



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**ΥΛΟΠΟΙΗΣΗ ΕΝΟΠΟΙΗΜΕΝΟΥ ΔΙΠΛΟΥ
ΠΟΛΛΑΠΛΑΣΙΑΣΤΗ-ΑΘΡΟΙΣΤΗ ΓΙΑ ΤΗΝ
ΕΠΙΤΑΧΥΝΣΗ DSP ΑΛΓΟΡΙΘΜΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΩΝΣΤΑΝΤΙΝΟΥ ΑΠΟΣΤΟΛΟΥ

Επιβλέπων : Κιαμάλ Πεκμεστζή

Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**ΥΛΟΠΟΙΗΣΗ ΕΝΟΠΟΙΗΜΕΝΟΥ ΔΙΠΛΟΥ
ΠΟΛΛΑΠΛΑΣΙΑΣΤΗ-ΑΘΡΟΙΣΤΗ ΓΙΑ ΤΗΝ
ΕΠΙΤΑΧΥΝΣΗ DSP ΑΛΓΟΡΙΘΜΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΩΝΣΤΑΝΤΙΝΟΥ ΑΠΟΣΤΟΛΟΥ

Επιβλέπων : Κιαμάλ Πεκμεστζή

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13^η Οκτωβρίου 2016.

.....
Κ. Πεκμεστζή
Καθηγητής Ε.Μ.Π.

.....
Δ. Σούντρης
Αναπλ. Καθηγητής Ε.Μ.Π.

.....
Γ. Γκούμας
Επικ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016

.....

ΚΩΝΣΤΑΝΤΙΝΟΣ ΑΠΟΣΤΟΛΟΥ

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

Copyright © Αποστόλου Κωνσταντίνος 2016

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Περίληψη

Πολλές εφαρμογές υψηλών προδιαγραφών σήμερα επιτάσσουν την αποδοτική υλοποίηση αλγορίθμων Ψηφιακής Επεξεργασίας Σήματος (ΨΕΣ). Η ανάπτυξη καινοτόμων hardware αρχιτεκτονικών που αποσκοπούν στην επιτάχυνση της εκτέλεσης τέτοιων αλγορίθμων έχει αποδειχθεί πως είναι αποδοτική όσον αφορά την επιφάνεια και την κατανάλωση ισχύος.

Στα πλαίσια της παρούσας διπλωματικής, προτείνεται μια νέα αρχιτεκτονική, ειδικά σχεδιασμένη για την επιτάχυνση της απεικόνισης αλγορίθμων ΨΕΣ και άλλων υπολογιστικά έντονων εφαρμογών. Ως βασικά επεξεργαστικά στοιχεία της αρχιτεκτονικής αυτής, εισάγονται δύο ξεχωριστές μονάδες, ο Ενοποιημένος Διπλός Πολλαπλασιαστής-Αθροιστής (UDMA), και μια τροποποιημένη εκδοχή αυτού που χρησιμοποιεί διοχέτευση δυο επιπέδων. Αυτές οι μονάδες αποτελούν ευέλικτα αριθμητικά κυκλώματα, βέλτιστα σχεδιασμένα ώστε να μειωθεί η κρίσιμη καθυστέρηση, τα οποία δύνανται να επεξεργαστούν δεδομένα εσωτερικά σε αναπαράσταση Σωσίματος-Κρατουμένου ενώ οι είσοδοι/έξοδοι είναι σε δυαδική μορφή 16-bit.

Η μονάδα UDMA μπορεί να εκτελέσει οποιοδήποτε υποσύνολο πράξεων από την ακολουθία $A_1 \times (X_1 \pm Y_1) + A_2 \times (X_2 \pm Y_2) + B_1 + B_2$ σε ένα κύκλο εκτέλεσης. Για την περίπτωση της διοχέτευσης, απαιτούνται δύο κύκλοι, ωστόσο αυξάνεται ο ρυθμός διεκπεραίωσης του κυκλώματος. Οι συγκεκριμένες πράξεις, που απαρτίζονται κυρίως από πολλαπλασιασμούς και προσθαιρέσεις, είναι ιδιαίτερα κοινές στους αλγόριθμους ΨΕΣ. Συνεπώς, η κατάλληλη ομαδοποίηση της εκτέλεσης των πράξεων αυτών μπορεί να οδηγήσει στην επιτάχυνση της απεικόνισης πυρήνων ΨΕΣ.

Η αρχιτεκτονική που βασίζεται στην UDMA διαφοροποιείται από τις έως τώρα προτεινόμενες στο ότι επιχειρεί την βελτιστοποίηση της σχεδίασης τόσο στο αρχιτεκτονικό όσο και στο αριθμητικό επίπεδο του κυκλώματος και επωφελείται της αποδοτικής ενοποίησης δύο πολλαπλασιαστών, που είναι και το κεντρικό σημείο της σχεδίασης της UDMA. Η αρχιτεκτονική τελικά αποτιμάται μέσω της σύγκρισης όσον αφορά την κρίσιμη καθυστέρηση, την επιφάνεια και την κατανάλωση ισχύος με την ιδιαίτερα αποδοτική αρχιτεκτονική που βασίζεται στην μονάδα FCU.

Λέξεις Κλειδιά:

UDMA, DSP, αλγόριθμος, επιτάχυνση, απεικόνιση, αρχιτεκτονική, βελτιστοποίηση

Abstract

Many high-end applications dictate the efficient implementation of Digital Signal Processing (DSP) algorithms. The development of innovative hardware architectures with the main purpose of accelerating the execution of such algorithms has been proven to be area and power efficient.

In the scope of this diploma thesis, a new architecture is proposed, specifically designed for the accelerated mapping of DSP algorithms and other computation intensive applications. As the core processing unit of this architecture, two distinctive modules are introduced, the Unified Double Multiplier-Adder (UDMA) and a modified variation of the UDMA that implements a two-stage pipeline. These two units are flexible arithmetic circuits optimally designed to reduce their critical delay, which can process data internally in Carry Save representation while using 16-bit binary I/O.

The UDMA is capable of executing any subset of the $A_1 \times (X_1 \pm Y_1) + A_2 \times (X_2 \pm Y_2) + B_1 + B_2$ sequence of operations within one execution cycle. In the pipelined case, two cycles will be required, but the circuit's throughput is increased. These certain operations, which mainly consist of multiplications and additions or subtractions, are common in DSP algorithms. Thus, by properly chaining the execution of these operations the accelerated mapping of DSP kernels can be achieved.

The UDMA-based architecture is different to the ones that have already been proposed, in that it targets the optimization of both the architectural and arithmetic levels of design, and that it seeks to take advantage of the efficient grouping of two multiplications together, which is the focal point of the UDMA's design. Its performance is evaluated by comparing the latency, area, and power consumption figures with the highly efficient FCU-based architecture.

Keywords:

UDMA, DSP, algorithm, acceleration, mapping, architecture, optimization

Πίνακας περιεχομένων

1	ΕΙΣΑΓΩΓΗ	1
1.1	<i>Εισαγωγή.....</i>	1
1.2	<i>Αντικείμενο διπλωματικής.....</i>	1
1.3	<i>Οργάνωση κειμένου</i>	3
2	ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	4
2.1	<i>Αριθμητικά συστήματα</i>	5
2.1.1	<i>Δυαδικό σύστημα</i>	5
2.1.2	<i>Αναπαράσταση σε μορφή συμπληρώματος ως προς δύο</i>	6
2.1.3	<i>Αναπαράσταση σε μορφή Σωσίματος-Κρατουμένου (Carry-Save).....</i>	10
2.1.4	<i>Αναπαράσταση σε μορφή Προσημασμένου ψηφίου (Signed Digit)</i>	11
2.2	<i>Κωδικοποιήσεις Booth και Modified Booth.....</i>	12
2.2.1	<i>Κωδικοποίηση Booth.....</i>	12
2.2.2	<i>Κωδικοποίηση Modified Booth.....</i>	14
2.3	<i>Είδη αθροιστών και πολλαπλασιαστών.....</i>	17
2.3.1	<i>Δομικά στοιχεία.....</i>	17
2.3.2	<i>Αθροιστές.....</i>	21
2.3.3	<i>Πολλαπλασιαστές.....</i>	24
2.4	<i>Αλγόριθμοι Ψηφιακής Επεξεργασίας Σήματος.....</i>	31
2.4.1	<i>Χαρακτηριστικά των DSP αλγορίθμων</i>	32
2.4.2	<i>Επιδράσεις στην προτεινόμενη αρχιτεκτονική</i>	32
3	UDMA ΚΑΙ UDMA ΜΕ PIPELINE: ΟΙ 2 ΒΑΣΙΚΕΣ ΕΠΕΞΕΡΓΑΣΤΙΚΕΣ ΜΟΝΑΔΕΣ ΤΩΝ ΠΡΟΤΕΙΝΟΜΕΝΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ.....	35
3.1	<i>Προεπισκόπηση</i>	35
3.2	<i>Δομή της μονάδαςUDMA.....</i>	35
3.2.1	<i>Είσοδοι-Έξοδοι τηςUDMA</i>	37
3.2.2	<i>Παρουσίαση των δομικών στοιχείων της UDMA-Διάταξη και διασύνδεση.....</i>	38
3.3	<i>Δομική ανάλυση των επιμέρους μονάδων</i>	39
3.3.1	<i>S-MB1 recoder USUM2MB1 και USUM2MB2</i>	39
3.3.2	<i>Partial Product Generator και Truncated Partial Product Generator ...</i>	46
3.3.3	<i>Δενδρικός CS αθροιστής 20:2 UTREE</i>	49

3.3.4	Τελικός CLA αθροιστής UFA	50
3.4	UDMA pipeline: Δομικές διαφορές και ανάλυση	50
3.5	Critical Path: Καθυστέρηση απόκρισης των προτεινόμενων μονάδων	52
3.6	Λειτουργίες της UDMA	53
4	ΜΕΘΟΔΟΛΟΓΙΑ ΤΗΣ ΑΠΕΙΚΟΝΙΣΗΣ ΑΛΓΟΡΙΘΜΩΝ ΣΤΙΣ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ.....	58
4.1	Προεπισκόπηση	58
4.2	Μεθοδολογία απεικόνισης αλγορίθμων- Παράμετροι που καθορίζουν τη μορφή των προτεινόμενων αρχιτεκτονικών	59
4.2.1	Ενοποιημένη απεικόνιση αλγορίθμων- Τελική μεθοδολογία που επιλέχθηκε.....	59
4.2.2	Πλήθος επεξεργαστικών πόρων.....	60
4.2.3	Αριθμός καταχωρητών ανά απεικόνιση	61
4.2.4	Μέθοδος ελαχιστοποίησης του χρόνου εκτέλεσης	61
5	ΑΠΕΙΚΟΝΙΣΗ DSP ΑΛΓΟΡΙΘΜΩΝ-ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	63
5.1	Προεπισκόπηση	63
5.2	Πειραματικά αποτελέσματα για τις μονάδες UDMA- Σύγκριση με Flexible Computational Unit (FCU)	64
5.3	Πειραματικά αποτελέσματα απεικόνισης DSP αλγορίθμων- Σύγκριση με Flexible Accelerator Architecture.....	69
6	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ	82
6.1	Σύνοψη και συμπεράσματα.....	82
6.2	Επεκτάσεις.....	84
7	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	86

1

ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή

Στη σύγχρονη εποχή, η αλματώδης διείσδυση της τεχνολογίας σε όλους τους τομείς της ανθρώπινης δραστηριότητας, είναι γεγονός. Αποτέλεσμα αυτής, είναι η συνεχής εξέλιξη των ψηφιακών συστημάτων ώστε να καλύψουν τις ολοένα αυξανόμενες ανάγκες των εφαρμογών.

Η συνεχής και επιτακτική ανάγκη για αύξηση ταχύτητας εκτέλεσης και ενεργειακή αποδοτικότητα σε πολλά σύγχρονα πεδία υπολογιστικά έντονων εφαρμογών οδήγησε αρχικά στην ραγδαία ανάπτυξη των πολυπύρηνων μικροεπεξεργαστών. Τέτοια πεδία περιλαμβάνουν την ψηφιακή επεξεργασία σήματος, την επεξεργασία γραφικών, αλγόριθμους κρυπτογραφίας, επεξεργασία διανυσμάτων και άλλους επιστημονικούς αλγόριθμους. Τελικά κατέληξε στην ευρύτερη αποδοχή πως οι ετερογενείς αρχιτεκτονικές μπορούν να επιφέρουν τα βέλτιστα αποτελέσματα σε αυτούς τους τομείς.

Η ετερογένεια αυτή επιτυγχάνεται με τον συνδυασμό επεξεργαστικών πυρήνων με εξειδικευμένο υλικό (specialized hardware), ενίοτε ευέλικτο, όπως οι επιταχυντές (accelerators) και οι Graphics Processing Units (GPUs). Αυτό γίνεται με σκοπό την ενσωμάτωση επιμέρους τμημάτων που είναι εξειδικευμένα να εκτελούν βέλτιστα συγκεκριμένες εργασίες. Αυξάνεται έτσι η πολυπλοκότητα και ο χρόνος σχεδίασης ενός συστήματος, ωστόσο αποδεικνύεται πως η συνολική απόδοση του συστήματος, η βελτίωση δηλαδή ενός χαρακτηριστικού χωρίς την ανάλογη επιβάρυνση κάποιου άλλου (πχ ταχύτητα έναντι κόστους, επιφάνειας ολοκληρωμένου, ακρίβειας κτλ), αυξάνεται σημαντικά σε σχέση με τη χρήση ενός γενικού σκοπού επεξεργαστή για τέτοιες εφαρμογές. Κεντρικό σημείο του εξειδικευμένου υλικού αποτελεί η ανάθεση τμημάτων που περιλαμβάνουν έντονους, συχνά επαναλαμβανόμενους, υπολογισμούς

σε μικρότερα επεξεργαστικά στοιχεία που λειτουργούν παράλληλα, με σκοπό την αύξηση του συγχρονισμού (concurrency) του συστήματος.

Όπως αναφέρθηκε, ιδιαίτερα σημαντικό πεδίο αποτελεί η Ψηφιακή Επεξεργασία Σήματος. Το πεδίο αυτό αφορά την επεξεργασία σημάτων και δεδομένων με σκοπό την ενίσχυση, την τροποποίηση, τον μετασχηματισμό ή την ανάλυσή τους για την εξαγωγή συγκεκριμένης πληροφορίας. Στα πλαίσια της διαδικασίας αυτής, εισάγεται ένα πλήθος μαθηματικών τεχνικών, το οποίο μπορεί να αναλυθεί περαιτέρω σε στοιχειώδεις πράξεις. Η αποδοτική επιτάχυνση εκτέλεσης των διάφορων αλγορίθμων ΨΕΣ είναι ένα πεδίο με ιδιαίτερο ερευνητικό ενδιαφέρον.

1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία αποσκοπεί στην παρουσίαση μιας εξειδικευμένης hardware αρχιτεκτονικής, ειδικά σχεδιασμένης για την απεικόνιση αλγορίθμων Ψηφιακής Επεξεργασίας Σήματος, με έμφαση στην επιτάχυνση της εκτέλεσής τους. Παρατίθεται και επεξηγείται αναλυτικά (δομή και λειτουργία) κάθε επιμέρους στάδιο της ανάπτυξής της, από τα βασικά επεξεργαστικά στοιχεία που περιλαμβάνει μέχρι την τελική αρχιτεκτονική που θα χρησιμοποιηθεί για την απεικόνιση.

Ως επεξεργαστικά στοιχεία της αρχιτεκτονικής, δυο ξεχωριστές μονάδες αναπτύσσονται και διερευνώνται ως προς την αποτελεσματικότητά τους: ο ενοποιημένος διπλός πολλαπλασιαστής-αθροιστής (UDMA) και ο ενοποιημένος διπλός πολλαπλασιαστής-αθροιστής με διοχέτευση (UDMA pipeline). Οι μονάδες αυτές αποτελούν σύνθετα, βέλτιστα σχεδιασμένα αριθμητικά κυκλώματα τα οποία λαμβάνουν υπόψη τα ιδιαίτερα χαρακτηριστικά των αλγορίθμων ΨΕΣ και άλλων data-intensive εφαρμογών, και κυρίως το γεγονός πως συνήθως αποτελούνται από συνεχείς πολλαπλασιασμούς και προσθαιρέσεις δεδομένων με ορισμένους συντελεστές.

Στόχος είναι, ύστερα από την μελέτη και κατασκευή της αρχιτεκτονικής, να εξαχθούν πειραματικά αποτελέσματα που αφορούν πρωτίστως την ταχύτητα της απεικόνισης που είναι και βασικό ζητούμενο, αλλά και την καταλαμβανόμενη επιφάνεια (area) και την ενέργεια (energy) που καταναλώνεται για την απεικόνιση κάθε DSP πυρήνα. Τα πειραματικά αποτελέσματα αυτά, θα αξιολογηθούν, και μέσω της σύγκρισης με την FCU-based αρχιτεκτονική που αποτελεί μία παρόμοιας λογικής αποδεδειγμένα αποδοτική υλοποίηση, θα εξαχθούν τα τελικά συμπεράσματα που αφορούν την αποτελεσματικότητα της προτεινόμενης αρχιτεκτονικής στην απεικόνιση των υπό εξέταση αλγορίθμων.

Όλες οι απαραίτητες περιγραφές υλικού για τις απεικονίσεις και τις επιμέρους μονάδες έγιναν σε γλώσσα Verilog. Για τους δενδρικούς αθροιστές και τους αθροιστές πρόβλεψης κρατούμενου χρησιμοποιήθηκαν τα standard ip blocks της DesignWare. Η προσομοίωση και η επιβεβαίωση της επιθυμητής λειτουργίας των

κυκλωμάτων πραγματοποιήθηκε με το πακέτο λογισμικού Modelsim. Για την σύνθεση των κυκλωμάτων και την εξαγωγή πειραματικών αποτελεσμάτων όσον αφορά την καθυστέρηση (latency) και την επιφάνεια (area) χρησιμοποιήθηκε ο Synopsys Design Compiler. Η σύνθεση έγινε με χρήση της βιβλιοθήκης TSMC στα 65 nm. Η κατανάλωση ισχύος (average calculation) μετρήθηκε με τη βοήθεια του εργαλείου Primetime-PX.

1.3 Οργάνωση κειμένου

Στο κεφάλαιο 2 γίνεται μια παρουσίαση της θεωρίας όλων των αριθμητικών συστημάτων και των κυκλωμάτων που θα χρησιμοποιηθούν στην παρούσα εργασία.

Στο κεφάλαιο 3 αναλύονται ενδελεχώς οι μονάδες UDMA μέσω όλων των επιμέρους δομικών μονάδων που τις απαρτίζουν. Υπολογίζεται θεωρητικά η κρίσιμη καθυστέρηση κάθε κυκλώματος και παρουσιάζονται οι λειτουργίες που επιτελούν.

Στο κεφάλαιο 4 παρουσιάζεται η μεθοδολογία που επιλέχθηκε για την σχεδίαση της αρχιτεκτονικής, και τι αυτή περιλαμβάνει.

Στο κεφάλαιο 5 παρουσιάζονται τα πειραματικά αποτελέσματα που προέκυψαν τόσο σε επίπεδο απόδοσης της μονάδας όσο και απεικόνισης των υπό εξέταση αλγορίθμων από την αρχιτεκτονική. Πραγματοποιείται σύγκριση με την μονάδα FCU και την αντίστοιχη αρχιτεκτονική.

Στο κεφάλαιο 6 γίνεται σύνοψη της εργασίας, αναλύονται τα συμπεράσματα που προέκυψαν και παρατίθενται πιθανές επεκτάσεις.

Στο κεφάλαιο 7 αναφέρεται η βιβλιογραφία που χρησιμοποιήθηκε κατά την εκπόνηση της παρούσας εργασίας.

2

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτό το κεφάλαιο καλύπτεται το θεωρητικό υπόβαθρο που είναι απαραίτητο για τη λεπτομερή μελέτη της προτεινόμενης ευέλικτης αρχιτεκτονικής και της απεικόνισης αλγορίθμων σε αυτή.

Αρχικά, παρουσιάζεται το δυαδικό σύστημα αρίθμησης, και πιο συγκεκριμένα η αναπαράσταση αριθμών στη μορφή συμπληρώματος του δύο, στην οποία βρίσκονται οι είσοδοι και οι έξοδοι της προτεινόμενης αρχιτεκτονικής και είναι και η πιο διαδεδομένη στα αριθμητικά κυκλώματα. Ιδιαίτερη αναφορά γίνεται και στην αριθμητική αναπαράσταση Carry-Save (CS), καθώς οι αριθμοί στο εσωτερικό της προτεινόμενης μονάδας βρίσκονται σε αυτή τη μορφή.

Επίσης, παρουσιάζονται οι κωδικοποιήσεις Booth και Modified Booth που προσφέρουν αποδοτικότερες υλοποιήσεις της πράξης του πολλαπλασιασμού, ελαττώνοντας την πολυπλοκότητα των κυκλωμάτων πολλαπλασιασμού.

Παρατίθενται επίσης μερικές βασικές τοπολογίες αριθμητικών κυκλωμάτων για την εκτέλεση των πράξεων της πρόσθεσης και του πολλαπλασιασμού και τονίζονται οι ιδιότητες τους βάσει των οποίων γίνεται η επιλογή για την ενσωμάτωσή τους στην προτεινόμενη αρχιτεκτονική.

Τέλος, αναλύονται τα είδη και τα χαρακτηριστικά των αλγορίθμων ψηφιακής επεξεργασίας σήματος (DSP) και επεξηγείται ο λόγος για τον οποίο η προτεινόμενη αρχιτεκτονική είναι αποτελεσματική στις εφαρμογές αυτές.

2.1 Αριθμητικά συστήματα

2.1.1 Δυαδικό σύστημα

Το δυαδικό σύστημα (binary system) είναι το πιο παλιό, απλό αλλά και ευρέως χρησιμοποιούμενο αριθμητικό σύστημα στα ψηφιακά κυκλώματα. Είναι η βάση πάνω στην οποία στηρίχθηκε η ανάπτυξη της ψηφιακής λογικής, και σχεδόν όλα τα επόμενα συστήματα αποτελούν παραλλαγές ή επεκτάσεις του.

Το δυαδικό σύστημα, θεωρούμενο κατ' αρχήν σαν σύστημα αρίθμησης, είναι ακριβώς όμοιο με το γνωστό μας δεκαδικό, με την μόνη διαφορά ότι χρησιμοποιεί μόνο δύο διαφορετικά ψηφία για την αναπαράσταση όλων των αριθμών. Το δυαδικό σύστημα καθιερώθηκε στον κόσμο των ψηφιακών συστημάτων διότι η χρήση δύο μόνο διακριτών καταστάσεων (που προφανώς μπορούν να αντιστοιχηθούν στα ψηφία 0 και 1) καθιστά πιο εύκολο τον διαχωρισμό τους σε επίπεδα τάσεων, κάτι που δεν θα ήταν καθόλου εύκολο αν χρησιμοποιούταν πχ ένα δεκαδικό ή οκταδικό σύστημα.

