίπου. Μια μονάδα υπολογισμού κρατουμένων των 8 δυαδικών ψηφίων για την περίπτωση των αθροιστών υπολοίπου, που ακολουθεί τη δομή που προτάθηκε από τον Zimmerman [176] απεικονίζεται στο σχήμα 6.1(α). Τα προβλήματα αυτά αντιμετωπίστηκαν μερικώς στην εργασία [77], όπου ο υπολογισμός των κρατουμένων στην περίπτωση των αθροιστών υπολοίπου περιελάμβανε την επανατροφοδότηση συγκεκριμένων ομάδων γέννησης και διάδοσης του κρατουμένου εσωτερικά στο δίκτυο παράλληλου προθέματος. Μ' αυτόν τον τρόπο δεν απαιτείται πλέον ένα επιπλέον στάδιο από τελεστές και ο υπολογισμός των κρατουμένων μπορεί να ολοκληρωθεί σε τόσα επίπεδα όσα και ενός απλού αθροιστή παράλληλου προθέματος. Η αντίστοιχη δομή των 8 δυαδικών ψηφίων, που υπολογίζει τα κρατούμενα στην περίπτωση του αθροιστή υπολοίπου και ακολουθεί τη δομή που προτάθηκε στην εργασία [77], φαίνεται στο σχήμα 6.1(β). Παρόμοιοι αθροιστές παρουσιάστηκαν στην εργασία [10] στοχεύοντας σε υλοποιήσεις όπου οι τελεστές έχουν περισσότερες από δύο εισόδους. Αν και στην εργασία [77] παρουσιάστηκε η βασική θεωρία της νέας μεθόδου, δε δόθηκε καμιά διαδικασία σχεδίασης η οποία να καλύπτει την περίπτωση που το πλήθος των δυαδικών ψηφίων της εισόδου δεν είναι δύναμη του 2. Φυσικά, στην περίπτωση μας κάτι τέτοιο είναι εξαιρετικά αναγκαίο, εφόσον το μέγεθος των αριθμών κινητής υποδιαστολής είναι 24 (23+1) και 53 (52+1) δυαδικά ψηφία στην περίπτωση της μονής και της διπλής ακρίβειας αντίστοιχα. Μια πρώτη προσπάθεια για τη γενίκευση της μεθόδου σε οποιοδήποτε μήκος λέξης παρουσιάστηκε στην εργασία [41].

6.2 Βασική θεωρία και επεκτάσεις

Στην ενότητα αυτή θα περιγράψουμε τη βασική θεωρία υλοποίησης των αθροιστών υπολοίπου με τεχνικές παράλληλου προθέματος. Στη συνέχεια, θα επεκτείνουμε τις τεχνικές αυτές με δύο τρόπους. Πρώτον, θα δείξουμε πως μπορούν τα κυκλώματα αυτά να επιταχυνθούν υιοθετώντας τα απλοποιημένα κρατούμενα του Ling και δεύτερον, θα παρουσιάσουμε μια συγκεντρωτική σχεδιαστική μεθοδολογία η οποία επιτρέπει τον καθορισμό μιας νέας οικογένειας από δένδρα παράλληλου προθέματος κατάλληλα για τον υπολογισμό των κρατουμένων στην περίπτωση των αθροιστών υπολοίπου.

Όπως αποδείχθηκε στην εργασία [77], κάθε κρατούμενο C_i^- στην περίπτωση των αθροιστών υπολοίπου ισούται με

$$C_i^- = G_{i:0} + P_{i:0} G_{n-1:i+1} \quad (6.1)$$

και μπορεί να υπολογιστεί με την τεχνική παράλληλου προθέματος ως εξής¹

$$C_i^- \leftrightarrow (G_i, P_i) \circ (G_{i-1}, P_{i-1}) \circ \dots \circ (G_0, P_0) \circ (G_{n-1}, P_{n-1}) \circ \dots \circ (G_{i+1}, P_{i+1}). \quad (6.2)$$

Όπως φαίνεται από τη σχέση (6.2), όλα τα κρατούμενα στην περίπτωση των αθροιστών υπολοίπου απαιτούν το συνδυασμό n όρων γέννησης και διάδοσης κρατουμένου (G_k, P_k) . Αντίθετα, στην περίπτωση της απλής πρόσθεσης, κάθε κρατούμενο που παράγεται από την i -οστή θέση συνδιάζει i ζεύγη γέννησης και διάδοσης κρατουμένου.

Εξαιτίας της μορφή της σχέσης (6.2), που περιγράφει τα κρατούμενα για τους αθροιστές υπολοίπου, ο ορισμός των ομάδων γέννησης και διάδοσης κρατουμένου $(G_{k:j}, P_{k:j})$ πρέπει να γενικευτεί και στην περίπτωση που $k < j$, $0 \leq k, j \leq n-1$

$$(G_{k:j}, P_{k:j}) = (G_{k:0}, P_{k:0}) \circ (G_{n-1:j}, P_{n-1:j}). \quad (6.3)$$

Έτσι, στη γενική περιπτώσή ($k > j$ ή $k < j$), το πλήθος των όρων γέννησης και διάδοσης κρατουμένου που συμμετέχουν στην αντίστοιχη ομάδα $(G_{k:j}, P_{k:j})$ είναι ίσος με $(k - j + 1) \bmod n$. Χρησιμοποιώντας αυτήν την παραδοχή μπορούμε να ορίσουμε πιο γενικά τη σχέση (6.2) που περιγράφει κάθε κρατούμενο C_i^- , αν θεωρήσουμε έναν ενδιάμεσο δείκτη k , $0 \leq k \leq n-1$.

$$C_i^- \leftrightarrow (G_{i:k}, P_{i:k}) \circ (G_{k-1:i+1}, P_{k-1:i+1}). \quad (6.4)$$

Στην περίπτωση της απλής πρόσθεσης, η απορροφητική ιδιότητα του τελεστή $\circ$ μας επέτρεπε να συνδιάσουμε ομάδες γέννησης και διάδοσης κρατουμένου οι οποίες μερικώς επικαλύπτονταν. Τί θα συνέβαινε όμως στην περίπτωση της πρόσθεσης υπολοίπου όπου δύο ομάδες μπορούν να αποτελούνται από κοινούς όρους σε πολλά σημεία; Το ερώτημα αυτό θα απαντηθεί αποδεικνύοντας την κυκλική μορφή της απορροφητικής ιδιότητας για την περίπτωση των αθροιστών υπολοίπου μέσω του θεωρήματος 1.

Θεώρημα 1. Έστω ότι

$$(G_{i:k}, P_{i:k}) = \begin{cases} (g_i, p_i) \circ (g_{i-1}, p_{i-1}) \circ \dots \circ (g_k, p_k), & \text{αν } i \geq k \\ (g_i, p_i) \circ (g_{i-1}, p_{i-1}) \circ \dots \circ (g_0, p_0) \circ (g_{n-1}, p_{n-1}) \circ \dots \circ (g_k, p_k), & \text{αν } i < k \end{cases}.$$

Τότε ισχύει πως $(G_{i:k}, P_{i:k}) \circ (G_{k-1:i+1}, P_{k-1:i+1}) \circ (G_{i:k}, P_{i:k}) = (G_{i:k}, P_{i:k}) \circ (G_{k-1:i+1}, P_{k-1:i+1})$.

Απόδειξη. Θεωρούμε ότι $(G_{i:k}, P_{i:k}) \circ (G_{k-1:i+1}, P_{k-1:i+1}) \circ (G_{i:k}, P_{i:k}) = T$. Ανοίγοντας πλήρως τον ορισμό του τελεστή $\circ$ προκύπτει ότι

$$\begin{aligned} T &= (G_{i:k} + P_{i:k} \cdot G_{k-1:i+1}, P_{i:k} \cdot P_{k-1:i+1}) \circ (G_{i:k}, P_{i:k}) \\ &= (G_{i:k} + P_{i:k} \cdot G_{k-1:i+1} + P_{i:k} \cdot P_{k-1:i+1} \cdot G_{i:k}, P_{i:k} \cdot P_{k-1:i+1} \cdot P_{i:k}) \end{aligned}$$

¹Για να αποφύγουμε τη διπλή αναπαράσταση του μηδενός που επιβάλλει η αναπαράσταση συμπληρώματος ως προς ένα, χωρίς να αλλάξουμε την πολυπλοκότητα του υπολογισμού, μπορούμε να αντικαταστήσουμε στη σχέση τον τελευταίο όρο (G_{i+1}, P_{i+1}) με (P_{i+1}, P_{i+1}) .

$$\begin{array}{c}
(G_{4:1}, P_{4:1}) \circ (G_{1:3}, P_{1:3}) \\
\downarrow \\
\overbrace{(g_4, p_4) \circ (g_3, p_3)} \circ (g_2, p_2) \circ \overbrace{(g_1, p_1) \circ (g_1, p_1)} \circ (g_0, p_0) \circ \overbrace{(g_4, p_4) \circ (g_3, p_3)} \\
\downarrow \\
(g_4, p_4) \circ (g_3, p_3) \circ (g_2, p_2) \circ (g_1, p_1) \circ (g_0, p_0)
\end{array}$$

Σχήμα 6.2: Ο κλασικός και ο κυκλικός ορισμός της απορροφητικής ιδιότητας.

Στην αριστερή πλευρά της παρένθεσης οι δύο όροι $G_{i:k}$ και $P_{i:k} \cdot P_{k-1:i+1} \cdot G_{i:k}$ είναι περιττοί και το λογικό τους OR ισούται με $G_{i:k}$. Αντίστοιχα στο δεξί μέρος της παρένθεσης απλοποιούμε το $P_{i:k} \cdot P_{k-1:i+1} \cdot P_{i:k}$ σε $P_{i:k} \cdot P_{k-1:i+1}$. Μετά από αυτές τις απλοποιήσεις ο όρος T γράφεται ως $T = (G_{i:k} + P_{i:k} \cdot G_{k-1:i+1}, P_{i:k} \cdot P_{k-1:i+1})$. Η σχέση αυτή αποτελεί ακριβώς τον ορισμό του τελεστή $\circ$ και επομένως μπορεί να γραφεί με ένα νέο τρόπο, ολοκληρώνοντας έτσι την απόδειξη.

$$T = (G_{i:k}, P_{i:k}) \circ (G_{k-1:i+1}, P_{k-1:i+1})$$

□

Το θεώρημα 1 μας επιτρέπει να απλοποιήσουμε οποιουσδήποτε όρους τυχόν προκύψουν που αποτελούνται από περισσότερους από n όρους γέννησης και διάδοσης κρατουμένου, σε όρους με μήκος ακριβώς ίσο με n . Για παράδειγμα, ας θεωρήσουμε την περίπτωση ενός αθροιστή υπολοίπου των 5 δυαδικών ψηφίων. Ο συνδιασμός των όρων $(G_{4:1}, P_{4:1})$ (μήκους 4) και $(G_{1:3}, P_{1:3})$ (μήκους 3) αναμένεται να οδηγήσει σε έναν όρο μεγέθους 7 επειδή σύμφωνα με τον κλασικό ορισμό της απορροφητικής ιδιότητας μόνο ο επικαλυπτόμενος όρος (G_1, P_1) μπορεί να απλοποιηθεί. Όμως, βάσει του θεωρήματος 1, ο τελικός όρος που προκύπτει είναι ο $(G_{4:0}, P_{4:0})$, ο οποίος έχει μήκος 5.

$$\begin{aligned}
(G_{4:1}, P_{4:1}) \circ (G_{1:3}, P_{1:3}) &= (G_{4:3}, P_{4:3}) \circ (G_{2:1}, P_{2:1}) \circ (G_{1:0}, P_{1:0}) \circ (G_{4:3}, P_{4:3}) \\
&= (G_{4:3}, P_{4:3}) \circ (G_{2:0}, P_{2:0}) \circ (G_{4:3}, P_{4:3}) \\
&= (G_{4:3}, P_{4:3}) \circ (G_{2:0}, P_{2:0}) = (G_{4:0}, P_{4:0}).
\end{aligned}$$

Το θεώρημα 1 είναι μια γενίκευση της βασικής απορροφητικής ιδιότητας που παρουσιάστηκε στην εργασία [106]. Η βασική θεώρηση ότι δύο ομάδες γέννησης και διάδοσης κρατουμένου για να συνδυαστούν δεν είναι απαγορευτικό να επικαλύπτονται, έχει επεκταθεί και στην περίπτωση που η επικάλυψη αυτή παίρνει κυκλική μορφή. Ένα παράδειγμα της κυκλικής επικάλυψης απεικονίζεται στο σχήμα (6.2) για την περίπτωση των ομάδων $(G_{4:1}, P_{4:1})$ και $(G_{1:3}, P_{1:3})$.

6.3 Σχεδιαστική μεθοδολογία

Στην ενότητα αυτή θα παρουσιάσουμε μια συστηματοποιημένη μεθοδολογία για το σχεδιασμό αθροιστών υπολοίπου παράλληλου προθέματος. Όλοι οι αθροιστές που μπορούν να δημιουργηθούν από την προτεινόμενη μεθοδολογία έχουν ελάχιστο λογικό βάθος $m = \lceil \log_2 n \rceil$ και επιτρέπεται για κάθε n η δημιουργία δένδρων παράλληλου προθέματος με διαφορετικές

Σχήμα 6.3: (α) Ο γράφος εξάρτησης μεγέθους LDG_4 που αφορά υλοποιήσεις με ελάχιστο βάθος τα 4 επίπεδα παράλληλου προθέματος και (β) ο αντίστοιχος γράφος σχεδίασης $DG_{10,\{8,5\}}$.

ιδιότητες όσον αφορά στο πλήθος των τελεστών και στο βαθμό οδήγησης των εσωτερικών γραμμών του κυκλώματος.

Σύμφωνα με την (6.1), το μήκος κάθε σχέσης κρατουμένου στην περίπτωση των αθροιστών υπολοίπου είναι ίσο με n . Έτσι, ενδιαφερόμαστε να παράγουμε εκείνες τις ομάδες γέννησης και διάδοσης κρατουμένου μέχρι και το επίπεδο $m - 1$, οι οποίες θα μας επιτρέψουν συνδυάζοντάς τις να υπολογίσουμε στο τελευταίο επίπεδο ομάδες μεγέθους μεγαλύτερου ή ίσου του n . Σε οποιαδήποτε άλλη περίπτωση η δημιουργία ενός έγκυρου κρατουμένου είναι αδύνατη. Ας υποθέσουμε ότι δύο οποιεσδήποτε ομάδες γέννησης και διάδοσης κρατουμένου με μήκος S_a και S_b αντίστοιχα, παράγονται στα πρώτα $m - 1$ επίπεδα του δένδρου παράλληλου προθέματος και επιλέγονται για να παράγουν μια σχέση κρατουμένου στο επίπεδο m . Τότε τα επιλεγμένα μεγέθη S_a και S_b πρέπει πάντα να ικανοποιούν τη σχέση (6.5),

$$S_a + S_b \geq n. \quad (6.5)$$

Ο τρόπος με τον οποίο μπορούμε να παράγουμε όλες τις δυνατές σχέσεις διάδοσης και γέννησης κρατουμένου στα πρώτα $m - 1$ επίπεδα μπορεί να αναπαρασταθεί γραφικά μέσω του γράφου εξάρτησης μεγέθους (Length Dependency Graph, LDG). Ο LDG είναι ο ίδιος για όλες τις τιμές του n που δίνουν το ίδιο ελάχιστο λογικό βάθος $m = \lceil \log_2 n \rceil$ και, για το λόγο αυτό, αναπαρίσταται ως LDG_m . Ο LDG_m είναι οργανωμένος σε m επίπεδα και περιέχει έναν κόμβο για κάθε ένα δυνατό μέγεθος των όρων γέννησης και διάδοσης κρατουμένου που μπορεί να παραχθεί στα πρώτα $m - 1$ επίπεδα της μονάδας υπολογισμού κρατουμένων. Ο αριθμός μέσα σε κάθε κόμβο συμβολίζει το μέγεθος που αναπαριστά. Στο επίπεδο 0 υπάρχει μόνο ένας κόμβος με τιμή 1 που αναπαριστά τις σχέσεις μεγέθους 1, δηλαδή τα αρχικά δυαδικά ψηφία γέννησης και διάδοσης κρατουμένου (G, P) , ενώ σε κάθε επόμενο επίπεδο i , $1 \leq i \leq m - 1$ ο LDG_m αποτελείται από 2^{i-1} κόμβους. Για παράδειγμα, ο LDG_4 έχει σχεδιαστεί στο σχήμα 6.3(α) και περιέχει όλα τα δυνατά μήκη των όρων γέννησης και διάδοσης κρατουμένου που μπορούν να παραχθούν στα 3 πρώτα επίπεδα του δένδρου παράλληλου προθέματος.

Τα βάρη στις ακμές του LDG_m περιγράφουν ποιά από τα δυνατά μήκη των υπο-εξέταση όρων μπορούν να συνδιαστούν για να παράγουν το ζητούμενο όρο γέννησης και διάδοσης κρατουμένου. Λόγου χάρη, η ακμή $(4) \rightarrow (6)$ με βάρη $\{2, 3, 4\}$ υποδηλώνει πως μια σχέση μεγέθους 6 μπορεί να υπολογιστεί στο τρίτο επίπεδο του δένδρου παράλληλου προθέματος

συνδυάζοντας έναν όρο μεγέθους 4 του δευτέρου επιπέδου με τον αντίστοιχο όρο με μέγεθος είτε 2 είτε 3 ή 4 αντίστοιχα. Σημειώνουμε πως ο συνδιασμός των όρων μεγέθους 4, 3, και 4, απαιτούν τη χρήση της απορροφητικής ιδιότητας ώστε να παράγουν τη ζητούμενη σχέση μεγέθους 6. Έτσι ακολουθώντας τις συνδέσεις του LDG_m και διαλέγοντας αυτά τα μεγέθη των σχέσεων $\{S_a, S_b\}$ που ικανοποιούν τη συνθήκη (6.5), μπορούμε να σχεδιάσουμε οποιονδήποτε αθροιστή υπολοίπου των n δυαδικών ψηφίων. Για να απλοποιήσουμε τη διαδικασία για κάθε επιλεγμένο ζεύγος $\{S_a, S_b\}$ εξάγουμε από τον LDG_m τον αντίστοιχο γράφο σχεδίασης (Design Graph, $DG_{n,\{S_a, S_b\}}$). Ο Design Graph, $DG_{n,\{S_a, S_b\}}$ είναι ένας υπογράφος του LDG_m και δημιουργείται αν ακολουθήσουμε με αντίστροφη φορά τις ακμές του LDG_m που ξεκινούν από τους κόμβους με τιμές S_a και S_b και διατρέξουμε τους αντίστοιχους ενδιάμεσους κόμβους φτάνοντας στον κόμβο με τιμή 1. Οι κόμβοι τους οποίους δεν επισκεφθήκαμε στη διαδρομή εξαιρούνται και δεν υπάρχουν στο $DG_{n,\{S_a, S_b\}}$. Για παράδειγμα, ο $DG_{10,\{8,5\}}$, στην περίπτωση του αθροιστή υπολοίπου $2^{10} - 1$, προέρχεται από τον LDG_4 του σχήματος 6.3(α), και παρουσιάζεται στο σχήμα 6.3(β). Από το σχήμα 6.3(β) αποδεικνύεται ότι η κατασκευή όρων με μεγέθη 8 και 5 στο τρίτο επίπεδο του δένδρου παράλληλου προθέματος αφήνει πολλές επιλογές στο σχεδιαστή ειδικά για την περίπτωση των όρων μεγέθους 5. Σημειώνουμε ότι για να παράγουμε όλα τα κρατούμενα στο τέταρτο επίπεδο του δένδρου παράλληλου προθέματος απαιτείται η δημιουργία όλων των δυνατών προθεμάτων (g_i, p_i) , $i = 0, 1, \dots, 9$, τα οποία θα προκύψουν από τον κατάλληλο συνδυασμό όρων μεγέθους 8 και 5 αντίστοιχα.

Στη γενική περίπτωση, ακόμα και αν η επιλογή των μεγεθών $\{S_a, S_b\}$ είναι έγκυρη, ο σχεδιαστής μένει και πάλι με μια πληθώρα επιλογών για τη σχεδίαση των αθροιστών υπολοίπου. Βάσει του LDG_m μπορούμε να παράγουμε εξαντλητικά όλες τις δυνατές λύσεις που αφορούν στη σχεδίαση ενός αθροιστή υπολοίπου των n δυαδικών ψηφίων και να επιλέξουμε αυτήν που ανταποκρίνεται καλύτερα στους περιορισμούς που έχουμε θέσει. Εφόσον όμως ο χώρος λύσεων είναι πολύ μεγάλος για μεγάλες τιμές του n , θα παρουσιάσουμε στη συνέχεια κάποιους σχεδιαστικούς περιορισμούς που μειώνουν τη σχεδιαστική προσπάθεια και κατευθύνουν το σχεδιαστή στην εύρεση αποδοτικών λύσεων.

6.3.1 Μείωση σχεδιαστικών επιλογών

Σύμφωνα με την προτεινόμενη μεθοδολογία, τα κρατούμενα των άρτιων (0,2,4,...) και των περιττών (1,3,5,...) θέσεων του αθροιστή υπολοίπου αντιμετωπίζονται χωριστά. Πιο συγκεκριμένα, επιβάλλουμε σε όλες τις ομάδες γέννησης και διάδοσης κρατουμένου που παράγονται στις άρτιες θέσεις του i -οστού επιπέδου του δένδρου παράλληλου προθέματος να έχουν όλες το ίδιο μέγεθος το οποίο συμβολίζεται ως $L_{\text{even}}(i)$. Το ίδιο συμβαίνει και με τις σχέσεις που υπολογίζονται στις περιττές θέσεις και το αντίστοιχο μέγεθος των σχέσεων συμβολίζεται ως $L_{\text{odd}}(i)$. Οι κανόνες μείωσης που θα περιγράψουμε, αφορούν στην παραγωγή των σχέσεων μέχρι το επίπεδο $m - 1$ της μονάδας υπολογισμού του κρατουμένου, ενώ οι συνδέσεις στο τελευταίο επίπεδο αντιμετωπίζονται χωριστά. Οι τελικές σχέσεις κρατουμένου παράγονται στο επίπεδο m , συνδιάζοντας τους όρους που προυπολογίστηκαν στις άρτιες και στις περιττές θέσεις του δένδρου παράλληλου προθέματος, ακολουθώντας τη βασική σχέση (6.1).