Η δυαδική αναπαράσταση ακολουθεί το ίδιο σύστημα με βάρη που χρησιμοποιεί και η δεκαδική αναπαράσταση, μόνο που σε αυτή την περίπτωση η βάση είναι το δύο (και όχι το δέκα) και επομένως τα δυνατά ψηφία ανήκουν στο σύνολο $\{0,1\}$ (αντί στο $\{0,1,2,3,4,5,6,7,8,9\}$) ενώ τα βάρη είναι και αυτά δυνάμεις του δύο (αντί του δέκα).

Η σχέση ισότητας που αντιστοιχεί έναν δεκαδικό και έναν δυαδικό αριθμό είναι η παρακάτω:

$$a_{(10)} = \sum_{i=0}^{n-1} b_i \cdot 2^i = b_{n-1}b_{n-2}\dots b_2b_1b_{0(2)}$$

όπου,

a: η ισοδύναμη δεκαδική τιμή του αριθμού

b_i : τα ψηφία του δυαδικού αριθμού που ανήκουν στο σύνολο $\{0,1\}$

n: το πλήθος των δυαδικών ψηφίων του αριθμού

Για παράδειγμα, ο δεκαδικός αριθμός 180 αντιστοιχίζεται στο δυαδικό σύστημα με τον εξής τρόπο:

$$180_{(10)} = 10110100_{(2)} = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

Η αναπαράσταση ενός αριθμού στο συμβατικό δυαδικό σύστημα είναι αμφιμονοσήμαντη, δηλαδή υπάρχει μια ένα προς ένα αντιστοιχία μεταξύ ενός δυαδικού και ενός δεκαδικού αριθμού. Ακολουθώντας τον ίδιο τρόπο γραφής με το δεκαδικό σύστημα, το πιο αριστερό ψηφίο ενός δυαδικού αριθμού έχει το μεγαλύτερο

βάρος (Most Significant Bit - MSB) ενώ το πιο δεξιό έχει το μικρότερο βάρος (Least Significant Bit - LSB) .

2.1.1.1 Πρόσθεση αριθμών

Η πρόσθεση των αριθμών στο συμβατικό δυαδικό σύστημα είναι ακριβώς όμοια με την πρόσθεση στο δεκαδικό σύστημα. Κάθε φορά αθροίζονται τα ψηφία ίδιου βάρους και το τυχόν κρατούμενο διαδίδεται στην αμέσως επόμενη στα αριστερά βαθμίδα με την μόνη προφανή διαφορά να είναι η διαφορετική βάση του συστήματος. Ένα παράδειγμα μιας τέτοιας πρόσθεσης είναι η πρόσθεση των αριθμών 154 και 185:

$$\begin{array}{r} 154 = 10011010 \\ +185 = +1011100 \\ \hline 339 = 10101001 \end{array}$$

Το συμβατικό δυαδικό σύστημα όμως, παρουσιάζει ένα σημαντικό μειονέκτημα, και αυτό είναι ότι δεν μπορεί να παρουσιάσει αρνητικούς αριθμούς. Έτσι το αποτέλεσμα μιας αφαίρεσης όπου ο δεύτερος αριθμός θα ήταν μεγαλύτερος από τον πρώτο δεν θα μπορούσε να παρασταθεί με αυτό το σύστημα.

2.1.2 Αναπαράσταση σε μορφή συμπληρώματος ως προς δύο

Η ανάγκη για απεικόνιση αντικειμένων πέραν των φυσικών αριθμών εισήγαγε και άλλες αναπαραστάσεις και κωδικοποιήσεις, όπως πχ ο κώδικας ASCII που αντιστοιχεί έναν χαρακτήρα σε έναν 8-bit δυαδικό αριθμό.

Η ανάγκη αναπαράστασης αρνητικών αριθμών δημιούργησε την απαίτηση για μια άλλη κωδικοποίηση, πέραν του συμβατικού δυαδικού συστήματος. Μια προφανής λύση είναι η χρήση ενός επιπλέον bit ως ένδειξη προσήμου (0 = + , 1 = -), η οποία είναι γνωστή σαν αναπαράσταση προσήμου-μέτρου. Δυστυχώς όμως, όσο κοντά κι αν είναι μια τέτοια αναπαράσταση στη διαίσθησή μας, δεν είναι ιδιαίτερα βολική στην εκτέλεση αριθμητικών υπολογισμών.

Η αναπαράσταση που έχει τα καλύτερα αποτελέσματα προς αυτή την κατεύθυνση, δίνει δηλαδή μια σχεδιαστικά εύκολη υλοποίηση τόσο της πρόσθεσης όσο και της αφαίρεσης, για θετικούς και αρνητικούς αριθμούς, σε συνδυασμό με μια απλή διαδικασία για την εύρεση του αντίθετου ενός αριθμού, ενώ είναι ταυτόχρονα κατανοητή με μια απλή ανάγνωση, είναι η μορφή του συμπληρώματος ως προς δυο.

Στην αναπαράσταση αυτή, το πιο σημαντικό ψηφίο του αριθμού (MSB) λαμβάνει το αντίστοιχο αρνητικό βάρος, μετατοπίζοντας έτσι το εύρος τιμών που μπορούν να αναπαρασταθούν με N bits από τις τιμές 0 ως 2^N-1 στις τιμές -2^{N-1} ως $2^{N-1}-1$.

Με αυτή την παραδοχή, η σχέση ισότητας που αντιστοιχεί έναν δεκαδικό αριθμό και έναν αριθμό σε μορφή συμπληρώματος ως προς δύο, είναι:

$$a_{(10)} = -b_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} b_i \cdot 2^i = b_{n-1}b_{n-2} \dots b_2b_1b_0$$

όπου,

a: η ισοδύναμη δεκαδική τιμή του αριθμού

b_i : τα ψηφία του δυαδικού αριθμού που ανήκουν στο σύνολο $\{0,1\}$

n: το πλήθος των δυαδικών ψηφίων του αριθμού

Όπως ακριβώς και στο συμβατικό δυαδικό σύστημα, έτσι και στην μορφή του συμπληρώματος του δύο η αναπαράσταση ενός αριθμού είναι αμφιμονοσήμαντη.

Η διαφορά των δύο αναπαραστάσεων φαίνεται με το επόμενο παράδειγμα, ο δυαδικός αριθμός 10110100 ενώ στο συμβατικό δυαδικό σύστημα αντιστοιχεί στην δεκαδική τιμή 180, στην αναπαράσταση σε μορφή συμπληρώματος ως προς δύο αντιστοιχεί σε αρνητικό δεκαδικό αριθμό ίσο με την τιμή

$$10011010_{(2)} = -1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = -76_{(10)}$$

Προφανώς, στην μορφή συμπληρώματος του δύο, όλοι οι θετικοί αριθμοί έχουν το πιο σημαντικό τους ψηφίο (MSB) ίσο με το μηδέν, και τα υπόλοιπα bits είναι τα ίδια με αυτά που θα είχαν στη συμβατική δυαδική αναπαράσταση.

2.1.2.1 Συμπλήρωμα (ως προς δύο)

Συμπλήρωμα ως προς δύο (two's complement) ενός αριθμού που έχει αναπαρασταθεί στο δυαδικό σύστημα ως προσημασμένος αριθμός, είναι ένας προσημασμένος δυαδικός αριθμός που έχει το ίδιο μέτρο αλλά αντίθετο πρόσημο.

Συγκεκριμένα, για να βρούμε το συμπλήρωμα (ως προς δύο) ενός αριθμού ακολουθούμε τα εξής βήματα:

A) Αντιστρέφουμε όλα τα ψηφία του αριθμού μετατρέποντας τα 0 σε 1 και τα 1 σε 0. Πλέον ο αριθμός βρίσκεται στην μορφή συμπληρώματος ως προς 1.

B) Έπειτα προσθέτουμε μία μονάδα στον αριθμό που προέκυψε.

Για παράδειγμα θα βρούμε το συμπλήρωμα ως προς δύο του αριθμού $10110100 = -76_{(10)}$.

Αντιστρέφοντας τα ψηφία του παίρνουμε τον αριθμό:

01001011

και προσθέτουμε μία μονάδα σε αυτόν τον αριθμό:

$$\begin{array}{r} 01001011 \\ + \\ 1 \\ \hline 01001100 \end{array}$$

Έτσι, βλέπουμε πως το συμπλήρωμα ως προς δύο του $-76_{(10)}$ είναι το $76_{(10)}$ που επαληθεύει τον ορισμό.

2.1.2.2 Πρόσθεση και αφαίρεση δύο αριθμών σε μορφή συμπληρώματος ως προς 2

Η πρόσθεση και η αφαίρεση δύο αριθμών σε μορφή συμπληρώματος ως προς δύο γίνεται με παρόμοιο τρόπο με την πρόσθεση και την αφαίρεση αριθμών στο συμβατικό δυαδικό σύστημα, με δύο όμως βασικές τροποποιήσεις:

A) Όλα τα bits των δύο αριθμών προστίθενται όπως τα bits των συμβατικών δυαδικών αριθμών, εκτός από τα MSB που έχουν αρνητικό βάρος. Για την αναπαράσταση του αποτελέσματος της πρόσθεσης στη βαθμίδα των MSB χρειαζόμαστε δύο ψηφία, τα οποία να καλύπτουν το εύρος τιμών $-2 \dots +1$. Ο λόγος που το αποτέλεσμα έχει αυτό το εύρος τιμών, είναι το ότι στη βαθμίδα αυτή, προστίθενται δύο ψηφία αρνητικού βάρους (τα MSB των αριθμών) και το κρατούμενο εισόδου από την προηγούμενη βαθμίδα, το οποίο είναι θετικού βάρους. Επομένως όλα τα πιθανά αποτελέσματα αναπαρίστανται σωστά με ένα bit θετικού βάρους στη βαθμίδα των MSB και ένα επιπλέον bit αρνητικού βάρους στην αμέσως μεγαλύτερη βαθμίδα. Το τελικό αποτέλεσμα προκύπτει από τη συνένωση των δύο αυτών bit με τα υπόλοιπα bits των χαμηλότερων βαθμίδων του αποτελέσματος που υπολογίστηκαν με το συμβατικό τρόπο.

B) Για να προστεθούν δύο αριθμοί και το αποτέλεσμα να είναι πάντα σωστό, θα πρέπει να έχουν τον ίδιο αριθμό ψηφίων. Για να συμβεί αυτό, αυξάνουμε τον αριθμό με τα λιγότερα ψηφία κατά όσα ψηφία χρειάζεται, έτσι ώστε οι δύο αριθμοί να έχουν τον ίδιο αριθμό ψηφίων. Η επαύξηση αυτή πραγματοποιείται τοποθετώντας τα επιπλέον bits στα αριστερά του MSB του αριθμού και δίνοντας σε όλα την τιμή του

MSB. Με αυτό τον τρόπο δεν μεταβάλλεται η δεκαδική τιμή του αριθμού. Η τεχνική αυτή ονομάζεται επέκταση προσήμου (sign extension).

Παρακάτω δίνεται ένα παράδειγμα για την πρόσθεση δυαδικών αριθμών σε μορφή συμπληρώματος του δύο:

Έστω ότι θέλουμε να προσθέσουμε τους $00110100_{(2)} = 52_{(10)}$ και $1000110_{(2)} = -58_{(10)}$.

Βλέπουμε πως ενώ ο πρώτος αριθμός έχει 8 ψηφία, ο δεύτερος έχει 7. Άρα όπως αναφέραμε παραπάνω, θα πραγματοποιήσουμε επέκταση προσήμου στον δεύτερο αριθμό. Έτσι ο δεύτερος αριθμός θα γίνει:

$1000110_{(2)} = (1)1000110_{(2)}$ που έχει και αυτός 8 ψηφία..

Στην συνέχεια προσθέτουμε όλα τα bit εκτός από τα MSB:

$$\begin{array}{r} (0)0110100 \\ +(1)1000110 \\ \hline 0\ 1111010 \end{array}$$

Το υπογραμμισμένο ψηφίο του αποτελέσματος είναι το κρατούμενο που προέρχεται από την προηγούμενη βαθμίδα των MSB, είναι θετικής αξίας και σε αυτή την περίπτωση έχει την τιμή 0.

Προσθέτουμε τα MSB και το κρατούμενο:

$$(-1) \cdot \text{MSB}_1 + (-1) \cdot \text{MSB}_2 + C_{\text{in}} = (-1) \cdot 0 + (-1) \cdot 1 + 0 = -1$$

Το αποτέλεσμα παριστάνεται ως 11 (και αυτό γιατί είναι ίσο με $(-1) \cdot 2^1 + 1 \cdot 2^0 = -1$) και το τελικό αποτέλεσμα προκύπτει από την συνένωση αυτού του αριθμού με το υπόλοιπο αποτέλεσμα που προέκυψε παραπάνω, όπου παραλείψαμε τα MSB και το κρατούμενο της προηγούμενης βαθμίδας.

Άρα έχουμε:

$$\{11, 1111010\} = 11111010_{(2)} = -6_{(10)} = 52_{(10)} + (-58_{(10)}).$$

Παρακάτω δίνεται ένα δεύτερο παράδειγμα για την αφαίρεση προσημασμένων δυαδικών αριθμών:

Έστω ότι θέλουμε να αφαιρέσουμε τον $0100110_{(2)} = 38_{(10)}$ από τον $01001100_{(2)} = 76_{(10)}$.

Πρώτο βήμα μας είναι να εφαρμόσουμε επέκταση προσήμου στον πρώτο αριθμό αφού έχει 7 και όχι 8 ψηφία όπως ο δεύτερος. Έτσι παίρνουμε τον δυαδικό: $00100110_{(2)} = 38_{(10)}$ και εν συνεχεία βρίσκουμε το συμπλήρωμα του ως προς 1, το οποίο είναι το $11011001_{(2)}$ και του προσθέτουμε μία μονάδα για να πάρουμε το συμπλήρωμα ως προς δύο: $11011010_{(2)}$.

Το τελικό αποτέλεσμα θα βρεθεί αν προσθέσουμε τον αριθμό που μόλις υπολογίσαμε με τον $01001100_{(2)} = 76_{(10)}$:

$$\begin{array}{r} (1)1011010 \\ + \\ (0)1001100 \\ \hline \underline{1\ 0100110} \end{array}$$

Όπου πάλι το υπογραμμισμένο ψηφίο του αποτελέσματος είναι το κρατούμενο που προέρχεται από την προηγούμενη βαθμίδα των MSB, είναι θετικής αξίας και σε αυτή την περίπτωση έχει την τιμή 1.

Προσθέτουμε τα MSB και το κρατούμενο:

$$(-1) \cdot MSB_1 + (-1) \cdot MSB_2 + C_{in} = (-1) \cdot 1 + (-1) \cdot 0 + 1 = -1 + 1 = 0$$

Το οποίο αναπαρίσταται με τον αριθμό 00.

Έτσι το τελικό αποτέλεσμα βρίσκεται ως εξής:

$$\{00, 0100110\} = 000100110_{(2)} = 38_{(10)} = 76_{(10)} - 38_{(10)}.$$

2.1.3 Αναπαράσταση σε μορφή Σωσίματος-Κρατουμένου (Carry-Save)

Η αναπαράσταση σε μορφή Carry-Save ανήκει στα αριθμητικά συστήματα με περίσσεια (redundant). Τα συστήματα αυτά έχουν ως βασικό χαρακτηριστικό ότι δεν υπάρχει αμφιμονοσήμαντη αντιστοιχία με το δεκαδικό σύστημα, συγκεκριμένα υπάρχουν πολλές διαφορετικές μορφές που έχουν την ίδια δεκαδική τιμή.

Ένας αριθμός σε μορφή Carry-Save αποτελείται από δύο ξεχωριστούς δυαδικούς αριθμούς, και η δεκαδική τιμή του προκύπτει από το άθροισμα των επιμέρους αριθμών. Η γενική μορφή του έχει ως εξής: $x^* = x^s + x^c$.

Γενικά, η Carry-Save αναπαράσταση απαιτεί τον διπλάσιο αριθμό bits για την παράσταση της πληροφορίας από ότι θα απαιτούσε πχ η αναπαράσταση σε μορφή συμπληρώματος του δυο. Ωστόσο, η Carry-Save αριθμητική επιτρέπει την εκτέλεση των πράξεων της πρόσθεσης και της αφαίρεσης χωρίς την διάδοση κρατουμένου, και επομένως προσφέρει μια ταχύτερη υλοποίηση των πράξεων αυτών.

2.1.4 Αναπαράσταση σε μορφή Προσημασμένου ψηφίου (Signed Digit)

Η Signed Digit αναπαράσταση ανήκει και αυτή στα redundant αριθμητικά συστήματα. Η γενική μορφή ενός Signed Digit αριθμού είναι η εξής :

$$B = \sum_{i=0}^{n-1} b_i \cdot r^i = b_{n-1}b_{n-2}\dots b_2b_1b_{0(SD)}$$

Όπου το b_i ανήκει στο σύνολο $\{-\alpha, \dots, -1, 0, 1, \dots, \alpha\}$.

Δηλαδή κάθε ψηφίο μπορεί να έχει θετικό, μηδενικό ή ακόμα και αρνητικό βάρος.

Προφανώς, σε ένα τέτοιο σύστημα κάθε ψηφίο b_i απαιτεί περισσότερα από ένα bits για την αναπαράστασή του.

Το μειονέκτημα των επιπλέον bits για την αναπαράσταση της ίδιας πληροφορίας σε αυτές τις περιπτώσεις αντισταθμίζεται από άλλες κυκλωματικές διευκολύνσεις, όπως την απαλοιφή των αλυσίδων διάδοσης κρατουμένου ή την μεγιστοποίηση του αριθμού των μηδενικών ψηφίων της αναπαράστασης.

2.1.4.1 Αναπαράσταση σε μορφή προσημασμένου μέτρου (Signed Magnitude representation, SM)

Στην αναπαράσταση σε μορφή προσημασμένου μέτρου το MSB καθορίζει το πρόσημο του αριθμού και τα υπόλοιπα bits καθορίζουν το μέτρο του, σαν έναν απρόσημο, συμβατικό δυαδικό αριθμό. Όλα τα bits του αριθμού ανήκουν στο σύνολο $\{0, +1\}$. Εάν το MSB είναι 0, τότε ο αριθμός είναι θετικός. Εάν το MSB είναι 1 τότε ο αριθμός είναι αρνητικός.

Για παράδειγμα, ο αριθμός 01011001 σε μορφή Signed Magnitude αντιστοιχεί στο δεκαδικό αριθμό 89. Για την αναπαράσταση του -89 σε μορφή Signed Magnitude απλώς αντιστρέφουμε το MSB από 0 σε 1. Έτσι ο αριθμός 11011001 αντιστοιχεί στον δεκαδικό -89.

Η αναπαράσταση Signed Magnitude είναι αμφιμονοσήμαντη με μοναδική εξαίρεση το δεκαδικό αριθμό 0. Δεδομένου ενός συγκεκριμένου αριθμού από bits για την αναπαράσταση σε μορφή Signed Magnitude, το δεκαδικό 0 μπορεί να αναπαρασταθεί με δύο τρόπους, είτε σαν +0 είτε σαν -0.

Η αναπαράγοντας αυτή βρίσκεται πιο κοντά στον τρόπο παράστασης προσημασμένων αριθμών του δεκαδικού συστήματος, σε σχέση με τη δυαδική μορφή συμπληρώματος ως προς 2. Παρουσιάζει όμως δύο σημαντικά μειονεκτήματα.

1. Οι πράξεις της πρόσθεσης και της αφαίρεσης εκτελούνται με πιο πολύπλοκο τρόπο, καθώς εάν οι δύο τελεστέοι είναι ετερόσημοι πρέπει να συγκριθούν τα μέτρα τους για να καθοριστεί το πρόσημο του αποτελέσματος.
2. Με δεδομένο αριθμό ψηφίων, η SM μορφή αναπαριστά πάντοτε ένα δεκαδικό αριθμό λιγότερο από τη δυαδική μορφή συμπληρώματος ως προς 2, καθώς το 0 αναπαρίσταται με 2 τρόπους.

2.2 Κωδικοποιήσεις Booth και Modified Booth

2.2.1 Κωδικοποίηση Booth

Ο αλγόριθμος που φέρει το όνομα του Andrew Donald Booth, επινοήθηκε το 1950 και συνίσταται στο εξής:

Έστω δυαδικός αριθμός X σε μορφή συμπληρώματος ως προς δύο. Όπως είδαμε, ο X δίνεται από την παρακάτω σχέση:

$$X_{(10)} = -x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i$$

Όπου x_i τα ψηφία του δυαδικού αριθμού X .

Ο αριθμός X μπορεί να γραφεί ισοδύναμα: $X = 2X - X$

$2X =$	$-x_{n-1}$	x_{n-2}	x_{n-3}	\dots	x_0	0
$-X =$	0	$-x_{n-1}$	$-x_{n-2}$	\dots	x_1	x_0
		z_{n-1}	z_{n-2}	\dots	z_1	z_0

όπου:

$$z_0 = 0 - x_0,$$

$$z_1 = x_0 - x_1$$

$$z_2 = x_1 - x_2$$

.

.

$$z_{n-2} = x_{n-3} - x_{n-2}$$

Το z_{n-1} που χρήζει περαιτέρω ανάλυσης προκύπτει από την παρακάτω σχέση:

$$z_{n-1} = -2x_{n-1} + x_{n-2} + x_{n-1} = x_{n-2} - x_{n-1}$$

Επομένως, ο αριθμός X δίνεται από την σχέση:

$$X = \sum_{i=0}^{n-1} z_i \cdot 2^i$$

όπου z_i είναι τα ψηφία που προέκυψαν από την κωδικοποίηση του X .

Αυτή είναι και η κωδικοποίηση Booth και το μεγάλο της πλεονέκτημα είναι ότι εφαρμόζεται σε κάθε αριθμό σε μορφή συμπληρώματος ως προς δύο, ανεξάρτητα από το πρόσημο του.

2.2.1.1 Πολλαπλασιασμός με την μέθοδο Booth

Έστω το γινόμενο $P = X \cdot Y$, με $X = x_{n-1}x_{n-2} \dots x_1x_0$ και $Y = y_{n-1}y_{n-2} \dots y_1y_0$.

Αν χρησιμοποιήσουμε την κωδικοποίηση Booth για τον παράγοντα Y , θα έχουμε:

$$P = X \cdot Y = \sum_{i=0}^{n-1} (y_{i-1} - y_i) \cdot X \cdot 2^i$$

Η παραπάνω σχέση υποδεικνύει ότι σε κάθε βήμα i , το X πολλαπλασιάζεται με ένα από τα στοιχεία του συνόλου $\{-1, 0, 1\}$ σύμφωνα με το αποτέλεσμα της αφαίρεσης των δύο διαδοχικών ψηφίων του πολλαπλασιαστή Y . Στην πραγματικότητα με την κωδικοποίηση Booth ελέγχουμε τις σειρές από μονάδες που παρουσιάζονται μέσα σε κάθε δυαδικό αριθμό. Όλες οι δυνατές περιπτώσεις, μαζί με τα κωδικοποιημένα ψηφία που προκύπτουν, καθώς και τις λογικές λειτουργίες που αντιπροσωπεύουν, φαίνονται αναλυτικά στον Πίνακα 2.1.