Κανόνες μείωσης για τις άρτιες θέσεις δυαδικών ψηφίων

Ε1. Στις $\left\lceil \frac{n}{2} \right\rceil$ άρτιες θέσεις του δένδρου παράλληλου προθέματος παράγονται μόνο σχέσεις

άρτιου μεγέθους.

- E2. Σε κάθε επίπεδο i πριν το $m - 1$ παράγουμε σχέσεις μεγέθους 2^i συνενώνοντας όρους μεγέθους 2^{i-1} από το προηγούμενο επίπεδο.
- E3. Στο επίπεδο $m - 1$ συνδυάζουμε τις σχέσεις μεγέθους 2^{m-2} του προηγούμενου επιπέδου που δημιουργήθηκαν στις άρτιες θέσεις για να παράγουμε τη σχέση του ζητούμενου μεγέθους είτε χρησιμοποιώντας την απορροφητική ιδιότητα είτε όχι. Για παράδειγμα η δημιουργία μιας σχέση μεγέθους 6 στο τρίτο επίπεδο του δένδρου επιβάλλει τη συνένωση δύο σχέσεων μεγέθους 4 που δημιουργήθηκαν στο δεύτερο επίπεδο.

Κανόνες μείωσης για τις περιττές θέσεις δυαδικών ψηφίων

- O1. Στις $\lfloor \frac{n}{2} \rfloor$ περιττές θέσεις του δένδρου παράλληλου προθέματος παράγονται μόνο σχέσεις περιττού μεγέθους.
- O2. Οι σχέσεις περιττού μήκους, το οποίο συμβολίζουμε με $L_{\text{odd}}(i)$, δημιουργούνται στο i -οστό επίπεδο του δένδρου με το συνδιασμό μιας σχέσης μήκους 2^{i-1} του επιπέδου $i - 1$ και μιας σχέσης μήκους $L_{\text{odd}}(i) - 2^{i-1}$, η οποία έχει παραχθεί σε ένα προηγούμενο επίπεδο k , $k < i$, όπου $k = \lceil \log_2 (L_{\text{odd}}(i) - 2^{i-1}) \rceil$. Ο κανόνας αυτός περιγράφει ότι οι τελεστές, οι οποίοι τοποθετούνται στο i -οστό επίπεδο στις περιττές θέσεις του δένδρου παράλληλου προθέματος, συνδυάζουν σχέσεις οι οποίες παράγονται σε μια άρτια και σε μια περιττή θέση των προηγούμενων επιπέδων αντίστοιχα.

Οι κανόνες μείωσης E.2 και O.2 καθορίζουν επακριβώς τον τρόπο με τον οποίο θα υπολογιστούν οι απαραίτητες σχέσεις τόσο στις άρτιες όσο και στις περιττές θέσεις του δένδρου. Η εφαρμογή των κανόνων αυτών στο γράφο σχεδίασης πραγματοποιείται από το τέλος προς την αρχή, ξεκινώντας από επίπεδο $(m - 1)$ και φτάνοντας στο πρώτο επίπεδο. Η διαδρομή που θα ακολουθήσουμε, καθορίζει το μέγεθος των ενδιάμεσων σχέσεων που πρέπει να παραχθούν στα αντίστοιχα επίπεδα του δένδρου παράλληλου προθέματος.

Αντιμετωπίζοντας χωριστά τις σχέσεις που μπορούν να παραχθούν στις άρτιες και στις περιττές θέσεις του δένδρου παράλληλου προθέματος, και εφαρμόζοντας τους κανόνες μείωσης, μπορούμε να προσδιορίσουμε ένα υποσύνολο των λύσεων που περιγράφονται από το LDG_m . Εφαρμόζοντας τους κανόνες μείωσης στο γράφο LDG_m , παράγουμε έναν απλοποιημένο γράφο εξάρτησης των μεγεθών ο οποίος συμβολίζεται ως SLDG_m . Ο αντίστοιχος SLDG_4 φαίνεται στο σχήμα 6.4(α). Οι κόμβοι του SLDG_m χωρίζονται σε δύο ομάδες. Στη μια ομάδα V_{even} ανήκουν οι κόμβοι με άρτιες τιμές, ενώ στην άλλη V_{odd} οι κόμβοι με περιττές τιμές, οι οποίοι αντιστοιχούν στις περιττού μήκους σχέσεις που μπορούν να παραχθούν στο υπο-σχεδίαση δένδρο παράλληλου προθέματος. Όμοια με τα S_a και S_b , ορίζουμε τα $S_{\text{even}} \in L_{\text{even}}(i)$ και $S_{\text{odd}} \in L_{\text{odd}}(i)$, $1 \leq i \leq m - 1$, που συμβολίζουν τα μήκη των όρων που επιλέγονται από τις άρτιες και τις περιττές θέσεις αντίστοιχα για να ολοκληρώσουν τον υπολογισμό του κρατουμένου. Επίσης, ορίζουμε μια νέα μεταβλητή που ονομάζουμε d_n και παίρνει τις εξής τιμές:

$$d_n = \begin{cases} n + 1, & \text{όταν το } n \text{ περιττό} \\ n, & \text{όταν το } n \text{ άρτιο} \end{cases}. \quad (6.6)$$

Σχήμα 6.4: (α) Η απλοποιημένη εκδοχή του γράφου εξάρτησης μεγέθους $SLDG_4$ που αφορά υλοποιήσεις με ελάχιστο βάθος τα 4 επίπεδα παράλληλου προθέματος και (β) ο αντίστοιχος απλοποιημένος γράφος σχεδίασης $SDG_{10, \{8,5\}}$.

Σύμφωνα με τη σχέση (6.5) οι τιμές των S_{even} και S_{odd} πρέπει να ικανοποιούν τη συνθήκη (6.7).

$$S_{\text{even}} + S_{\text{odd}} \geq d_n. \quad (6.7)$$

Η μεταβλητή d_n χρησιμοποιείται για να υποδηλώσει ότι, στην περίπτωση που το n είναι περιττός αριθμός, οφείλουμε να εμφανίσουμε ένα πιο αυστηρό όριο όσον αφορά στο μήκος των σχέσεων που θα συνδυαστούν για να υπολογίσουν το τελικό κρατούμενο. Έτσι για κάθε ζεύγος $\{S_{\text{even}}, S_{\text{odd}}\}$ που ικανοποιεί τη συνθήκη (6.7) παράγουμε έναν απλοποιημένο γράφο σχεδίασης ($SDG_{n, \{S_{\text{even}}, S_{\text{odd}}\}}$) από τον $SLDG_m$. Ο $SDG_{10, \{8,5\}}$ που εξάγουμε από τον $SLDG_4$ παρουσιάζεται στο σχήμα 6.4(β). Ο $SDG_{10, \{8,5\}}$ επιτρέπει τη σχεδίαση ενός αθροιστή υπολοίπου των 10 δυαδικών ψηφίων με έναν απλό τρόπο και είναι λιγότερο περίπλοκος από τον αντίστοιχο γράφο σχεδίασης $DG_{10, \{8,5\}}$ του σχήματος 6.3(β), αφού πολλές από τις πιθανές λύσεις έχουν απαλειφθεί με την εφαρμογή των κανόνων μείωσης που περιγράψαμε.

6.3.2 Διαδικασία σχεδίασης

Μετά τη δημιουργία του $SDG_{n, \{S_{\text{even}}, S_{\text{odd}}\}}$, η διαδικασία σχεδίασης περιλαμβάνει 4 επιπλέον βήματα, στα οποία περιγράφεται πλήρως η κατασκευή του δένδρου παράλληλου προθέματος και ο τρόπος ένωσης στο τελευταίο επίπεδο των ενδιάμεσων σχέσεων για την παραγωγή των τελικών κρατουμένων C_i^- όπως περιγράφονται από τη σχέση (6.1).

Βήμα 1: Ορισμοί

Θέτουμε $m = \lceil \log_2 n \rceil$, $\beta = \min\{S_{\text{even}}, S_{\text{odd}}\}$, και $\gamma = S_{\text{even}} + S_{\text{odd}}$. Με $L(u)$ συμβολίζουμε την τιμή του κόμβου u στο γράφο $SDG_{n, \{S_{\text{even}}, S_{\text{odd}}\}}$.

Βήμα 2: Πρώτο επίπεδο του δένδρου παράλληλου προθέματος

Τοποθετήσε $\lceil \frac{n}{2} \rceil$ τελεστές στις άρτιες θέσεις του πρώτου επιπέδου του δένδρου παράλληλου προθέματος. Κάθε τελεστής $(1, j)$ με $j \in \{0, 2, \dots, d_n - 2\}$ συνδέεται με τους κόμβους $(0, j)$ και $(0, (j - 1) \bmod n)$. Όπου δεν έχω τοποθετήσει τελεστές, τότε οι αντίστοιχοι κόμβοι $(1, j)$ συνδέονται μόνο με τον αμέσως από πάνω τους $(0, j)$.

Βήμα 3: Επόμενα $m - 2$ επίπεδα

Εξέτασε το i -οστό επίπεδο του $\text{SDG}_{n, \{S_{\text{even}}, S_{\text{odd}}\}}$.

- Αν υπάρχει κόμβος u ο οποίος να ανήκει στο V_{even} , τότε τοποθέτησε $\lfloor \frac{n}{2} \rfloor$ τελεστές στις άρτιες θέσεις του i -οστού επιπέδου. Κάθε τελεστής (i, j) , με $j \in \{0, 2, \dots, d_n - 2\}$ συνδέεται με τους τελεστές $(i - 1, j)$ και $(i - 1, (j - L(u) + 2^{i-1}) \bmod d_n)$.
- Αν υπάρχει κόμβος u ο οποίος να ανήκει στο V_{odd} , τότε τοποθέτησε $\lfloor \frac{n}{2} \rfloor$ τελεστές στις περιττές θέσεις του i -οστού επιπέδου. Κάθε τελεστής (i, j) με $j \in \{1, 3, \dots, d_n - 3\}$ συνδέεται με τους τελεστές $(i - 1, j)$ και $(i - 1, (j - L(u) + 2^{i-1}) \bmod d_n)$. Επιπλέον, αν n άρτιο πρόσθεσε τον τελεστή $(i, d_n - 1)$ και συνέδεσέ τον με τους τελεστές $(i - 1, d_n - 1)$ και $(i - 1, (2^{i-1} - L(u) - 1) \bmod d_n)$, αντίστοιχα.

Βήμα 4: Συνδέσεις του τελευταίου επιπέδου

Το τελευταίο επίπεδο του δένδρου παράλληλου προθέματος κατασκευάζεται ως εξής:

- Κάθε τελεστής (m, j) με $j \in \{0, 2, \dots, d_n - 2\}$ δέχεται είσοδο από τους τελεστές $(m - 1, j)$ και $(m - 1, (j - \beta + \gamma) \bmod d_n)$.
- Κάθε τελεστής (m, j) με $j \in \{1, 3, \dots, d_n - 3\}$ δέχεται είσοδο από τους τελεστές $(m - 1, j)$ και $(m - 1, (j - 1 - \beta + \gamma) \bmod d_n)$. Επιπλέον, αν n άρτιο πρόσθεσε τον τελεστή $(m, d_n - 1)$ και συνέδεσέ τον με τους τελεστές $(m - 1, d_n - 1)$ και $(m - 1, (\gamma - \beta - 2) \bmod d_n)$ του προηγούμενου επιπέδου αντίστοιχα.

Οι αθροιστές υπολοίπου των δέκα δυαδικών ψηφίων που σχεδιάζονται ακολουθώντας την προτεινόμενη μεθοδολογία για διάφορες τιμές των $\{S_{\text{even}}, S_{\text{odd}}\}$ φαίνονται στο σχήμα 6.5 μαζί με τους αντίστοιχους γράφους SLDG_4 που καθόρισαν τη σχεδιάσή τους. Μπορούμε εύκολα να παρατηρήσουμε πως οι διαφορετικές λύσεις που πήραμε αναμένεται να έχουν διαφορετική συμπεριφορά τόσο όσον αφορά στην καθυστέρηση όσο και στην κατανάλωση ενέργειας. Αυτό οφείλεται στο διαφορετικό πλήθος τελεστών που απαιτούνται και στη διαφορετική πολυπλοκότητα των καλωδίων. Στο μέλλον σκοπός μας είναι να επαυξήσουμε τη σχεδιαστική μεθοδολογία, ώστε να οδηγεί πάντα στις πιο αποδοτικές λύσεις, δηλαδή αυτές με την απλούστερη δομή.

6.4 Υιοθέτηση του κρατούμενου του Ling

Όπως δείξαμε στο κεφάλαιο 3, η χρησιμοποίηση των κρατούμενων που προτάθηκαν από τον Ling οδηγεί σε απλοποιημένες δομές παράλληλου προθέματος με αποδοτικά χαρακτηριστικά. Στην ενότητα αυτή θα περιγράψουμε πώς τα κρατούμενα της πρόσθεσης υπολοίπου μπορούν να απλοποιηθούν υιοθετώντας τις σχέσεις του Ling, επιταχύνοντας έτσι τον υπολογισμό τους μέσω των βελτιστοποιημένων δομών παράλληλου προθέματος που παρουσιάστηκαν στο κεφάλαιο 3. Θα παρουσιάσουμε την προτεινόμενη μέθοδο χρησιμοποιώντας σαν παράδειγμα τον αθροιστή υπολοίπου των 8 δυαδικών ψηφίων. Σύμφωνα με τη σχέση (6.1), το κρατούμενο C_2^- στην περίπτωση των 8 δυαδικών ψηφίων στην πλήρως ανοιγμένη του μορφή είναι ίσο με

$$C_2^- = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0G_7 + \dots + P_2P_1P_0P_7P_6P_5P_4G_3 \quad (6.8)$$

Επειδή $G_i = P_i \cdot G_i$ η σχέση (6.8) μπορεί να γραφεί ως εξής.

$$C_2^- = P_2(G_2 + G_1 + P_1G_0 + P_1P_0G_7 + \dots + P_1P_0P_7P_6P_5P_4G_3) = P_2H_2^-.$$

Σχήμα 6.5: Οι αθροιστές υπολοίπου των 10 δυαδικών ψηφίων που προέκυψαν χρησιμοποιώντας τους (α) $SDG_{10,\{8,3\}}$, (β) $SDG_{10,\{8,5\}}$, (γ) $SDG_{10,\{6,5\}}$, (δ) $SDG_{10,\{4,7\}}$, (ε) $SDG_{10,\{6,7\}}$, και (ζ) $SDG_{10,\{8,7\}}$.

Ο όρος H_2^- μπορεί να θεωρηθεί σαν το ψευδο-κρατούμενο του Ling για την περίπτωση των αθροιστών υπολοίπου. Εφαρμόζοντας την ίδια λογική με αυτή που παρουσιάστηκε στο κεφάλαιο 3, οι γειτονικοί όροι του H_2^- μπορούν να ομαδοποιηθούν

$$H_2^- = (G_2 + G_1) + (P_1 P_0)(G_0 + G_7) + (P_1 P_0)(P_7 P_6)(G_6 + G_5) + (P_1 P_0)(P_7 P_6)(P_5 P_4)(G_4 + G_3).$$

Σύμφωνα με τον ορισμό των όρων G_i^* και P_i^* και με την επιπλέον θεώρηση ότι $G_{-1} = G_{n-1}$ και $P_{-1} = P_{n-1}$ η εξίσωση (6) μετασχηματίζεται στην εξής μορφή.

$$H_2 = G_2^* + P_1^* G_0^* + P_1^* P_7^* G_6^* + P_1^* P_7^* P_5^* G_4^* \quad (6.9)$$

Με την ίδια διαδικασία μπορούν να προκύψουν οι σχέσεις που υπολογίζουν και τα υπόλοιπα ψευδο-κρατούμενα H_i^- . Έτσι, για όλες τις θέσεις του αθροιστή υπολοίπου των 8 δυαδικών

Σχήμα 6.6: Η μονάδα υπολογισμού των ψευδο-κρατούμενων του Ling για έναν αθροιστή υπολοίπου των 8 δυαδικών ψηφίων.

ψηφίων τα ψευδο-κρατούμενα υπολογίζονται με βάση τον τελεστή $\circ$ σύμφωνα με τις παρακάτω σχέσεις.

$$\begin{aligned}
 H_0 &\leftrightarrow (G_0^*, P_7^*) \circ (G_6^*, P_5^*) \circ (G_4^*, P_3^*) \circ (G_2^*, P_1^*) \\
 H_2 &\leftrightarrow (G_2^*, P_1^*) \circ (G_0^*, P_7^*) \circ (G_6^*, P_5^*) \circ (G_4^*, P_3^*) \\
 H_4 &\leftrightarrow (G_4^*, P_3^*) \circ (G_2^*, P_1^*) \circ (G_0^*, P_7^*) \circ (G_6^*, P_5^*) \\
 H_6 &\leftrightarrow (G_6^*, P_5^*) \circ (G_4^*, P_3^*) \circ (G_2^*, P_1^*) \circ (G_0^*, P_7^*) \\
 H_1 &\leftrightarrow (G_1^*, P_0^*) \circ (G_7^*, P_6^*) \circ (G_5^*, P_4^*) \circ (G_3^*, P_2^*) \\
 H_3 &\leftrightarrow (G_3^*, P_2^*) \circ (G_1^*, P_0^*) \circ (G_7^*, P_6^*) \circ (G_5^*, P_4^*) \\
 H_5 &\leftrightarrow (G_5^*, P_4^*) \circ (G_3^*, P_2^*) \circ (G_1^*, P_0^*) \circ (G_7^*, P_6^*) \\
 H_7 &\leftrightarrow (G_7^*, P_6^*) \circ (G_5^*, P_4^*) \circ (G_3^*, P_2^*) \circ (G_1^*, P_0^*)
 \end{aligned}$$

Από τις σχέσεις αυτές είναι προφανές ότι χρησιμοποιώντας τα νέα ζεύγη γέννησης και διάδοσης κρατούμενου (G_i^*, P_{i-1}^*) , όπου $G_{-1}^* = G_7^*$ και $P_{-1}^* = P_7^*$, μπορούμε να υπολογίσουμε τα ψευδο-κρατούμενα της πρόσθεσης υπολοίπου ανεξάρτητα στις άρτιες και στις περιττές στήλες, ακολουθώντας τη γενική δομή του σχήματος 3.5. Σημειώνουμε, πως στην περίπτωση των νέων αθροιστών υπολοίπου, το στάδιο της προεπεξεργασίας και του υπολογισμού του αθροίσματος είναι ίδια με αυτά των απλών αθροιστών που χρησιμοποιούν τα κρατούμενα του Ling. Οι νέες δομές παράλληλου προθέματος υπολογισμού των ψευδο-κρατούμενων στην περίπτωση των 8 δυαδικών ψηφίων φαίνονται στο σχήμα 6.6. Είναι εμφανές πως και στην περίπτωση των αθροιστών υπολοίπου, η υιοθέτηση των κρατούμενων του Ling διατηρεί όλα εκείνα τα πλεονεκτήματα που αναδείχθηκαν στο κεφάλαιο 3 για την περίπτωση των απλών αθροιστών ακέραιων αριθμών.

6.5 Αραιές δομές

Όπως και στην περίπτωση των παραδοσιακών αθροιστών, έτσι και στους αθροιστές υπολοίπου, μπορούμε να σχεδιάσουμε υβριδικές δομές παράλληλου προθέματος και επιλογής

Σχήμα 6.7: Αραιές υβριδικές δομές παράλληλου προθέματος και επιλογής κρατούμενου για την περίπτωση των αθροιστών υπολοίπου των 32 δυαδικών ψηφίων.

κρατούμενου. Οι δομές αυτές χρειάζονται ιδιαίτερα στην περίπτωση των αθροιστών υπολοίπου, εξαιτίας του μεγάλου αριθμού καλωδίων που υπάρχει μέσα στις δομές παράλληλου προθέματος. Ακολουθώντας την ίδια λογική που περιγράψαμε στο κεφάλαιο 3, όλες οι δομές που περιγράψαμε μπορούν εύκολα να μετασχηματιστούν σε αραιές δομές παράλληλου προθέματος και επιλογής κρατούμενου. Για παράδειγμα, οι αραιές εκδοχές των αθροιστών υπολοίπου, που προτάθηκαν στις εργασίες [176] και [77], φαίνονται στο σχήμα 6.7. Η σχεδίαση αυτών των δομών βασίζεται στην κυκλική απορροφητική ιδιότητα που αποδείξαμε στην ενότητα 6.2. Η απόδοση των κυκλωμάτων αυτών, συμπεριλαμβανομένης της αραιής δομής των νέων αθροιστών υπολοίπου που υιοθετούν τα κρατούμενα του Ling, θα αναλυθούν, μέσω των πειραματικών πειραμάτων που συλλέξαμε, στην επόμενη ενότητα.

6.6 Πειραματικά αποτελέσματα

Τα πειραματικά αποτελέσματα για την περίπτωση των αθροιστών υπολοίπου, βασίζονται αποκλειστικά σε υλοποιήσεις που προέκυψαν μέσω της αυτοματοποιημένης ροής σχεδιασμού

Πίνακας 6.1: Το εμβαδόν (μm^2) και η καθυστέρηση των αθροιστών υπολοίπου. Με LF συμβολίζεται η δομή παράλληλου προθέματος Ladner-Fischer και με KS η δομή Kogge-Stone.

n	[176] LF		[176] KS		[41]		Proposed		
	Area	Time	Area	Time	Area	Time	SDG _{n}	Area	Time
24	18805	13.3	35099	12.5	30378	10.9	SDG _{24,{16,9}}	24156	11.6
53	48382	15.1	77500	14.1	93289	12.6	SDG _{56,{32,25}}	58081	13.2

με τη βοήθεια των εργαλείων Synopsys και Cadence. Η διαδικασία που ακολουθήσαμε και ο τρόπος με τον οποίο μετρήσαμε την καθυστέρηση και το εμβαδόν των κυκλωμάτων είναι ακριβώς ίδιος με αυτόν που περιγράψαμε στη ενότητα 3.4. Η επιλογή αυτή πραγματοποιήθηκε για να μπορέσουμε να συμπεριλάβουμε στις μετρήσεις μας την επίδραση της αντίστασης και της χωρητικότητας των καλωδίων, με όση το δυνατόν περισσότερη ακρίβεια (η πληροφορία αυτή εξάγεται από το φυσικό σχέδιο του κυκλώματος, μέσω του εργαλείου Cadence). Η διαδικασία της βελτιστοποίησης μέσω του γεωμετρικού προγραμματισμού είναι πιο δύσκολη σ' αυτήν την περίπτωση, εξαιτίας της σύνθετης δομής των καλωδίων που εμφανίζονται στα δένδρα παράλληλου προθέματος των αθροιστών υπολοίπου, δυσκολεύοντας έτσι την εκτίμηση των χαρακτηριστικών τους.