Πίνακας 2.1 Αντιστοίχιση δυαδικών ψηφίων με λειτουργίες και κωδικοποίηση Booth

Y_i	Y_{i-1}	Κωδικοποιημένα ψηφία z_i ($y_{i-1} - y_i$)	Αποτέλεσμα ελέγχου
0	0	0	Δεν υπάρχει σειρά από 1
0	1	1	Τέλος σειράς από 1
1	0	-1	Αρχή σειράς από 1
1	1	0	Μέση σειράς από 1

Είναι φανερό πως η κωδικοποίηση Booth ανήκει στα συστήματα με αναπαράσταση προσημασμένου ψηφίου (signed digit).

Συνοψίζοντας, η κωδικοποίηση Booth παρουσιάζει τα παρακάτω πλεονεκτήματα:

Η κωδικοποίηση Booth τόσο των θετικών και όσο και των αρνητικών αριθμών στη μορφή συμπληρώματος του δύο γίνεται ακριβώς με τον ίδιο τρόπο, όπως αναφέραμε και πιο πάνω.

Κάθε ψηφίο του κωδικοποιημένου αριθμού παράγεται ανεξάρτητα από τα άλλα, αφού είναι συνάρτηση μόνο των δύο bit $Y_i Y_{i-1}$. Αυτό σημαίνει ότι και κάθε μερικό γινόμενο σε έναν πολλαπλασιασμό όπου χρησιμοποιείται η κωδικοποίηση Booth παράγεται αμέσως. Το γεγονός αυτό βοηθά στη γρήγορη εκτέλεση του πολλαπλασιασμού.

Το μεγαλύτερο πλεονέκτημα όμως, είναι ότι η κωδικοποίηση Booth μπορεί να μειώσει τον αριθμό των μη μηδενικών μερικών γινομένων και έτσι να μειωθούν οι προσθέσεις που πρέπει να γίνουν εάν ο αριθμός που κωδικοποιήθηκε είχε μεγάλο αριθμό μονάδων. Ωστόσο, αν ο αριθμός είχε απομονωμένες μονάδες, τότε είναι πολύ πιθανόν η κωδικοποίηση Booth να μην δώσει τα αναμενόμενα αποτελέσματα και να αυξήσει τον αριθμό των μη μηδενικών μερικών γινομένων και έτσι να επιβαρύνει το κύκλωμα υλοποίησης της πράξης στην οποία λαμβάνει μέρος ο αριθμός.

Για να αντιμετωπιστεί αυτό το πρόβλημα, προτάθηκε μια διαφοροποιημένη μέθοδος κωδικοποίησης, η οποία παρουσιάζεται στη συνέχεια.

2.2.2 Κωδικοποίηση Modified Booth

Μια επέκταση του απλού αλγορίθμου Booth είναι ο τροποποιημένος αλγόριθμος Booth που επεκτείνει το σύνολο των ψηφίων κωδικοποίησης από το σύνολο $\{-1, 0, +1\}$ στο σύνολο $\{-2, -1, 0, +1, +2\}$ και προκύπτει αλγεβρικά από τον απλό αλγόριθμο του Booth.

Όπως και η απλή κωδικοποίηση Booth, έτσι και η Modified Booth κωδικοποίηση ανήκει στα συστήματα με αναπαράσταση προσημασμένου ψηφίου (signed digit).

Στην αμέσως προηγούμενη παράγραφο είδαμε πως ισχύει η σχέση:

$$Y = \sum_{i=0}^{n-1} z_i \cdot 2^i$$

με z_i να είναι τα ψηφία που προέκυψαν από την απλή κωδικοποίηση Booth του Y .

Αυτή η σχέση μπορεί να αναλυθεί περαιτέρω ως εξής:

$$\begin{aligned}
 Y &= \sum_{i=0}^{n-1} z_i \cdot 2^i = \sum_{j=0}^{\frac{n-1}{2}} (z_{2j} \cdot 2^{2j} + z_{2j+1} \cdot 2^{2j+1}) = \sum_{j=0}^{\frac{n-1}{2}} (z_{2j} + z_{2j+1} \cdot 2) \cdot 2^{2j} \\
 &= \sum_{j=0}^{\frac{n-1}{2}} (w_j) \cdot 4^j
 \end{aligned}$$

Με το w_i να είναι το κωδικοποιημένο ψηφίο κατά Modified Booth, το οποίο μπορεί να αναλυθεί ως εξής:

$$w_i = y_{2i} - 2 \cdot y_{2i+1} + y_{2i-1}$$

Όπως φαίνεται και από την παραπάνω σχέση, ο τροποποιημένος αλγόριθμος Booth χρησιμοποιεί τριάδες ψηφίων του δυαδικού αριθμού για την κωδικοποίηση. Οι τριάδες αυτές επικαλύπτονται κατά ένα ψηφίο και επειδή για $i=0$ χρειάζεται μια τιμή y_{-1} για την σωστή κωδικοποίηση, πρέπει να θεωρήσουμε ότι $y_{-1} = 0$ για να έχουμε σωστά αποτελέσματα.

Όλες οι δυνατές περιπτώσεις, μαζί με τα κωδικοποιημένα ψηφία που προκύπτουν, καθώς και την αντιστοιχία με την κωδικοποίηση Booth, φαίνονται αναλυτικά στον Πίνακα 2.2.

Πίνακας 2.2 Αντιστοίχιση δυαδικών ψηφίων με απλή κωδικοποίηση Booth & Modified Booth

Y_{2i+1}	Y_{2i}	Y_{2i-1}	Κωδικοποίηση Booth		Κωδικοποίηση Modified
			Z_{2i+1}	Z_{2i}	Booth: W_i
0	0	0	0	0	0
0	0	1	0	+1	+1
0	1	0	+1	-1	+1
0	1	1	+1	0	+2
1	0	0	-1	0	-2
1	0	1	-1	+1	-1
1	1	0	0	-1	-1
1	1	1	0	0	0

2.2.2.1 Πολλαπλασιασμός με την μέθοδο Modified Booth

Η κωδικοποίηση Modified Booth χρησιμοποιείται ευρέως για την υλοποίηση πολλαπλασιαστών. Κάθε κωδικοποιημένο ψηφίο καθορίζει τη λειτουργία που πρόκειται να γίνει στον πολλαπλασιασμό, ακριβώς όπως και στον απλό αλγόριθμο κωδικοποίησης Booth. Έστω ότι πρόκειται να πολλαπλασιάσουμε δύο αριθμούς 8-bit A και B, όπου B είναι ο πολλαπλασιαστής. Τα ψηφία του B κωδικοποιούνται κατά Modified Booth. Τα Modified Booth ψηφία που προκύπτουν, ανάλογα με την τιμή τους καθορίζουν τις λειτουργίες που θα εκτελεστούν. Οι λειτουργίες αυτές φαίνονται στον Πίνακα 2.3.

Πίνακας 2.3 Αντιστοιχία λειτουργιών και κωδικοποιημένων ψηφίων

Κωδικοποιημένο ψηφίο	Λειτουργία
0	Πρόσθεσε το 0 στο μερικό γινόμενο
+1	Πρόσθεσε το (A) στο μερικό γινόμενο
+2	Πρόσθεσε το (2A) στο μερικό γινόμενο
-2	Αφαίρεσε το (2A) από το μερικό γινόμενο
-1	Αφαίρεσε το (A) από το μερικό γινόμενο

Για να γίνει κατανοητή η διαδικασία πολλαπλασιασμού δύο αριθμών, παρατίθεται ένα παράδειγμα όπου ο πολλαπλασιαστής κωδικοποιείται με τον αλγόριθμο Modified Booth:

Έστω $A = 10110101_{(2)} = -75_{(10)}$ και $B = 01110010_{(2)} = 114_{(10)}$.

Ο B κωδικοποιημένος σύμφωνα με τον αλγόριθμο Modified Booth γίνεται:

$$B = 2 \bar{1} 1 \bar{2}.$$

Επειδή θα μας χρειαστούν οι αριθμοί $2A$, \bar{A} και $2\bar{A}$ τους υπολογίζουμε:

$$2A = 101101010, \bar{A} = 01001011 \text{ και } 2\bar{A} = 010010110.$$

Ο πολλαπλασιασμός του A με τον κωδικοποιημένο B φαίνεται στον Πίνακα 2.4.

Πίνακας 2.4 Παράδειγμα πολλαπλασιασμού με χρήση κωδικοποίησης Modified Booth

A = 10110101 B = 01110010	
Μερικό γινόμενο = 0 0 0 0 0 0 0 0	
0 1	$\begin{array}{r} \underline{0} 0 1 0 0 1 0 1 1 0 \\ 1 0 1 1 0 1 \end{array}$
	<p>-2 Πρόσθεσε -2A (Πρώτο μερικό γινόμενο)</p> <p>+1 Πρόσθεσε A (Δεύτερο μερικό γινόμενο)</p>
	$\begin{array}{r} \underline{11} 1 1 0 1 1 0 1 0 1 0 \\ 0 1 0 0 1 0 1 1 \end{array}$
	<p>Άθροισμα των δύο πρώτων μερικών γινομένων</p> <p>-1 Πρόσθεσε -A (Τρίτο μερικό γινόμενο)</p>
	$\begin{array}{r} \underline{000} 0 1 0 0 0 0 0 1 1 0 1 0 \\ 1 0 1 1 0 1 0 1 0 \end{array}$
	<p>Άθροισμα των τριών μερικών γινομένων</p> <p>+2 Πρόσθεσε +2A (Τέταρτο μερικό γινόμενο)</p>
1 0	$1 0 1 1 1 1 0 1 0 0 1 1 0$ <p>Τελικό αποτέλεσμα = -8550</p>

Όπου τα υπογραμμισμένα bits αποτελούν την επέκταση προσήμου για να βγει σωστό το αποτέλεσμα.

Ο Modified Booth έχει σαν πλεονέκτημα το ότι εφαρμόζεται ανεξάρτητα από το αν οι αριθμοί είναι σε απλή δυαδική αναπαράσταση ή σε μορφή συμπληρώματος ως προς δυο, μαζί με τα υπόλοιπα πλεονεκτήματα της απλής κωδικοποίησης Booth.

Το μεγαλύτερο όμως πλεονέκτημα που επιτυγχάνεται με τον τροποποιημένο αλγόριθμο του Booth, είναι το ότι έχουμε μείωση του αριθμού των μερικών γινομένων στο μισό, σε σχέση με τις προηγούμενες κωδικοποιήσεις, και επομένως λιγότερες προσθέσεις να εκτελέσουμε, με προφανές όφελος το κέρδος σε ταχύτητα λειτουργίας του πολλαπλασιαστή αλλά και την μείωση της επιφάνειας που καταλαμβάνει. Η κωδικοποίηση αυτή είναι και η οποία χρησιμοποιείται στους πολλαπλασιαστές της προτεινόμενης αρχιτεκτονικής.

2.3 Είδη αθροιστών και πολλαπλασιαστών

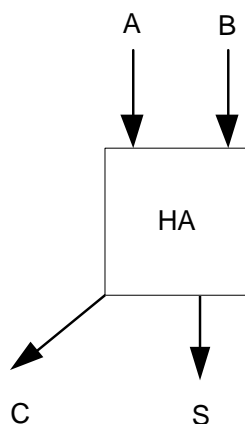
2.3.1 Δομικά στοιχεία

Τα περισσότερα κυκλώματα που επιτελούν τις πράξεις της πρόσθεσης ή του πολλαπλασιασμού αποτελούνται από κάποια βασικά στοιχεία τα οποία χρησιμοποιούνται σαν μονάδες πρόσθεσης των επιμέρους όρων (μερικών γινομένων).

Οι βασικότερες από αυτές τις μονάδες είναι ο ημιαθροιστής και ο πλήρης αθροιστής, που παρουσιάζονται στη συνέχεια. Επίσης, παρουσιάζεται και η γεννήτρια πρόβλεψης κρατουμένου, που αποτελεί βασικό στοιχείο των αθροιστών πρόβλεψης κρατουμένου.

2.3.1.1 Ημιαθροιστής

Ο ημιαθροιστής (Half Adder) είναι ένα στοιχειώδες κύκλωμα που επιτελεί την πράξη της πρόσθεσης δύο bits. Ως εκ τούτου, έχει δύο εισόδους, οι οποίες είναι ψηφία του ίδιου βάρους και επιτελεί την πρόσθεση αυτών των δύο ψηφίων, παράγοντας δύο εξόδους, η μία του ίδιου βάρους με τις εισόδους (Save, S) και μία άλλη στο αμέσως μεγαλύτερο (Carry, C). Το σχήμα του ημιαθροιστή φαίνεται αμέσως παρακάτω:



Σχήμα 2.1 Ημιαθροιστής

Οι σχέσεις που συνδέουν τις εισόδους και τις εξόδους του ημιαθροιστή είναι οι εξής:

$$S = A \oplus B$$

$$C = A \cdot B$$

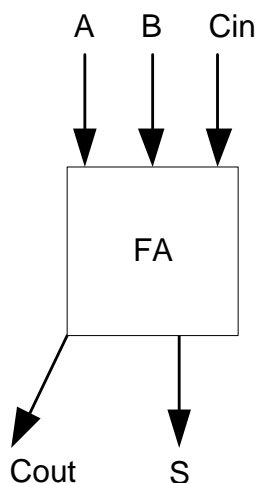
Οι οποίες προκύπτουν από τον παρακάτω πίνακα αληθείας:

Πίνακας 2.5 Πίνακας αληθείας ημιαθροιστή

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

2.3.1.2 Πλήρης αθροιστής

Ο πλήρης αθροιστής (Full Adder) είναι ένα στοιχειώδες κύκλωμα που επιτελεί την πράξη της πρόσθεσης τριών bits, γι' αυτό αναφέρεται στην βιβλιογραφία και ως 3-2 συμπιεστής. Ως εκ τούτου, έχει τρεις εισόδους, οι οποίες είναι ψηφία του ίδιου βάρους και επιτελεί την πρόσθεση αυτών των τριών ψηφίων, παράγοντας δύο εξόδους, η μία του ίδιου βάρους με τις εισόδους (Save, S) και μία άλλη στο αμέσως μεγαλύτερο (Carry, C). Το σχήμα του πλήρους αθροιστή φαίνεται αμέσως παρακάτω:



Σχήμα 2.2 Πλήρης αθροιστής

Οι σχέσεις που συνδέουν τις εισόδους και τις εξόδους του πλήρους αθροιστή είναι οι εξής:

$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = (A \cdot B) + (A \cdot C_{in}) + (B \cdot C_{in})$$

Οι οποίες προκύπτουν από τον παρακάτω πίνακα αληθείας:

Πίνακας 2.6 Πίνακας αληθείας πλήρους αθροιστή

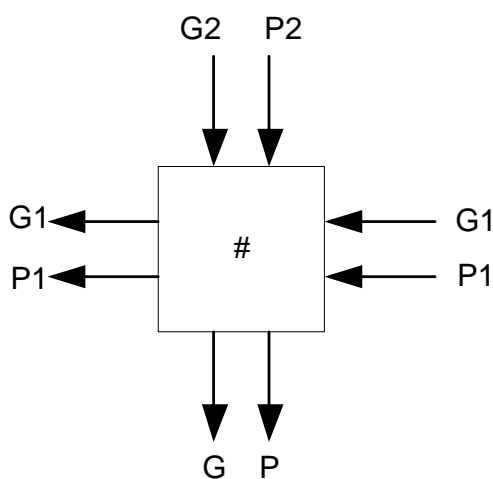
A	B	C _{in}	C _{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

2.3.1.3 Κύκλωμα πρόβλεψης κρατουμένου

Το κύκλωμα πρόβλεψης κρατουμένου αποτελεί βασικό δομικό στοιχείο της γεννήτριας κρατουμένου, η οποία είναι η «καρδιά» ενός αθροιστή πρόβλεψης κρατουμένου.

Δέχεται ως εισόδους τα σήματα G1, P1, G2, P2 τα οποία είναι τα σήματα γέννησης (G, Generation) και διάδοσης (P, Propagation) κρατουμένου δύο βαθμίδων διαφορετικής αξίας, ενώ δίνει σαν έξοδο τα σήματα G, P τα οποία αποτελούν δύο νέα σήματα γεννήσεως και διάδοσης κρατουμένου, στην μεγαλύτερη από τις δύο βαθμίδες.

Το σχήμα του τελεστή # (κύκλωμα πρόβλεψης κρατουμένου) φαίνεται αμέσως παρακάτω:



Σχήμα 2.3 Κύκλωμα πρόβλεψης κρατουμένου

Οι σχέσεις που συνδέουν τις εισόδους και τις εξόδους του κυκλώματος πρόβλεψης κρατούμενου είναι οι εξής:

$$G = G2 + (G1 \cdot P2)$$

$$P = P1 \cdot P2$$

2.3.2 Αθροιστές

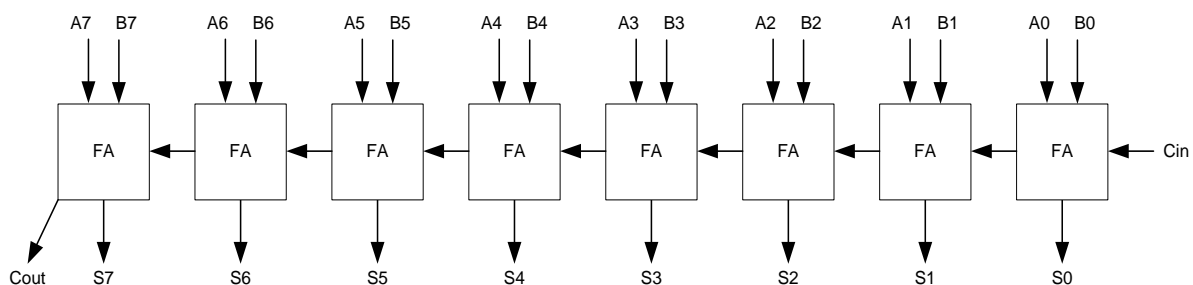
2.3.2.1 Αθροιστής διάδοσης κρατούμενου (Carry Propagate Adder)

Ο αθροιστής διάδοσης κρατούμενου (Carry Propagate Adder) πραγματοποιεί την πράξη της πρόσθεσης με βάση τον αλγόριθμο που ξέρουμε για την πρόσθεση δεκαδικών αριθμών, δηλαδή προσθέτει δύο ψηφία ίσου βάρους και παράγει ένα ψηφίο ίσου βάρους, το οποίο είναι το άθροισμά τους, καθώς και ένα κρατούμενο με το αμέσως μεγαλύτερο βάρος.

Κυκλωματικά, αποτελείται από ένα δίκτυο πλήρων αθροιστών συνδεδεμένων σε σειρά. Κάθε πλήρης αθροιστής αντιστοιχεί σε μία βαθμίδα συγκεκριμένου βάρους. Το κρατούμενο εξόδου (Cout) του κάθε πλήρους αθροιστή συνδέεται με το κρατούμενο εισόδου (Cin) του πλήρους αθροιστή της επόμενης βαθμίδας, και το αποτέλεσμα της πράξης αποτελείται από τα Save ψηφία όλων των πλήρων αθροιστών, καθώς και το κρατούμενο εξόδου της πιο σημαντικής βαθμίδας.

Με αυτόν τον τρόπο τα κρατούμενα διαδίδονται μέσα στην δομή του αθροιστή από την χαμηλότερη μέχρι την υψηλότερη βαθμίδα, και αυτός είναι ο λόγος που ο αθροιστής πήρε αυτό το όνομα.

Παρακάτω φαίνεται ένα κύκλωμα αθροιστή διάδοσης κρατούμενου 8-bit, το οποίο μπορεί να δεχτεί σαν είσοδο είτε δύο δυαδικούς αριθμούς είτε έναν carry save αριθμό. Η έξοδος του είναι ένας δυαδικός αριθμός που αποτελείται από τα ψηφία {Cout,S7,S6,S5,S4,S3,S2,S1,S0}.



Σχήμα 2.4 Κύκλωμα αθροιστή διάδοσης κρατούμενου

Το παραπάνω κύκλωμα μπορεί να λειτουργήσει και για δυαδικούς αριθμούς σε μορφή συμπληρώματος του δύο, αν προστεθεί έλεγχος για τις περιπτώσεις υπερχειλίσης, ή αν απλά αντιστραφούν τα ψηφία αρνητικής αξίας {A7, B7, Cout}.

Αν και ο αθροιστής διάδοσης κρατουμένου έχει πολύ απλή δομή και πολύ μικρή επιφάνεια, η διάδοση του κρατουμένου τον καθιστά απαγορευτικά αργό για τις περισσότερες εφαρμογές καθώς η καθυστέρηση (critical path) ενός N-bit αθροιστή είναι N επίπεδα πλήρων αθροιστών.

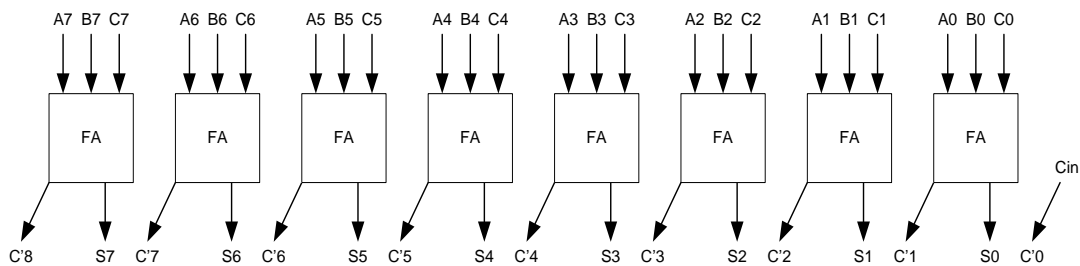
2.3.2.2 Αθροιστής σωσίματος κρατουμένου (Carry Save Adder)

Ο αθροιστής σωσίματος κρατουμένου (Carry Save Adder, CSA) λειτουργεί παρόμοια με τον αθροιστή διάδοσης κρατουμένου, όμως δεν παράγει έναν δυαδικό αριθμό ως αποτέλεσμα, αντί αυτού διατηρεί το αποτέλεσμα σε μορφή Carry-Save όπως προκύπτει από τις επιμέρους αθροίσεις κάθε βαθμίδας.

Η δομή του αθροιστή σωσίματος κρατουμένου είναι όμοια με τον αθροιστή διάδοσης κρατουμένου, με τη διαφορά ότι το κρατούμενο εξόδου (Cout) κάθε πλήρους αθροιστή δεν είναι είσοδος σε κάποιον άλλο, αλλά αποτελεί αποτέλεσμα στην συγκεκριμένη βαθμίδα, μαζί με το αντίστοιχο Save ψηφίο.

Παρακάτω φαίνεται ένα κύκλωμα αθροιστή σωσίματος κρατουμένου 8-bit, το οποίο μπορεί να δεχτεί σαν είσοδο είτε τρεις δυαδικούς αριθμούς είτε ένα δυαδικό και έναν Carry-Save αριθμό. Η έξοδός του είναι ένας Carry-Save αριθμός που αποτελείται από τα ψηφία

$C = \{C'8, C'7, C'6, C'5, C'4, C'3, C'2, C'1, C'0\}$ και $S = \{S7, S6, S5, S4, S3, S2, S1, S0\}$.



Σχήμα 2.5 Κύκλωμα αθροιστή σωσίματος κρατουμένου

Το παραπάνω κύκλωμα μπορεί να λειτουργήσει και για δυαδικούς αριθμούς σε μορφή συμπληρώματος του δύο, αν αντιστραφούν τα ψηφία αρνητικής αξίας {A7, B7, C7, S7, C'8}.

Η παραπάνω δομή εξηγεί και την ονομασία του πλήρους αθροιστή ως 3-2 συμπιεστή. Ο αθροιστής σωσίματος κρατουμένου έχει απλή δομή και την ίδια επιφάνεια με τον αθροιστή διάδοσης κρατουμένου. Επίσης μπορεί να χειριστεί τρεις αντί για δύο δυαδικούς αριθμούς ταυτόχρονα. Το μεγάλο πλεονέκτημά του όμως είναι ότι η

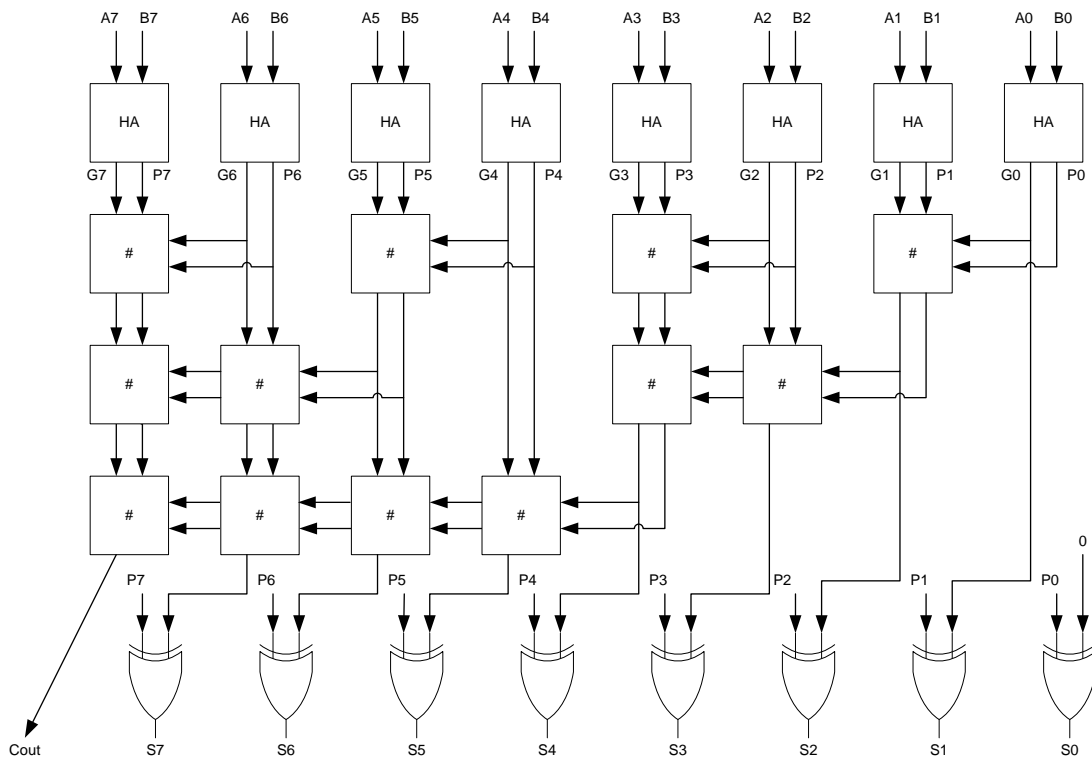
καθυστέρηση (critical path) ενός N-bit CSA αθροιστή είναι μόνο 1 επίπεδο πλήρους αθροιστή (FA) ενώ ενός N-bit CPA αθροιστή είναι N επίπεδα. Λόγω αυτών, οι αθροιστές-συμπιεστές που χρησιμοποιούνται από κάθε μονάδα της προτεινόμενης αρχιτεκτονικής είναι CS adders. Το βασικό μειονέκτημα του είναι ότι το αποτέλεσμα δεν βρίσκεται σε δυαδική μορφή, οπότε θα χρειαστεί περαιτέρω επεξεργασία.

2.3.2.3 Αθροιστής πρόβλεψης κρατουμένου (Carry Lookahead Adder)

Ο αθροιστής πρόβλεψης κρατουμένου (Carry Lookahead Adder, CLA) ακολουθεί μια διαφορετική φιλοσοφία άθροισης ώστε να αποφύγει την καθυστέρηση από την διάδοση κρατουμένου. Χρησιμοποιεί κυκλώματα ημιαθροιστών για την δημιουργία των σημάτων P και G, κυκλώματα πρόβλεψης κρατουμένου (#) για την πρόβλεψη του τελικού κρατουμένου και ένα τελευταίο στάδιο που είναι η γεννήτρια του τελικού αθροίσματος, η οποία στην απλοποιημένη περίπτωση που δεν έχουμε κρατούμενο εισόδου, αποτελείται μόνο από N πύλες αποκλειστικού-ή (XOR), όπου N το πλήθος των ψηφίων του αθροιστή.

Το κύκλωμα δημιουργίας κρατουμένου, εκμεταλλεύεται την προσεταιριστικότητα του τελεστή # για να υπολογίσει παράλληλα τα τελικά κρατούμενα. Αυτό επιτυγχάνεται με την διάταξη των κυκλωμάτων πρόβλεψης κρατουμένου σε μια δενδρική δομή, και επομένως την παραγωγή των τελικών αποτελεσμάτων με καθυστέρηση $\log N$ κυκλωμάτων #.

Παρακάτω φαίνεται ένα κύκλωμα αθροιστή πρόβλεψης κρατουμένου 8-bit χωρίς κρατούμενο εισόδου, το οποίο μπορεί να δεχτεί σαν είσοδο είτε δυο δυαδικούς αριθμούς είτε έναν Carry-Save αριθμό. Η έξοδος του είναι ένας δυαδικός αριθμός που αποτελείται από τα ψηφία {Cout,S7,S6,S5,S4,S3,S2,S1,S0}.



Σχήμα 2.6 Κύκλωμα αθροιστή πρόβλεψης κρατουμένου

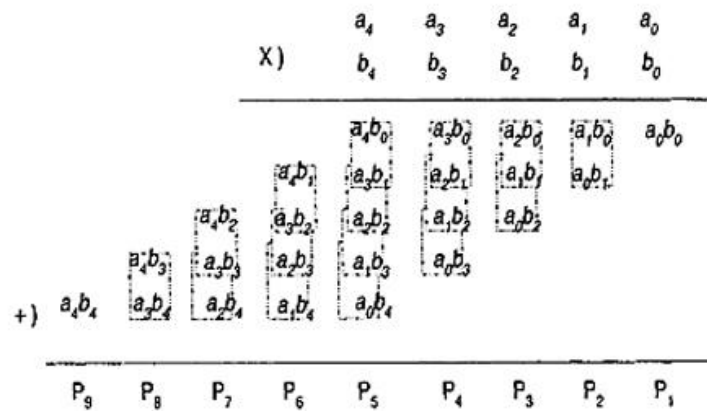
Το παραπάνω κύκλωμα μπορεί να λειτουργήσει και για δυαδικούς αριθμούς σε μορφή συμπληρώματος του δύο, αν αντιστραφούν τα ψηφία αρνητικής αξίας $\{A_7, B_7, \text{Cout}\}$.

Ο αθροιστής πρόβλεψης κρατουμένου έχει πιο πολύπλοκη δομή, αλλά είναι πιο γρήγορος από τον αθροιστή διάδοσης κρατουμένου, καθώς αποφεύγει την διάδοση κρατουμένου και παράγει τα αποτελέσματά του με καθυστέρηση ενός HA συν $\log N$ κυκλωμάτων # συν μιας πύλης XOR. Το μειονέκτημα του αθροιστή πρόβλεψης κρατουμένου είναι η επιφάνειά του, η οποία αυξάνει πολύ σε μεγάλα μήκη λέξης. Συγκεκριμένα, ο αριθμός των κυκλωμάτων πρόβλεψης κρατουμένου (#) που απαιτούνται είναι $N \cdot \log N / 2$, όπου N το πλήθος των ψηφίων του αθροιστή. Σε γενικές γραμμές ωστόσο, το κέρδος σε ταχύτητα είναι πιο σημαντικό από την επιβάρυνση σε επιφάνεια, για αυτό και χρησιμοποιείται από τις μονάδες της προτεινόμενης αρχιτεκτονικής για τη 'γρήγορη' τελική μετατροπή των CS αριθμών σε binary μορφή.

2.3.3 Πολλαπλασιαστές

Η πράξη του πολλαπλασιασμού στο δυαδικό σύστημα εκτελείται ακριβώς όμοια με το δεκαδικό. Επειδή όμως στο δυαδικό σύστημα έχουμε μόνο δυο ψηφία $\{0,1\}$, ο πολλαπλασιασμός ενός N -bit αριθμού μπορεί να έχει μόνο δύο δυνατά αποτελέσματα, είτε τον αριθμό αυτούσιο, είτε ένα μηδενικό μερικό γινόμενο. Ως εκ τούτου, η παραγωγή των μερικών γινομένων γίνεται με πύλες AND.

Ο πολλαπλασιασμός δύο αριθμών A και B των πέντε bit φαίνεται στο παρακάτω σχήμα:

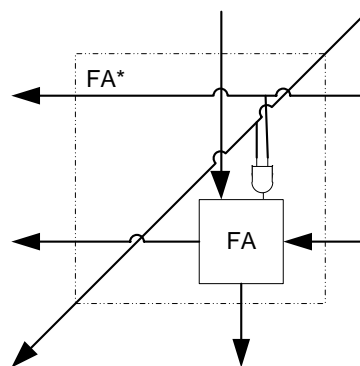


Σχήμα 2.7 Πράξεις πολλαπλασιασμού

2.3.3.1 Παράλληλοι πολλαπλασιαστές τύπου πίνακα (Array Multipliers)

Η παραπάνω διαδικασία υποδεικνύει και μια βασική τοπολογία κυκλώματος πολλαπλασιασμού. Συγκεκριμένα, παράγονται παράλληλα N μερικά γινόμενα, αποτελούμενα από N-bits το καθένα, και αθροίζονται τα ψηφία ίσου βάρους, κάνοντας χρήση ενός δικτύου N^2 πλήρων αθροιστών σε μορφή πίνακα.

Θα παρουσιάσουμε δύο βασικές τοπολογίες αυτής της οικογενείας, τον παράλληλο πολλαπλασιαστή με διάδοση κρατουμένου και τον παράλληλο πολλαπλασιαστή με σώσιμο κρατουμένου. Και οι δύο τοπολογίες κάνουν χρήση της ίδιας βασικής δομικής μονάδας που αποτελείται, όπως προαναφέραμε, από μια πύλη AND, για την παραγωγή ενός ψηφίου ενός μερικού γινομένου, και έναν πλήρη αθροιστή. Η δομική αυτή μονάδα, παρουσιάζεται αμέσως παρακάτω.

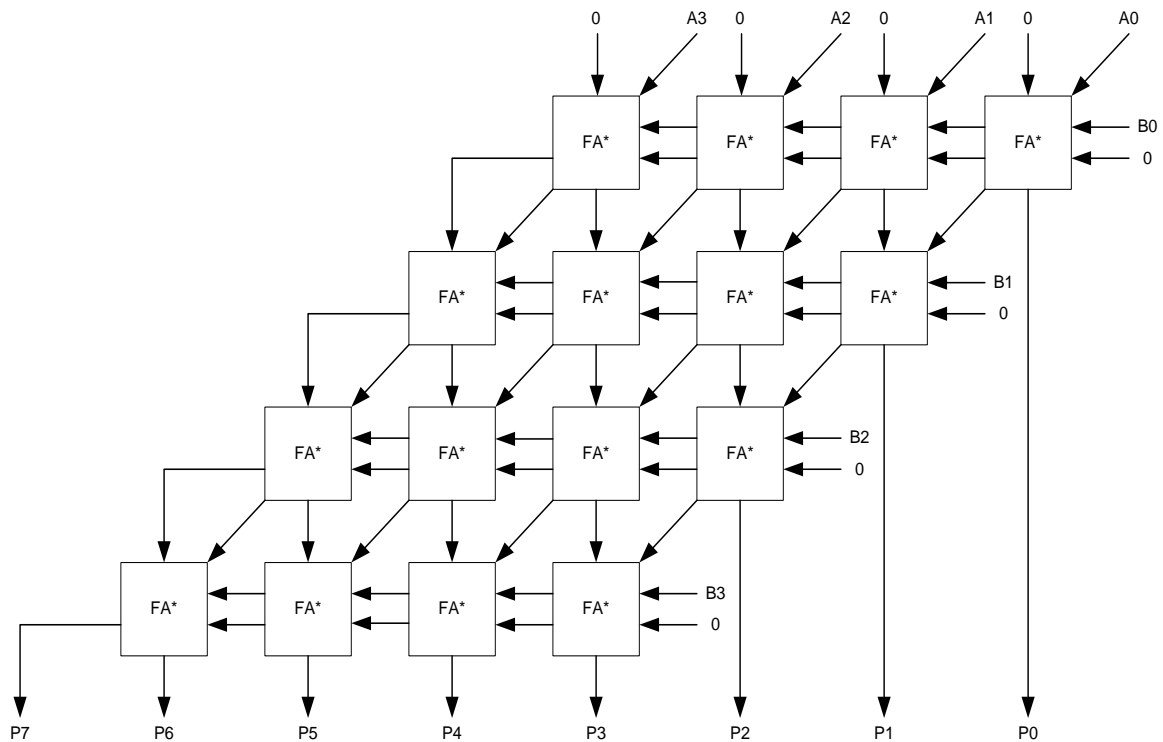


Σχήμα 2.8 Κύκλωμα FA*

Βασικό πλεονέκτημα και των δύο αυτών σχημάτων είναι πως η δομή των πολλαπλασιαστών παραμένει κανονική και επίσης είναι αρκετά εύκολη η μετατροπή τους σε συνεχούς διοχέτευσης και συστολικούς.

A. Παράλληλος πολλαπλασιαστής με διάδοση κρατουμένου

Ο παράλληλος πολλαπλασιαστής με διάδοση κρατουμένου, αποτελείται από αθροιστές διάδοσης κρατουμένου συνδεδεμένους σε σειρά (οι έξοδοι του ενός δηλαδή είναι είσοδοι του επομένου) αλλά ο κάθε ένας από αυτούς είναι ολισθημένος κατά μία θέση αριστερά, ως προς τον προηγούμενο. Αμέσως παρακάτω παρουσιάζεται ένας παράλληλος πολλαπλασιαστής με διάδοση κρατουμένου, τεσσάρων bit.

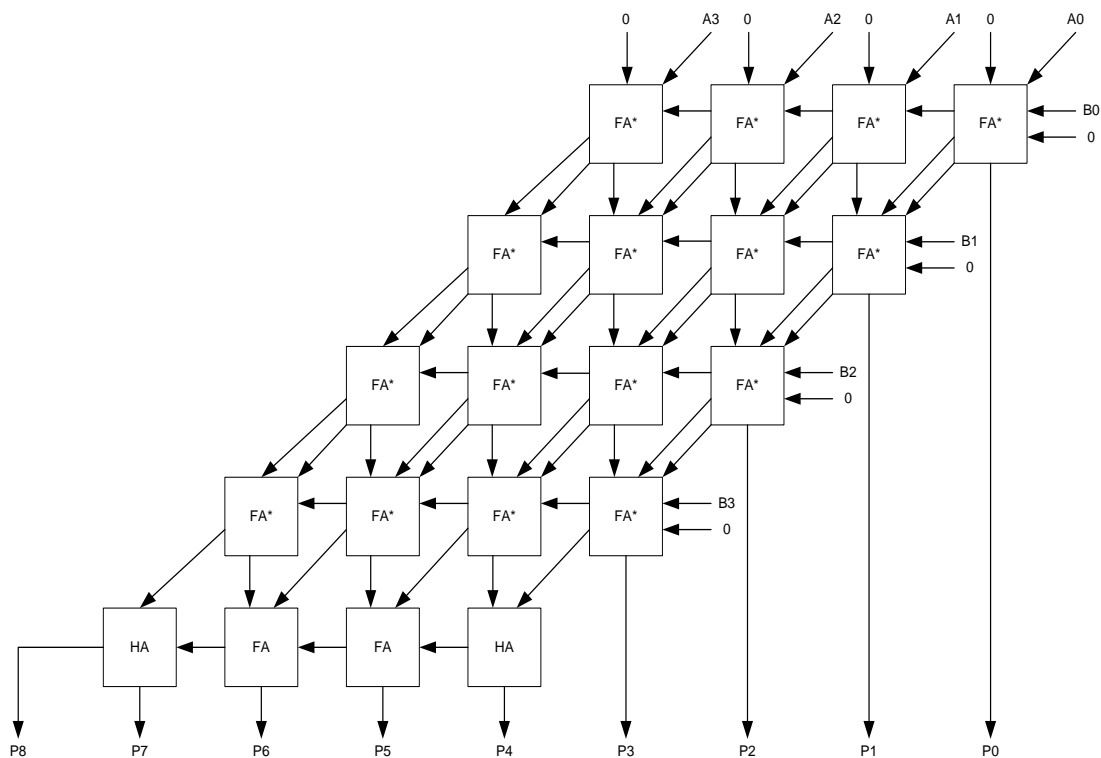


Σχήμα 2.9 Κύκλωμα παράλληλου πολλαπλασιαστή με διάδοση κρατουμένου

Ο παράλληλος πολλαπλασιαστής με διάδοση κρατουμένου τεσσάρων bit παράγει τα αποτελέσματά του με καθυστέρηση 10 FA*, όπως φαίνεται και από το παραπάνω σχήμα. Στην γενική περίπτωση, η επιφάνεια του πολλαπλασιαστή με διάδοση κρατουμένου είναι N^2 FA* και η καθυστέρησή του είναι $3N-2$ FA*, όπου N το μήκος λέξης των όρων A και B .

B. Παράλληλος πολλαπλασιαστής με σώσιμο κρατουμένου

Ο παράλληλος πολλαπλασιαστής με σώσιμο κρατουμένου, αποτελείται από αθροιστές σωσίματος κρατουμένου συνδεδεμένους σε σειρά (οι έξοδοι του ενός δηλαδή είναι είσοδοι του επομένου) αλλά ο κάθε ένας από αυτούς είναι ολισθημένος κατά μία θέση αριστερά, ως προς τον προηγούμενο. Επειδή στο τελευταίο στάδιο της άθροισης το πιο σημαντικό τμήμα του αποτελέσματος είναι σε μορφή Carry-Save, πρέπει να ενσωματώσουμε έναν επιπλέον αθροιστή διάδοσης κρατουμένου για να έχουμε το σωστό δυαδικό αποτέλεσμα. Αμέσως παρακάτω παρουσιάζεται ένας παράλληλος πολλαπλασιαστής με σώσιμο κρατουμένου, τεσσάρων bit.



Σχήμα 2.10 Κύκλωμα παράλληλου πολλαπλασιαστή με σώσιμο κρατουμένου

Ο παράλληλος πολλαπλασιαστής με σώσιμο κρατουμένου τεσσάρων bit παράγει τα αποτελέσματά του με καθυστέρηση 4 FA* συν 2 FA συν 2 HA, όπως φαίνεται και από το παραπάνω σχήμα. Στην γενική περίπτωση, η επιφάνεια του πολλαπλασιαστή με σώσιμο κρατουμένου είναι N^2 FA* συν 2 HA συν $N-2$ FA και η καθυστέρησή του είναι N FA* συν 2 HA συν $N-2$ FA, όπου N το μήκος λέξης των όρων A και B .

Ο παράλληλος πολλαπλασιαστής με σώσιμο κρατουμένου επομένως, καταλαμβάνει ελαφρώς περισσότερη επιφάνεια από τον παράλληλο πολλαπλασιαστή με διάδοση κρατουμένου, όμως είναι σημαντικά ταχύτερος.

2.3.3.2 Δενδρικοί πολλαπλασιαστές (Tree Multipliers)

Οι δενδρικοί πολλαπλασιαστές χρησιμοποιούν μια διαφορετική φιλοσοφία άθροισης, στοχεύοντας στην παραγωγή κυκλωμάτων πολλαπλασιασμού με την ελάχιστη καθυστέρηση. Η δημιουργία των μερικών γινομένων στους δενδρικούς πολλαπλασιαστές γίνεται ανεξάρτητα από το στάδιο της πρόσθεσης. Αφού παραχθούν όλα τα μερικά γινόμενα, οδηγούνται σε ένα δίκτυο FA και HA, από το οποίο παράγεται ένας αριθμός σε μορφή σωσίματος - κρατουμένου. Στο τελικό στάδιο, μπορεί να χρησιμοποιηθεί ένας αθροιστής διάδοσης κρατουμένου ή πρόβλεψης κρατουμένου για να παραχθεί το τελικό αποτέλεσμα σε δυαδική μορφή.

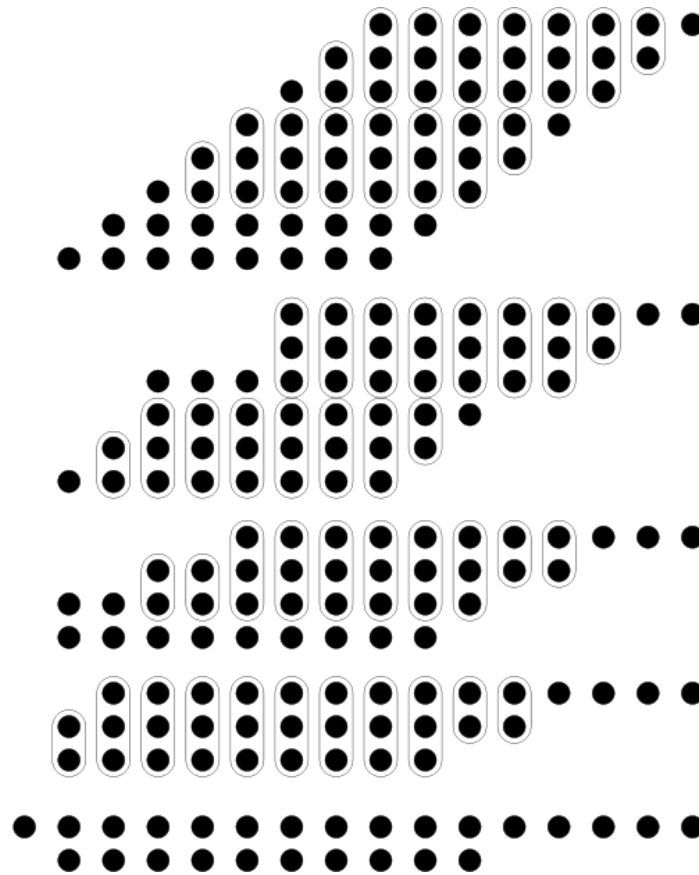
Θα παρουσιάσουμε δύο τύπους δενδρικών πολλαπλασιαστών, που χρησιμοποιούν διαφορετικό αλγόριθμο για την δημιουργία του δέντρου συμπίεσης, τους πολλαπλασιαστές Wallace και τους πολλαπλασιαστές Dadda.