Αρχικά, διερευνήσαμε την απόδοση των δομών παράλληλου προθέματος για την περίπτωση που το μήκος της λέξης εισόδου δεν είναι δύναμη του 2. Σ' αυτήν την περίπτωση χρησιμοποιήσαμε τις δομές που προέκυψαν μετά από την εφαρμογή της σχεδιαστικής μεθοδολογίας που παρουσιάστηκε στην ενότητα 2.3. Οι δομές αυτές συγκρίθηκαν στην τεχνολογία UMC-VST 25 των 250nm (FO4 = 135ps) με τους αθροιστές υπολοίπου που προτάθηκαν στις εργασίες [176] και [41]. Η τεχνολογία αυτή είναι η μόνη που είχαμε στη διάθεση μας την περίοδο σχεδίασης των προτεινόμενων αθροιστών υπολοίπου (έτος 2003). Οι αθροιστές που σχεδιάστηκαν έχουν εύρος ίσο με αυτό που χρησιμοποιούν οι αθροιστές μεγεθών των μονάδων κινητής υποδιαστολής μονής και διπλής ακρίβειας. Τα κυκλώματα αυτά βελτιστοποιήθηκαν, ώστε να πετύχουν τη μικρότερη δυνατή καθυστέρηση, ενώ τα αποτελέσματα που πήραμε φαίνονται στον πίνακα 6.1.

Οι προτεινόμενοι αθροιστές επιβάλλουν μικρότερο φορτίο στις εσωτερικές γραμμές του κυκλώματος, σε σχέση με τις δομές που προτάθηκαν στην εργασία [176], και είναι πάντα πιο γρήγορες από αυτές κατά 10% κατά μέσο όρο. Επίσης, συγκρινόμενες με τις ταχύτερες εκδοχές των αθροιστών υπολοίπου του [176], απαιτούν μικρότερο εμβαδόν χωροθέτησης περίπου της τάξης του 20%. Αντίθετα, οι προτεινόμενες δομές είναι μεγαλύτερες από τις πιο αργές εκδοχές του [176] και απαιτούν περίπου 20% περισσότερο χώρο στο ολοκληρωμένο. Τέλος, οι προτεινόμενοι αθροιστές υπολοίπου είναι κάπως πιο αργοί από τους αθροιστές του [41], επειδή αυξάνουν κάπως το φορτίο που πρέπει να οδηγήσουν οι πύλες του κυκλώματος. Παρολα αυτά, εξαιτίας της νέας μεθοδολογίας σχεδίασης, οι δομές που προτείνουμε μειώνουν το απαιτούμενο εμβαδόν χωροθέτησης περισσότερο από 28% κατά μέσο όρο.

Στη συνέχεια, συγκρίναμε τις αραιές δομές αθροιστών υπολοίπου, ώστε να ποσοτικοποιήσουμε τη μείωση της καθυστέρησης που επιτυγχάνουν οι νέοι αθροιστές που χρησιμοποιούν τα κρατούμενα του Ling αντί αυτών που προτάθηκαν στην εργασία [77]. Σημειώνουμε πως οι μετρήσεις που ακολουθούν είναι πιο πρόσφατες και πραγματοποιήθηκαν στην τεχνολογία-

Πίνακας 6.2: Το εμβαδόν (μm^2) και η καθυστέρηση των αθροιστών υπολοίπου που βασίζονται σε αραιές δομές παράλληλου προθέματος.

bits		Sparse [77]	Ling Sparse	Reduction (%)
32	Delay	14	12.6	10
	Area	10815	13339	-18
64	Delay	15.8	14.5	8
	Area	21991	31044	-29

(α)

bits		Sparse [77]	Ling Sparse	Reduction (%)
32	Delay	14	14	0
	Area	10815	12606	14
64	Delay	15.8	15.8	0
	Area	21991	29165	24

(β)

α των 180nm που χρησιμοποιήθηκε και στην περίπτωση των κλασικών αθροιστών που παρουσιάστηκαν στο κεφάλαιο 3. Οι αθροιστές υπολοίπου που σχεδιάστηκαν αφορούν στην περίπτωση των 32 και των 64 δυαδικών ψηφίων και τα αποτελέσματα που συλλέξαμε απεικονίζονται στον πίνακα 6.2.

Και στις δύο περιπτώσεις οι αραιές δομές που βασίζονται στα κρατούμενα του Ling είναι πιο γρήγορες. Τα κυκλώματα που επιτυγχάνουν τη μικρότερη καθυστέρηση απαιτούν κατά μέσο όρο πάνω από 20% περισσότερο εμβαδόν χωροθέτησης. Αντίθετα, στην περίπτωση που τα υπο-εξέταση κυκλώματα βελτιστοποιηθούν για την ίδια καθυστέρηση (πίνακας 6.2(β)), οι προτεινόμενοι αθροιστές οδηγούν σε πιο αποδοτικές υλοποιήσεις σε σχέση με τους αθροιστές υπολοίπου της εργασίας [77]. Σ' αυτές τις περιπτώσεις η μέση μείωση του εμβαδού χωροθέτησης που επιτυγχάνουμε ισούται με 19%.

Μονάδα Κανονικοποίησης

Η κανονικοποίηση του αποτελέσματος απαιτείται ώστε να φέρει το μέγεθος του αριθμού κινητής υποδιαστολής στην αναγκαία, σύμφωνα με το πρότυπο IEEE-754, κανονικοποιημένη του μορφή $1.xxx$. Όπως αναφέρθηκε στο κεφάλαιο 5, μας ενδιαφέρουν εκείνες οι περιπτώσεις της κανονικοποίησης στις οποίες δεν είναι εξ αρχής γνωστό πόσες θέσεις ολίσθησης απαιτούνται ώστε να επαναφέρουμε το αποτέλεσμα στην κανονικοποιημένη του μορφή. Οι πιο απλές περιπτώσεις μπορούν να πραγματοποιηθούν με ελάχιστες πύλες και δεν μας απασχολούν, αφού δεν επηρεάζουν ουσιαστικά την απόδοση της μονάδας κινητής υποδιαστολής.

Το πρόβλημα της κανονικοποίησης του αποτελέσματος μπορεί να λυθεί με δύο τρόπους οι οποίοι απεικονίζονται στο σχήμα 7.1. Σύμφωνα με το σχήμα 7.1(α), θα μπορούσαμε να μετρήσουμε τα προπορευόμενα μηδενικά (leading zeros) του αποτελέσματος του αθροιστή, αυτά δηλαδή που βρίσκονται μπροστά από τον πιο σημαντικό άσσο, ο οποίος καλείται ψηφίο θέσης κανονικοποίησης (leading digit), και να ολισθήσουμε το αποτέλεσμα τόσες φορές προς τ' αριστερά όσο και το πλήθος των μηδενικών που μετρήσαμε. Αν ολισθήσουμε το αποτέλεσμα k θέσεις προς τ' αριστερά είναι σαν το πολλαπλασιάζουμε με το 2^k . Έτσι για να ολοκληρώσουμε τη διαδικασία πρέπει να αφαιρέσουμε την τιμή k από την αντίστοιχη τιμή του εκθέτη που έχουμε για το αποτέλεσμα μας. Η μέθοδος αυτή είναι αρκετά αργή και σπάνια χρησιμοποιείται. Αν επιμείνουμε να την χρησιμοποιήσουμε, πρέπει υποχρεωτικά ο υπολογισμός των προπορευόμενων μηδενικών και η ολίσθηση να πραγματοποιηθούν σε δύο διαφορετικά στάδια του pipeline, αυξάνοντας έτσι το χρόνο παραγωγής του πρώτου αποτελέσματος (latency).

Για να επιταχύνουμε τη διαδικασία μπορούμε να χρησιμοποιήσουμε τη δομή που φαίνεται στο σχήμα 7.1(β). Σύμφωνα με αυτή την τεχνική, προβλέπουμε, παράλληλα με την πρόσθεση των πραγματικών τιμών, ένα ψευδό-αποτέλεσμα το οποίο έχει περίπου τον ίδιο αριθμό από προπορευόμενα μηδενικά με το τελικό πραγματικό αποτέλεσμα. Έτσι, αντί να περιμένουμε

Σχήμα 7.1: Οι δύο επιλογές για την υλοποίηση της κανονικοποίησης.

να ολοκληρωθεί η πραγματική λειτουργία, -π.χ. αφαίρεση- μπορούμε παράλληλα με αυτή να μετρήσουμε τον αριθμό των προπορευόμενων μηδενικών του ψευδο-αποτελέσματος. Η μέτρηση αυτή θα ολοκληρωθεί περίπου στον ίδιο χρόνο με τον υπολογισμό του τελικού αποτελέσματος και έτσι οι δύο αυτές τιμές μπορούν να δωθούν ταυτόχρονα στον ολισθητή ώστε να ολοκληρωθεί η κανονικοποίηση. Μ' αυτόν τον τρόπο, το κρίσιμο μονοπάτι έχει τουλάχιστον θεωρητικά απαλλαγεί από την καθυστέρηση που εισάγει το κύκλωμα μέτρησης των προπορευόμενων μηδενικών (Leading Zero Counter, LZC) και αποτελείται από την καθυστέρηση του αθροιστή και του ολισθητή. Στην περίπτωση χρησιμοποίησης του pipeline, το πλήθος των απαιτούμενων σταδίων μειώνεται κατά ένα.

Η πρόβλεψη του ψευδο-αποτελέσματος πραγματοποιείται από τη λογική πρόβλεψης της θέσης του ψηφίου κανονικοποίησης (Leading Zero Anticipation, LZA). Για τη σχεδίαση της λογικής LZA έχουν αναφερθεί πολλές υλοποιήσεις τα τελευταία χρόνια [18], [83], [67] [96], [140]. Μια πολύ κατανοητή σύνοψη της λειτουργίας των πιο αποδοτικών μεθόδων LZA παρουσιάζεται στην εργασία [140]. Όπως γνωρίζουμε, το μέγεθος των αριθμών κινητής υποδιαστολής ακολουθεί την αναπαράσταση του προσημασμένου μεγέθους, όπου η πληροφορία του προσήμου του αριθμού αναπαρίσταται με ένα επιπλέον ψηφίο. Έτσι, η τιμή του μεγέθους είναι πάντα θετική ακόμα και αν ο αριθμός είναι αρνητικός. Για να προκύψει πάντα θετικό αποτέλεσμα όταν η πραγματική πράξη είναι αφαίρεση, πρέπει ο μικρότερος από τους δύο αριθμούς να αφαιρεθεί από το μεγαλύτερο. Η εναλλαγή αυτή όμως εισάγει πολύ μεγάλη καθυστέρηση και δεν μπορεί να πραγματοποιηθεί. Έτσι, η LZA λογική πρέπει να είναι σε θέση να προβλέψει τη θέση του ψηφίου κανονικοποίησης είτε το αποτέλεσμα είναι αρνητικό είτε θετικό. Αν το μη-κανονικοποιημένο αποτέλεσμα που προέκυψε είναι θετικό τότε θα έχει τη μορφή $0.0 \dots 01XXX$ και το ψηφίο της θέσης κανονικοποίησης θα έχει την τιμή 1. Αντίθετα, στην περίπτωση που το αποτέλεσμα είναι αρνητικό, τότε θα έχει την μορφή $1.1 \dots 10XXX$. Αυτό δε σημαίνει πως το αποτέλεσμα είναι σωστά κανονικοποιημένο, γιατί όταν στη συνέχεια αντιστραφεί για να συμβαδίσει με την αναπαράσταση προσημασμένου μεγέθους θα έρθει και πάλι στην γνωστή μη-κανονικοποιημένη μορφή. Στην δεύτερη περίπτωση, το ψηφίο της θέσης κανονικοποίησης έχει την τιμή 0 και μπροστά του βρίσκονται προπορευόμενοι άσσοι και όχι μηδενικά. Στη γενική

περίπτωση, η οποία είναι αυτή που μας ενδιαφέρει, η λογική LZA πρέπει να αντιμετωπίζει ταυτόχρονα την περίπτωση των προπορευόμενων άσσων και των μηδενικών. Υπάρχουν μέθοδοι οι οποίες προσπαθούν να αντιμετωπίσουν και τις δύο πιθανές εκδοχές του αποτελέσματος (θετικό/αρνητικό) με έναν ενιαίο και αποτελεσματικό τρόπο [83], [67], [140], [134]. Όμως σε κάποιες άλλες περιπτώσεις, όταν η γενική LZA λογική εισάγει περισσότερη από την αναμενόμενη καθυστέρηση, χρησιμοποιούνται δύο χωριστές μονάδες πρόβλεψης [141], [96], μία για την περίπτωση των προπορευόμενων μηδενικών και μία για τους προπορευόμενους άσσους αντίστοιχα. Επίσης, για τον ίδιο λόγο, σε πολλά κυκλώματα οι σχεδιαστές επέλεξαν να παράγουν δύο εκδοχές του ίδιου αποτελέσματος. Για παράδειγμα, στην περίπτωση της αφαίρεσης υπολογίζονται το $A - B$ και το $B - A$ και η τελική τιμή επιλέγεται ανάλογα με την τιμή του προσήμου του τελικού αποτελέσματος. Έτσι, είναι σίγουρο πως κάποιο από τα δύο αποτελέσματα θα είναι θετικό και επομένως δεν απαιτείται παρά μόνο η πρόβλεψη των προπορευόμενων μηδενικών [124], [53].

Η εγγενής αδυναμία όλων των τεχνικών πρόβλεψης, που έχουν μέχρι σήμερα προταθεί, είναι ότι η θέση του προπορευόμενου ψηφίου του ψευδο-αποτελέσματος μπορεί να διαφέρει κατά ένα από αυτό του πραγματικού αποτελέσματος. Έτσι, υπάρχει πάντα η περίπτωση η κανονικοποίηση που εκτελέστηκε μετρώντας τα προπορευόμενα μηδενικά ή τους άσσους του ψευδο-αποτελέσματος να είναι λανθασμένη και να απαιτείται μια επιπλέον ολίσθηση του αποτελέσματος κατά μία θέση. Η ολίσθηση αυτή θα είναι είτε προς τ' αριστερά είτε προς τα δεξιά ανάλογα με το αν τα προπορευόμενα μηδενικά ή άσσοι υπολείπονται ή υπερβαίνουν αντίστοιχα αυτά του πραγματικού αποτελέσματος. Η αντιμετώπιση του λάθους πρόβλεψης μπορεί να πραγματοποιηθεί με πολλούς τρόπους, οι οποίοι θα αναλυθούν στην ενότητα 7.3.2.

Στις επόμενες ενότητες αυτού του κεφαλαίου θα παρουσιάσουμε τις προτεινόμενες μονάδες μέτρησης και πρόβλεψης των προπορευόμενων μηδενικών ή άσσων. Οι νέες τεχνικές προέκυψαν μέσα από μια μαθηματική αντιμετώπιση του προβλήματος, η οποία ανέδειξε ταυτόχρονα την περιττή επεξεργασία που εμπεριέχουν οι ήδη γνωστές υλοποιήσεις. Θα ξεκινήσουμε την περιγραφή μας κάνοντας μια ανασκόπηση των προηγούμενων αρχιτεκτονικών και θα την ολοκληρώσουμε με την παρουσίαση των πειραματικών αποτελεσμάτων, τα οποία συλλέξαμε μετά από υλοποιήσεις των υπο-εξέταση κυκλωμάτων σε στατική και σε δυναμική CMOS λογική.

7.1 Μέτρηση προπορευόμενων μηδενικών - Σύνοψη ανασκόπηση

Η μέτρηση των προπορευόμενων μηδενικών αναφέρεται στη δυαδική κωδικοποίηση του πλήθους των ψηφίων που είναι ίσα με μηδέν στο αριστερό πιο σημαντικό μέρος της λέξης εισόδου μπροστά από τη θέση κανονικοποίησης. Το αντίθετο ισχύει για την περίπτωση των προπορευόμενων άσσων. Εφόσον οι δύο περιπτώσεις είναι συμμετρικές θα περιορίσουμε τη συζήτηση για τη μέτρηση των προπορευόμενων ψηφίων ενός αριθμού στα μηδενικά.

Μια μονάδα μέτρησης των προπορευόμενων μηδενικών δέχεται σαν είσοδο n δυαδικά ψηφία $A = A_{n-1}A_{n-2} \dots A_0$, όπου το A_{n-1} είναι το πιο σημαντικό ψηφίο και υπολογίζει τα $\log_2 n$ δυαδικά ψηφία της δυαδικής αναπαράστασης του πλήθους των προπορευόμενων μηδενικών που συμβολίζουμε Z . Επίσης, υπολογίζεται ένα σήμα V το οποίο υποδηλώνει την περίπτωση που όλα τα δυαδικά ψηφία της εισόδου είναι ίσα με μηδέν.

Η πρώτη μέθοδος που χρησιμοποιείται για τον υπολογισμό του πλήθους των προπορευόμενων μηδενικών βασίζεται σε μια διαδικασία κωδικοποίησης της θέσης του προπορευόμενου

ψηφίου που εκτελείτε σε δύο βήματα [140]. Αρχικά, η θέση του πιο σημαντικού άσσου μαρκάρεται, ενώ τα υπόλοιπα ψηφία μηδενίζονται (αναπαράσταση one hot). Για παράδειγμα, για την είσοδο 00110100 η θέση του πιο σημαντικού άσσου καθορίζεται από την one hot λέξη 00100000. Για να παράγουμε αυτή την κωδικοποίηση χρησιμοποιούμε μια ενδιάμεση αναπαράσταση μέσω της λέξης S . Τα λιγότερο σημαντικά δυαδικά ψηφία του S που ακολουθούν τον πιο σημαντικό άσσο τίθενται όλα στη τιμή 1. Για παράδειγμα, για την ίδια τιμή της εισόδου η λέξη S είναι ίση με 00111111. Αν συμβολίσουμε με S_i το i -οστό, $0 \leq i \leq n-1$, δυαδικό ψηφίο της ενδιάμεσης λέξης S , τότε κάθε S_i υπολογίζεται από τα δυαδικά ψηφία της εισόδου σύμφωνα με την σχέση (7.1).

$$S_i = A_{n-1} + A_{n-2} + \dots + A_{i+1} + A_i \quad (7.1)$$

Κάθε δυαδικό ψηφίο S_i αποκαλύπτει την ύπαρξη τουλάχιστον ενός άσσου μεταξύ της πιο σημαντικής θέσης και της θέσης i . Η one-hot αναπαράσταση της θέσης του πιο σημαντικού άσσου (λέξη L) μπορεί να προκύψει ανιχνεύοντας το ζευγάρι τιμών $(0, 1)$ μεταξύ δύο οποιονδήποτε διαδοχικών ψηφίων της λέξης S . Έτσι προκύπτει ότι,

$$L_i = \overline{S}_{i+1} \cdot S_i = \overline{S}_{i+1} \cdot A_i \quad \text{για} \quad 0 \leq i < n-1 \quad \text{και} \quad L_{n-1} = S_{n-1} \quad (7.2)$$

Το κύκλωμα που υπολογίζει τη λέξη L είναι γνωστό και ως κωδικοποιητής προτεραιοτήτων (priority encoder) [169]. Για να πάρουμε την τελική δυαδική αναπαράσταση του πλήθους των προπορευόμενων μηδενικών η λέξη L δίνεται σαν είσοδος σε ένα κύκλωμα που ονομάζεται κωδικοποιητής (encoder)[108]. Για την περίπτωση που η λέξη εισόδου αποτελείται από 8 δυαδικά ψηφία, τα δυαδικά ψηφία του Z δίνονται από τις σχέσεις (7.3)–(7.5).

$$Z_2 = L_3 + L_2 + L_1 + L_0 \quad (7.3)$$

$$Z_1 = L_5 + L_4 + L_1 + L_0 \quad (7.4)$$

$$Z_0 = L_6 + L_4 + L_2 + L_0 \quad (7.5)$$

Αντίστοιχα, το σήμα V είναι ίσο με $\overline{S}_0$ υποδηλώνοντας ότι δεν υπάρχει κανένα ψηφίο της λέξης εισόδου A που να είναι ίσο με ένα. Η προσέγγιση αυτή είναι η πιο ευρέως χρησιμοποιούμενη, όταν για την υλοποίηση της μονάδας κινητής υποδιαστολής επιλέγεται η δυναμική λογική. Στην περίπτωση αυτή, το πλήθος των προπορευόμενων μηδενικών μπορεί να υπολογιστεί σε λίγα λογικά επίπεδα, χρησιμοποιώντας δυναμικές πύλες OR πολλών εισόδων για τον υπολογισμό τόσο των δυαδικών ψηφίων του S όσο και του Z . Στη συνέχεια του κεφαλαίου η συγκεκριμένη τεχνική θα ονομάζεται μονάδα μέτρησης βασισμένη σε κυκλώματα κωδικοποιητή.

Η δεύτερη μέθοδος βασίζεται σε μια αλγοριθμική προσέγγιση και προτάθηκε από τον Oklobdzija στην εργασία [126]. Το ίδιο κύκλωμα χρησιμοποιήθηκε και στην περίπτωση της εργασίας [154] όπως αποδείχθηκε στη συνέχεια στο δημοσιευμένο σχόλιο [127]. Αρχικά, η είσοδος χωρίζεται σε $n/2$ ομάδες αποτελούμενες από ζεύγη διαδοχικών δυαδικών ψηφίων. Σε κάθε ομάδα μετράμε και κωδικοποιούμε τον αριθμό των προπορευόμενων άσσων. Έχουμε ένα προπορευόμενο μηδενικό μόνο στην περίπτωση 01. Αντίστοιχα, η περίπτωση 00 κωδικοποιείται με ένα επιπλέον σήμα “όλα μηδέν” υποδηλώνοντας ότι όλα τα δυαδικά ψηφία της ομάδας είναι μηδενικά. Ταυτόχρονα, το σήμα αυτό δείχνει ότι ο πιο σημαντικός άσσος αποκλείεται να είναι στην ομάδα αυτή. Στο επόμενο επίπεδο, οι μετρήσεις των προπορευόμενων μηδενικών

Σχήμα 7.2: Η μονάδα μέτρησης των προπορευόμενων μηδενικών των 16 δυαδικών ψηφίων που ακολουθεί τη δομή που προτάθηκε στην εργασία (126).