Επειδή η δημιουργία των μερικών γινομένων γίνεται ανεξάρτητα από το στάδιο της δενδρικής συμπίεσης, είναι δυνατή η κωδικοποίηση του ενός από τους δύο αριθμούς ή και των δύο με βάση κάποιο άλλο σύστημα. Με αυτόν τον τρόπο μπορεί να μειωθεί ο αριθμός των μερικών γινομένων έτσι ώστε να αυξηθεί η ταχύτητα του δενδρικού πολλαπλασιαστή.

A. Δενδρικός πολλαπλασιαστής Wallace

Στόχος του δενδρικού συμπίεστη Wallace είναι να γίνει η συμπίεση των μερικών γινομένων μετά από όσο το δυνατόν λιγότερα επίπεδα FA και HA και συνεπώς σε όσο το δυνατόν μικρότερο χρονικό διάστημα. Για το σκοπό αυτό, σε κάθε επίπεδο, τα ψηφία ίδιου βάρους ομαδοποιούνται ανά τρία και εισέρχονται σαν είσοδοι σε έναν FA, εάν περισσέψουν δύο, τότε ομαδοποιούνται ανά δύο και εισέρχονται σαν είσοδοι σε έναν HA, ενώ εάν περισσέψει μόνο ένα, μεταφέρεται στο επόμενο επίπεδο.

Αμέσως παρακάτω παρουσιάζεται το τμήμα συμπίεσης των μερικών γινομένων ενός 8x8 bit πολλαπλασιαστή Wallace. Κάθε κουκκίδα συμβολίζει ένα bit του αντίστοιχου μερικού γινομένου. Όπου υπάρχει ομαδοποίηση τριών bit, τα τρία αυτά bit εισέρχονται σαν είσοδοι σε έναν FA και όπου υπάρχει ομαδοποίηση δύο bit, τα δύο αυτά bit εισέρχονται σαν είσοδοι σε έναν HA. Προφανώς, κάθε FA και HA παράγουν ως αποτέλεσμα ένα bit ίδιου βάρους (Save) και ένα bit με το αμέσως μεγαλύτερο βάρος (Carry).

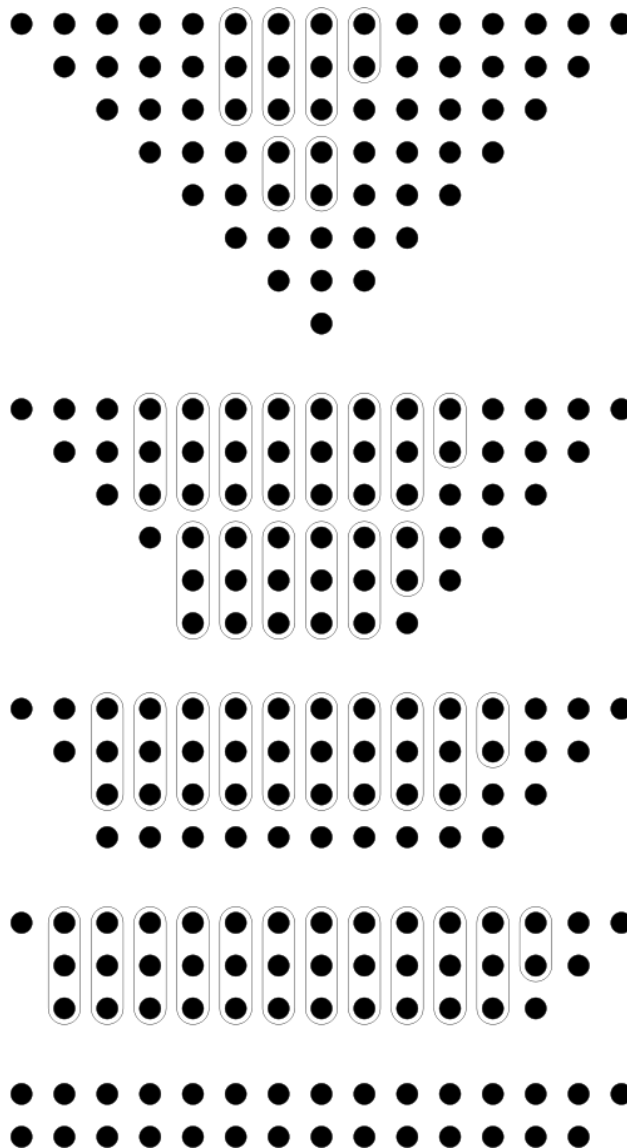


Σχήμα 2.12 Wallace δενδρικός συμπεστής

B. Δενδρικός πολλαπλασιαστής Dadda

Ο δενδρικός συμπεστής Dadda ακολουθεί μία ελαφρώς διαφορετική προσέγγιση στην συμπίεση των bit των μερικών γινομένων. Στόχος του είναι να διατηρήσει σε κάθε βήμα τα bits που έχουν το ίδιο βάρος όσο το δυνατόν πλησιέστερα σε ένα πολλαπλάσιο του τρία, ώστε να χρησιμοποιήσει τους ελάχιστους δυνατούς FA και HA επιτυγχάνοντας παράλληλα την ίδια μείωση του ύψους του δέντρου των μερικών γινομένων με τον Wallace. Ως εκ τούτου, επιτυγχάνει μικρότερη επιφάνεια, καθώς χρησιμοποιεί πιο αποδοτικά την 3:2 συμπίεση που προσφέρει ο FA, έναντι της 2:2 που προσφέρει ο HA. Επιπλέον, νεότερες μελέτες έδειξαν ότι επιτυγχάνει και μικρότερη καθυστέρηση έναντι του δενδρικού συμπεστή Wallace.

Αμέσως παρακάτω παρουσιάζεται το τμήμα συμπίεσης των μερικών γινομένων ενός 8x8 bit πολλαπλασιαστή Dadda. Κάθε κουκκίδα συμβολίζει ένα bit του αντίστοιχου μερικού γινομένου. Όπου υπάρχει ομαδοποίηση τριών bit, τα τρία αυτά bit εισέρχονται σαν είσοδοι σε έναν FA και όπου υπάρχει ομαδοποίηση δύο bit, τα δύο αυτά bit εισέρχονται σαν είσοδοι σε έναν HA. Προφανώς, κάθε FA και HA παράγουν ως αποτέλεσμα ένα bit ίδιου βάρους (Save) και ένα bit με το αμέσως μεγαλύτερο βάρος (Carry).



Σχήμα 2.13 Dadda δενδρικός συμπιεστής

Συνοψίζοντας, έχοντας δει τις βασικές μορφές πολλαπλασιαστών, συμπεραίνουμε ότι οι δενδρικοί πολλαπλασιαστές έχουν τα καλύτερα αποτελέσματα στον τομέα της ταχύτητας καθώς έχουν λιγότερα επίπεδα FA, οι πολλαπλασιαστές τύπου πίνακα προσφέρουν μια κανονική δομή, η οποία μπορεί να οδηγήσει σε συστολικούς πολλαπλασιαστές συνεχούς διοχέτευσης (pipelining), οι οποίοι έχουν σημαντικά μικρότερη επιφάνεια και μπορούν να λειτουργήσουν σε μεγαλύτερες συχνότητες, λόγω του μικρότερου κρίσιμου μονοπατιού (critical path). Για την προτεινόμενη αρχιτεκτονική ένας array πολλαπλασιαστής θα ήταν πολύ αργός, ενώ και η δυνατότητα που προσφέρει για pipeline δεν είναι πρακτική στη σχεδίασή μας. Επιλέχθηκε λοιπόν ένας Wallace-based δενδρικός πολλαπλασιαστής καθώς το αποτέλεσμα των πολλαπλασιασμών που εκτελεί κάθε μονάδα περικόπτεται και κατά τη συμπίεση των μερικών γινομένων πρέπει να συμπιεστούν όσο το δυνατόν περισσότερα bits από τις χαμηλότερες βαθμίδες για μεγαλύτερη ακρίβεια στο

αποτέλεσμα. Ένας Dadda πολλαπλασιαστής θα καταλάμβανε ελαφρώς μικρότερη επιφάνεια, αλλά θα αγνοούσε πολλά από τα bits των χαμηλότερων βαθμίδων οδηγώντας σε μικρότερη ακρίβεια στο αποτέλεσμα.

2.4 Αλγόριθμοι Ψηφιακής Επεξεργασίας Σήματος

Στην παράγραφο αυτή παρουσιάζονται τα κυριότερα χαρακτηριστικά των αλγορίθμων ψηφιακής επεξεργασίας σήματος (Digital signal processing algorithms, DSP). Η παρουσίαση αυτή γίνεται διότι η προτεινόμενη αρχιτεκτονική στοχεύει πρωτίστως στην αποδοτική απεικόνιση αυτής της οικογένειας αλγορίθμων (DSP specific). Η εξοικείωση με τα χαρακτηριστικά των DSP αλγορίθμων σε αυτό το σημείο θα συνεισφέρει στην κατανόηση των στόχων της προτεινόμενης αρχιτεκτονικής στα επόμενα κεφάλαια.

Η ψηφιακή επεξεργασία σήματος έχει να κάνει με την αναπαράσταση των σημάτων από μια ακολουθία ψηφίων και εν συνεχεία την επεξεργασία τους. Συχνά, τα σήματα που θέλουμε να επεξεργαστούμε βρίσκονται σε αναλογική μορφή. Τα σήματα αυτά μετατρέπονται σε ψηφιακά μέσω ειδικών κυκλωμάτων που ονομάζονται μετατροπείς αναλογικού σε ψηφιακό (analog to digital converters). Στη συνέχεια, εφαρμόζονται οι DSP αλγόριθμοι, έτσι ώστε τα σήματα αυτά να υποστούν την επιθυμητή επεξεργασία πριν οδηγηθούν στην έξοδο. Συνήθως, η έξοδος πρέπει να είναι κι αυτή σε αναλογική μορφή και για αυτό το σκοπό χρησιμοποιούνται μετατροπείς ψηφιακού σε αναλογικό (digital to analog converters).

Υπάρχουν πολλά είδη και κατηγορίες DSP αλγορίθμων. Τα κυριότερα πεδία εφαρμογών τους είναι τα εξής:

- Επεξεργασία ήχου και ομιλίας
- Συστήματα Ραντάρ
- Αισθητήρες
- Επεξεργασία στοχαστικών σημάτων
- Ψηφιακή επεξεργασία εικόνας
- Επεξεργασία σημάτων στις Τηλεπικοινωνίες
- Επεξεργασία ιατρικών σημάτων

Οι DSP αλγόριθμοι εκτελούνταν αρχικά σε κοινούς επεξεργαστές. Αργότερα κατασκευάστηκαν εξειδικευμένοι επεξεργαστές (DSP processors) και εξειδικευμένο υλικό (ASICs) για την εκτέλεσή τους. Σήμερα υπάρχουν επιπρόσθετες τεχνολογίες για την εκτέλεση των αλγορίθμων αυτών, όπως είναι για παράδειγμα οι εξειδικευμένοι επιταχυντές υλικού.

2.4.1 Χαρακτηριστικά των DSP αλγορίθμων

Οι DSP αλγόριθμοι έχουν ορισμένα χαρακτηριστικά τα οποία τους διαχωρίζουν από τις υπόλοιπες ψηφιακές εφαρμογές:

1. Στην πλειονότητά τους είναι υπολογιστικά έντονες εφαρμογές (computation-intensive applications). Σε αυτούς τους αλγόριθμους εκτελείται μεγάλος αριθμός πράξεων πάνω σε μικρό αριθμό δεδομένων.
2. Η ταχύτητα της εκτέλεσης τους εξαρτάται από τον αριθμό των διαθέσιμων υπολογιστικών πόρων της αρχιτεκτονικής (resource dominated applications).
3. Το συντριπτικά μεγαλύτερο ποσοστό των πράξεων που εκτελούνται σε αυτούς αποτελείται από προσθαφαιρέσεις ακολουθούμενες από πολλαπλασιασμούς ή το αντίστροφο.
4. Δεν έχουν μεγάλες απαιτήσεις όσον αφορά τη σχολαστική ακρίβεια των αποτελεσμάτων καθώς η φύση των αλγορίθμων αυτών (λόγω του ότι τα σήματα που επεξεργάζονται έχουν μετατραπεί από αναλογικά σε ψηφιακά και υφίστανται κβαντισμό) είναι τέτοια ώστε δεν είναι δυνατή η αναπαράσταση των εισόδων τους με απόλυτη ακρίβεια.

2.4.2 Επιδράσεις στην προτεινόμενη αρχιτεκτονική

Η προτεινόμενη αρχιτεκτονική σχεδιάστηκε λαμβάνοντας υπόψη τα ιδιαίτερα χαρακτηριστικά των DSP αλγορίθμων και τη συμπεριφορά της FCU based αρχιτεκτονικής στην απεικόνιση των αλγορίθμων αυτών.

Η βασική μονάδα της προτεινόμενης αρχιτεκτονικής εκτελεί συνεπυγμένα την πράξη της διπλής πρόσθεσης-διπλού πολλαπλασιασμού-πρόσθεσης, συγκεκριμένα $A_1 \times (X_1 \pm Y_1) + A_2 \times (X_2 \pm Y_2) + B_1 + B_2$. Η μονάδα UDMA σχεδιάστηκε έτσι ώστε οι λειτουργίες που μπορεί να εκτελέσει σε κάθε κύκλο να αντιστοιχούν σε ακολουθίες πράξεων που συναντώνται πολύ συχνά στους DSP αλγόριθμους. Έτσι, μπορεί να εκτελέσει τον ίδιο αριθμό πράξεων σε πολύ λιγότερους κύκλους σε σχέση με μια κοινή αρχιτεκτονική που βασίζεται σε βασικά δομικά στοιχεία (αθροιστές και πολλαπλασιαστές) ή ακόμα και με μια πιο σύνθετη που υλοποιεί όμως μόνο έναν πολλαπλασιασμό όπως συμβαίνει στην FCU based αρχιτεκτονική.

Για παράδειγμα, ας θεωρήσουμε την εξής ακολουθία πράξεων:

$$R_1 = A + B$$

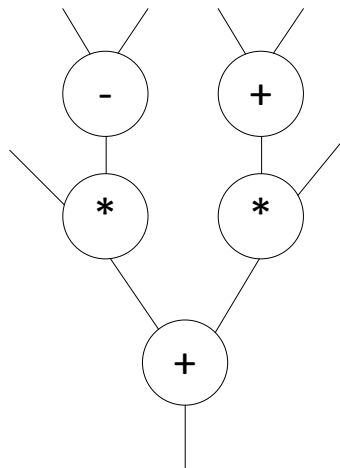
$$R_2 = C - D$$

$$R_3 = R_1 \cdot E$$

$$R_4 = R_2 \cdot F$$

$$R_5 = R_3 + R_4$$

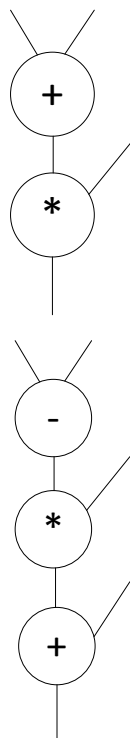
Ο γράφος ροής δεδομένων για αυτή την ακολουθία φαίνεται στο σχήμα 2.14:



Σχήμα 2.14 Παράδειγμα Γράφου Ροής Δεδομένων

Μια συμβατική αρχιτεκτονική, αποτελούμενη από βασικά επεξεργαστικά στοιχεία, θα χρειαζόταν τρεις κύκλους για την εκτέλεση αυτής της ακολουθίας. Ένα κύκλο για τον υπολογισμό των R_1, R_2 ένα κύκλο για τα R_3, R_4 και τέλος ένα κύκλο για τον υπολογισμό του R_5 .

Μια αρχιτεκτονική βασισμένη στην μονάδα FCU, θα χρειαζόταν δυο κύκλους. Ένα κύκλο για τους υπολογισμούς των R_3 , και/η R_4 και ένα για τον υπολογισμό του R_5 , όπως φαίνεται στα παρακάτω FCU templates:



Σχήμα 2.15 Παράδειγμα Γράφου Ροής Δεδομένων

Μια μονάδα UDMA όμως μπορεί να εκτελέσει τις παραπάνω πράξεις σε μόνο ένα κύκλο λειτουργίας. Η εσωτερική του αρχιτεκτονική επιτρέπει την ακολουθιακή εκτέλεση τριών 'ζευγών' πράξεων, 2 προσθαιρέσεων, 2 πολλαπλασιασμών και 2 προσθέσεων ανά κύκλο λειτουργίας. Επειδή αυτές οι ακολουθίες, κυρίως ο σχηματισμός πολλών γινομένων και οι συνεχείς προσθέσεις τους (sum of products) συναντώνται πολύ συχνά στους DSP αλγόριθμους, στοχεύουμε στο σχεδιασμό μιας αρχιτεκτονικής με πολύ καλύτερες επιδόσεις τόσο σε σχέση με μια συμβατική αρχιτεκτονική όσο και συγκεκριμένα με την FCU based αρχιτεκτονική.

Τέλος, ένα άλλο χαρακτηριστικό το οποίο καθιστά την μονάδα UDMA κατάλληλη για DSP αλγόριθμους είναι το γεγονός ότι η αποκοπή στα αποτελέσματα της εξόδου της μονάδας (έτσι ώστε οι εξόδοι να έχουν ίδιο μήκος λέξης με τις εισόδους) και η απώλεια ακρίβειας που συνεπάγεται δεν αποτελεί σημαντικό πρόβλημα, αφού δεν υπάρχει απαίτηση για σχολαστική ακρίβεια από τις εφαρμογές. Εξάλλου η απώλεια ακρίβειας της μονάδας είναι πολύ μικρή αφού έχει προβλεφθεί κυκλωματικά τόσο η στρογγυλοποίηση όσο και ο συνυπολογισμός μιας εκτίμησης του μέρους που αποκόπτεται στους πολλαπλασιασμούς.

3

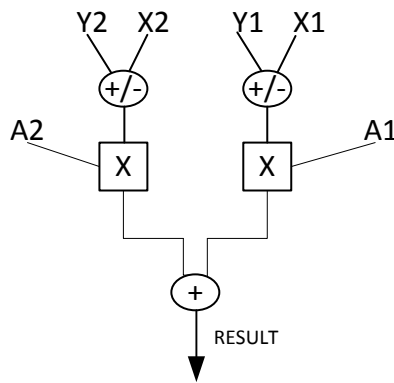
UDMA ΚΑΙ UDMA ΜΕ PIPELINE: ΟΙ 2 ΒΑΣΙΚΕΣ ΕΠΕΞΕΡΓΑΣΤΙΚΕΣ ΜΟΝΑΔΕΣ ΤΩΝ ΠΡΟΤΕΙΝΟΜΕΝΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ

3.1 Προεπισκόπηση

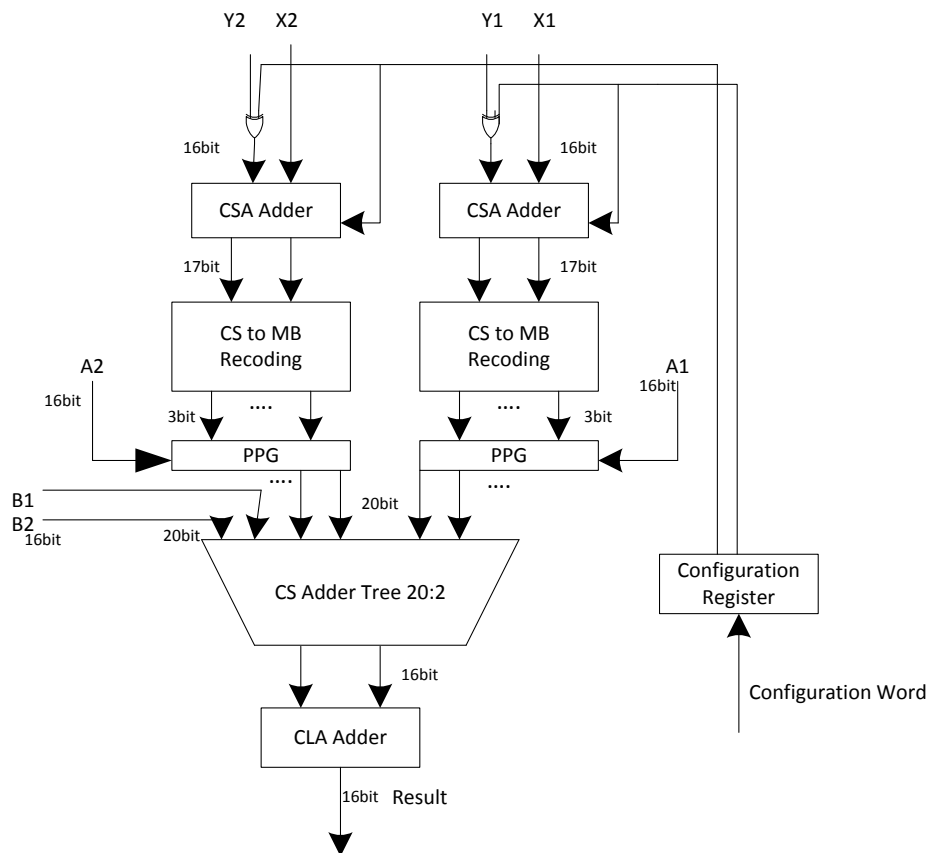
Στο κεφάλαιο αυτό αναλύονται διεξοδικά οι δύο βασικές επεξεργαστικές μονάδες των προτεινόμενων αρχιτεκτονικών: ο ενοποιημένος διπλός πολλαπλασιαστής-αθροιστής ή UDMA και ο ενοποιημένος διπλός πολλαπλασιαστής-αθροιστής με διοχέτευση ή UDMA pipeline. Αρχικά γίνεται συνοπτική παρουσίαση της δομής της μονάδας UDMA και των διασυνδέσεων μεταξύ των επιμέρους τμημάτων, και στη συνέχεια αναλύονται διεξοδικά τα επιμέρους δομικά στοιχεία της κατά σειρά, από το υψηλότερο έως το τελικό επίπεδο. Ακολουθεί μια συνοπτική παρουσίαση και ανάλυση για την μονάδα που χρησιμοποιεί pipeline, όπου τονίζονται κυρίως οι διαφορές σε σχέση με την πρώτη μονάδα. Μετά από την επεξήγηση των σχεδιαστικών επιλογών για τις δυο μονάδες πραγματοποιείται μια θεωρητική ανάλυση της καθυστέρησης απόκρισης των δυο μονάδων. Τέλος, γίνεται η παρουσίαση του συνόλου των λειτουργιών (ακολουθιών πράξεων) που μπορούν να εκτελέσουν σε ένα κύκλο λειτουργίας.

3.2 Δομή της μονάδας UDMA

Η δομή της προτεινόμενης μονάδας UDMA φαίνεται στα σχήματα 3.1 και 3.2. Είναι μια επεξεργαστική μονάδα που αποτελείται από μικρότερες υπομονάδες οι οποίες έχουν σχεδιασθεί και διαταχθεί έτσι ώστε να προσδίδουν στην UDMA την επιθυμητή λειτουργικότητα. Η δομή της καθεμιάς παρατίθεται αναλυτικά στην παράγραφο 3.3.



Σχήμα 3.1 Η μονάδα UDMA: Abstract view



Σχήμα 3.2 Η μονάδα UDMA: Detailed view

Γίνεται αρχικά παρουσίαση των εισόδων και εξόδων της μονάδας που είναι πάντοτε 16-bit δυαδικοί αριθμοί σε μορφή συμπληρώματος ως προς δύο, ώστε να γίνεται πιο εύκολα κατανοητή η δομική ανάλυση των επιμέρους μονάδων, δια μέσου της ροής δεδομένων από τις εισόδους προς την έξοδο.

3.2.1 Είσοδοι-Έξοδοι της UDMA

Η αποδοτικότερη δυνατή σχεδίαση του ενοποιημένου διπλού πολλαπλασιαστή-αθροιστή, βάσει των DSP αλγορίθμων για τους οποίους θα χρησιμοποιηθεί, απαιτεί συνολικά 8 εισόδους και μια έξοδο. Συγκεκριμένα, υπάρχουν:

1. Μια είσοδος X_1 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2. Αποτελεί τον πρώτο από τους 2 όρους που αθροίζονται πριν εισέλθουν στον πρώτο πολλαπλασιαστή όπως θα δούμε στη συνέχεια.
2. Μια είσοδος Y_1 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2. Μαζί με την X_1 συνθέτουν το ζευγάρι που εισέρχεται στην αθροιστική είσοδο του πρώτου πολλαπλασιαστή.
3. Μια είσοδος A_1 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2. Αποτελεί τον δεύτερο όρο που συμμετέχει στον πρώτο πολλαπλασιασμό.
4. Μια είσοδος X_2 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2. Αποτελεί τον πρώτο από τους 2 όρους που αθροίζονται πριν εισέλθουν στον δεύτερο πολλαπλασιαστή όπως θα δούμε στη συνέχεια.
5. Μια είσοδος Y_2 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2. Μαζί με την X_2 συνθέτουν το ζευγάρι που εισέρχεται στην αθροιστική είσοδο του δεύτερου πολλαπλασιαστή.
6. Μια είσοδος A_2 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2. Αποτελεί τον δεύτερο όρο που συμμετέχει στον δεύτερο πολλαπλασιασμό.
7. Μια είσοδος B_1 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2, και προέρχεται από το register file.
8. Μια είσοδος B_2 που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2, και προέρχεται από το register file.
9. Μια είσοδος CL (CL_0, CL_1) των 2 bit. Κάθε ένα από τα bit αυτά αποτελεί σήμα ελέγχου για τις αθροιστικές εισόδους των πολλαπλασιαστών και καθορίζουν ποια πράξη θα εκτελεστεί σε αυτές.

Η μονάδα UDMA έχει μια μοναδική έξοδο result που είναι ένας 16-bit binary αριθμός σε μορφή συμπληρώματος ως προς 2.

3.2.2 Παρουσίαση των δομικών στοιχείων της UDMA – Διάταξη και διασύνδεση

Στην παράγραφο αυτή παρουσιάζονται τα δομικά στοιχεία που απαρτίζουν την μονάδα UDMA. Για κάθε στοιχείο αναφέρονται οι εισοδοί που δέχεται και οι έξοδοι που παράγει, προκειμένου να δοθεί μια ξεκάθαρη εικόνα της διάταξης των στοιχείων αυτών μέσα στη βασική επεξεργαστική μονάδα. Η παρουσίαση των στοιχείων ακολουθεί την ροή των δεδομένων από τις εισόδους ως την έξοδο της μονάδας. Η τελική δομή της UDMA δίνεται στο σχήμα 3.1. Η αναλυτική επεξήγηση της δομής των στοιχείων και του τρόπου με τον οποίο επεξεργάζονται τα δεδομένα των εισόδων τους γίνεται στην επόμενη παράγραφο.

S-MB1 recoder USUM2MB1

Ο S-MB1 recoder USUM2MB1 δέχεται τις 16-bit εισόδους της μονάδας UDMA X_1 και Y_1 καθώς και το bit ελέγχου CL_0 . Η έξοδος του είναι οι 8 τριάδες δυαδικών αριθμών *one*, *two*, *sign*, οι οποίες αντιπροσωπεύουν την κωδικοποίηση Modified Booth των ψηφίων της αθροιστικής εισόδου $X_1 + Y_1$.

Γεννήτρια Μερικών Γινομένων (Partial Product Generator) pp

Η υπομονάδα pp που δημιουργεί τα μερικά γινόμενα βάσει της παραπάνω κωδικοποίησης, δέχεται ως εισόδους μια 16-bit binary είσοδο (A_1 ή A_2) που αποτελεί τον πολλαπλασιαστή, και μια τριάδα ψηφίων της ίδιας βαθμίδας *one*, *two*, *sign* που παράχθηκε από τον recoder. Η έξοδος της γεννήτριας μερικών γινομένων pp είναι ένας 17-bit binary αριθμός (βάσει του w που έχει επιλεγθεί σύμφωνα με τον αλγόριθμο των Chen/Chang όπως θα επεξηγήσουμε στην επόμενη παράγραφο), που αποτελεί ένα μερικό γινόμενο του πολλαπλασιασμού.

Γεννήτρια Περικομένων Μερικών Γινομένων (Truncated Partial Product Generator) tpp

Η υπομονάδα tpp δημιουργεί τα περικομμένα μερικά γινόμενα βάσει της παραπάνω κωδικοποίησης και σύμφωνα με την προσέγγιση των Chen/Chang. Δέχεται ως εισόδους μια 16-bit binary είσοδο (A_1 ή A_2) που αποτελεί τον πολλαπλασιαστή, και μια τριάδα ψηφίων της ίδιας βαθμίδας *one*, *two*, *sign* που παράχθηκε από τον recoder. Η έξοδος της γεννήτριας μερικών γινομένων pp είναι ένας μεταβλητού μήκους binary αριθμός (βάσει του w που έχει επιλεγθεί σύμφωνα με τον αλγόριθμο των Chen/Chang ορίζεται διαφορετικό τμήμα αποκοπής, όπως θα επεξηγήσουμε στην επόμενη παράγραφο), που αποτελεί ένα (περικομμένο) μερικό γινόμενο του πολλαπλασιασμού.

S-MB1 recoder USUM2MB2

Ο S-MB1 recoder USUM2MB2 δέχεται τις 16-bit εισόδους της μονάδας UDMA X_2 και Y_2 καθώς και το bit ελέγχου CL_1 . Η έξοδος του είναι οι 8 τριάδες δυαδικών αριθμών *one*, *two*, *sign*, οι οποίες αντιπροσωπεύουν την κωδικοποίηση Modified Booth των ψηφίων της αθροιστικής εισόδου $X_2 + Y_2$.

Δενδρικός αθροιστής 20:2 UTREE

Ο δενδρικός CS αθροιστής UTREE δέχεται 20 εισόδους των 20 bit. Οι 18 από αυτές είναι τα μερικά γινόμενα που παράχθηκαν στο προηγούμενο στάδιο και οι δύο είναι οι εισοδοί B_1, B_2 με επέκταση προσήμου ώστε να εκτείνονται στα 20 bit. Η έξοδος του είναι ένας 16 bit CS αριθμός (C και S τμήμα).

Τελικός Αθροιστής UFA

Ο τελικός αθροιστής UFA είναι ένας CLA adder που δέχεται ως είσοδο ένα 16 bit CS αριθμό και δίνει ως έξοδο ένα 16 bit binary αριθμό που είναι και το τελικό αποτέλεσμα που παράγει η μονάδα.

3.3 Δομική ανάλυση των επιμέρους μονάδων

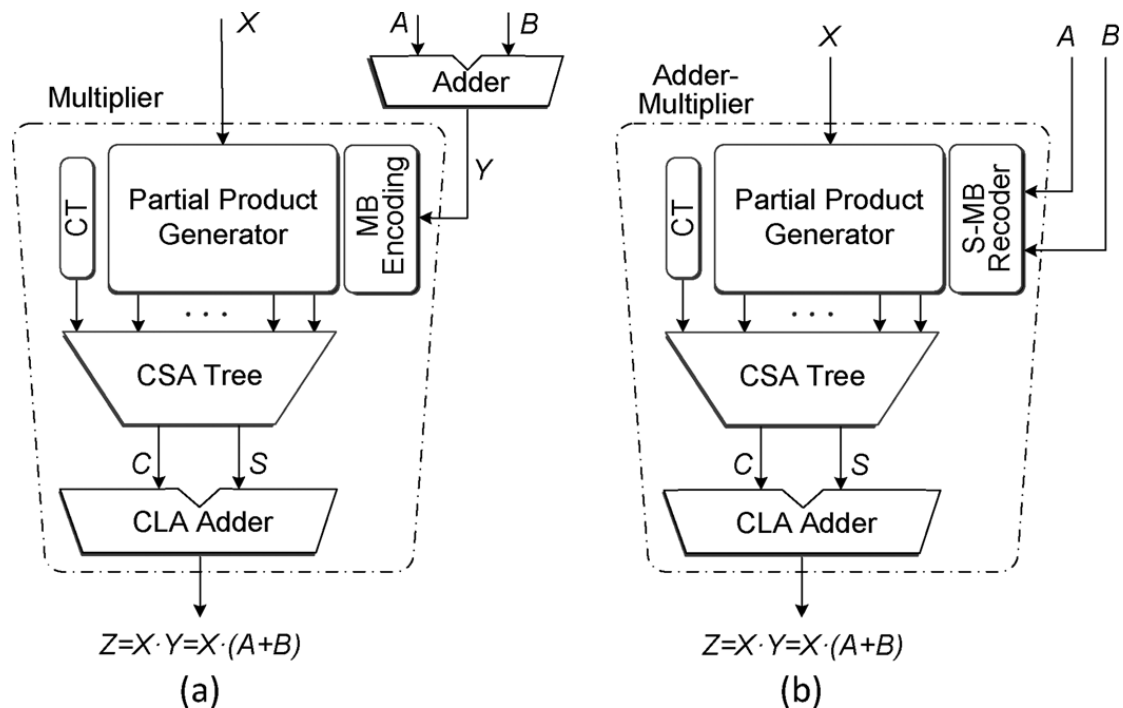
Στην παρούσα παράγραφο γίνεται αναλυτική παρουσίαση της εσωτερικής δομής κάθε επιμέρους μονάδας που συνθέτει την UDMA. Ακολουθείται μια top-down μεθοδολογία ανάλυσης δηλαδή, σύμφωνα με τη ροή των δεδομένων, κάθε μονάδα που συναντάται αναλύεται δομικά. Για τις υπομονάδες που επαναλαμβάνονται στην εσωτερική δομή της UDMA, παρουσιάζεται η δομή μιας από αυτές αντιπροσωπευτικά και επισημαίνονται οι περιπτώσεις που εμφανίζεται ξανά η συγκεκριμένη δομή.

3.3.1 S-MB1 recoder USUM2MB1 και USUM2MB2

Στην επεξεργασία DSP αλγορίθμων δύο ακολουθίες που συναντώνται πολύ συχνά είναι μια πρόσθεση μετά από ένα πολλαπλασιασμό ή αλλιώς Multiply-Add ή Multiply-Accumulate (MAD ή MAC operation) και ένας πολλαπλασιασμός μετά από μια πρόσθεση ή αλλιώς Add-Multiply (AM operation). Η μονάδα UDMA είναι σχεδιασμένη ώστε να υλοποιεί όσο γίνεται πιο αποδοτικά τις 2 αυτές λειτουργίες για την βέλτιστη απεικόνιση των DSP αλγορίθμων.

Η AM λειτουργία υλοποιείται στο πρώτο στάδιο της επεξεργασίας δεδομένων από τη μονάδα, όπου οι εισοδοί X_1, Y_1 και X_2, Y_2 εισέρχονται στην αθροιστική είσοδο του κάθε πολλαπλασιαστή.

Τυπικά, σε τέτοιες περιπτώσεις η αθροιστική είσοδος είναι συνήθως ένας CS adder το αποτέλεσμα του οποίου (είτε απευθείας ως CS αριθμός είτε με μετατροπή σε binary) κωδικοποιείται στη συνέχεια κατά MB στα πλαίσια του πολλαπλασιασμού που ακολουθεί. Επιλέχθηκε ωστόσο μια εναλλακτική υλοποίηση που εστιάζει στην απευθείας μετατροπή του αθροίσματος δυο αριθμών σε MB μορφή, δηλαδή στην FAM (fused Add-Multiply) λειτουργία, η οποία έχει ως στόχο την εξάλειψη του αρχικού αθροιστή και της καθυστέρησης που εισάγει (λόγω της διάδοσης των κρατουμένων σε αυτόν) έτσι ώστε η τελική καθυστέρηση να είναι μικρότερη και ανεξάρτητη του μήκους της λέξης εισόδου. Στο σχήμα 3.3 φαίνεται η υλοποίηση της AM λειτουργίας βάσει της συμβατικής σχεδίασης και βάσει της συγχωνευμένης (fused) σχεδίασης που χρησιμοποιεί η UDMA.



Σχήμα 3.3 (a) Συμβατική υλοποίηση AM λειτουργίας
(b) Fused Add-Multiply λειτουργία

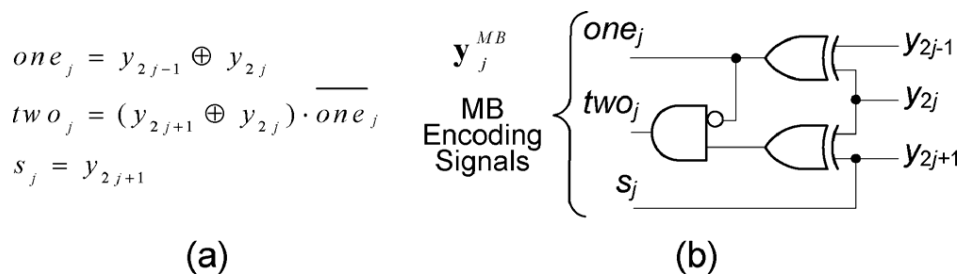
Συγκεκριμένα, χρησιμοποιήθηκε μια πρόσφατα προταθείσα τεχνική, ο απευθείας μετασχηματισμός αθροίσματος σε MB μορφή (Sum to MB ή S-MB recoder) που είναι πιο αποδοτικός όσον αφορά την καθυστέρηση του κρίσιμου μονοπατιού (critical path delay), την επιφάνεια πυριτίου (area) και την κατανάλωση ισχύος (power consumption).

Με την συγκεκριμένη τεχνική, επανακωδικοποιούμε (recode) το άθροισμα 2 συνεχόμενων bits της εισόδου A (με 2 συνεχόμενα bits της εισόδου B σε ένα MB

ψηφίο. Ο πίνακας 3.1 είναι ο πίνακας αληθείας σημάτων κωδικοποίησης Modified Booth, ενώ στο σχήμα 3.4 παρουσιάζονται οι αντίστοιχες εξισώσεις και το κύκλωμα που υλοποιεί τα σήματα κωδικοποίησης MB.

Πίνακας 3.1 Πίνακας αληθείας σημάτων κωδικοποίησης Modified Booth

Y_{2i+1}	Y_{2i}	Y_{2i-1}	neg_i	one_i	two_i	Booth: W_i
0	0	0	0	0	0	0
0	0	1	0	1	0	+1
0	1	0	0	1	0	+1
0	1	1	0	0	1	+2
1	0	0	1	0	1	-2
1	0	1	1	1	0	-1
1	1	0	1	1	0	-1
1	1	1	1	0	0	-0



Σχήμα 3.4 (a) Εξισώσεις σημάτων MB κωδικοποίησης
(b) Κύκλωμα υλοποίησης σημάτων MB κωδικοποίησης

Όπως φαίνεται και στην αντίστοιχη σχέση στο κεφάλαιο 2.2.2, ένα MB ψηφίο δημιουργείται από 3 bits, ένα αρνητικής αξίας (negative weight) και 2 θετικής αξίας (positive weight). Για να μετατρέψουμε λοιπόν τα δυο παραπάνω ζεύγη bits σε MB μορφή πρέπει να χρησιμοποιήσουμε αριθμητική προσημασμένου bit (signed bit arithmetic). Για το σκοπό αυτό εισάγουμε δύο τροποποιημένους FA, προσημασμένους σε επίπεδο bit, τους οποίους ονομάζουμε FA^* και FA^{**} . Οι πίνακες 3.2 και 3.3 αποτελούν τους πίνακες αληθείας για τους FA^* και FA^{**} αντιστοίχως ενώ στο σχήμα 3.5 παρουσιάζεται η εσωτερική δομή τους (σε σχέση με τον συμβατικό FA που παρουσιάστηκε στο κεφάλαιο 2) και οι λογικές εξισώσεις τους.

Πίνακας 3.2 Πίνακας αληθείας του FA*

<i>Inputs</i>			<i>Output</i>	<i>Outputs</i>	
<i>p (+)</i>	<i>q (-)</i>	<i>c_i (+)</i>	<i>Value</i> ¹	<i>c_o (+)</i>	<i>s (-)</i>
0	0	0	0	0	0
0	0	1	+1	1	1
0	1	0	-1	0	1
0	1	1	0	0	0
1	0	0	+1	1	1
1	0	1	+2	1	0
1	1	0	0	0	0
1	1	1	+1	1	1

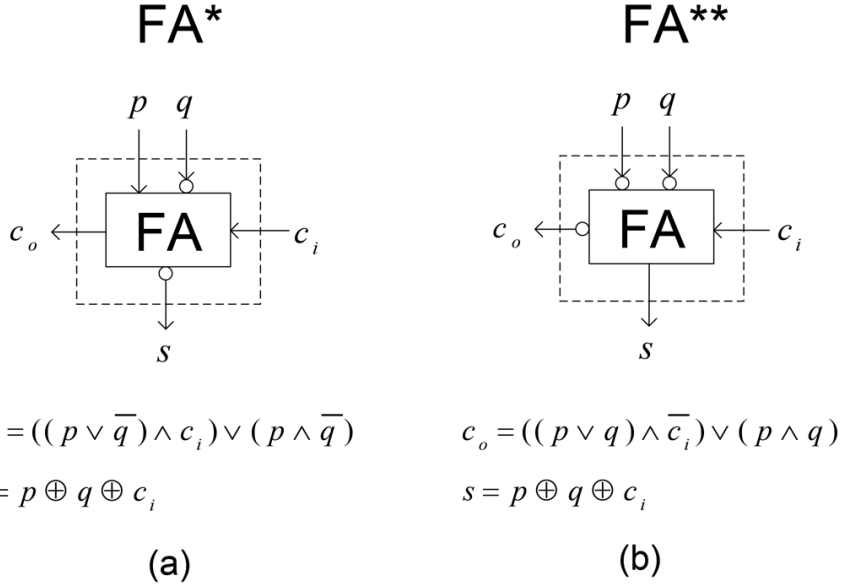
$$^1 \text{Output Value} = 2 \cdot c_o - s = p - q + c_i$$

Πίνακας 3.3 Πίνακας αληθείας του FA**

<i>Inputs</i>			<i>Output</i>	<i>Outputs</i>	
<i>p (-)</i>	<i>q (-)</i>	<i>c_i (+)</i>	<i>Value</i> ²	<i>c_o (-)</i>	<i>s (+)</i>
0	0	0	0	0	0
0	0	1	+1	0	1
0	1	0	-1	1	1
0	1	1	0	0	0
1	0	0	-1	1	1
1	0	1	0	0	0
1	1	0	-2	1	0
1	1	1	-1	1	1

$$^2 \text{Output Value} = -2 \cdot c_o + s = -p - q + c_i$$

Ο FA* υλοποιεί τη σχέση $2 \times c_o - s = p - q - c_i$ όπου τα bits s,q έχουν αρνητικό πρόσημο ενώ ο FA** υλοποιεί τη σχέση $-2 \times c_o + s = -p - q + c_i$, όπου τα bits p,q έχουν αρνητικό πρόσημο. Όπως φαίνεται στους πίνακες 3.2 και 3.3 οι έξοδοι FA* και FA** έχουν εύρος τιμών $\{-1,0,+1,+2\}$ και $\{-2,-1,0,+1\}$ αντιστοίχως.



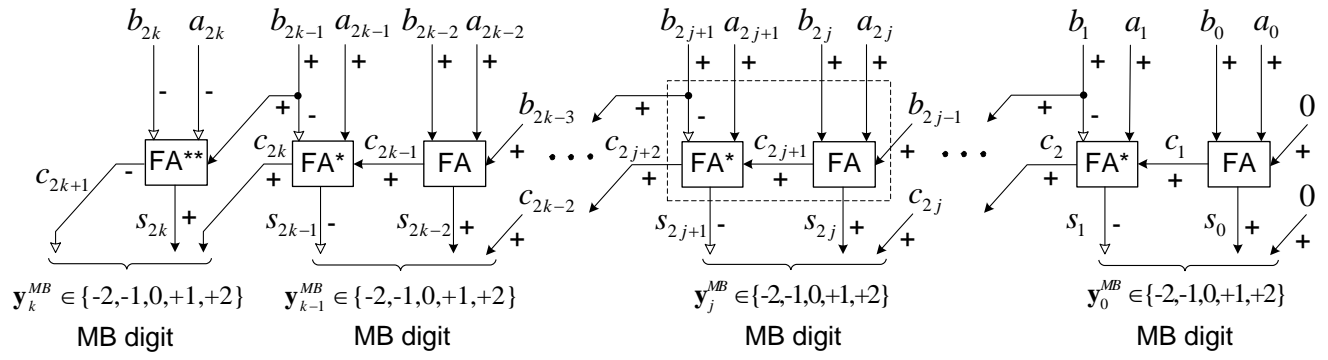
Σχήμα 3.5 (a) Εσωτερική δομή και εξισώσεις του FA*
 (b) Εσωτερική δομή και εξισώσεις του FA**

Το κύκλωμα S-MB1 φαίνεται στο σχήμα 3.6. Είναι εμφανές ότι το άθροισμα των A,B δίνεται από τη σχέση : $Y = A + B = y_k \times 2^{2k} + \sum_{j=0}^{k-1} y_j^{MB} \times 2^{2j}$

Όπου $y_j^{MB} = -2s_{2j+1} + s_{2j} + c_{2j}$.