που προέκυψαν από τις γειτονικές ομάδες των δύο δυαδικών ψηφίων συνδυάζονται ώστε να υπολογιστεί η αντίστοιχη μέτρηση για τις ομάδες των τεσσάρων δυαδικών ψηφίων. Στην ουσία επιλέγεται η κωδικοποίηση είτε της αριστερής είτε της δεξιάς ομάδας, ανάλογα με την τιμή του σήματος “όλα μηδέν” της αριστερής. Επίσης, ένα νέο σήμα ένδειξης της κατάστασης “όλα μηδέν” υπολογίζεται συνδυάζοντας τα δύο αντίστοιχα σήματα των μικρότερων ομάδων. Η διαδικασία αυτή επαναλαμβάνεται διαδοχικά για $\log_2 n$ επίπεδα. Το κύκλωμα έχει κανονική (regular) δομή και είναι η πιο γρήγορη έως τώρα λύση για υλοποιήσεις σε στατική CMOS. Η υλοποίηση αυτής της LZC μονάδας για την περίπτωση των 16 δυαδικών ψηφίων φαίνεται στο σχήμα 7.2.

Η τελευταία μέθοδος για τον υπολογισμό του πλήθους των προπορευόμενων μηδενικών παρουσιάστηκε από τους Bruguera και Lang στην εργασία [93]. Ο αλγόριθμος που πρότειναν υπολογίζει αναδρομικά το i -οστό ψηφίο του πλήθους των προπορευόμενων μηδενικών βασιζόμενος στην τιμή του πιο σημαντικού ψηφίου Z_{i+1} , το οποίο υπολογίζεται γρηγορότερα. Για να αναλύσουμε τη λειτουργία του αλγορίθμου θα χρησιμοποιήσουμε σαν παράδειγμα την περίπτωση των 8 δυαδικών ψηφίων για την είσοδο A . Σε πρώτη φάση, υπολογίζεται η λογική συνάρτηση OR των $n/2$ πιο σημαντικών ψηφίων. Το σήμα αυτό θα είναι ίσο με μηδέν όταν όλα τα δυαδικά ψηφία $A_7 A_6 A_5 A_4$ είναι ίσα με μηδέν. Συνεπώς, αυτό το σήμα αντεστραμμένο δηλώνει ότι υπάρχουν τουλάχιστον τέσσερα προπορευόμενα μηδενικά στην είσοδο A και επομένως $Z_2 = 1$. Στο επόμενο βήμα, ο αλγόριθμος χρησιμοποιεί την τιμή του Z_2 για να καθορίσει το πλήθος των υπόλοιπων προπορευόμενων μηδενικών. Στην περίπτωση που $Z_2 = 1$, σημαίνει ότι πρέπει να ψάξουμε για περισσότερα προπορευόμενα μηδενικά στα δυαδικά ψηφία $A_3 A_2 A_1 A_0$. Σε αντίθετη περίπτωση, $Z_2 = 0$, υπάρχουν λιγότερα από τέσσερα προπορευόμενα μηδενικά τα οποία είναι τοποθετημένα στο αριστερό κομμάτι της λέξης $A_7 A_6 A_5 A_4$. Στη συνέχεια, για να καθορίσουμε την τιμή του Z_1 υπολογίζουμε την λογική συνάρτηση OR των δυαδικών ψηφίων A_3, A_2 και A_7, A_6 αντίστοιχα, και διαλέγουμε μία από τις δύο χρησιμοποιώντας έναν πολυπλέκτη ο οποίος ελέγχεται από το Z_2 . Εφόσον το λογικό OR των μικρότερων ομάδων υπολογίζεται

νωρίτερα απ' ότι το σήμα Z_2 ο υπολογισμός των Z_2 και Z_1 επικαλύπτεται χρονικά οδηγώντας σε γρήγορες λύσεις. Στα επόμενα επίπεδα ο αλγόριθμος λειτουργεί ακριβώς με τον ίδιο τρόπο και οι τέσσερις δυνατοί συνδυασμοί των Z_2Z_1 καθορίζουν, χρησιμοποιώντας ένα δένδρο από πολυπλέκτες, αν το επόμενο προπορευόμενο μηδενικό που αναζητούμε βρίσκεται στις θέσεις A_7, A_5, A_3 ή A_1 . Σ' αυτό το σημείο καλό είναι να σημειώσουμε ότι στις δύο τελευταίες μεθόδους που περιγράψαμε, όταν όλα τα δυαδικά ψηφία της εισόδου είναι μηδενικά, τότε η σημαία V ενεργοποιείται και τα δυαδικά ψηφία του Z αντιμετωπίζονται σαν αδιάφορες τιμές. Αυτό επιτρέπεται να συμβεί, γιατί όταν το $A = 0$ οποιαδήποτε περιττή κανονικοποίηση και αν υποστεί δεν θα αλλάξει την τιμή του. Όπως θα φανεί στην επόμενη ενότητα, αυτή η προσέγγιση ακολουθείται και από την προτεινόμενη αρχιτεκτονική.

7.2 Προτεινόμενες μονάδες μέτρησης προπορευόμενων μηδενικών

Στην ενότητα αυτή θα παρουσιάσουμε τις προτεινόμενες μονάδες μέτρησης των προπορευόμενων μηδενικών. Για την ανακάλυψη των κυκλωμάτων ακολουθήσαμε μια καθαρά μαθηματική προσέγγιση, φέρνοντας τις σχέσεις που περιγράφουν τα δυαδικά ψηφία Z στην απλούστερη δυνατή μορφή τους. Η μορφή των νέων σχέσεων μας επιτρέπει να υλοποιήσουμε τα αντίστοιχα κυκλώματα μέτρησης χρησιμοποιώντας απλές τεχνικές πρόβλεψης κρατουμένου.

Θα παρουσιάσουμε την προτεινόμενη μέθοδο μέσω ενός παραδείγματος. Ως παράδειγμα θα χρησιμοποιηθεί η σχεδίαση μιας μονάδας μέτρησης προπορευόμενων μηδενικών των 8 δυαδικών ψηφίων. Ξεκινώντας, αντικαθιστούμε στις σχέσεις (7.3)–(7.5), που περιγράφουν τα δυαδικά ψηφία Z , τον ορισμό των ψηφίων L ο οποίος δίνεται από τη σχέση (7.2). Οι σχέσεις που προκύπτουν μετά την αντικατάσταση είναι οι ακόλουθες:

$$Z_2 = \overline{S}_4 \cdot S_3 + \overline{S}_3 \cdot S_2 + \overline{S}_2 \cdot S_1 + \overline{S}_1 \cdot S_0 \quad (7.6)$$

$$Z_1 = \overline{S}_6 \cdot S_5 + \overline{S}_5 \cdot S_4 + \overline{S}_2 \cdot S_1 + \overline{S}_1 \cdot S_0 \quad (7.7)$$

$$Z_0 = \overline{S}_7 \cdot S_6 + \overline{S}_5 \cdot S_4 + \overline{S}_3 \cdot S_2 + \overline{S}_1 \cdot S_0 \quad (7.8)$$

Εφόσον τα δυαδικά ψηφία του S ακολουθούν πάντα τη μορφή $00 \dots 01 \dots 11$ (μονότονα αύξουσα ακολουθία), τότε ένα ζεύγος δυαδικών ψηφίων του S έστω (S_i, S_j) όπου $i > j$ δεν μπορεί να πάρει ποτέ την τιμή $(1, 0)$. Το γεγονός αυτό μας οδηγεί σε δύο πολύ βασικές ιδιότητες των δυαδικών ψηφίων του S . Για κάθε $i > j$ ισχύει ότι $S_i \cdot S_j = S_i$ και $S_i + S_j = S_j$. Χρησιμοποιώντας τις ιδιότητες αυτές μπορούμε να απλοποιήσουμε περαιτέρω τις σχέσεις που περιγράφουν τα δυαδικά ψηφία Z . Για παράδειγμα, ο όρος $\overline{S}_4 \cdot S_3 + \overline{S}_3 \cdot S_2$ της (7.6) μπορεί να γραφεί ως $\overline{S}_4 \cdot (S_3 + \overline{S}_3 \cdot S_2)$ μιας και $\overline{S}_4 \cdot \overline{S}_3 = \overline{S}_3$. Επίσης, η σχέση μέσα στην παρένθεση είναι ισοδύναμη με την $S_3 + S_2$, η οποία σύμφωνα με τη δεύτερη ιδιότητα των δυαδικών ψηφίων του S μειώνεται σε S_2 . Τέτοιες αλλαγές μπορούν να εφαρμοστούν επαναληπτικά και στις υπόλοιπες σχέσεις που περιγράφουν το πλήθος των προπορευόμενων μηδενικών, οδηγώντας στις επόμενες εξισώσεις:

$$Z_2 = \overline{S}_4 \cdot S_0 \quad (7.9)$$

$$Z_1 = \overline{S}_6 \cdot S_4 + \overline{S}_2 \cdot S_0 \quad (7.10)$$

$$Z_0 = \overline{S}_7 \cdot S_6 + \overline{S}_5 \cdot S_4 + \overline{S}_3 \cdot S_2 + \overline{S}_1 \cdot S_0 \quad (7.11)$$

Σύμφωνα με τις (7.9)–(7.11), όταν η είσοδος είναι ίση με 0 όλα τα δυαδικά ψηφία του Z

παίρνουν με τη σειρά τους την τιμή 0, υποδεικνύοντας έτσι ότι το αποτέλεσμα δεν χρειάζεται να κανονικοποιηθεί. Έτσι, όσο το σήμα V είναι ενεργοποιημένο μπορούμε να θέσουμε τα δυαδικά ψηφία του Z σε μια οποιαδήποτε τιμή. Διαλέξουμε, λοιπόν, να θέσουμε σε αυτήν την περίπτωση $V = \overline{S}_0 = 1$ τα δυαδικά ψηφία του Z στην τιμή 1. Μ' αυτόν τον τρόπο οι σχέσεις (7.9)–(7.11) μπορούν να γραφούν ως εξής:

$$Z_2 = \overline{S}_4 \cdot S_0 + \overline{S}_0 \quad (7.12)$$

$$Z_1 = \overline{S}_6 \cdot S_4 + \overline{S}_2 \cdot S_0 + \overline{S}_0 \quad (7.13)$$

$$Z_0 = \overline{S}_7 \cdot S_6 + \overline{S}_5 \cdot S_4 + \overline{S}_3 \cdot S_2 + \overline{S}_1 \cdot S_0 + \overline{S}_0 \quad (7.14)$$

Στις νέες σχέσεις εμφανίζονται περιττοί όροι της μορφής $\overline{S}_i \cdot S_0 + \overline{S}_0$, $i > 0$. Κάθε τέτοιος όρος ισούται με $\overline{S}_i + \overline{S}_0$, ο οποίος σύμφωνα με την πρώτη ιδιότητα των δυαδικών ψηφίων του S μπορεί ισοδύναμα να γραφτεί ως $\overline{S}_i$. Εφαρμόζοντας αυτές τις απλοποιήσεις, οι νέες σχέσεις που περιγράφουν τα δυαδικά ψηφία του Z δίνονται στις (7.15)–(7.17).

$$Z_2 = \overline{S}_4 \quad (7.15)$$

$$Z_1 = \overline{S}_6 \cdot S_4 + \overline{S}_2 \quad (7.16)$$

$$Z_0 = \overline{S}_7 \cdot S_6 + \overline{S}_5 \cdot S_4 + \overline{S}_3 \cdot S_2 + \overline{S}_1 \quad (7.17)$$

Ο τελικός στόχος είναι να υπολογίσουμε το Z απευθείας από τα δυαδικά ψηφία της εισόδου A . Έτσι, ακολουθώντας τη σχέση (7.1) εκφράζουμε κάθε S_i των σχέσεων (7.15)–(7.17) συναρτήσει των δυαδικών ψηφίων της εισόδου. Αφού αντιστρέψουμε τα δυαδικά ψηφία του Z και συνεχίσουμε με απλές αλγεβρικές πράξεις, φτάνουμε στις τελικές σχέσεις που περιγράφουν τα δυαδικά ψηφία του πλήθους των προπορευόμενων μηδενικών για την περίπτωση των 8 δυαδικών ψηφίων.

$$\overline{Z}_2 = A_7 + A_6 + A_5 + A_4 \quad (7.18)$$

$$\overline{Z}_1 = A_7 + A_6 + \overline{A}_5 \cdot \overline{A}_4 \cdot (A_3 + A_2) \quad (7.19)$$

$$\overline{Z}_0 = A_7 + \overline{A}_6 \cdot A_5 + \overline{A}_6 \cdot \overline{A}_4 \cdot A_3 + \overline{A}_6 \cdot \overline{A}_4 \cdot \overline{A}_2 \cdot A_1 \quad (7.20)$$

Σημειώνουμε σ' αυτό το σημείο πως όταν το $A = 0$, τότε όλα τα δυαδικά ψηφία του Z παίρνουν την τιμή 1. Αυτός είναι ο λόγος που το λιγότερο σημαντικό ψηφίο της εισόδου A_0 δεν συμμετέχει σε καμιά από τις σχέσεις που περιγράφουν το πλήθος των προπορευόμενων μηδενικών. Το A_0 χρησιμοποιείται μόνο μέσω του S_0 για τον υπολογισμό του σήματος V .

Στη συνέχεια θα δείξουμε (αν και δεν είναι άμεσα φανερό) πως στην ουσία όλα τα δυαδικά ψηφία του Z ακολουθούν την ίδια μορφή. Θυμίζουμε ότι το λιγότερο σημαντικό ψηφίο Z_0 καθορίζει αν το πλήθος των προπορευόμενων μηδενικών είναι άρτιος ή περιττός αριθμός. Θα θέλαμε να εκμεταλλευτούμε αυτή την απλή ιδιότητα. Για το λόγο αυτό ορίζουμε ένα νέο τελεστή $F()$ ο οποίος έχει την ίδια μορφή με τη σχέση που περιγράφει το Z_0 και ορίζεται ως εξής:

$$F(X_{n-1}, X_{n-2}, \dots, X_1, X_0) = X_{n-1} + \overline{X}_{n-2} \cdot X_{n-3} + \overline{X}_{n-2} \cdot \overline{X}_{n-4} \cdot X_{n-5} + \dots + \overline{X}_{n-2} \cdot \overline{X}_{n-4} \cdot \overline{X}_{n-6} \cdot \dots \cdot \overline{X}_2 \cdot X_1 \quad (7.21)$$

Οι τιμές της εισόδου, για τις οποίες ο τελεστής $F()$ ικανοποιείται (ισούται με 1), αποτελούνται

Σχήμα 7.3: Ένα παράδειγμα των 8 δυαδικών ψηφίων του νέου τρόπου υπολογισμού των προπορευόμενων μηδενικών μιας λέξης.

από άρτιο αριθμό προπορευόμενων μηδενικών. Καλό είναι να τονιστεί ότι το λιγότερο σημαντικό ψηφίο της εισόδου X_0 δε χρησιμοποιείται από τον τελεστή $F()$. Αυτό εξηγείται από την επιλογή που κάναμε να θεωρήσουμε τα δυαδικά ψηφία του Z ίσα με 1 όταν η είσοδος X είναι ίση με 0. Ο υπολογισμός των δυαδικών ψηφίων του Z δεν είναι τίποτε άλλο από την εφαρμογή του τελεστή $F()$ σε διαφορετικές ομάδες των δυαδικών ψηφίων της εισόδου A . Συγκεκριμένα στη περίπτωση που το A αποτελείται από 8 δυαδικά ψηφία προκύπτει ότι

$$\overline{Z}_0 = F(A_7, A_6, A_5, A_4, A_3, A_2, A_1, A_0)$$

$$\overline{Z}_1 = F(A_7 + A_6, A_5 + A_4, A_3 + A_2, A_1 + A_0)$$

$$\overline{Z}_2 = F(A_7 + A_6 + A_5 + A_4, A_3 + A_2 + A_1 + A_0)$$

και

$$\overline{V} = A_7 + A_6 + A_5 + A_4 + A_3 + A_2 + A_1 + A_0$$

Στο σχήμα 7.3 φαίνεται ένα παράδειγμα εφαρμογής της προτεινόμενης μεθόδου στην είσοδο 00000101. Στο πρώτο στάδιο, εφαρμόζοντας τον τελεστή $F()$ σε όλα τα δυαδικά ψηφία της εισόδου υπολογίζουμε αν το πλήθος των προπορευόμενων μηδενικών είναι άρτιος ή περιττός αριθμός. Εφόσον η είσοδος αποτελείται από 5 προπορευόμενα μηδενικά τότε η σχέση του $F()$ είναι ψευδής και έτσι η τιμή του Z_0 είναι ίση με 1. Στη συνέχεια υπολογίζουμε ανά δύο το λογικό OR των γειτονικών ψηφίων της εισόδου και καταλήγουμε σε μια νέα ακολουθία που αποτελείται από μισά δυαδικά ψηφία σε σχέση με την αρχική. Με τον ίδιο τρόπο και χρησιμοποιώντας τον τελεστή $F()$ αποφασίζουμε για τη ακολουθία 0011 που προέκυψε αν το πλήθος των προπορευόμενων μηδενικών είναι άρτιος αριθμός. Εφόσον η λέξη 0011 έχει άρτιο αριθμό από προπορευόμενα μηδενικά το δυαδικό ψηφίο Z_1 τίθεται στην τιμή 0. Συνδυάζοντας και πάλι με την λογική πράξη OR τα διαδοχικά δυαδικά ψηφία του 0011 και εφαρμόζοντας τον τελεστή $F()$ στη νέα λέξη 01 που προκύπτει, μπορούμε να υπολογίσουμε ότι $Z_2 = 1$. Στη γενική περίπτωση η διαδικασία εφαρμόζεται επαναληπτικά μέχρι να μείνει μόνο ένα ψηφίο από τα αρχικά n δυαδικά ψηφία της εισόδου, το οποίο είναι το σήμα V που συμβολίζει, αν η

είσοδος αποτελείται μόνο από μηδενικά δυαδικά ψηφία. Η έξοδος της προτεινόμενης μονάδας μέτρησης των προπορευόμενων μηδενικών είναι ίση με $Z_2Z_1Z_0 = 101$, η οποία αναπαριστά σωστά τα 5 προπορευόμενα μηδενικά της εισόδου του παραδείγματος. Στο σχήμα τα λιγότερα σημαντικά ψηφία όλων των ενδιάμεσων λέξεων που προκύπτουν παρουσιάζονται σκιασμένα. Ο λόγος αυτής της διαφοροποίησης είναι γιατί θέλουμε να τονίσουμε πως δεν χρησιμοποιούνται στις διαδοχικές εφαρμογές του τελεστή $F()$ και παίρνουν μέρος μόνο στον υπολογισμό του σήματος V .

7.2.1 Η οργάνωση του νέου κυκλώματος μέτρησης

Στη συνέχεια θα παρουσιάσουμε δύο μεθόδους οργάνωσης του υλικού της μονάδας μέτρησης των προπορευόμενων μηδενικών. Μπορεί εύκολα να παρατηρήσει κανείς πως η σχέση (7.21) που περιγράφει τον τελεστή $F()$ έχει την ίδια μορφή με την κλασσική εξίσωση πρόβλεψης κρατουμένου

$$G_i + P_iG_{i-1} + \dots + P_iP_{i-1} \dots P_1G_0$$

θεωρώντας πως τα δυαδικά ψηφία γέννησης και διάδοσης κρατουμένου (G_i, P_i) της θέσης i είναι ίσα με $(X_i, \overline{X}_{i-1})$. Κάνοντας αυτήν την αντιστοίχιση καταφέρνουμε να μετατρέψουμε τον υπολογισμό του Z_i σε πρόβλημα πρόβλεψης κρατουμένου. Έτσι, κάθε Z_i μπορεί να υπολογιστεί χρησιμοποιώντας ένα δένδρο παράλληλου υπολογισμού κρατουμένου, το οποίο μπορεί να υλοποιηθεί αποτελεσματικά τόσο σε στατική όσο και σε δυναμική CMOS λογική.

Η πλήρης μονάδα μέτρησης των προπορευόμενων μηδενικών αποτελείται από $\log_2 n$ δένδρα πρόβλεψης κρατουμένου με διαφορετικά μεγέθη. Για κάθε δυαδικό ψηφίο Z_i χρησιμοποιείται ένα δένδρο με $n/2^i$ εισόδους. Το Z_0 υπολογίζεται απευθείας από τα δυαδικά ψηφία της εισόδου. Αντίθετα, τα δένδρα πρόβλεψης κρατουμένου που υπολογίζουν τα υπόλοιπα δυαδικά ψηφία του Z οδηγούνται από τους ενδιάμεσους κόμβους ενός δυαδικού δένδρου από OR πύλες, οι οποίες συνδυάζουν σε κάθε επίπεδο τα δυαδικά ψηφία της εισόδου, όπως φάνηκε στο παράδειγμα του σχήματος 7.3. Η υλοποίηση της μονάδας μέτρησης των προπορευόμενων μηδενικών των 16 δυαδικών ψηφίων φαίνεται στο σχήμα 7.4. Κάθε δένδρο πρόβλεψης κρατουμένου αποτελείται από τα γνωστά κελιά ένωσης κρατουμένων (CM) (Σχήμα 3.3). Ο υπολογισμός των ενδιάμεσων λέξεων από το δένδρο των πυλών OR και του πλήθους των προπορευόμενων μηδενικών επικαλύπτονται χρονικά με αποτέλεσμα τα δυαδικά ψηφία Z και το σήμα V να υπολογίζονται σχεδόν ταυτόχρονα.

Μπορούμε επίσης να σχεδιάσουμε ένα πιο απλό κύκλωμα μέτρησης των προπορευόμενων μηδενικών βασιζόμενοι στις ιδιότητες που έχουν τα σήματα διάδοσης κρατουμένου της αρχιτεκτονικής που προτάθηκε στην προηγούμενη παράγραφο. Όπως φαίνεται στο σχήμα 7.4, στο πρώτο επίπεδο του δένδρου πρόβλεψης κρατουμένου που χρησιμοποιείται για τον υπολογισμό του Z_0 , παράγεται ο όρος $A_{15} + \overline{A}_{14} \cdot A_{13}$ όπου το $\overline{A}_{14}$ έχει χρησιμοποιηθεί ως ψηφίο διάδοσης κρατουμένου. Θα μπορούσαμε να έχουμε υπολογίσει τον ίδιο όρο ακόμα και αν χρησιμοποιούσαμε το $A_{15} + A_{14}$ στη θέση του A_{14} , αφού η σχέση $A_{15} + \overline{A}_{14} \cdot A_{13}$ είναι ισοδύναμη με την $A_{15} + \overline{(A_{15} + A_{14})} \cdot A_{13}$. Μπορούμε, όμως, εύκολα να παρατηρήσουμε ότι ο όρος $A_{15} + A_{14}$ έχει ήδη παραχθεί από το δυαδικό δένδρο των πυλών OR και δίνεται σαν είσοδος στο δένδρο πρόβλεψης κρατουμένου που υπολογίζει το δυαδικό ψηφίο Z_1 . Έτσι, συμπεραίνουμε πως οι όροι που παράγονται από τους ενδιάμεσους κόμβους του δυαδικού δένδρου των πυλών

Σχήμα 7.4: Η πρώτη εκδοχή της προτεινόμενης μονάδας μέτρησης των προπορευόμενων μηδενικών των 16 δυαδικών ψηφίων. Το κυκλωμένο κόμματα του κυκλώματος αναφέρεται στην αντίστοιχη μονάδα μέτρησης των 8 δυαδικών ψηφίων.

OR μπορούν να επαναχρησιμοποιηθούν και να λειτουργήσουν σαν διαμοιραζόμενα σήματα διάδοσης κρατούμενου για τα δένδρα πρόβλεψης κρατούμενου που υπολογίζουν τα δυαδικά ψηφία του Z . Αυτή η ιδιότητα ισχύει και για τα επόμενα επίπεδα τόσο των δένδρων πρόβλεψης κρατούμενου όσο και του δυαδικού δένδρου από πύλες OR. Για παράδειγμα, η σχέση $\overline{A_{15}} + A_{14} + A_{13} + A_{12}$ μπορεί να χρησιμοποιηθεί στη θέση του $\overline{A_{14}} \cdot \overline{A_{12}}$ για τον υπολογισμό του όρου $A_{15} + \overline{A_{14}} \cdot A_{13} + \overline{A_{14}} \cdot \overline{A_{12}} \cdot A_{11}$. Το κύκλωμα που ακολουθεί τη λογική των διαμοιρασμένων σημάτων διάδοσης κρατούμενου παρουσιάζεται στο σχήμα 7.5. Η δεύτερη οργάνωση του κυκλώματος μέτρησης των προπορευόμενων μηδενικών που παρουσιάστηκε μοιάζει αρκετά με την αρχιτεκτονική που παρουσιάστηκε από τον Oklobdija στην εργασία [126]. Η κυριότερη διαφορά μεταξύ των δύο κυκλωμάτων είναι πως στην δικιά μας περίπτωση οι περιττές πύλες invert-AND του πρώτου επιπέδου έχουν αφαιρεθεί και οι πολυπλέκτες έχουν μετατραπεί σε απλούστερες πύλες AND-OR.

Ο απευθείας υπολογισμός του πλήθους των προπορευόμενων μηδενικών (Σχήμα 7.4) αναμένεται να είναι ελαφρά γρηγορότερος από το κύκλωμα που διαμοιράζει τα σήματα διάδοσης κρατούμενου (Σχήμα 7.5). Αυτό μπορεί εύκολα να εξηγηθεί αν παρατηρήσουμε το δένδρο

Σχήμα 7.5: Η δεύτερη εκδοχή της προτεινόμενης μονάδας μέτρησης των προπορευόμενων μηδενικών των 16 δυαδικών ψηφίων, η οποία χρησιμοποιεί τη λογική του μοιράσματος των ενδιάμεσων σημάτων διάδοσης κρατούμενου.

πρόβλεψης κρατούμενου που υπολογίζει το δυαδικό ψηφίο Z_0 και στα δύο σχήματα 7.4 και 7.5. Στην πρώτη περίπτωση, το πρώτο επίπεδο από πύλες AND-OR οδηγείται απευθείας από τα δυαδικά ψηφία της εισόδου και μέσω ενός αντιστροφέα. Αντίθετα, στη δεύτερη περίπτωση ο υπολογισμός των πυλών AND-OR ξεκινά αφού οι πύλες OR που συνδυάζουν τα δυαδικά ψηφία της εισόδου έχουν ολοκληρώσει τη λειτουργία τους. Επίσης, οι πύλες του πρώτου κυκλώματος έχουν να οδηγήσουν λιγότερο χωρητικό φορτίο στη έξοδο τους εν' αντιθέση με τις πύλες του δεύτερου κυκλώματος που οδηγούν σε κάθε επίπεδο περισσότερες από δύο πύλες. Η καθυστέρηση και η κατανάλωση ενέργειας των δύο μορφών του προτεινόμενου κυκλώματος μέτρησης των προπορευόμενων μηδενικών θα αναλυθεί σε βάθος στην ενότητα 7.4.

Μέχρι τώρα, η παρουσίαση της σχεδίασης των προτεινόμενων μονάδων μέτρησης των προπορευόμενων μηδενικών περιορίστηκε στις περιπτώσεις όπου το πλήθος των δυαδικών ψηφίων της εισόδου είναι δύναμη του δύο. Όπως θα φανεί σε αυτή την παράγραφο, αυτό έγινε γιατί η σχεδίαση μιας μονάδας μέτρησης των προπορευόμενων μηδενικών για οποιοδήποτε μήκος λέξης της εισόδου n μπορεί εύκολα να προκύψει από την αντίστοιχη μεγέθους 2^k δυαδικών ψηφίων όπου $k = \lceil \log_2 n \rceil$. Για παράδειγμα, η σχεδίαση μιας μονάδας μέτρησης των 11 δυαδικών ψηφίων δεν είναι τίποτε άλλο από μια απλοποιημένη εκδοχή της αντίστοιχης μονάδας των 16 δυαδικών ψηφίων. Για να πετύχουμε αυτή την απλοποίηση το μόνο που χρειάζεται να κάνουμε είναι να δημιουργήσουμε μια εικονική λέξη των 16 δυαδικών ψηφίων, η οποία όμως θα έχει τον ίδιο αριθμό από προπορευόμενα μηδενικά, όπως και η λέξη των

Σχήμα 7.6: Ένα αριθμητικό παράδειγμα της κωδικοποίησης των 3 προπορευόμενων μηδενικών μιας λέξης των 11 δυαδικών ψηφίων.

11 δυαδικών ψηφίων που μας ενδιαφέρει. Αυτό μπορεί να συμβεί αν τοποθετήσουμε τα 11 δυαδικά ψηφία της λέξης που θέλουμε να επεξεργαστούμε στις 11 πιο σημαντικές θέσεις της αντίστοιχης λέξης των 16 δυαδικών ψηφίων και θεωρήσουμε πως τα εναπομείναντα 5 δυαδικά ψηφία είναι ίσα με 0. Ένα παράδειγμα της εφαρμογής αυτής της μεθόδου φαίνεται στο σχήμα 7.6. Η αρχική λέξη των 11 δυαδικών ψηφίων έχει 3 προπορευόμενα μηδενικά, τα οποία κωδικοποιούνται σωστά από το προτεινόμενο κύκλωμα μέτρησης ($Z_3Z_2Z_1Z_0 = 0011$). Το απλοποιημένο κύκλωμα μέτρησης των 11 δυαδικών ψηφίων μπορεί να παραχθεί αν διαδόσουμε τα 5 λιγότερα σημαντικά μηδενικά της εισόδου των 16 δυαδικών ψηφίων και διαγράψουμε τις πύλες που υπολογίζουν σε σταθερή τιμή. Η υλοποίηση της προτεινόμενης μονάδας μέτρησης των προπορευόμενων μηδενικών μεγέθους 11 δυαδικών ψηφίων που ακολουθεί τη δομή του διαμοιρασμένου σήματος διάδοσης κρατουμένου φαίνεται στο σχήμα 7.7.

7.3 Πρόβλεψη της θέσης του προπορευόμενου ψηφίου

Το κύκλωμα πρόβλεψης της θέσης του ψηφίου κανονικοποίησης προσπαθεί να προβλέψει από τα δυαδικά ψηφία της εισόδου A και B μια λέξη των n δυαδικών ψηφίων (ψευδο-αποτέλεσμα), η οποία θα έχει τον ίδιο αριθμό από προπορευόμενα ψηφία (μηδενικά ή άσσους) σε σχέση με το πραγματικό αποτέλεσμα. Όταν το αποτέλεσμα είναι αρνητικό προκύπτουν προπορευόμενοι άσσοι $11 \dots 10\{x\}$, ενώ όταν το αποτέλεσμα είναι θετικό προκύπτουν προπορευόμενα μηδενικά $00 \dots 01\{x\}$. Η λέξη πρόβλεψης υπολογίζεται χρησιμοποιώντας τα σήματα διάδοσης P_i , γέννησης G_i , και εξολόθρευσης κρατουμένου K_i που προκύπτουν από τα δυαδικά ψηφία της εισόδου A_i και B_i .

$$T_i = A_i \oplus B_i \quad G_i = A_i \cdot B_i \quad K_i = \overline{A_i} \cdot \overline{B_i} \quad (7.22)$$

Ενδιαφερόμαστε για την γενική περίπτωση της LZA λογικής, η οποία είναι σε θέση να καθορίσει τη θέση του ψηφίου κανονικοποίησης είτε αυτός είναι άσσος (πρόβλεψη αρνητικού αποτελέσματος) είτε είναι μηδέν (πρόβλεψη θετικού αποτελέσματος) χωρίς να γνωρίζει εξ αρχής

Σχήμα 7.7: Η υλοποίηση σε επίπεδο πυλών της προτεινόμενης μονάδας μέτρησης των προπορευόμενων μηδενικών των 11 δυαδικών ψηφίων, η οποία ακολουθεί τη λογική της διαμοίρασης των σημάτων διάδοσης κρατουμένου.

το πρόσημο του πραγματικού αποτελέσματος.

Σύμφωνα με τη μέθοδο που παρουσιάστηκε από τον Knowles στην εργασία [83], κάθε δυαδικό ψηφίο της λέξης πρόβλεψης είναι ίσο με την τιμή ενός δείκτη f_i , $0 \leq i \leq n-1$. Ο δείκτης f_i προβλέπει τη θέση του προπορευόμενου ψηφίου χρησιμοποιώντας την πληροφορία που είναι διαθέσιμη από τα γειτονικά δυαδικά ψηφία των θέσεων $i+1$ και $i-1$. Ο ορισμός του δείκτη δίνεται από τη σχέση (7.23).

$$f_i = T_{i+1} \cdot (G_i \cdot \overline{K}_{i-1} + K_i \cdot \overline{G}_{i-1}) + \overline{T}_{i+1} \cdot (K_i \cdot \overline{K}_{i-1} + G_i \cdot \overline{G}_{i-1}) \quad \text{και} \quad f_{n-1} = \overline{T}_{n-1} \cdot T_{n-2} \quad (7.23)$$

Όταν ο δείκτης στη θέση i είναι αληθής και κανένας άλλος πιο σημαντικός δείκτης δεν είναι ταυτόχρονα αληθής σημαίνει ότι το προπορευόμενο ψηφίο βρίσκεται είτε στη θέση i είτε στην αμέσως λιγότερο σημαντική $i-1$. Ένα παράδειγμα ανίχνευσης της θέσης του προπορευόμενου ψηφίου φαίνεται στο σχήμα 7.8(α).

Μια άλλη μέθοδος για την πρόβλεψη της θέσης του προπορευόμενου ψηφίου προτάθηκε στην εργασία [140] και υλοποιήθηκε στην [96]. Η εναλλακτική αυτή μέθοδος χρησιμοποιεί δύο χωριστές μονάδες πρόβλεψης, η μία προβλέπει την περίπτωση το προπορευόμενο ψηφίο να είναι μηδέν και η άλλη να είναι ένα. Οι έξοδοι των δύο μονάδων συνδυάζονται τελικά για να παράγουν μια ενιαία λέξη πρόβλεψης. Κάθε μια μονάδα χρησιμοποιεί τους δικούς της δείκτες. Στην περίπτωση που αναζητούμε προπορευόμενα μηδενικά χρησιμοποιούμε τον δείκτη f_i^z , ο οποίος ισούται με $T_i \oplus \overline{K}_{i-1}$. Στην αντίθετη περίπτωση, χρησιμοποιούμε τον δείκτη f_i^o , ο οποίος

A :	0	0	0	0	1	0	0	0	0	1	0	0
B :	1	1	1	1	1	0	0	0	0	1	0	0
Propagate	T	1	1	1	1	0	0	0	0	0	0	0
Generate	G	0	0	0	0	1	0	0	0	0	1	0
Kill	K	0	0	0	0	0	1	1	1	1	0	1
Indicator	f	0	0	0	0	0	0	0	0	1	1	0
Real Sum		0	0	0	0	0	0	0	0	1	0	0

Correctly predicted Leading Digit

(α)

A :	0	0	0	0	1	0	0	0	0	1	0	0
B :	1	1	1	1	1	0	0	0	0	1	0	0
Propagate	T	1	1	1	1	0	0	0	0	0	0	0
Generate	G	0	0	0	0	1	0	0	0	0	1	0
Kill	K	0	0	0	0	0	1	1	1	1	0	1
Indicator	f ^z	0	0	0	0	0	0	0	0	1	0	0
Indicator	f ^o	0	0	0	1	1	1	1	1	0	1	1
OR from MSB for f ^z		0	0	0	0	0	0	0	0	1	1	1
OR from MSB for f ^o		0	0	0	1	1	1	1	1	1	1	1
Bitwise AND		0	0	0	0	0	0	0	0	1	1	1
Real Sum		0	0	0	0	0	0	0	0	1	0	0

Correctly predicted Leading Digit

(β)

Σχήμα 7.8: Αριθμητικά παραδείγματα της λειτουργίας των ήδη γνωστών μονάδων πρόβλεψης. (α) Η περίπτωση του ενιαίου δείκτη και (β) των χωριστών δεικτών.

ορίζεται ως εξής $T_i \oplus \overline{G}_{i-1}$. Αρχικά οι δείκτες σε κάθε μονάδα γίνονται OR από αριστερά προς τα δεξιά με σκοπό τη δημιουργία δύο λέξεων, οι οποίες ακολουθούν τη μονότονα αύξουσα μορφή $0 \dots 0011 \dots 1$. Πιο συγκεκριμένα, το i -οστό δυαδικό ψηφίο της λέξης που αφορά την περίπτωση των προπορευόμενων μηδενικών είναι ίσο με

$$f_{n-1}^z + f_{n-2}^z + \dots + f_{i+1}^z + f_i^z. \quad (7.24)$$

Το ίδιο ισχύει και για την περίπτωση των προπορευόμενων άσσων αντικαθιστώντας στην εξίσωση (7.24) τους δείκτες f_i^z με τους αντίστοιχους f_i^o . Στη συνέχεια, οι δύο λέξεις που έχουν υπολογιστεί χωριστά από τις δύο μονάδες ενώνονται σε μία μέσω της λογικής πράξης AND. Το πιο σημαντικό ψηφίο, που είναι ίσο με 1 της νέας λέξης που προκύπτει υποδηλώνει τη θέση του προπορευόμενου ψηφίου. Αν η πρόβλεψη είναι λάθος, τότε το προπορευόμενο ψηφίο βρίσκεται στην αμέσως λιγότερη σημαντική θέση προς τα δεξιά. Στο σχήμα 7.8(β) παρουσιάζεται ένα παράδειγμα της λειτουργίας της δεύτερης τεχνικής πρόβλεψης που μόλις περιγράψαμε.

Η πρόβλεψη της θέσης του προπορευόμενου ψηφίου μπορεί να αντιμετωπιστεί και σαν πρόβλημα ανίχνευσης προτύπων (pattern matching) χρησιμοποιώντας ειδικά κατασκευασμένα αυτόματα πεπερασμένων καταστάσεων (Finite State Machine) [67]. Αυτή όμως η εκδοχή δεν

χρησιμοποιείται πλέον γιατί οδηγεί σε λύσεις με χειρότερη απόδοση τόσο από πλευράς εμβαδού χωροθέτησης όσο και καθυστέρησης [140].

Αφού έχει ολοκληρωθεί η πρόβλεψη της θέσης του προπορευόμενου ψηφίου, πρέπει η θέση αυτή να κωδικοποιηθεί σε δυαδική τιμή, ώστε να καθορίσει την κανονικοποίηση του πραγματικού αποτελέσματος. Οποιαδήποτε από τις μονάδες μέτρησης των προπορευόμενων μηδενικών που παρουσιάστηκαν στις ενότητες 6.1 και 6.2 μπορεί να χρησιμοποιηθεί για την αναγκαία κωδικοποίηση. Στην περίπτωση όμως της LZA λογικής, που χρησιμοποιεί δύο χωριστές μονάδες πρόβλεψης, δεν απαιτείται η υλοποίηση ενός πλήρους μετρητή, αλλά, είναι πιο συμφέρον να χρησιμοποιηθεί το τελευταίο μέρος του κυκλώματος μέτρησης που στηρίζεται στη λογική του κωδικοποιητή. Αυτό μπορεί να συμβεί γιατί η λέξη πρόβλεψης αυτής της μεθόδου είναι ήδη σε μονότονα αύξουσα μορφή. Στην περίπτωση που η μονάδα LZA υλοποιηθεί σε δυναμική λογική, το κύκλωμα με τις δύο ανεξάρτητες μονάδες πρόβλεψης αναμένεται να δώσει υλοποιήσεις με μικρότερη καθυστέρηση, γιατί χρησιμοποιεί απλούστερες λογικές σχέσεις για τους δείκτες f^z και f^o και βασίζεται σε λογικές σχέσεις OR, η υλοποίηση των οποίων γίνεται πολύ απλή με τη χρήση δυναμικών πυλών. Σε υλοποιήσεις στατικής CMOS, ο ενιαίος δείκτης της σχέσης (7.23) μαζί με μια γρήγορη μονάδα μέτρησης, σαν αυτή που προτείνουμε, είναι η πιο ενδεδειγμένη επιλογή. Πάντως, και στις δύο περιπτώσεις τα κυκλώματα πρέπει να βελτιστοποιηθούν στο μέγιστο βαθμό, ώστε η έξοδος τους να μπορεί να υπολογίζεται ταυτόχρονα ή συντομότερα από αυτή του κανονικού αποτελέσματος.

7.3.1 Οι προτεινόμενες μονάδες πρόβλεψης

Στη συνέχεια θα παρουσιάσουμε μια νέα τεχνική για το σχεδιασμό της LZA λογικής, η οποία, όπως θα αποδειχθεί στη συνέχεια, είναι πιο αποδοτική ανεξάρτητα με το αν η υλοποίηση πραγματοποιηθεί σε στατική ή δυναμική λογική CMOS. Η νέα μονάδα βασίζεται στη λειτουργία της λογικής LZA με τις δύο μονάδες πρόβλεψης, χωρίς όμως να απαιτεί τον υπολογισμό μονότονα αύξουσων ακολουθιών οι οποίες δηλώνουν τη θέση του προπορευόμενου ψηφίου. Στην προτεινόμενη περίπτωση η δυαδική αναπαράσταση της θέσης του προπορευόμενου ψηφίου υπολογίζεται απευθείας από τους δείκτες f^z και f^o .

Στην αρχική εκδοχή της LZA λογικής με τις δύο μονάδες πρόβλεψης, το πρώτο βήμα του αλγορίθμου ήταν να υπολογίσει από τους αντίστοιχους δείκτες δύο μονότονα αύξουσες ακολουθίες της μορφής $00 \dots 01 \dots 11$ σύμφωνα με τη σχέση (7.24) (Σχήμα 7.8(β)). Μπορούμε εύκολα να παρατηρήσουμε ότι η (7.24) έχει ακριβώς την ίδια μορφή με τον ορισμό της λέξης S όπως αυτός δίνεται στην εξίσωση (7.1). Η μόνη διαφορά είναι ότι στη θέση των δυαδικών ψηφίων του A εμφανίζονται οι δείκτες για τη περίπτωση των προπορευόμενων μηδενικών f^z . Έχουμε δείξει πως ξεκινώντας από τη μονότονα αύξουσα ακολουθία S μπορούμε να δημιουργήσουμε πολύ απλές σχέσεις για τον υπολογισμό του πλήθους των προπορευόμενων μηδενικών απευθείας από τα δυαδικά ψηφία της εισόδου. Έτσι, συμπεραίνουμε ότι η δημιουργία μονότονα αύξουσων λέξεων από τους δείκτες f^z ή f^o είναι περιττή και ότι το πλήθος των προπορευόμενων ψηφίων μπορεί να υπολογιστεί απευθείας από τους δείκτες f^z ή f^o αντίστοιχα. Μια τέτοια βελτιστοποίηση δεν ήταν δυνατή έως τώρα γιατί δεν υπήρχε καμιά σχέση που να συνδέει τα δυαδικά ψηφία του πλήθους των προπορευόμενων μηδενικών με τα δυαδικά ψηφία της εισόδου.

Η προτεινόμενη LZA λογική αποτελείται από δύο επιμέρους μονάδες πρόβλεψης. Η μια ανιχνεύει την περίπτωση των προπορευόμενων μηδενικών χρησιμοποιώντας τους δείκτες f^z ενώ

η δεύτερη ανιχνεύει τους προπορευόμενους άσσους μέσω των δεικτών f^o . Αφού οι τιμές των δεικτών υπολογιστούν, δίνονται σε δύο χωριστά κυκλώματα μέτρησης των προπορευόμενων μηδενικών χωρίς περαιτέρω επεξεργασία. Μ' αυτόν το τρόπο υπολογίζονται δύο δυαδικές αναπαραστάσεις των προπορευόμενων ψηφίων (είτε μηδενικών είτε άσσων), εκ των οποίων μόνο μία είναι σωστή. Ποιά από τις δύο αναπαραστάσεις περιέχει τελικά την σωστή πληροφορία, μπορεί να επιλεγεί με δύο τρόπους.