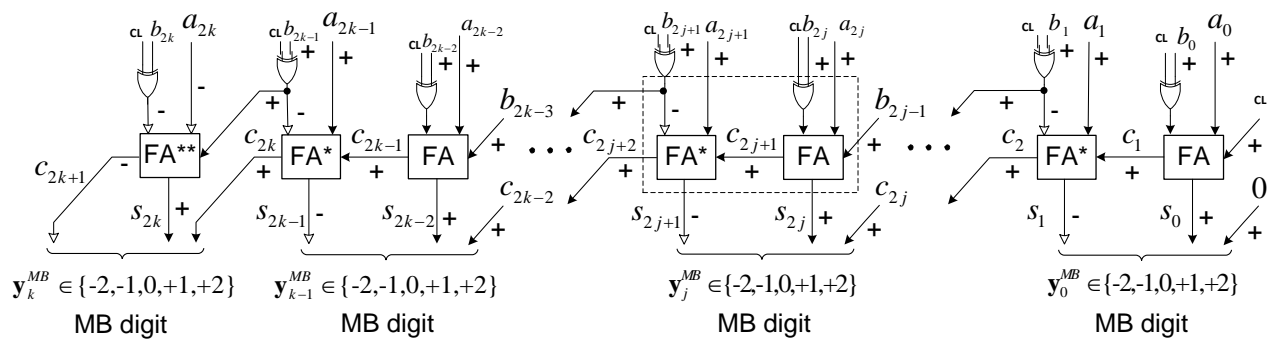
Η κωδικοποίηση των MB ψηφίων y_j^{MB} της παραπάνω σχέσης γίνεται με βάση την ανάλυση στο Κεφάλαιο 2.

Τα bits s_{2j+1}, s_{2j} παράγονται από το j-οστό κύτταρο (cell) του S-MB1 που φαίνεται στο σχήμα 3.6. Για την παραγωγή του αθροίσματος s_{2j} και του κρατουμένου c_{2j+1} χρησιμοποιείται ένας συμβατικός FA με εισόδους a_{2j}, b_{2j}, b_{2j-1} ενώ για το bits s_{2j+1} που πρέπει να έχει αρνητικό πρόσημο χρησιμοποιούμε τον FA* με εισόδους $a_{2j+1}, b_{2j+1}, c_{2j+1}$. Για το σχηματισμό του πιο σημαντικού ψηφίου (MSD) χρησιμοποιούμε τον FA** με εισόδους a_{2k}, b_{2k}, b_{2k-1} .



Σχήμα 3.6 Το κύκλωμα του S-MB1 recoder

Για την επέκταση του κυκλώματος ώστε η αθροιστική είσοδος να υλοποιεί και την πράξη της αφαίρεσης, χρησιμοποιούμε δυο σήματα ελέγχου CL_0, CL_1 (επίτρευση) που όταν παίρνουν την τιμή 1 τότε οι είσοδοι B_1, B_2 αντιστοίχως αντιστρέφονται ενώ η τιμή του σήματος ελέγχου προστίθεται στον πρώτο FA (αποτελεί δηλαδή το c_0) ώστε να λάβουμε το συμπλήρωμα ως προς 2 της αντίστοιχης εισόδου. Το κύκλωμα του S-MB recoder τελικά τροποποιείται όπως φαίνεται στο σχήμα 3.7.



Σχήμα 3.7 Το τροποποιημένο κύκλωμα του S-MB1 recoder

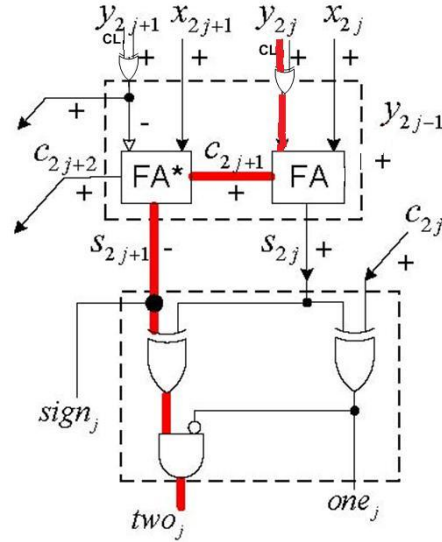
Για τον υπολογισμό της κρίσιμης καθυστέρησης του τελικού κυκλώματος θα χρησιμοποιηθεί η μετρική που φαίνεται στον πίνακα 3.4.

Πίνακας 3.4 Στοιχεία των κυκλωμάτων των μονάδων και η καθυστέρηση που εισάγουν

Components	Delay
NAND-2, NOR-2	1
XOR, XNOR	2
Half Adder	2
Full Adder	4
2 to 1 multiplexer (MUX)	2
Register	2

Όπως προκύπτει και από το σχήμα 3.8 η καθυστέρηση του κυκλώματος είναι:

$$T_{S-MB1} = 2 \times T_{XOR} + T_{FA,carry} + T_{FA^*,sum} + T_{NAND} = 10.$$



Σχήμα 3.8 Κρίσιμο μονοπάτι της αθροιστικής εισόδου

3.3.2 Partial Product Generator και Truncated Partial Product Generator

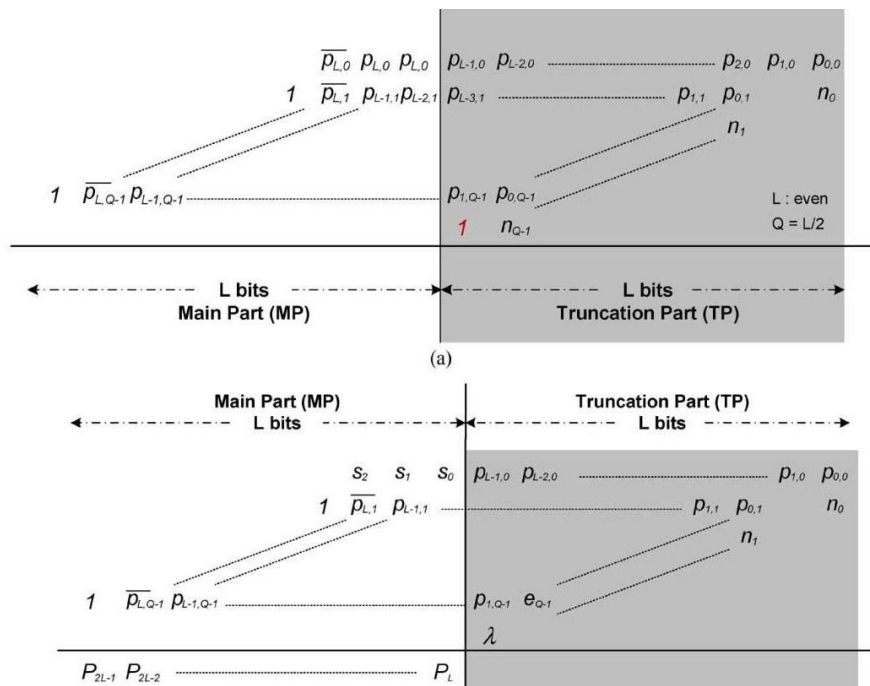
Οι πολλαπλασιαστές που χρησιμοποιούν οι μονάδες UDMA συναποτελούνται από τα κυκλώματα κωδικοποίησης Modified Booth (που παρουσιάστηκε στην προηγούμενη παράγραφο), τη γεννήτρια των μερικών γινομένων και το τελικό κύκλωμα άθροισής τους. Για την δημιουργία των μερικών γινομένων υιοθετείται η προσέγγιση των Chen και Chang, η οποία χρησιμοποιεί ένα εκτιμητή βασισμένο στην πιθανότητα υπό δέσμευση, για να επιτύχει τη βέλτιστη επίδοση όσον αφορά το trade-off μεταξύ ακρίβειας αποτελεσμάτων και κόστους σε επιφάνεια και ισχύ.

Το ζητούμενο είναι η ελάττωση της απαιτούμενης επιφάνειας και ισχύος μέσω της περικοπής υλικού (truncation) αλλά παράλληλα η διασφάλιση της καλύτερης δυνατής ακρίβειας. Για το λόγο αυτό, ο πίνακας των μερικών γινομένων διαχωρίζεται σε δύο μέρη 16 στηλών το καθένα: το κύριο μέρος (main part, MP) και το περικομμένο μέρος (truncated part, TP). Από το περικομμένο μέρος ένας μεταβλητός αριθμός στηλών w μπορεί να επιλεγθεί για να συνυπολογιστεί στο τελικό αποτέλεσμα, ώστε να υπάρχει δυνατότητα προσαρμογής ανάλογα με τις ανάγκες του συστήματος, ενώ η $(MP + w + 1)^{th}$ στήλη χρησιμοποιείται για την εκτίμηση των τιμών αποζημίωσης

(compensation values). Για την παρούσα εργασία βρέθηκε ότι η επιλογή $w=3$ δίνει τα καλύτερα αποτελέσματα. Συνεπώς, τα μερικά γινόμενα που θα παραχθούν θα έχουν μήκος λέξης: $MP + w + 1 = 16 + 3 + 1 = 20$ bits.

Στο σχήμα 3.9 φαίνονται τα μερικά γινόμενα, το κύριο μέρος και το περικομμένο μέρος, ενώ χρησιμοποιείται η απεικόνιση:

$$\{s_2, s_1, s_0, \lambda, e_{q-1}\} = \{\overline{p_{L,0}}, p_{L,0}, p_{L,0}, 1, p_{0,q-1}\} + \{0,0,0,0, n_{q-1}\}.$$



Σχήμα 3.9 Το κύριο μέρος και το περικομμένο μέρος των μερικών γινομένων

Με n_i συμβολίζονται οι όροι που προστίθενται στα μερικά γινόμενα για τον υπολογισμό του συμπληρώματος ενώ η μονάδα προστίθεται για τη στρογγυλοποίηση του αποτελέσματος (rounding off operator).

Η κωδικοποίηση MB παρουσιάστηκε στο κεφάλαιο 2 και στην προηγούμενη παράγραφο. Για τις ανάγκες του εκτιμητή (που υπολογίζει τις τιμές αποζημίωσης) χρησιμοποιείται η μεταβλητή nz_i που είναι 0 όταν το ψηφίο y_i είναι 0 και 1 όταν δεν είναι, όπως φαίνεται και στον πίνακα 3.5. Οι τιμές αυτές δίνουν πληροφορία για το τμήμα που τελικά αποκόπτεται, ώστε να διατηρηθεί όσο το δυνατόν καλύτερη ακρίβεια. Συγκεκριμένα, οι μεταβλητές nz_i και e_{q-1} χρησιμοποιούνται για την εκτίμηση της προσδοκώμενης υπό συνθήκη τιμής του βάρους κάθε διαμέρισης TP_j .

Πίνακας 3.5 Non Zero τιμές με βάση την κωδικοποίηση MB

y_{2i+1}	y_{2i}	y_{2i-1}	y_i	nz_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	2	1
1	0	0	-2	1
1	0	1	-1	1
1	1	0	-1	1
1	1	1	0	0

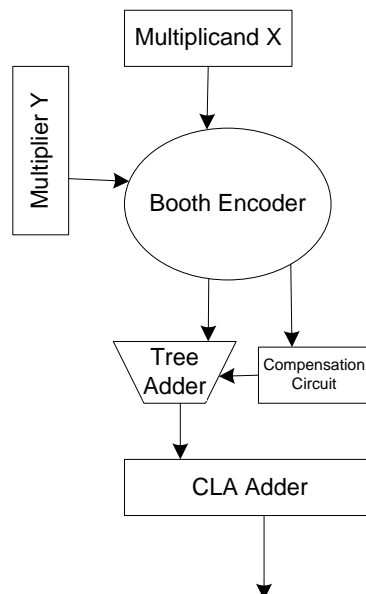
Το τελικό αποτέλεσμα δίνεται από τη σχέση:

$$P = MP + 2^L \times (TP_w + 2^{-w} \times \sum_{j=0}^{q-1-\frac{w}{2}} \frac{z_j}{2}).$$

Για το κύριο μέρος MP και το περικομμένο μέρος (τρεις στήλες) TP_w δημιουργούνται και υπολογίζονται τα μερικά γινόμενα και τα περικομμένα μερικά γινόμενα, ενώ για το

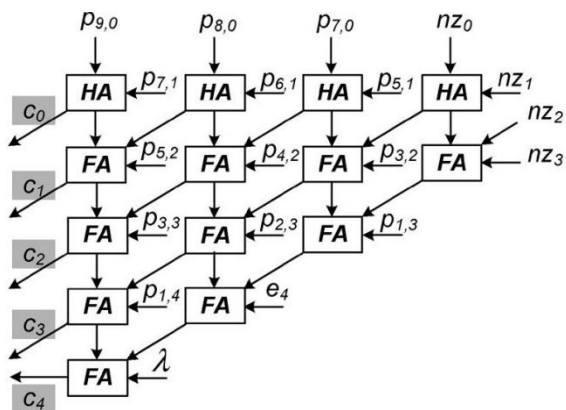
$$z_j \text{ ισχύει: } \begin{cases} z_j = nz_j, & q - 2 \geq j \geq 0 \\ z_j = e_{q-1}, & j = q - 1 \end{cases}.$$

Στο σχήμα 3.10 παρουσιάζεται η γενική μορφή ενός τέτοιου πολλαπλασιαστή.



Σχήμα 3.10 Πολλαπλασιαστής που χρησιμοποιεί κύκλωμα εκτιμητή

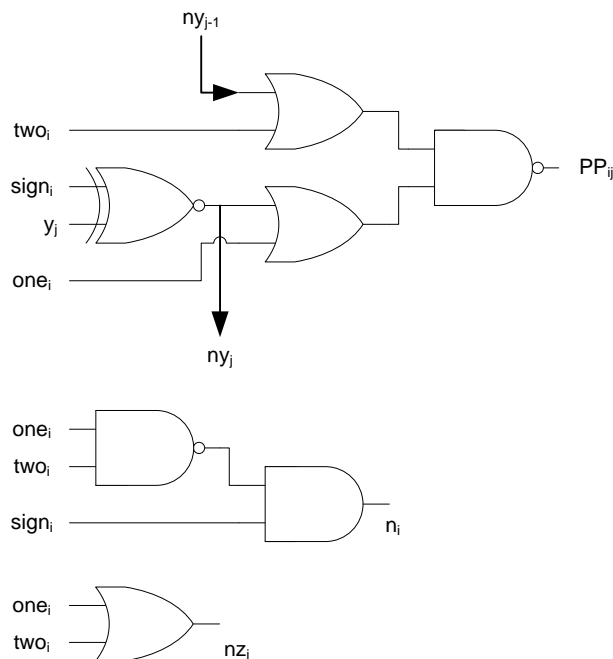
Το κύκλωμα του εκτιμητή για $w=3$ υπολογίζεται με βάση τις μη μηδενικές nz_i τιμές του πίνακα 3.5 και φαίνεται στο σχήμα 3.11.



Σχήμα 3.11 Το κύκλωμα του εκτιμητή (estimator)

Τα κυκλώματα δημιουργίας των μερικών γινομένων pp_i και tpp_i , των διορθωτικών όρων n_i και των μη μηδενικών όρων nz_i φαίνονται αντιστοίχως στο σχήμα 3.12. Σημειώνεται πως οι όροι n_i και nz_i παράγονται παράλληλα με τα ψηφία του μερικού γινομένου και συνεπώς δεν συνεισφέρουν στη συνολική καθυστέρηση του κυκλώματος καθώς πρόκειται για απλούστερα κυκλώματα, με μικρότερη κρίσιμη καθυστέρηση.

Άρα η τελική συνολική καθυστέρηση για την παραγωγή των μερικών γινομένων είναι: $T_{PPG} = T_{XNOR} + T_{OR} + T_{NAND} = 4$.



Σχήμα 3.12 Τα κυκλώματα δημιουργίας των μερικών γινομένων, των διορθωτικών όρων και των μη μηδενικών τιμών

3.3.3 Δενδρικός CS αθροιστής 20:2 UTREE

Το δεύτερο κομμάτι του (κάθε) πολλαπλασιαστή είναι ο αθροιστής των μερικών γινομένων που έχουν παραχθεί.

Η μονάδα χρησιμοποιεί το IP block DW02_TREE από την βιβλιοθήκη της Designware. Πρόκειται για ένα δενδρικό Wallace-based συμπιεστή (Κεφ. 2.3.3.2), που δέχεται 20 εισόδους (18 μερικά γινόμενα και τις εισόδους B_1, B_2) και δίνει ως έξοδο το άθροισμά τους σε CS μορφή. Οι εισοδοί έχουν μήκος 20 bit όπως είδαμε για μεγαλύτερη ακρίβεια, οπότε από το αποτέλεσμα του δενδρικού αθροιστή τελικά αποκόπτονται bits ώστε η έξοδος να είναι ένας 16 bit CS αριθμός.

Επιλέχθηκε ένας ενιαίος δενδρικός αθροιστής για όλα τα μερικά γινόμενα, καθώς πειραματικά αποδεικνύεται πως καταλαμβάνει συνολικά μικρότερο area ενώ και η κρίσιμη καθυστέρηση είναι ελαφρώς μικρότερη ($-0.16 ns$) συγκριτικά με την χρησιμοποίηση ενός δενδρικού αθροιστή 9 σε 2 για κάθε πολλαπλασιαστή και στη συνέχεια ενός νέου δενδρικού αθροιστή 6 σε 2. Σύμφωνα με τη θεωρητική ανάλυση, όπως φαίνεται και στον πίνακα 3.6, η καθυστέρηση θα έπρεπε να ήταν παρόμοια για τις δυο περιπτώσεις αφού έχουμε συνολικό βάθος $D(h) = 7$, ωστόσο η μικρή διαφορά πιθανώς εξηγείται από την υλοποίηση του δενδρικού αθροιστή η οποία εναπόκειται στον Design Compiler. Σε κάθε περίπτωση, το κύκλωμα θα έχει θεωρητικά αναμενόμενο βάθος 7 FA δηλαδή η κρίσιμη καθυστέρηση θα είναι: $T_{tree} = 7 \times T_{FA} = 28$.

Πίνακας 3.6 Βάθος δένδρου Wallace

h	4	5-6	7-9	10-13	14-19	20-28	29-42
D(h)	2	3	4	5	6	7	8

3.3.4 Τελικός CLA αθροιστής

Το τελικό στάδιο επεξεργασίας δεδομένων στη μονάδα UDMA είναι ένας αθροιστής για τη μετατροπή των CS αποτελεσμάτων σε ένα 16 bit binary αριθμό που είναι και η έξοδος της μονάδας.

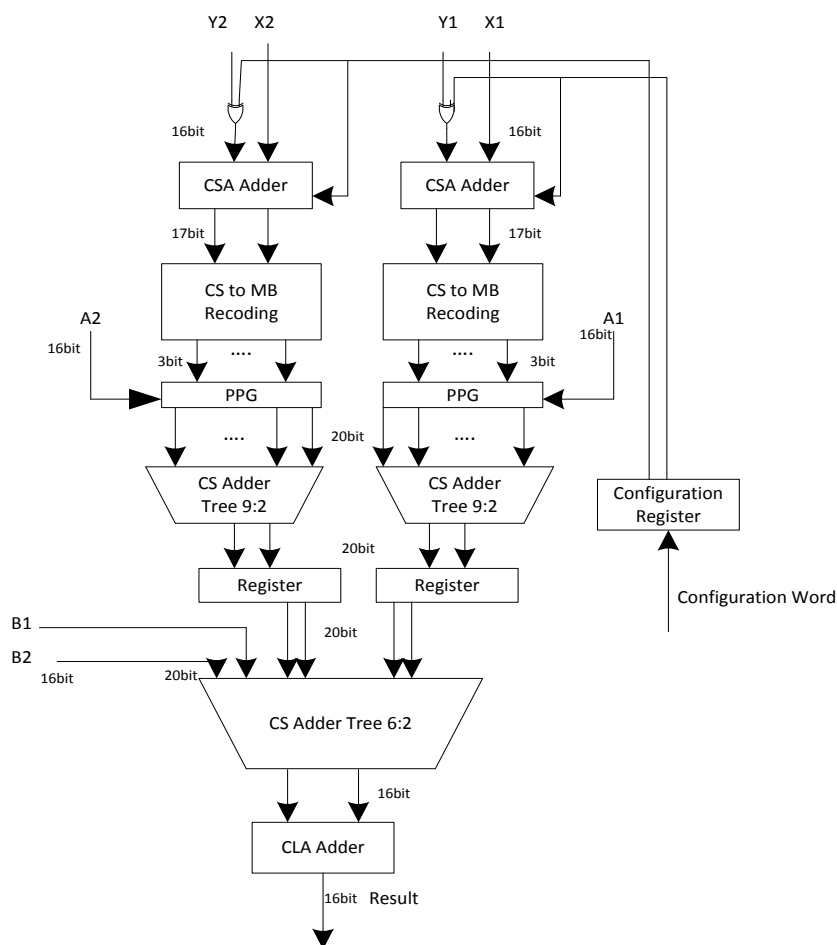
Χρησιμοποιήθηκε το IP block DW01_ADD από την βιβλιοθήκη της Designware. Όμοια με πριν, ο Design Compiler υλοποιεί βέλτιστα με βάση τα optimizations που έχουμε θέσει ένα μοντέλο carry lookahead αθροιστή (Brent-Kung, Sklansky, Kogge-Stone, Han-Carlson, Knowles κλπ), η κρίσιμη καθυστέρηση του οποίου γενικά όπως

είδαμε και στο Κεφ. 2.3.2.3 είναι λογαριθμική συνάρτηση του μήκους εισόδου:
 $T_{add} = (2 + \log_2 n) \times T_{FA} = 24$.

Η εσωτερική δομή του Brent-Kung adder -όχι απαραίτητα το μοντέλο που χρησιμοποιείται κατά τη σύνθεση του κυκλώματος- έχει παρουσιαστεί στο σχήμα 2.6 του Κεφ. 2.3.2.3.

3.4 UDMA pipeline: Δομικές διαφορές και ανάλυση

Η δεύτερη προτεινόμενη μονάδα ενοποιημένου διπλού πολλαπλασιαστή-αθροιστή χρησιμοποιεί διοχέτευση (pipeline). Στόχος είναι η δημιουργία ενός κυκλώματος με μικρότερο critical path συγκριτικά με την αρχικά προτεινόμενη UDMA και η διερεύνηση της αποδοτικότητας της απεικόνισης των υπό εξέταση αλγορίθμων με μια τέτοια μονάδα καθώς και του tradeoff μεταξύ της ελάττωσης του latency και της αύξησης των area και power consumption. Η δομή της μονάδας UDMA με pipeline φαίνεται στο σχήμα 3.13.



Σχήμα 3.13 Η μονάδα UDMA (pipeline): Detailed View

Όπως είναι εμφανές, οι εισοδοί/έξοδοι, τα βασικά δομικά στοιχεία και οι λειτουργίες που επιτελεί η μονάδα είναι πανομοιότυπες με αυτές που παρουσιάστηκαν στις προηγούμενες παραγράφους. Οι βασικές δομικές διαφορές σε σχέση με την UDMA εστιάζονται στη χρήση ενός ξεχωριστού δενδρικού αθροιστή (9 σε 2) για τα μερικά γινόμενα κάθε πολλαπλασιαστή και ενός δενδρικού αθροιστή 6 σε 2 για την τελική άθροιση. Σε κάθε περίπτωση, χρησιμοποιείται το ip block DW02_TREE από την βιβλιοθήκη της Designware. Οι καταχωρητές (registers) για την αποθήκευση των ενδιάμεσων αποτελεσμάτων εισάγονται μετά τους δενδρικούς αθροιστές κάθε πολλαπλασιαστή. Αυτό εξασφαλίζει –όπως προκύπτει πειραματικά αλλά και από τη θεωρητική ανάλυση της κρίσιμης καθυστέρησης του κυκλώματος- το καλύτερο ‘μοίρασμα’ του latency ανάμεσα στα 2 επίπεδα της μονάδας. Επιτυγχάνεται, δηλαδή, η ελαχιστοποίηση της κρίσιμης καθυστέρησης και παράλληλα του χρονικού διαστήματος που το δεύτερο επίπεδο (που έχει το μικρότερο delay) μένει άεργο κατά τη διάρκεια ενός κύκλου λειτουργίας, με αποτέλεσμα να αξιοποιείται βέλτιστα η μονάδα. Το μειονέκτημα είναι πως χρειάζονται τέσσερις 20 bit registers καθώς απαιτείται η αποθήκευση 2 carry save αριθμών για τους οποίους όπως δείξαμε διατηρούνται 20 bits με σκοπό την αύξηση της ακρίβειας.