- Ο πρώτος τρόπος επιλογής της δυαδικής αναπαράστασης που κωδικοποιεί σωστά τα προπορευόμενα ψηφία χρησιμοποιεί το πρόσημο του πραγματικού αποτελέσματος. Εφόσον το αποτέλεσμα είναι θετικό, τότε επιλέγεται η κωδικοποίηση των δεικτών f^z . Στην αντίθετη περίπτωση επιλέγουμε την κωδικοποίηση των δεικτών f^o , που προβλέπουν τους προπορευόμενους άσσους.
- Ο δεύτερος τρόπος επιλογής της σωστής αναπαράστασης του πλήθους των προπορευόμενων ψηφίων επιβάλλει τη σύγκριση των δύο προβλέψεων και την επιλογή του μεγαλύτερου. Η ορθότητα αυτής της προσέγγισης μπορεί πολύ εύκολα να αποδειχθεί αν παρατηρήσουμε μια κρυμμένη λεπτομέρεια στη λειτουργία του αρχικού LZA αλγορίθμου με δύο μονάδες πρόβλεψης. Θυμηθείτε πως το τελευταίο βήμα του αλγορίθμου προσπαθούσε να συνενώσει τις δύο μονότονα αύξουσες λέξεις πρόβλεψης σε μία. Η τελική λέξη υπολογιζόταν εφαρμόζοντας τη λογική πράξη AND μεταξύ των δύο λέξεων (σχήμα 7.8(β)). Έτσι, η τελική λέξη πρόβλεψης ήταν ίση με τη λέξη που είχε το μικρότερο αριθμό από συνεχόμενους άσσους στο λιγότερο σημαντικό κομμάτι της ή αντίθετα με τη λέξη που έχει το μεγαλύτερο αριθμό από προπορευόμενα μηδενικά. Στη δικιά μας περίπτωση δεν έχουμε μονότονα αύξουσες λέξεις αλλά την αντίστοιχη κωδικοποίηση των προπορευόμενων ψηφίων. Έτσι, εφόσον θέλουμε να πετύχουμε την ίδια λειτουργία με τον αρχικό αλγόριθμο, αν επιλέξουμε την μεγαλύτερη αριθμητικά κωδικοποίηση των προπορευόμενων ψηφίων είναι ισοδύναμο με το να εκτελούμε τη λογική πράξη AND. Ένα παράδειγμα της λειτουργία της προτεινόμενης μονάδας LZA φαίνεται στο σχήμα 7.9.

7.3.2 Αντιμετώπιση του λάθους πρόβλεψης

Όπως ήδη έχουμε αναφέρει σε συγκεκριμένες περιπτώσεις, η πρόβλεψη της θέσης του ψηφίου κανονικοποίησης μπορεί να διαφέρει από την πραγματική κατά μία θέση. Τότε το τελικό αποτέλεσμα θα είναι λάθος, αν κανονικοποιηθεί με βάση την τιμή της πρόβλεψης και θα απαιτείται μία επιπλέον ολίσθηση προς τ' αριστερά ή τα δεξιά, ανάλογα με τη λειτουργία της μονάδας πρόβλεψης. Οι μονάδες πρόβλεψης που έχουμε παρουσιάσει στην περίπτωση λάθους υπερεκτιμούν τη θέση του ψηφίου κανονικοποίησης και το αποτέλεσμα πρέπει να μετακινηθεί μία θέση προς τα δεξιά. Ταυτόχρονα, στην περίπτωση αυτή, ο εκθέτης του αποτελέσματος θα πρέπει να αυξηθεί κατά ένα. Το λάθος αυτό της πρόβλεψης μπορεί να αντιμετωπιστεί με τρεις διαφορετικούς τρόπους.

Η πρώτη μέθοδος (LZE I) αντιμετώπισης του προβλήματος είναι η πιο απλή και δίνει σχετικά καλά αποτελέσματα. Εφόσον υπάρχει περίπτωση να έχει υπερεκτιμηθεί ο αριθμός των προπορευόμενων ψηφίων, τότε στον ολισθητή δίνεται η τιμή του μεγέθους ολισθημένης εξαρχής μία θέση προς τα δεξιά. Επομένως, αν έχει γίνει λάθος το τελικό αποτέλεσμα της ολίσθησης θα είναι σωστό. Αν όμως το ποσό της ολίσθησης είχε προβλεφθεί σωστά τότε θα

Σχήμα 7.9: Ένα αριθμητικό παράδειγμα της λειτουργίας της προτεινόμενης LZA μονάδας, η οποία χρησιμοποιεί δύο χωριστούς δείκτες πρόβλεψης και τις προτεινόμενες μονάδες μέτρησης των προπορευόμενων μηδενικών. Θεωρούμε ότι η πραγματική πράξη που εκτελείται είναι αφαίρεση και ότι $A > B$ ενώ τα δυαδικά ψηφία του B είναι ήδη ανεστραμμένα. Το τελικό αποτέλεσμα είναι θετικό πράγμα που σημαίνει ότι η κωδικοποίηση των δεικτών f^Z περιέχει την απαιτούμενη πληροφορία για την κανονικοποίηση.

απαιτείται μία επιπλέον ολίσθηση προς τ' αριστερά. Το μόνο που έχουμε να κάνουμε είναι να ελέγξουμε το πιο σημαντικό ψηφίο της εξόδου του ολισθητή κανονικοποίησης. Αν είναι ίσο με 0 σημαίνει ότι απαιτείται ακόμη μια ολίσθηση ενώ αν είναι ίσο με 1, το αποτέλεσμα είναι σωστά κανονικοποιημένο. Για να μειώσουμε την καθυστέρηση της διαδικασίας μπορούμε να ελέγξουμε ενδιάμεσα σήματα του ολισθητή για να προ-υπολογίσουμε παράλληλα με την ολίσθηση την τιμή του πιο σημαντικού ψηφίου.

Η δεύτερη μέθοδος (LZE II) συνδυάζει την πληροφορία που παράγεται από τη λογική του LZA και σήματα από τον αθροιστή/αφαιρέτη που υπολογίζει το πραγματικό αποτέλεσμα ώστε να εντοπίσει συντομότερα (πριν ξεκινήσει η κανονικοποίηση) την περίπτωση πιθανού λάθους. Στη μια περίπτωση, χρησιμοποιείται η one-hot αναπαράσταση της λέξης πρόβλεψης, εάν είναι διαθέσιμη, η οποία συνδυάζεται με τα δυαδικά ψηφία του πραγματικού αποτελέσματος μέσα από την λογική πράξη AND, όπως φαίνεται στο παράδειγμα που ακολουθεί.

Πραγματικό Αποτέλεσμα: 00001010110

Λάθος πρόβλεψη: 00010000000

Λογική πράξη AND: 00000000000

Έτσι, αν το αποτέλεσμα της λογικής πράξης AND αποτελείται μόνο από μηδενικά ψηφία τότε η λέξη πρόβλεψης που υπολογίστηκε δεν είναι σωστή. Η περίπτωση αυτή μπορεί να ανιχνευτεί

από ένα απλό κύκλωμα, το οποίο λειτουργεί παράλληλα με τον ολισθητή κανονικοποίησης και θα κατευθύνει τη διόρθωση του αποτελέσματος [124]. Θυμίζουμε ότι θεωρούμε πως η λογική LZA πάντα υπερεκτιμά τη θέση του leading digit. Δηλαδή, αν στο πραγματικό αποτέλεσμα η θέση του πιο σημαντικού άσσου βρισκόταν στη θέση i τότε στην περίπτωση λανθασμένης πρόβλεψης το LZA θα έδειχνε ότι ο πιο σημαντικός άσος βρίσκεται στη θέση $i + 1$. Σε μια άλλη περίπτωση, [134], [133] αντί για το τελικό άθροισμα χρησιμοποιούνται τα κρατούμενα που παράγονται κατά τον υπολογισμό του πραγματικού αποτελέσματος και συνδυάζονται με τους δείκτες f για τον υπολογισμό νέων σημάτων ένδειξης λάθους e_i . Αν έστω και ένα από τα σήματα εντοπισμού του λάθους ενεργοποιηθεί τότε η πρόβλεψη του πιο σημαντικού άσσου έχει πέσει έξω κατά μία θέση. Η συνθήκη αυτή μπορεί επίσης να υλοποιηθεί εύκολα με ένα δένδρο από OR πύλες που θα δέχεται σαν είσοδο τα δυαδικά ψηφία e_i και το οποίο θα λειτουργεί παράλληλα με τον ολισθητή κανονικοποίησης.

Η τελευταία μέθοδος (LZE III) παράγει ένα σήμα λάθος πρόβλεψης παράλληλα με την LZA λογική και τον αθροιστή που παράγει το τελικό αποτέλεσμα [18], [173]. Αυτή η προσέγγιση χρησιμοποιεί τους δείκτες της LZA λογικής για να ανιχνεύσει συγκεκριμένες περιπτώσεις της εισόδου, οι οποίες προκαλούν το λάθος. Τα κυκλώματα που έχουν μέχρι τώρα προταθεί έχουν σημαντική καθυστέρηση και απαιτούν μεγάλο εμβαδόν χωροθέτησης σε σχέση με τις λύσεις LZE I και II και δεν αναμένεται να χρησιμοποιηθούν στο μέλλον αν πρώτα δεν βελτιωθούν σημαντικά.

Η προτεινόμενη μονάδα κανονικοποίησης αντιμετωπίζει το λάθος της πρόβλεψης με έναν απλό αλλά αποτελεσματικό τρόπο. Το μόνο κύκλωμα που προσθέτουμε είναι αυτό που υπολογίζει το λιγότερο σημαντικό ψηφίο του πλήθους των προπορευόμενων μηδενικών του πραγματικού μη κανονικοποιημένου αποτελέσματος. Δεν είναι τίποτε άλλο από ένα δυαδικό δένδρο από απλές AND-OR πύλες. Το κύκλωμα αυτό λειτουργεί παράλληλα με τον ολισθητή κανονικοποίησης και η έξοδος του οδηγεί ένα επιπλέον επίπεδο ολίσθησης προς τα δεξιά, ώστε να διορθώσει το πιθανό λάθος. Αντίστοιχα, θα μπορούσε να συνδυαστεί με το λιγότερο σημαντικό ψηφίο της λέξης πρόβλεψης του πλήθους των προπορευόμενων μηδενικών και να καθορίσει αντίστοιχα την κατεύθυνση (δεξιά η αριστερά) της ολίσθησης του τελευταίου σταδίου του ολισθητή. Η λύση είναι απλή στη σχεδίαση της και απαιτεί λιγότερο υλικό από την έως τώρα πιο αποδοτική μέθοδο LZE II.

7.4 Βελτιστοποίηση της ενέργειας και της καθυστέρησης

Οι προτεινόμενες αρχιτεκτονικές σχεδιάστηκαν στην τεχνολογία υλοποίησης των 130nm, θεωρώντας τυπικές συνθήκες λειτουργίας και κατασκευής. Για να εξερευνήσουμε την ενέργεια και την καθυστέρηση των κυκλωμάτων, χρησιμοποιήσαμε τη διαδικασία βελτιστοποίησης που περιγράψαμε στο κεφάλαιο 2. Οι βελτιστοποιήσεις πραγματοποιήθηκαν θεωρώντας ότι ο λόγος της χωρητικότητας εξόδου και εισόδου του κυκλώματος είναι σε όλες τις περιπτώσεις 100/25fF. Η θεώρηση αυτή είναι αρκετά ρεαλιστική και καλύπτει ένα ευρύ φάσμα σχεδιασμών μονάδων επεξεργασίας δεδομένων [50]. Οι αντίστοιχες μετρήσεις της ενέργειας και της καθυστέρησης για όλα τα υπο-εξέταση κυκλώματα πραγματοποιήθηκαν σε Spice λαμβάνοντας υπόψη και την αντίστοιχη επιβάρυνση των γραμμών διασύνδεσης.

Αρχικά συγκρίναμε τις προτεινόμενες μονάδες μέτρησης των προπορευόμενων μηδενικών των 64 δυαδικών ψηφίων με την πιο αποδοτική έως τώρα λύση για υλοποιήσεις σε στατική CMOS [126]. Η καθυστέρηση και η ενέργεια των υπο-εξέταση κυκλωμάτων φαίνεται στο

Σχήμα 7.10: Οι καμπύλες της ενέργειας ως προς την καθυστέρηση για τις δύο εκδοχές των προτεινόμενων μονάδων μέτρησης των προπορευόμενων μηδενικών των 64 δυαδικών ψηφίων και της αρχιτεκτονικής, η οποία προτάθηκε στην εργασία [126].

σχήμα 7.10. Από το διάγραμμα μπορούμε να παρατηρήσουμε ότι η προτεινόμενη μεθοδολογία οδηγεί αφενός σε ταχύτερα κυκλώματα (της τάξης του 7%) και αφετέρου σε σημαντικά μικρότερη κατανάλωση ενέργειας σε σχέση με τα κυκλώματα που σχεδιάζονται με βάση τη δομή που προτάθηκε στην εργασία [126]. Για ίση καθυστέρηση το κέρδος σε κατανάλωση ενέργειας κυμαίνεται μεταξύ 15% και 51%. Η απόδοση των προτεινόμενων κυκλωμάτων προέρχεται από το μειωμένο αριθμό πυλών που απαιτείται για την υλοποίηση της ίδιας λειτουργίας και από το γεγονός ότι στο κρίσιμο μονοπάτι εμφανίζονται ταχύτερες πύλες σε σχέση με το [126]. Από το ίδιο σχήμα μπορούμε να παρατηρήσουμε πως όταν η καθυστέρηση είναι μικρότερη των 8.5FO4, η τεχνική διαμοίρασης του σήματος διάδοσης κρατουμένου δίνει κυκλώματα τα οποία απαιτούν περισσότερη ενέργεια σε σχέση με την απευθείας υλοποίηση της μονάδας μέτρησης. Ο λόγος γι' αυτή τη συμπεριφορά είναι ότι το κύκλωμα με τα διαμοιρασμένα σήματα διάδοσης κρατουμένου απαιτεί μεγαλύτερα μεγέθη πυλών για να πετύχει μικρές καθυστερήσεις, εξαιτίας του μεγαλύτερου χωρητικού φορτίου στην έξοδο της κάθε πύλης. Αυτό δεν ισχύει για καθυστερήσεις μεγαλύτερες από 8.5FO4, όπου η δεύτερη εκδοχή των προτεινόμενων κυκλωμάτων οδηγεί στις πιο συμφέρουσες υλοποιήσεις. Σ' αυτό το σημείο οφείλουμε να σημειώσουμε πως δεν συμπεριλάβαμε μετρήσεις για τη συμπεριφορά των μονάδων μέτρησης που βασίζονται σε κωδικοποιητή και στη δομή που προτάθηκε στην εργασία [93]. Πήραμε αυτή την απόφαση γιατί τα πειράματα που πραγματοποιήσαμε σε στατική CMOS για τις μονάδες αυτές, έδειξαν, ότι η ελάχιστη δυνατή καθυστέρηση τους είναι σχεδόν διπλάσια από τις καθυστερήσεις των πρώτων κυκλωμάτων που εξετάσαμε (περίπου 11FO4). Λόγω της πολύ μεγάλης διαφοράς στην καθυστέρηση οποιαδήποτε άλλη σύγκριση δεν θα είχε νόημα .

Η ελάχιστη καθυστέρηση των 6.6FO4 την οποία πετυχαίνει η προτεινόμενη μονάδα μέτρησης των προπορευόμενων μηδενικών επιτρέπει με άνεση την εισαγωγή οποιονδήποτε δεικτών

Σχήμα 7.11: Η ενέργεια και η καθυστέρηση για τις δύο μορφές της LZA λογικής των 64 δυαδικών ψηφίων υλοποιημένης σε στατική CMOS και η αντίστοιχη καμπύλη για τους αθροιστές παράλληλου προθέματος που υπολογίζουν τα κρατούμενα του Ling.

πρόβλεψης¹. Για να μπορέσουμε να ποσοτικοποιήσουμε την χρονική επιβάρυνση που εισάγει η LZA λογική, υλοποιήσαμε σε στατική CMOS την προτεινόμενη μονάδα πρόβλεψης των προπορευόμενων μηδενικών θεωρώντας ότι οδηγείται από την έξοδο του κυκλώματος που υπολογίζει τους ενοποιημένους δείκτες της σχέσης (7.23). Ο υπολογισμός των f_i απαιτεί τουλάχιστον τρία επιπλέον επίπεδα λογικής, τα οποία επηρεάζουν όπως είναι φυσικό το κρίσιμο μονοπάτι του κυκλώματος. Η καθυστέρηση και η ενέργεια του συνδυασμένου κυκλώματος πρόβλεψης και μέτρησης φαίνεται στο σχήμα 7.11. Σύμφωνα με τις τιμές που απεικονίζονται στο σχήμα 7.11, το προτεινόμενο κύκλωμα μέτρησης που οδηγείται από τους συνδυασμένους δείκτες f_i μπορεί να σχεδιαστεί ώστε να είναι ταχύτερο από τον πιο γρήγορο αθροιστή του κεφαλαίου 3 και απαιτεί τουλάχιστον $2.8\times$ λιγότερη ενέργεια. Οι τιμές της ενέργειας και της καθυστέρησης του αθροιστή που φαίνονται στο σχήμα 7.11 αναμένονται να είναι μεγαλύτερες αν συνυπολογίσουμε την επιβάρυνση σε καθυστέρηση και ενέργεια που εισάγει η επανατροφοδότηση του κρατουμένου, ώστε να υπολογιστεί η απόλυτη τιμή του τελικού αποτελέσματος που είναι αναγκαία στην περίπτωση της αναπαράστασης προσημασμένου μεγέθους (Κεφάλαιο 6).

Οι επιδόσεις της προτεινόμενης μονάδας πρόβλεψης, η οποία χρησιμοποιεί τους απλοποιημένους δείκτες f_i^z και f_i^o και δύο ανεξάρτητες μονάδες μέτρησης, φαίνονται επίσης στο σχήμα 7.11. Η προτεινόμενη μονάδα πρόβλεψης πετυχαίνει τη μικρότερη καθυστέρηση απαιτώντας τουλάχιστον 30% περισσότερη ενέργεια από αυτήν που χρησιμοποιεί τους συνδυασμένους δείκτες f_i . Σημειώνουμε πως για την υλοποίηση της προτεινόμενης μονάδας πρόβλεψης η σωστή κωδικοποίηση του πλήθους των προπορευόμενων μηδενικών (σχήμα 7.9) επιλέγεται με τη χρήση

¹Ο χρόνος που έχουμε στη διάθεση μας καθορίζεται από την ταχύτητα του αθροιστή, ο οποίος υπολογίζει το πραγματικό μη κανονικοποιημένο αποτέλεσμα, και εκτιμάται πως είναι μεγαλύτερη από 10FO4 στην περίπτωση των 64 δυαδικών ψηφίων

Σχήμα 7.12: Οι καμπύλες της ενέργειας ως προς την καθυστέρηση για τη μονάδα μέτρησης των 64 δυαδικών ψηφίων που η σχεδίαση σε δυναμική CMOS βασίστηκε στα κυκλώματα κωδικοποιητή και στην προτεινόμενη δομή.

2-σε-1 πολυπλεκτών, σύμφωνα με την τιμή του πραγματικού προσήμου του αποτελέσματος.

Οι μονάδες μέτρησης των προπορευόμενων μηδενικών που βασίζονται στη χρήση κωδικοποιητή υλοποιούνται αποδοτικότερα σε δυναμική CMOS τεχνολογία. Σχεδιάσαμε μια τέτοια μονάδα μέτρησης των 64 δυαδικών ψηφίων χρησιμοποιώντας δυναμικές πύλες OR των 8 δυαδικών ψηφίων². Η δομή του κυκλώματος είναι όμοια με αυτή που παρουσιάστηκε στην εργασία [96]. Οι φάσεις προφόρτισης και υπολογισμού των δυναμικών πυλών καθορίζονται από επικαλυπτόμενες φάσεις ρολογιού ακολουθώντας τις σχεδιαστικές οδηγίες που περιγράφονται στο [60], με σκοπό την ελαχιστοποίηση της χρονικής επιβάρυνσης. Η ενέργεια και η καθυστέρηση της μονάδας αυτής παρουσιάζονται στο σχήμα 7.12. Επίσης, υλοποιήσαμε σε δυναμική CMOS και την προτεινόμενη μονάδα μέτρησης των προπορευόμενων μηδενικών των 64 δυαδικών ψηφίων, η δομή της οποίας απεικονίζεται στο σχήμα 7.4. Για τον υπολογισμό του πλήθους των προπορευόμενων μηδενικών οι πύλες AND-OR των σταδίων ένωσης του κρατούμενου απαιτούν στην είσοδο τους τόσο κανονικά όσο και ανεστραμμένα σήματα. Το γεγονός αυτό περιπλέκει τη σχεδίαση με μονές δυναμικές πύλες (single-rail dynamic logic) και έτσι διαλέξαμε να σχεδιάσουμε την προτεινόμενη μονάδα χρησιμοποιώντας διαφορικές δυναμικές πύλες (differential, dual-rail dynamic logic), ώστε να αποφύγουμε τα όποια προβλήματα μονοτονικότητας³. Το κύκλωμα αποτελείται από δυναμικές και στατικές πύλες

²Η σχεδίαση δυναμικών πυλών OR με περισσότερες εισόδους αποφεύγεται στις σύγχρονες τεχνολογίες εξαιτίας του αυξημένου ρεύματος υποκατωφλίου που ρέει στα παράλληλα NMOS τρανζίστορ προς τη γή με αποτέλεσμα τον κίνδυνο αποφόρτισης του δυναμικού κόμβου. Η ανθεκτικότητα του δυναμικού κόμβου μπορεί φυσικά να βελτιωθεί αυξάνοντας το μέγεθος του τρανζίστορ κρατήματος (keeper). Κάτι τέτοιο, όμως, δεν επηρεάζει αρνητικά την ταχύτητα της δυναμικής πύλης και αυξάνει σημαντικά την κατανάλωση ενέργειας.

³Η υλοποίηση των προτεινόμενων κυκλωμάτων είναι εφικτή και με μονές δυναμικές πύλες, αν χρησιμοποιήσουμε έξυπνα τις φάσεις προφόρτισης και υπολογισμού των πυλών. Οι τεχνικές αυτές όμως είναι λιγότερο ασφαλείς και πρέπει να χρησιμοποιούνται με πολύ μεγάλη προσοχή [169].

Σχήμα 7.13: Η ενέργεια και η καθυστέρηση για τις δύο μορφές της LZA λογικής των 64 δυαδικών ψηφίων υλοποιημένης σε δυναμική CMOS.

οι οποίες τοποθετούνται εναλλάξ. Επίσης, για να αυξήσουμε την ταχύτητα των κυκλωμάτων και την ανοχή στη καθυστέρηση μεταξύ των ακμών του ρολογιού (skew) οι δυναμικές πύλες οδηγούνται από επικαλυπτόμενες φάσεις ρολογιού [60]. Η ενέργεια και η καθυστέρηση των κυκλωμάτων που προέκυψαν για την περίπτωση των 64 δυαδικών ψηφίων φαίνονται επίσης στο σχήμα 7.12. Από τις τιμές του σχήματος μπορούμε εύκολα να διαπιστώσουμε ότι η προτεινόμενη προσέγγιση οδηγεί σε κυκλώματα τόσο με μικρότερη καθυστέρηση όσο και με μικρότερη κατανάλωση ενέργειας σε σχέση με τη δυναμική υλοποίηση της μονάδας μέτρησης που βασίζεται σε μονάδες κωδικοποίησης. Η ελάχιστη δυνατή καθυστέρηση του προτεινόμενου κυκλώματος είναι 27% μικρότερη σε σχέση με τα κυκλώματα κωδικοποίησης, ενώ το κέρδος σε κατανάλωση ενέργειας κυμαίνεται μεταξύ 13% και 58%. Η μικρότερη ταχύτητα και κατανάλωση ενέργειας των προτεινόμενων κυκλωμάτων είναι άμεσο αποτέλεσμα της απλούστερης δομής τους και των ταχύτερων πυλών που συνθέτουν το κρίσιμο μονοπάτι.

Τέλος, συγκρίναμε τις δυναμικές CMOS υλοποιήσεις των αρχιτεκτονικών πρόβλεψης των προπορευόμενων μηδενικών που χρησιμοποιούν δύο χωριστές μονάδες πρόβλεψης, θεωρώντας πως το μήκος των λέξεων της εισόδου είναι 64 δυαδικά ψηφία. Στην πρώτη περίπτωση, παράγεται μία μονότονα αύξουσα λέξη πρόβλεψης (σχήμα 7.8(β)) η οποία κωδικοποιείται στην αντίστοιχη δυαδική της αναπαράστασης χρησιμοποιώντας ένα κύκλωμα κωδικοποιητή (οι ισοδύναμες σχέσεις στα 64 δυαδικά ψηφία των (7.3)–(7.5)). Στη δεύτερη περίπτωση, υλοποιούμε την προτεινόμενη μονάδα πρόβλεψης. Οι υλοποιήσεις σε επίπεδο κυκλώματος και για τις δύο μονάδες ακολουθούν τους ίδιους κανόνες που αναφέραμε στην προηγούμενη παράγραφο. Οι καμπύλες της ενέργειας ως προς την καθυστέρηση των δύο κυκλωμάτων παρουσιάζονται στο σχήμα 7.13. Οι προτεινόμενες μονάδες πρόβλεψης μπορούν να σχεδιαστούν ώστε να είναι 12% ταχύτερες σε σχέση με αυτές που βασίζονται σε κωδικοποιητή, ενώ ταυτόχρονα η κατανάλωση ενέργειας μπορεί να μειωθεί σημαντικά. Η μείωση της κατανάλωσης ενέργειας είναι μεταξύ 12% και 65%, συγκρίνοντας τα δύο κυκλώματα στην ίδια καθυστέρηση. Τα νούμερα αυτά

ποσοτικοποιούν σε γενικές γραμμές το ποσοστό της περιττής πληροφορίας που περιλαμβάνουν οι προηγούμενες αρχιτεκτονικές [140], την οποία καταφέρνουμε να αφαιρέσουμε με έναν απλό και αποδοτικό τρόπο.

Σύνοψη και συμπεράσματα

Στην ενότητα αυτή θα περιγράψουμε συνοπτικά τα ερευνητικά αποτελέσματα που προέκυψαν στα πλαίσια της διδακτορικής διατριβής, ενώ παράλληλα να αναφερθούμε στα νέα ερωτήματα που ανέκυψαν και στις πιθανές μελλοντικές ερευνητικές κατευθύνσεις για την περαιτέρω βελτίωση των μονάδων επεξεργασίας δεδομένων. Η πειραματική διαδικασία που ακολουθήσαμε, καλύπτει ένα ευρύ φάσμα υλοποιήσεων, τόσο ελεύθερης σχεδίασης όσο και αυτοματοποιημένης ροής σχεδιασμού, και στηρίχθηκε στην εξερεύνηση της συμπεριφοράς των κυκλωμάτων με τη χρήση των βέλτιστων καμπυλών της ενέργειας ως προς την καθυστέρηση. Η ανάλυση αυτή είναι η πιο έγκυρη μέθοδος σύγκρισης μεταξύ δύο διαφορετικών αρχιτεκτονικών και επιτρέπει την εξαγωγή ποικίλων συμπερασμάτων για τις ιδιότητες τους.

Αρχικά, παρουσιάσαμε νέα κυκλώματα υλοποίησης της δυαδικής πρόσθεσης, οι οποίες καταφέρνουν να ενοποιήσουν αποτελεσματικά τις γρήγορες δομές παράλληλου προθέματος για τον υπολογισμό των κρατουμένων με την απλοποιημένη μορφή των κρατουμένων που προτάθηκαν από τον Ling. Το αποτέλεσμα της ένωσης των δύο δομών, είναι νέοι γρήγοροι αθροιστές, οι οποίοι καταφέρνουν να μειώσουν την καθυστέρηση περισσότερο από 10% σε σχέση με τις παραδοσιακές δομές των 64 δυαδικών ψηφίων, χωρίς να αυξήσουν την κατανάλωση ισχύος του κυκλώματος. Η απόδοση των νέων δομών μένει να αναλυθεί σε ακόμη πιο μικρές τεχνολογίες κατασκευής έως και 32nm [174], ώστε να διερευνηθεί σε βάθος η επίδραση της αύξησης των ρευμάτων διαρροής και πώς μπορεί να αντιμετωπιστεί με νέες τεχνικές σε επίπεδο κυκλώματος. Οι αθροιστές των μικροεπεξεργασιών υψηλών επιδόσεων είναι από τα πρώτα κυκλώματα που επανασχεδιάζονται κατά τη μετάβαση σε μια νέα τεχνολογία υλοποίησης και η καθυστέρηση τους αποτελεί δείκτη της απόδοσης που περιμένουμε από τα υπόλοιπα κυκλώματα. Οι μελλοντικές τεχνολογίες υλοποίησης αναμένεται να χρησιμοποιούν διαφορετικές δομές τρανζίστορ με πιο πιθανή εκδοχή τα τρανζίστορ διπλής πύλης [72], [57]. Οι κυκλωματικές τεχνικές που απαιτούνται σε αυτές τις περιπτώσεις είναι κάπως διαφορετικές σε σχέση με την σημερινή κοινή

πρακτική. Για το λόγο αυτό, θα ήταν πολύ χρήσιμο να μελετηθούν από τώρα οι τεχνικές εκείνες, οι οποίες θα επιτρέπουν την αποτελεσματική υλοποίηση των μονάδων επεξεργασίας δεδομένων και ιδιαιτέρως των αθροιστών. Άλλωστε μη ξεχνάμε πως οι αθροιστές ανήκουν σε εκείνη την κατηγορία κυκλωμάτων που συναντούμε στην πλειοψηφία των ψηφιακών ολοκληρωμένων κυκλωμάτων ανεξαρτήτως εφαρμογής.

Στη συνέχεια παρουσιάσαμε αναλυτικά τις προτεινόμενες δομές αναδιοργάνωσης δεδομένων. Η εντολή που υλοποιούν, είναι γενική στη χρήση της, και όπως έχουμε διαπιστώσει μετά την ολοκλήρωση της διδακτορικής διατριβής, με κάποιες μικρές παραλλαγές μπορεί να αντικαταστήσει μια πληθώρα από ήδη υπάρχουσες εντολές αναδιοργάνωσης χωρίς να αυξάνει το κόστος της υλοποίησης. Τα νέα κυκλώματα βασίζουν τη λειτουργία τους στις επαυξημένες εκδοχές των δικτύων ταξινόμησης, οι οποίες μειώνουν σημαντικά την καθυστέρηση σε σχέση με προηγούμενες υλοποιήσεις. Δείξαμε, πως ανάλογα με το ελάχιστο μέγεθος υπολέξης, που υποστηρίζει ο ορισμός της εντολής, απαιτούνται διαφορετικές δομές, ώστε να επιτύχουμε τη βέλτιστη συμπεριφορά ως προς την ενέργεια και την καθυστέρηση. Κάθε ένα από τα προτεινόμενα κυκλώματα, σχεδιάζεται χρησιμοποιώντας ένα μόνο κελί υπολογισμού (οι διάφορες εκδοχές των κελιών σύγκρισης και εναλλαγής), ενώ παράλληλα διατηρεί ομοιόμορφες και επαναλαμβανόμενες διασυνδέσεις, επιτρέποντας έτσι την πυκνή χωροθέτηση του. Επίσης, το κέρδος σε απόδοση που επιτυγχάνουν οι προτεινόμενες δομές αναδιοργάνωσης των δεδομένων, δεν προέρχεται από κάποιες ιδιαίτερες κυκλωματικές τεχνικές, οι οποίες ίσως να εφαρμόζονται με δυσκολία στο μέλλον, αλλά είναι αποτέλεσμα μιας νέας αλγοριθμικής αντιμετώπισης του προβλήματος σε λογικό επίπεδο. Συνεπώς, πιστεύουμε πως οι δομές αυτές αναμένεται να διατηρήσουν τα πλεονεκτήματά τους και σε μελλοντικές τεχνολογίες VLSI υλοποίησης βελτιώνοντας αποτελεσματικά τις λειτουργίες αναδιοργάνωσης των δεδομένων. Τέλος, χρήσιμη θα ήταν η περαιτέρω διερεύνηση των γραμμικών δομών αναδιοργάνωσης των δεδομένων αφού χρησιμοποιούν την απλούστερη μορφή καλωδίωσης και πετυχαίνουν πολύ ελπιδοφόρα αποτελέσματα ιδίως στις πιο πρακτικές περιπτώσεις.

Κλείνοντας, στα κεφάλαια 6 και 7 της διατριβής αυτής παρουσιάσαμε αποτελεσματικές υλοποιήσεις των πιο κρίσιμων, από πλευράς καθυστέρησης, σταδίων της πρόσθεσης αριθμών κινητής υποδιαστολής. Από τη μια βελτιώσαμε τα κυκλώματα που πραγματοποιούν την πρόσθεση των μεγεθών και από την άλλη παρουσιάσαμε μια νέα ολοκληρωμένη προσέγγιση για τη σχεδίαση της μονάδας κανονικοποίησης των αριθμών κινητής υποδιαστολής. Και στις δύο περιπτώσεις οι μονάδες που προτείνουμε επιφέρουν σημαντικά κέρδη τόσο σε ενέργεια όσο και σε καθυστέρηση σε σχέση με τις προηγούμενες υλοποιήσεις. Κύριο μέλημα μας για τη συνέχεια αυτής της ερευνητικής προσπάθειας είναι η σχεδίαση μιας ολοκληρωμένης μονάδας κινητής υποδιαστολής, η οποία θα χρησιμοποιεί τα προτεινόμενα κυκλώματα και θα συνδιάζει τις επιμέρους δυνατότητές τους, ώστε να επιτύχει ένα συνολικά αποδοτικό σχεδιασμό.

Bibliograf'ia

- [1] D. H. Allen, S. H. Dhong, H. P. Hofstee, J. Leenstra, K. J. Nowka, D. L. Stasiak, and D. F. Wendel. Custom circuit design as a driver of microprocessor performance. *IBM Journal of Research and Development*, 44(12):799–822, Nov. 2000.
- [2] American National Standards Institute and Institute of Electrical and Electronic Engineers. IEEE standard for binary floating-point arithmetic. *ANSI/IEEE Standard, Std 754-1985*, New York, 1985.
- [3] B. Amrutur and M. A. Horowitz. Fast low-power decoders for RAMs. *IEEE Journal of Solid-State Circuits*, 36:1506–1514, 2001.
- [4] Krste Asanovic. *Vector Microprocessors*. PhD thesis, University of California, Berkeley, 1998.
- [5] Krste Asanovic, Ras Bodik, Bryan Christopher Catanzaro, Joseph James Gebis, Parry Husbands, Kurt Keutzer, David A. Patterson, William Lester Plishker, John Shalf, Samuel Webb Williams, and Katherine A. Yelick. The landscape of parallel computing research: A view from berkeley. Technical report, University of California, Berkeley, 2006.
- [6] Todd Austin, David Blaauw, Scott Mahlke, Trevor Mudge, Chaitali Chakrabarti, and Wayne Wolf. Mobile supercomputers. *IEEE Computer*, pages 81–83, May 2004.
- [7] Magma Design Automation. Deep-submicron signal integrity. White Paper, 2002.
- [8] Magma Design Automation. Early silicon performance. White Paper, 2002.
- [9] K. E. Batcher. Sorting Networks and their Applications. In *AFIPS Joint Computer Conference*, pages 307–314, 1968.
- [10] A. Beaumont-Smith and C. C. Lim. Parallel-Prefix Adder Design. In *15th Symp. on Computer Arithmetic*, pages 218–225, June 2001.
- [11] W. Belluomini, D. Jamsek, A. K. Martin, C. McDowell, R. K. Montoye, H. C. Ngo, and J. Sawada. Limited switch dynamic logic circuits for high-speed low-power circuit design. *IBM Journal of Research and Development*, 50:277–286, 2006.
- [12] Aart J. C. Bik. *The software vectorization handbook: Applying Multimedia Extensions for Maximum Performance*. Intel Press, June 2004.

- [13] Shekhar Y. Borkar, Pradeep Dubey, Kevin C. Kahn, David J. Kuck, Hans Mulder, Stephen S. Pawlowski, and Justin R. Rattner. Platform 2015: Intel processor and platform evolution for the next decade. White Paper, 2005.
- [14] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [15] S. P. Boyd, S. J. Kim, D. Patil, and M. A. Horowitz. Digital Circuit Optimization via Geometric Programming. *Operations Research*, 53(6):899–932, Nov.-Dec. 2005.
- [16] Stephen P. Boyd, Seung Jean Kim, Lieven Vandenberghez, and Arash Hassibix. A tutorial on geometric programming. *Optimization and Engineering*, 2005.
- [17] R. P. Brent and H. T. Kung. A Regular Layout for Parallel Adders. *IEEE Trans. on Comp.*, 31(3):260–264, Mar. 1982.
- [18] Javier D. Bruguera and Tomas Lang. Leading-one prediction with concurrent position correction. *IEEE Transactions on Computers*, 48:1083–1097, October 1999.
- [19] N. Burgess. The flagged prefix adder and its applications in integer arithmetic. *Journal of VLSI Signal Processing*, 31(3):263–271, Aug. 2002.
- [20] Neil Burgess. Accelerated carry-skip adders with low hardware cost. In *Proc. of Asilomar Conference on Signals, Systems and Computers*, pages 852–856, 2001.
- [21] Neil Burgess. New models of prefix adder topologies. *Journal of VLSI Signal Processing Systems*, 40:125–141, May 2005.
- [22] Neil Burgess. Prenormalization rounding in IEEE floating-point operations using a flagged prefix adder. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13:266–277, 2005.
- [23] A. Chandrakasan, S. Sheng, and R. Brodersen. Low-power CMOS digital design. *IEEE Journal Solid State Circuits*, 27(4):473–484, April 1992.
- [24] Anantha Chandrakasan, William J. Bowhill, and Frank Fox, editors. *Design of High-Performance Microprocessor Circuits*. Wiley, September 2000.
- [25] David Chinnery. *Low Power Design Automation*. PhD thesis, University of California, Berkeley, 2006.
- [26] David Chinnery and Kurt Keutzer. *Closing the gap between ASIC and Custom: Tools and techniques for high-performance ASIC design*. Kluwer Academic Publishers, 2002.
- [27] David Chinnery, B. Nikolic, and K. Keutzer. Achieving 550 MHz in an ASIC methodology. In *Design Automation Conference*, pages 420–425, June 2001.

- [28] A. R. Conn, I. M. Elfadel, Jr. W. W. Molzen, P. R. O'Brien, P. N. Strenski, C. Visweswariah, and C. B. Whan. Gradient-based optimization of custom circuits using a static-timing formulation. In *Design Automation Conference (DAC)*, pages 452–459, June 1999.
- [29] T. Conte. Challenges to Combining General-Purpose and Multimedia Processors. *IEEE Computer*, 12(30):33–37, Dec. 1997.
- [30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd Edition*. MIT Press, 1990.
- [31] William J. Dally and Andrew Chang. The role of custom design in ASIC chips. In *Proceedings of the 37th Design Automation Conference*, 2000.
- [32] H.Q. Dao, B. R. Zeydel, and V. G. Oklobdzija. Energy optimization of pipelined digital systems using circuit sizing and supply scaling. *IEEE Transaction on VLSI Systems*, 14:122–134, 2006.
- [33] Florentin Dartu, Noel Menezes, and Lawrence T. Pileggi. Performance computation for precharacterized cmos gates with rc loads. *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, 15:544–553, 1996.
- [34] K. Diefendorff and P. K. Dubey. How Mutimedia Will Change Processor Design. *IEEE Computer*, 30(9):43–45, Sept. 1997.
- [35] K. Diefendorff, P. K. Dubey, R. Hochsprung, and H. Scales. AltiVec Extensions to PowerPC Accelerates Media Processing. *IEEE Micro*, (2):85–95, March 2000.
- [36] G. Dimitrakopoulos, K. Galanopoulos, C. Mavrokefalidis, and D. Nikolos. Low-power leading zero counting and anticipation logic for high-speed floating point unit. *IEEE Transactions on VLSI Systems*, 2006. under review.
- [37] G. Dimitrakopoulos, C. Mavrokefalidis, K. Galanopoulos, and D. Nikolos. An energy-delay efficient subword permutation unit. In *IEEE 17th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 245–252, September 2006.
- [38] G. Dimitrakopoulos, C. Mavrokefalidis, K. Galanopoulos, and D. Nikolos. Fast Bit Permutation Unit for Media-Enhanced Microprocessors. In *Proc. of IEEE International Symposium on Circuits and Systems*, pages 49–52, May 2006.
- [39] G. Dimitrakopoulos, D. G. Nikolos, D. Nikolos, H. T. Vergos, and C. Efstathiou. New architectures for modulo $2^n - 1$ adders. In *IEEE International Conference on Electronics, Circuits, and Systems*, 2005.
- [40] G. Dimitrakopoulos, H. T. Vergos, D. Nikolos, and C. Efstathiou. A family of parallel prefix modulo $2^n - 1$ adders. In *IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP)*, pages 326–336, 2003.

- [41] G. Dimitrakopoulos, H. T. Vergos, D. Nikolos, and C. Efstathiou. A systematic methodology for designing area-time efficient modulo $2^n - 1$ adders. In *Proc. of IEEE International Symposium on Circuits and Systems*, pages 225-228, May 2003.
- [42] Giorgos Dimitrakopoulos, Christos Mavrokefalidis, Kostas Galanopoulos, and Dimitris Nikolos. Sorter-based permutation unit for media-enhanced microprocessor. *IEEE Transactions on VLSI Systems*, 2007. under second round of reviews.
- [43] Giorgos Dimitrakopoulos and Dimitris Nikolos. High-speed parallel prefix VLSI Ling adders. *IEEE Transactions on Computers*, 54(2):225-231, February 2005.
- [44] R. W. Doran. Variants of an improved carry-lookahead adder. *IEEE Trans. on Comp.*, C-37:1110-1113, 1988.
- [45] C. Efstathiou, D. Nikolos, and J. Kalamatianos. Area-Time Efficient Modulo $2^n - 1$ Adder Design. *IEEE Trans. on Circuits and Systems II*, 41(7):463-467, Jul. 1994.
- [46] Alexandre E. Eichenberger, Peng Wu, and Kevin O'Brien. Vectorization for simd architectures with alignment constraints. volume 39, pages 82-93, New York, NY, USA, 2004. ACM Press.
- [47] W. C. Elmore. The transient response of damped linear networks. *Journal of Applied Physics*, 19:55-63, 1948.
- [48] M. Ergecovac and T. Lang. *Digital Arithmetic*. Morgan-Kaufman Publishers, 2003.
- [49] J. Park et. al. 470ps 64-bit parallel binary adder. In *Digest of Technical Papers, Symposium on VLSI Circuits.*, 2000.
- [50] E. S. Fetzer, M. Gibson, A. Klein, N. Calick, Z. Chengyu, E. Busta, and B. Mohammad. A fully bypassed six-issue integer datapath and register file on the itanium-2 microprocessor. *IEEE Journal of Solid-State Circuits*, 37(11):1433-1430, 2002.
- [51] J. P. Fishburn and A. E. Dunlop. TILOS : A posynomial programming approach to transistor sizing. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pages 326-328, 1985.
- [52] Joseph A. Fisher, Paolo Faraboschi, and Cliff Young. *Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools*. Morgan Kaufmann, 2004.
- [53] Guenter Gerwig and Michael Kroener. Floating-point unit in standard cell design with 116 bit wide dataflow. In *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, pages 266-273, Apr. 1999.
- [54] T. Ghani and et al. 100 nm gate length high performance/low power CMOS transistor structure. In *Technical Digest, International Electron Devices Meeting*, pages 415-418, 1999.

- [55] J. Grad and J. E. Stine. Hybrid emodl ling addition. In *Asilomar Conf. Signals Systems and Computers*, pages 1624–1628, 2002.
- [56] Paul E. Gronowski, William J. Bowhill, Ronald P. Preston, Michael K. Gowan, and Randy L. Allmon. High-performance microprocessor design. *IEEE Journal Solid State Circuits*, 33:676–686, 1998.
- [57] Z. Guo, S. Balasubramanian, R. Zlatanovici, T.-J. King, and B. Nikolic. Finfet-based sram design. In *International Symposium on Low Power Design (ISLPED)*, pages 2–7, 2005.
- [58] T. Han and D. Carlson. Fast area-efficient VLSI adders. In *Symp. on Computer Arithithmetic*, pages 49–56, May 1987.
- [59] David Harris. A taxonomy of parallel prefix networks. In *Proc. of Asilomar Conference on Signals, Systems and Computers*, pages 2213–2217, 2003.
- [60] David Harris and Mark Horowitz. Skew-tolerand domino circuits. *IEEE Journal of Solid State Circuits*, 32:1702–1711, Nov. 1997.
- [61] Seongmoo Heo and Krste Asanovic. Power-optimal pipelining in deep submicron technology. In *International Symposium on Low Power Electronics and Design (ISLPED)*, pages 218–223, 2004.
- [62] Y. Hilewitz, Z. J. Shi, and R. B. Lee. Comparing Fast Implementations of Bit Permutation Instructions. In *Proc. of Asilomar Conference on Signals, Systems and Computers*, pages 1856–1863, Nov. 2004.
- [63] Yedidya Hilewitz and Ruby B. Lee. Fast bit compression and expansion with parallel extract and parallel deposit instructions. In *Proceedings of the IEEE 17th International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*, pages 65–72, September 2006.
- [64] Ron Ho, Ken Mai, and Mark Horowitz. The future of wires. *Proceedings of the IEEE*, pages 4901–504, April 2001.
- [65] David Hodges, Horace Jackson, and Resve Saleh. *Analysis and Design of Digital Integrated Circuits: In Deep Submicron Technology*. McGraw-Hill, 2003.
- [66] H. Hofstee. Power-constrained microprocessor design. In *International Conference on Computer Design (ICCD)*, pages 14–16, September 2002.
- [67] E. Hokenek and R. K. Montoye. Leading-Zero Anticipator (LZA) in the IBM RISC System/6000 floating point execution unit. *IBM Journal of Research and Development*, 34:71–77, January 1990.
- [68] Z. Hong and R. Sedgewick. Notes on Merging Networks. In *Proc. of the 14th Annual ACM Symposium on Theory of Computing*, pages 296–302, May 1982.

- [69] M. Horowitz. VLSI Scaling for Architects. *Invited Talk at University of Texas Austin* <http://www-vlsi.standord.edu>, Feb. 2000.
- [70] M. Horowitz, E. Alon, D. Pati, S. Naffziger, R. Kumar, and K. Bernstein. Scaling, power, and the future of CMOS. In *IEEE International Electron Devices Meeting*, December 2005.
- [71] M. Horowitz, T. Indermaur, and R. Gonzalez. Low-power digital design. In *IEEE Symposium on Low Power Electronics*, pages 8–11, October 1994.
- [72] Mark Horowitz. Life after silicon: An oxymoron?, February 2000.
- [73] Steven K. Hsu. *Advanced Microarchitecture and Circuit Design Techniques for On-Chip Memories in CMOS Technology*. PhD thesis, Oregon State University, 2006.
- [74] Z. Huang and M. D. Ercegovac. Effect of wire delay on the design of prefix adders in deep-dubmicron technology. In *Asilomar Conference on Signals, Systems and Computers*, pages 1713–1717, 2000.
- [75] IBM. Powerpc microprocessor family: Vector/SIMD multimedia extensions technology programming environment manual. Technical report, 2005.
- [76] F. Ishiharam, F. Sheikh, and B. Nikolic. Level-conversion for dual-supply systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(2):185–195, February 2004.
- [77] L. Kalampoukas, D. Nikolos, C. Efstathiou, H. T. Vergos, and J. Kalamatianos. High-Speed Parallel-Prefix Modulo $2^n - 1$ Adders. *IEEE Trans. on Computers*, 49(7):673–680, Jul. 2000.
- [78] S. Kao, R. Zlatanovici, and B. Nikolic. A 240ps 64-bit carry-lookahead adder in 90nm cmos. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 1735 – 1744, 2006.
- [79] S. Kao, R. Zlatanovici, and B. Nikolic. A 250ps 64-bit carry-lookahead adder in 90nm cmos. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 438–439, February 2006.
- [80] S. K. Karandikar and S. S. Sapatnekar. Fast comparisons of circuit implementations. *IEEE Transactions on VLSI Systems*, 13:1329–1339, 2005.
- [81] K. Kasamsetty, M. Ketkar, and S. S. Sapatnekar. A new class of convex functions for delay modeling and their application to the transistor sizing problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19:779–788, 2000.
- [82] Fabian Klass, Chaim Amir, Ashutosh Das, Kathirgamar Aingaran, Cindy Truong, Richard Wang, Anup Mehta, Ray Heald, and Gin Yee. A new family of semidynamic and dynamic flip-flops with embedded logic for high-performance process. *IEEE Journal of Solid-State Circuits*, 34:712–716, 1999.

- [83] S. C. Knowles. Arithmetic processor design for the T9000 transputer. In *Proc. SPIE Vol. 1566, Advanced Signal Processing Algorithms, Architectures, and Implementations II*, pages 230–243, December 1991.
- [84] Simon Knowles. A family of adders. In *IEEE Symposium on Computer Arithmetic*, pages 30–34, April 1999.
- [85] P. M. Kogge and H. S. Stone. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Transactions on Computers*, C-22:786–793, Aug. 1973.
- [86] I. Koren. *Computer Arithmetic Algorithms*. A. K. Peters, Ltd., 2002.
- [87] R.K. Krishnamurthy, A. Alvandpour, V. De, and S. Borkar. High-performance and low-power challenges for sub-70nm microprocessor circuits. In *IEEE Custom Integrated Circuits Conference*, pages 125–128, May 2002.
- [88] I. Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. In *ACM Symposium on FPGAs*, pages 21–30, February 2006.
- [89] Tadahiro Kuroda. 10 tips for low power CMOS design. Design Automation Conference, Tutorial, June 2003.
- [90] Ding-Ming Kwai. Review of 6t sram cell. larc.ee.nthu.edu.tw/dtc/doc/940603.pdf, June 2005.
- [91] R. E. Ladner and M. J. Fisher. Parallel Prefix Computation. *Journal of The ACM*, 27(4):831–838, October 1980.
- [92] S. Lakshmivarahan and S. K. Dhall. *Parallel Computing: Using the Prefix Problem*. Oxford University Press, 1994.
- [93] Tomás Lang and Javier D. Bruguera. Floating-point multiply-add-fused with reduced latency. *IEEE Trans. Computers*, 53:988–1003, 2004.
- [94] D. Lee, D. Blaauw, and D. Sylvester. Runtime leakage minimization through probability-aware optimization. *IEEE Transactions on VLSI Systems*, pages 1075–1088, 2006.
- [95] Dongwoo Lee, David Blaauw, and Dennis Sylvester. Gate oxide leakage current analysis and reduction for VLSI circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12:155–166, 2004.
- [96] Kyung T. Lee and Kevin J. Nowka. 1GHz leading-zero anticipator using indendent sign-bit determination logic. In *Symposium on VLSI Circuits Digest of Technical Papers*, pages 194–195, 2000.
- [97] R. B. Lee. Subword Parallelism with MAX-2. *IEEE Micro*, pages 51–59, Aug. 1996.

- [98] R. B. Lee and A. M. Fiskiran. PLX: An Instruction Set Architecture and Testbed for Multimedia Information Processing. *Journal of VLSI Signal Processing*, 40(1):85–108, May 2005.
- [99] R. B. Lee, R. L. Rivest, M.J.B. Robshaw, Z.J. Shi, and Y.L. Yin. On permutation operations in cipher design. In *Proceedings of the International Conference on Information Technology (ITCC)*, pages 569 – 577, April 2004.
- [100] R. B. Lee, Z. Shi, and X. Yang. Efficient Permutation Instructions for Fast Software Cryptography. *IEEE Micro*, 21(6):56–69, Dec 2001.
- [101] Ruby Lee. *Instruction Set Architecture for Multimedia Signal Processing*, chapter 39. CRC Press, December 2001.
- [102] Yuan Lin, Hyunseok Lee, Mark Woh, Yoav Harel, Scott Mahlke, and Trevor Mudge. Soda: A lowpower architecture for software radio. In *International Symposium on Computer Architecture*, June 2006.
- [103] Hans Lindkvist and Per Andresson. Techniques for fast CMOS-based conditional sum adders. In *International Conference on Computer Design (ICCD)*, pages 626–635, 1994.
- [104] H. Ling. High-Speed Binary Adder. *IBM Journal of R & D*, 25:156–166, May 1981.
- [105] Jianhua Liu, Yi Zhu, Haikun Zhu, Chung-Kuan Cheng, and John Lillis. Optimum prefix adders in a comprehensive area, timing and power design space. In *Asian-Pacific Design Automation Conference (ASPDAC)*, January 2007.
- [106] T. Lynch and E. Swartzlander. A Spanning Tree Carry Lookahead Adder. *IEEE Trans. on Comp.*, C-41(8):931–939, August 1992.
- [107] N. Mading, J. Leenstra, J. Pille, R. Sautter, S. Buttner, S. Ehrenreich, and W. Haller. The vector fixed point unit of the synergistic processor element of the cell architecture processor. In *Europe Solid-State Circuits Symposium (ESSCIRC)*, pages 203–206, 2005.
- [108] Morris Mano and Michael D. Ciletti. *Digital Design*. Prentice Hall, 2006.
- [109] Dejan Markovic, Vladimir Stojanovic, Borivoje Nikolic, Mark A. Horowitz, and Robert W. Brodersen. Methods for true energy-performance optimization. *IEEE Journal of Solid-State Circuits*, 39:1282–1293, 2004.
- [110] Dejan Marko Markovic. *A Power/Area Optimal Approach to VLSI Signal Processing*. PhD thesis, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2006.
- [111] S. Mathew, M. Anders, R. Krishnamurty, and 2002 S. Borkar, p.126-127. A 4GHz 130nm address generation unit with 32-bit sparse-tree adder core. *IEEE Journal of Solid-State Circuits*, 38:689–695, 2003.

- [112] S. Mathew, R. Krishnamurthy, M. Anders, R. Rios, K. Mistry, and K. Soumyanath. Sub-500-ps 64-b ALUs in 0.18- μ m SOI/Bulk CMOS: Design and Scaling Trends. *IEEE Journal of Solid-State Circuits*, 36, 2001.
- [113] Sun Microsystems. *The VIS Instruction Set: White Paper*. June 2002.
- [114] M. Mittal, A. Peleg, and U. Weiser. MMX Technology Architecture Overview. *Intel Technology Journal*, 1(1), (Q3) 1997.
- [115] S. M. Mueller, C. Jacobi, H-J Oh, K. D. Tran, S. R. Cottier, B. W. Michael, H. Nishikawa, Y. Totsuka, T. Namatane, N. Yano, T. Machida, and S. H. Dhong. The vector floating-point unit in a synergistic processor element of a CELL processor. In *Proceedings of the IEEE Symposium on Computer Arithmetic*, pages 59–67, June 2005.
- [116] Almir Mutapcic, Kwangmoo Koh, Seungjean Kim, and Stephen Boyd. GGPLAB: A MatLab toolbox for geometric programming. <http://www.stanford.edu/~boyd/ggplab/>, 2006.
- [117] S. Naffziger. A Sub-Nanosecond 0.5 μ m 64b Adder Design. In *IEEE Solid-State Circuits Conference.*, pages 362–363, Feb. 1996.
- [118] Samuel Naffziger. High-performance processors in a power-limited world. In *Symposium VLSI Circuits*, pages 93 – 97, 2006.
- [119] Samuel D. Naffziger, Glenn Colon-Bonet, Timothy Fischer, Reid Riedlinger, Thomas J. Sullivan, and Tom Grutkowski. The implementation of the itanium 2 microprocessor. *IEEE Journal Solid-State Circuits*, 37(11):1448–1460, November 2002.
- [120] Siva Narendra and Anatha Chandrakasan, editors. *Leakage in Nanometer CMOS Technologies*. Springer-Verlag, 2006.
- [121] Kevin Nowka. *Computer Engineering Handbook*, chapter Issues in High-Frequency Processor Design. CRC Press, 2002.
- [122] Dorit Nuzman, Ira Rosen, and Ayal Zaks. Auto-vectorization of interleaved data for simd. In *Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation*, pages 132 – 143, June 2006.
- [123] S. Oberman, G. Favor, and F. Weber. AMD 3DNow! Technology: Architecture and Implementations. *IEEE Micro*, 19(2):37–48, Mar-April 1999.
- [124] H-J. Oh, S. M. Mueller, C. Jacobi, K. D. Tran, S. R. Cottier, B. W. Michael, H. Nishikawa, Y. Totsuka, T. Namatane, N. Yano, T. Machida, and S. H. Dhong. A fully-pipelined single-precision floating point unit in the synergistic processor element of a CELL processor. *IEEE Journal of Solid-State Circuits*, 41:759–771, Apr. 2006.
- [125] V. G. Oklobdzija and R. K. Krishnamurthy, editors. *High-Performance Energy-Efficient Microprocessor Design*. Springer, July 2006.

- [126] Vojin Oklobdzija. An algorithmic and novel design of a leading zero detector circuit : Comparison with logic synthesis. *IEEE Transactions on VLSI Systems*, 2:124–128, March 1994.
- [127] Vojin Oklobdzija. Comment on “Leading-zero anticipatory logic for high-speed floating point addition”. *IEEE Journal of Solid-State Circuits*, 32:292–293, Feb. 1997.
- [128] L.T. Pang and B. Nikolic. Impact of layout on 90nm CMOS process parameter fluctuations. In *Symposium on VLSI Circuits*, 2006.
- [129] B. Parhami. *Computer Arithmetic - Algorithms and Hardware Designs*. Oxf. Univ. Press, 2000.
- [130] Marcel Pelgrom, Aad C. J. Duinmaijer, and Andanton P. G. Welbers. Matching properties of the MOS transistor. *IEEE Journal of Solid-State Circuits*, 24:1433–1440, 1989.
- [131] Dac C. Pham, Tony Aipperspach, David Boerstler, Mark Bolliger, Rajat Chaudhry, Dennis Cox, Paul Harvey, H. Peter Hofstee, Charles Johns, Jim Kahle, Atsushi Kameyama, John Keaty, Yoshio Masubuchi, Mydung Pham, Jurgen Pille, Stephen Posluszny, Mack Riley, Daniel L. Stasiak, Masakazu Suzuoki, Osamu Takahashi, James Warnock, Stephen Weitzel, Dieter Wendel, and Kazuaki Yazawa. Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor. *IEEE Journal of Solid-State Circuits*, 41:179–196, 2006.
- [132] N. T. Quach and M. J. Flynn. High-Speed Addition in CMOS. *IEEE Trans. on Comp.*, 41(12):1612–1615, December 1992.
- [133] Nhon Quach. *Reducing the latency of floating-point arithmetic operations*. PhD thesis, Stanford University, 1993.
- [134] Nhon Quach and Michael J. Flynn. Leading one prediction - implementation, generalization, and application. In *Technical Report, CSL-TR-91-463, Stanford University*, March 1991.
- [135] R.M. Ramanathan, Ron Curry, Srinivas Chennupaty, Robert L. Cross, Shihjong Kuo, and Mark J. Buxton. Extending the world’s most popular processor architecture: New innovations that improve the performance and energy efficiency of intel architecture. Intel White Paper, 2006.
- [136] Gang Ren, Peng Wu, and David Padua. Optimizing data permutations for simd devices. In *Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation*, pages 118 – 131, June 2006.
- [137] Stefan Rusu. Circuit technologies for multi-core processor design. Presentation Talk, IEEE Santa Clara Valley (SCV) Solid State Circuits Society, April 2006.

- [138] Takayasu Sakurai and Richard Newton. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *IEEE Journal of Solid-State Circuits*, 25:584–594, 1990.
- [139] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang. An exact solution to the transistor sizing problem for CMOS circuits using convex optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12:1621–1634, 1993.
- [140] Martin S. Schmookler and Kevin J. Nowka. Leading zero anticipation and detection : A comparison of methods. In *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, pages 7–12, July 2001.
- [141] Peter-Michael Seidel and Guy Even. On the design of fast ieee floating-point adders. *IEEE Transactions on Computers*, 53:97–113, Feb. 2004.
- [142] Z. Shi and R. B. Lee. Bit Permutation Instructions for Accelerating Software Cryptography. In *Proc. of IEEE Conference on Application-Specific Systems, Architectures and Processors*, pages 138–148, July 2000.
- [143] Z. Shi and R. B. Lee. Subword Sorting with Versatile Permutation Instructions. In *Proc. of IEEE International Conference on Computer Design*, pages 234–241, Sept. 2002.
- [144] Z. J. Shi. *Bit Permutation Instructions: Architecture, Implementation and Cryptographic Properties*. PhD thesis, Princeton University, 2004.
- [145] Z. J. Shi and R. B. Lee. Implementation complexity of bit permutation instructions. In *Proc. of Asilomar Conference on Signals, Systems and Computers*, pages 879–886, Nov 2003.
- [146] Y. Shimazaki, R. Zlatanovici, and B. Nikolic. A shared-well dual-supply-voltage 64-bit ALU. *IEEE Journal of Solid-State Circuits*, 39:494–500, 2004.
- [147] D. Singh and et al. Power conscious CAD tools and methodologies: a perspective. *Proceedings of the IEEE*, 83:570–594, 1995.
- [148] N. T. Slingerland and A. J. Smith. Multimedia Extensions for General Purpose Microprocessors: A Survey. *Microprocessors and Microsystems*, (29):225–246, 2005.
- [149] Eric Sprangle and Doug Carmean. Increasing processor performance by implementing deeper pipelines. In *Proceedings of the 29th annual international symposium on Computer architecture*, pages 25–34, Washington, DC, USA, 2002. IEEE Computer Society.
- [150] Viji Srinivasan, David Brooks, Michael Gschwind, Pradip Bose, Victor Zyuban, Philip N. Strenski, and Philip G. Emma. Optimizing pipelines for power and performance. In *MICRO 35: Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, pages 333–344, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

- [151] Peter Suarisa, Taeho Kgil, Keith A. Bowman, Vivek De, and Trevor N. Mudge. Total power-optimal pipelining and parallel processing under process variations in nanometer technology. In *International Conference on Computer Design*, pages 535–540, 2005.
- [152] A. K. Sultania, D. Sylvester, and S. S. Sapatnekar. Gate oxide leakage and delay tradeoffs for dual tox circuits. *IEEE Transactions on VLSI Systems*, 13:1362–1375, 2005.
- [153] I. Sutherland, R. Sproull, and D. Harris. *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.
- [154] H. Suzuki, H. Morinaka, H. Makino, Y. Nakase, K. Mashiko, and T. Sumi. Leading-zero anticipatory logic for high-speed floating point addition. *IEEE Journal of Solid-State Circuits*, 31:1157–1164, August 1996.
- [155] O. Takahashi, R. Cook, S. Cottier, S. H. Dhong, B. Flachs, K. Hirairi, A. Kawasumi, H. Murakami, H. Murakami, Noro, H. Oh, S. Onishi, J. Pille, and J. Silberman. The circuit design of the synergistic processor element of a cell processor. In *International Conference on Computer-Aided Design (ICCAD)*, 2005.
- [156] Deepu Talla, Ching-Yu Hung, Raj Talluri, Frank Brill, David Smith, David Brier, Bruce Xiong, and Derek Huynh. Anatomy of a portable digital mediaprocessor. *IEEE Micro*, pages 32–39, 2004.
- [157] S. Thakkar and T. Huff. The Internet Streaming SIMD Extensions. *Intel Technology Journal*, 3(2), May 1999.
- [158] UMC-18. *eSi-Route/11 0.18 μ m Standard Cell Library*. Virtual Silicon Technology, Jan. 2001.
- [159] K. Usami and M. Horowitz. Clustered voltage scaling technique for low-power design. In *International Symposium on Low Power Design*, pages 3–8, 1995.
- [160] Jan-Willem van de Waerdt. *The TM3270 Media-processor*. PhD thesis, Delft Technical University, 2006.
- [161] J.W. van de Waerdt, S. Vassiliadis, and N. Engin. The TM3270 media-processor. In *MICRO 05: Proceedings of the 38th International Symposium on Microarchitecture*, pages 331–342, November 2005.
- [162] S. R. Vangal, Y. V. Hoskote, N. Y. Borkar, and A. Alvandpour. A 6.2-gflops floating-point multiply-accumulator with conditional normalization. *IEEE Journal of Solid-State Circuits*, 41:2314–2323, 2006.
- [163] S. Vassiliadis. Recursive Equations for Hardwired Binary Adders. *Journal of Electronics*, 67(2):201–213, Aug. 1989.

- [164] H. Veendrick. Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. *IEEE Journal of Solid-State Circuits*, 19:468–473, 1984.
- [165] H. Veendrick. Digital goes analog. In *European Solid-State Circuits Conference*, pages 44–50, 1998.
- [166] C. Visweswariah and A. R. Conn. Formulation of static circuit optimization with reduced size, degeneracy and redundancy by timing graph manipulation. In *IEEE International Conference on Computer-Aided Design*, pages 244–291, November 1999.
- [167] Y. Wang, C. Pai, and X. Song. The design of hybrid carry-lookahead/carry-select adders. *IEEE Trans. Circuits Syst. II*, 46:16–24, 2002.
- [168] Z. Wang, G. A. Jullien, W. C. Miller, J. Wang, and S. S. Bizzan. Fast adders using enhanced multiple-output domino logic. *IEEE Journal of Solid-State Circuits*, 32:206–214, 1997.
- [169] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison Wesley, 2005.
- [170] Sapumal B. Wijeratne, Nanda Siddaiah, Sanu K. Mathew, Mark A. Anders, Ram K. Krishnamurthy, Jeremy Anderson, Matthew Ernest, and Mark Nardin. A 9-GHz 65-nm intel pentium 4 processor integer execution unit. *IEEE Journal of Solid-State Circuits*, 42(1):26–37, January 2007.
- [171] A. Ye, J. Rose, and D. Lewis. Architecture of datapath-oriented coarse-grain logic and routing for FPGA. In *Custom Integrated Circuits Conference*, pages 61–64, September 2003.
- [172] Andy Ye. *Field-Programmable Gate Array Architectures and Algorithms Optimized for Implementing Datapath Circuits*. PhD thesis, University of Toronto, 2004.
- [173] Ge Zhang, Zichu Qi, and Weiwu Hu. A novel design of leading zero anticipation circuit with parallel error detection. In *Proc. of IEEE International Symposium on Circuits and Systems*, pages 676–679, May 2005.
- [174] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45nm design exploration. In *International Symposium on Quality Electronic Design (ISQED)*, pages 585–590, 2006.
- [175] Haikun Zhu, Chung-Kuan Cheng, and Ronald Graham. Constructing zero-deficiency parallel prefix adder of minimum depth. In *Asian-Pacific Design Automation Conference (ASPDAC)*, pages 883–888, 2005.
- [176] R. Zimmerman. Efficient VLSI Implementation of Modulo $(2^n \pm 1)$ Addition and Multiplication. In *Proc. of 14th IEEE Symposium Computer Arithmetic*, pages 158–167, April 1999.

- [177] Reto Zimmermann. Non-heuristic optimization and synthesis of parallel prefix adders. In *International Workshop on Logic and Architecture Synthesis*, December 1996.
- [178] Reto Zimmermann. *Binary adder architectures for cell-based VLSI and their synthesis*. PhD thesis, Swiss Federal Institute of Technology, Zurich (ETHZ), 1997.
- [179] Reto Zimmermann and David Q. Tran. Optimized synthesis of sum-of-products. In *Proc. of Asilomar Conference on Signals, Systems and Computers*, pages 867–872, 2003.
- [180] R. Zlatanovici and B. Nikolic. Power-performance optimization for custom digital circuits. In *Proc. PATMOS'05, Lecture Note in Computer Science*, pages 404–414, 2005.
- [181] Radu Zlatanovici. Power - performance optimization for digital circuits. Technical report, University of California, Berkeley, 2006.