3.5 Critical Path: Καθυστέρηση απόκρισης των προτεινόμενων μονάδων

Έχοντας ολοκληρώσει τη δομική ανάλυση των δυο προτεινόμενων μονάδων και των επιμέρους στοιχείων που τις αποτελούν, μπορούμε να προχωρήσουμε στον προσδιορισμό του κρίσιμου μονοπατιού και της συνολικής καθυστέρησης απόκρισης.

Η καθυστέρηση απόκρισης των επιμέρους μονάδων σε στοιχειώδεις λογικές πύλες όπως έχουμε δείξει κατά την ανάλυση των κυκλωμάτων είναι:

- S-MB1 recoder: 10
- Γεννήτρια Μερικών Γινομένων: 4
- Δενδρικός αθροιστής 20:2: 28
- Δενδρικός αθροιστής 9:2: 16
- Δενδρικός αθροιστής 6:2: 12
- Τελικός CLA αθροιστής: 24

Παρατηρώντας τα σχήματα 3.2 και 3.9 και με βάση την ανάλυση της προηγούμενης παραγράφου συμπεραίνουμε ότι η διαδρομή η οποία παρουσιάζει τη μεγαλύτερη καθυστέρηση απόκρισης από την είσοδο ως την έξοδο κάθε μονάδας (critical path), περνάει από τα εξής στάδια επεξεργασίας:

UDMA

1. S-MB1 recoder
2. Γεννήτρια Μερικών Γινομένων
3. Δενδρικός αθροιστής 20:2
4. Τελικός CLA αθροιστής
- 5.

UDMA pipeline

1. S-MB1 recoder
2. Γεννήτρια Μερικών Γινομένων
3. Δενδρικός αθροιστής 9:2
4. Register

Επομένως η καθυστέρηση απόκρισης για κάθε μονάδα υπολογίζεται ως το άθροισμα των καθυστερήσεων όλων των στοιχείων που βρίσκονται μέσα στο κρίσιμο μονοπάτι. Άρα τελικά:

Καθυστέρηση απόκρισης UDMA: 66

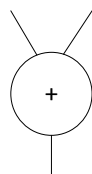
Καθυστέρηση απόκρισης UDMA pipeline: 32

3.6 Λειτουργίες της UDMA

Στην παράγραφο αυτή παρουσιάζονται όλες οι ακολουθίες πράξεων που μπορούν να εκτελέσουν οι προτεινόμενες μονάδες σε ένα κύκλο λειτουργίας. Συνολικά, εάν εκμεταλλευτούμε πλήρως τις δυνατότητες της σχεδίασης, κάθε μονάδα μπορεί να εκτελέσει την πράξη $A_1 \times (X_1 \pm Y_1) + A_2 \times (X_2 \pm Y_2) + B_1 + B_2$. Με τις κατάλληλες εισόδους και σήματα ελέγχου μπορεί να εκτελέσει οποιοδήποτε υποσύνολο πράξεων αυτής της ακολουθίας. Για κάθε μια από τις ακολουθίες πράξεων, που θα αναφέρονται ως UDMA templates, παρουσιάζονται οι είσοδοι και τα σήματα ελέγχου που απαιτούνται, οι πράξεις που εκτελούνται και ο αντίστοιχος γράφος, όπου αυτός είναι διαφορετικός.

Σημειώνεται πως για όλες τις περιπτώσεις που εκτελούνται μόνο προσθέσεις πάντα υπεισέρχεται σε αυτές και τουλάχιστον ένας πολλαπλασιασμός με την μονάδα (εκτός από την περίπτωση $B_1 + B_2$).

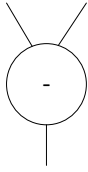
- 1) $X_1 + Y_1$ (ή $X_2 + Y_2$ ή $B_1 + B_2$ ομοίως)



Είσοδοι: $X_1, Y_1, A_1 = 1$ (οι υπόλοιπες είναι 0)

Σήματα ελέγχου: $CL_0 = 0, CL_1 = 0$.

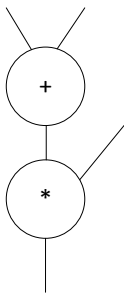
2) $X_1 - Y_1$ (ή $X_2 - Y_2$ ομοίως)



Είσοδοι: $X_1, Y_1, A_1 = 1$ (οι υπόλοιπες είναι 0)

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 0$ (ή αντίστροφα για να εκτελεστεί $X_2 - Y_2$)

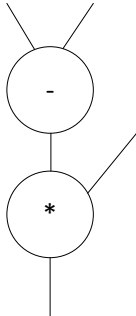
3) $A_1 \times (X_1 + Y_1)$ ή $A_2 \times (X_2 + Y_2)$



Είσοδοι: X_1, Y_1, A_1 (οι υπόλοιπες είναι 0)

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

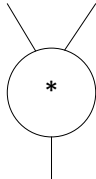
4) $A_1 \times (X_1 - Y_1)$ ή $A_2 \times (X_2 - Y_2)$



Είσοδοι: X_1, Y_1, A_1 (οι υπόλοιπες είναι 0)

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 0$

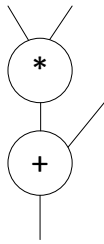
5) $A_1 \times X_1$ ή $A_2 \times X_2$



Είσοδοι: X_1, A_1 (οι υπόλοιπες είναι 0)

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

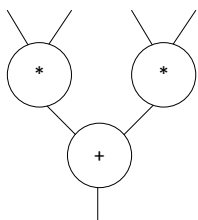
6) $A_1 \times X_1 + B_1 + B_2$



Είσοδοι: X_1, A_1, B_1, B_2

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

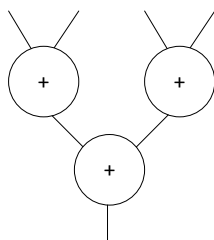
7) $A_1 \times X_1 + A_2 \times X_2 (+B_1 + B_2)$



Είσοδοι: X_1, A_1, X_2, A_2 (B_1, B_2 προαιρετικά, ο γράφος μένει ίδιος)

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

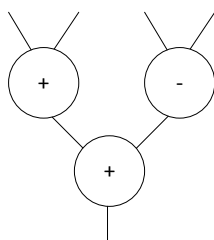
8) $(X_1 + Y_1) + (X_2 + Y_2) \quad (+B_1 + B_2)$



Είσοδοι: $X_1, Y_1, A_1 = 1, X_2, Y_2, A_2 = 1$ (B_1, B_2 προαιρετικά, ο γράφος μένει ίδιος)

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

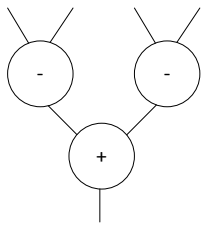
9) $(X_1 + Y_1) + (X_2 - Y_2) \quad (+B_1 + B_2)$



Είσοδοι: $X_1, Y_1, A_1 = 1, X_2, Y_2, A_2 = 1$ (B_1, B_2 προαιρετικά, ο γράφος μένει ίδιος)

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 0$

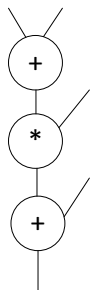
$$10) (X_1 - Y_1) + (X_2 - Y_2) \quad (+B_1 + B_2)$$



Είσοδοι: $X_1, Y_1, A_1 = 1, X_2, Y_2, A_2 = 1$ (B_1, B_2 προαιρετικά, ο γράφος μένει ίδιος)

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 1$

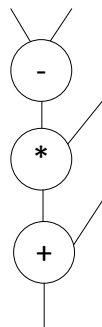
$$11) A_1 \times (X_1 + Y_1) + B_1 + B_2$$



Είσοδοι: X_1, A_1, Y_1, B_1, B_2

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

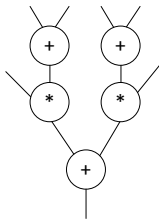
$$12) A_1 \times (X_1 - Y_1) + B_1 + B_2$$



Είσοδοι: X_1, A_1, Y_1, B_1, B_2

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 0$

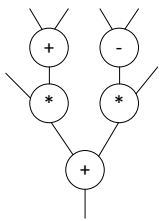
$$13) A_1 \times (X_1 + Y_1) + A_2 \times (X_2 + Y_2) + B_1 + B_2$$



Είσοδοι: $X_1, A_1, Y_1, X_2, A_2, Y_2, B_1, B_2$

Σήματα ελέγχου : $CL_0 = 0, CL_1 = 0$

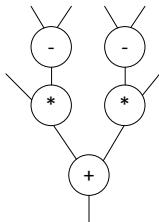
$$14) A_1 \times (X_1 - Y_1) + A_2 \times (X_2 + Y_2) + B_1 + B_2$$



Είσοδοι: $X_1, A_1, Y_1, X_2, A_2, Y_2, B_1, B_2$

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 0$

$$15) A_1 \times (X_1 - Y_1) + A_2 \times (X_2 - Y_2) + B_1 + B_2$$



Είσοδοι: $X_1, A_1, Y_1, X_2, A_2, Y_2, B_1, B_2$

Σήματα ελέγχου : $CL_0 = 1, CL_1 = 1$

4

ΜΕΘΟΔΟΛΟΓΙΑ ΤΗΣ ΑΠΕΙΚΟΝΙΣΗΣ ΑΛΓΟΡΙΘΜΩΝ ΣΤΙΣ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ

4.1 Προεπισκόπηση

Στο παρόν κεφάλαιο γίνεται η παρουσίαση της μεθοδολογίας απεικόνισης (mapping) DSP αλγορίθμων που ακολουθεί η προτεινόμενη αρχιτεκτονική. Παρατίθενται οι παράμετροι που πρέπει να ληφθούν υπόψη κατά τη σχεδίαση μιας αρχιτεκτονικής, πως αυτές επηρεάζουν την τελική σχεδίαση και επεξηγούνται οι τελικές επιλογές στις οποίες καταλήγουμε. Η τελική μορφή της αρχιτεκτονικής όπως αναλύεται σε αυτό το κεφάλαιο είναι κοινή και για τις δυο μονάδες (UDMA και UDMA με pipeline) και είναι αυτή στην οποία γίνονται οι απεικονίσεις των DSP αλγορίθμων στο επόμενο κεφάλαιο.

4.2 Μεθοδολογία απεικόνισης αλγορίθμων- Παράμετροι που καθορίζουν τη μορφή των προτεινόμενων αρχιτεκτονικών

4.2.1 Ενοποιημένη απεικόνιση αλγορίθμων- Τελική μεθοδολογία που επιλέχθηκε

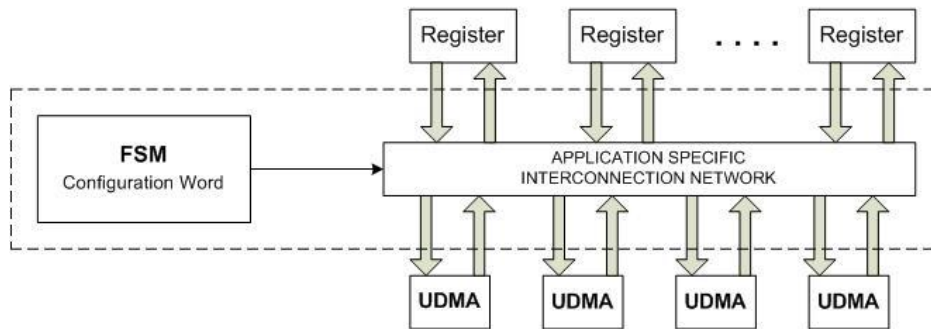
Μια πολύ βασική παράμετρος που οφείλει να λάβει υπόψη η μεθοδολογία, είναι το αν η αρχιτεκτονική θα υποστηρίζει μια ενοποιημένη απεικόνιση των DSP αλγορίθμων. Οι δυο δυνατότητες που υφίστανται ως προς αυτή τη παράμετρο είναι η επιλογή μιας πολύ ευέλικτης μορφής αρχιτεκτονικής που μπορεί να απεικονίσει οποιονδήποτε από τους υπό εξέταση αλγόριθμους και η επιλογή μιας προσέγγισης στην οποία για την απεικόνιση κάθε αλγορίθμου απαιτείται μια εξειδικευμένη αρχιτεκτονική. Μια συμβιβαστική λύση μεταξύ των 2 αυτών επιλογών μπορεί να υπάρξει (ως τρίτη επιλογή): η ομαδοποίηση των αλγορίθμων και η απεικόνιση του κάθε αλγόριθμου σε μια αρχιτεκτονική εξειδικευμένη για τον αλγόριθμο-υπερσύνολο της εκάστοτε ομάδας.

Επιλέχθηκε η δεύτερη μεθοδολογία απεικόνισης, δηλαδή αυτή των εξειδικευμένων αρχιτεκτονικών απεικονίσεων για κάθε αλγόριθμο, διότι ανταποκρίνεται καλύτερα στους στόχους της προτεινόμενης αρχιτεκτονικής. Η συγκεκριμένη μεθοδολογία ακολουθείται και στην FCU based αρχιτεκτονική με την οποία θα γίνει η τελική σύγκριση. Μια εξειδικευμένη αρχιτεκτονική μπορεί να απεικονίσει μόνο τον αλγόριθμο για τον οποίο προορίζεται, ενώ για την απεικόνιση ενός νέου DSP αλγόριθμου απαιτείται η σχεδίαση μιας νέας αποκλειστικής εξειδίκευσης.

Μια αρχιτεκτονική αυτής της μορφής έχει τα εξής χαρακτηριστικά:

- Έναν (σταθερό) αριθμό από μονάδες UDMA που αποτελούν τα επεξεργαστικά στοιχεία της αρχιτεκτονικής
- Έναν αριθμό από καταχωρητές (Register file) που χρησιμοποιούνται για την τροφοδοσία των εισόδων των UDMA και την αποθήκευση των ενδιάμεσων αποτελεσμάτων. Ο αριθμός τους διαφέρει ανά απεικόνιση, αφού εξαρτάται από τον εκάστοτε αλγόριθμο.
- Μια Finite State Machine (FSM) η οποία καθορίζει το δίκτυο διασύνδεσης μεταξύ των καταχωρητών και των μονάδων UDMA και στέλνει σε κάθε κύκλο λειτουργίας τα κατάλληλα σήματα ελέγχου.

Η προτεινόμενη αρχιτεκτονική, η οποία βασίζεται σε αυτή τη μεθοδολογία και συνυπολογίζει και τις παραμέτρους που παρουσιάζονται στις επόμενες παραγράφους, φαίνεται στο Σχήμα 4.1 (abstract form).



Σχήμα 4.1 Η προτεινόμενη αρχιτεκτονική (abstract form)

Από την άλλη πλευρά, η επιλογή της πρώτης μεθοδολογίας θα οδηγούσε σε μια αρχιτεκτονική που θα καταλάμβανε πολύ μεγάλη επιφάνεια λόγω του αυξημένου αριθμού διασυνδέσεων μεταξύ των στοιχείων (Crossbar) ενώ η τρίτη μεθοδολογία παρουσιάζει το μειονέκτημα πως οι πιο απλοί αλγόριθμοι θα απεικονίζονται σε πιο σύνθετα και μεγαλύτερης επιφάνειας κυκλώματα που έχουν σχεδιαστεί για την απεικόνιση πιο σύνθετων αλγορίθμων.

4.2.2 Πλήθος Επεξεργαστικών Πόρων

Η δεύτερη παράμετρος που πρέπει να ληφθεί υπόψη είναι το πλήθος των επεξεργαστικών πόρων, δηλαδή των μονάδων UDMA, που χρησιμοποιεί η αρχιτεκτονική. Για τον προσδιορισμό του πλήθους απαιτείται η μελέτη της ακολουθίας των πράξεων που χαρακτηρίζει κάθε ένα από τους αλγόριθμους, ώστε να ισορροπήσουμε μεταξύ του βέλτιστου παραλληλισμού εντολών και της καταλαμβανόμενης επιφάνειας. Επιλέγεται λοιπόν μια αρχιτεκτονική με 4 μονάδες UDMA, αριθμός ο οποίος παραμένει σταθερός σε όλες τις απεικονίσεις. Η FCU based αρχιτεκτονική χρησιμοποιεί επίσης 4 επεξεργαστικές μονάδες FCU (και μια μονάδα μετατροπής σε binary), οπότε έτσι μπορούμε άμεσα να συγκρίνουμε τις επιδόσεις και στην ταχύτητα εκτέλεσης των αλγορίθμων. Θα μπορούσε βέβαια ενδεχομένως να χρησιμοποιηθεί εναλλακτικά μια αρχιτεκτονική με 2 μονάδες UDMA (εφόσον μπορούν να εκτελέσουν τους ίδιους πολλαπλασιασμούς με 4 FCU) και χρησιμοποιώντας το ίδιο mapping με την FCU (χωρίς δηλαδή να προβούμε σε μεθόδους ελαχιστοποίησης του χρόνου εκτέλεσης που να είναι συγκεκριμένοι για την μονάδα UDMA, Κεφ.4.2.4) να συγκριθούν οι δυο αρχιτεκτονικές καθαρά στο κομμάτι της ενεργειακής απόδοσης.

4.2.3 Αριθμός Καταχωρητών ανά Απεικόνιση

Ο αριθμός των καταχωρητών (registers) που χρησιμοποιείται σε μια εξειδικευμένη αρχιτεκτονική για κάποιο αλγόριθμο, διαφέρει από τον αριθμό που χρησιμοποιούνται από μια άλλη εξειδικευμένη αρχιτεκτονική για την απεικόνιση κάποιου άλλου αλγορίθμου. Μπορούμε να διακρίνουμε τους καταχωρητές σε κατηγορίες ανάλογα με το ρόλο που επιτελούν:

- Καταχωρητές οι οποίοι αποθηκεύουν τις πιθανές καταστάσεις της FSM.
- Καταχωρητές αποθήκευσης των εισόδων που δίνονται κατά το χρόνο εκτέλεσης του αλγορίθμου.
- Καταχωρητές οι οποίοι αποθηκεύουν παραμέτρους του αλγορίθμου που είναι σταθερές και πρέπει να φορτωθούν πριν το χρόνο εκτέλεσης (δεν υπάρχουν σε όλους τους αλγόριθμους). Η FSM αφιερώνει ένα συγκεκριμένο αριθμό κύκλων για να αρχικοποιήσει αυτές τις παραμέτρους πριν από τους κύκλους εκτέλεσης της απεικόνισης.
- Καταχωρητές αποθήκευσης των εισόδων που οδηγούνται σε κάθε κύκλο στις μονάδες UDMA.
- Καταχωρητές αποθήκευσης ενδιάμεσων αποτελεσμάτων. Οι καταχωρητές αυτοί χρησιμοποιούνται όταν τα αποτελέσματα εξόδου κάποιας μονάδας στο τέλος ενός κύκλου θα πρέπει να χρησιμοποιηθούν ως εισοδοί κάποιας άλλης μονάδας μετά από 2 ή περισσότερους κύκλους. Μια πολύ σημαντική παράμετρος που υπεισέρχεται στη μεθοδολογία και επηρεάζει άμεσα την τελική απεικόνιση στην οποία καταλήξαμε είναι ο περιορισμός των καταχωρητών αυτών στο ελάχιστο δυνατό (με αποτελεσματικό mapping), όσο η μείωση του δεν συνεπάγεται την αύξηση των κύκλων εκτέλεσης. Είναι οι μοναδικοί καταχωρητές της αρχιτεκτονικής που με προσεκτική σχεδίαση μπορούν να υποβληθούν σε διαδικασία ελαχιστοποίησης.

4.2.4 Μέθοδος Ελαχιστοποίησης του Χρόνου Εκτέλεσης

Η ελαχιστοποίηση του χρόνου εκτέλεσης της απεικόνισης αποτελεί πρωταρχική προτεραιότητα της προτεινόμενης αρχιτεκτονικής και κεντρικό σημείο της παρούσας εργασίας. Επικεντρώνεται δηλαδή στην ελαχιστοποίηση του αριθμού των κύκλων λειτουργίας που απαιτούνται για να ολοκληρωθεί η απεικόνιση και να παραχθούν τα αποτελέσματα εξόδου.

Η μεθοδολογία που ακολουθείται είναι η εξής: αρχικά μελετάται ο κώδικας που περιγράφει τον εκάστοτε αλγόριθμο (που είναι γραμμένος σε γλώσσα υψηλού

επιπέδου) και ο κύριος βρόχος (και υποβρόχοι που τυχόν περιέχει) ξεδιπλώνεται και γράφεται αναλυτικά η ακολουθία των στοιχειωδών πράξεων. Οι πράξεις αυτές, που είναι πολλαπλασιασμοί και προσθαιρέσεις, υφίστανται όπου είναι δυνατό παραγοντοποίηση ενώ οι ανεξάρτητοι πολλαπλασιασμοί-προσθέσεις (χωρίς data dependencies) γίνονται ζευγάρια. Ο σκοπός των δυο αυτών κινήσεων είναι οι ακολουθίες στοιχειωδών πράξεων που θα απεικονίζονταν σε πιο απλά templates να απεικονιστούν τελικά σε πιο σύνθετα, ώστε να έχουμε πληρέστερη αξιοποίηση των δυνατοτήτων των μονάδων της αρχιτεκτονικής και τελικά την μείωση των απαιτούμενων κύκλων εκτέλεσης. Τέλος, εξετάζεται η αποδοτικότητα της απεικόνισης, δηλαδή ο συνολικός αριθμός των κύκλων εκτέλεσης και το ποσοστό χρησιμοποίησης των μονάδων, και αν κριθεί απαραίτητο μπορεί να αναλυθεί όχι μια, αλλά N επαναλήψεις του κύριου υπολογιστικού βρόχου. Σε κάθε περίπτωση, κατά τη διαδικασία της ανάθεσης των στοιχειωδών πράξεων στις UDMA σε κάθε κύκλο, πρωταρχικό μέλημα αποτελεί η ελαχιστοποίηση των κύκλων εκτέλεσης ενώ δευτερευόντως επιδιώκεται η ελαχιστοποίηση των καταχωρητών αποθήκευσης των ενδιάμεσων αποτελεσμάτων που απαιτούνται.

5

ΑΠΕΙΚΟΝΙΣΗ DSP ΑΛΓΟΡΙΘΜΩΝ - ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

5.1 Προεπισκόπηση

Στο παρόν κεφάλαιο παρουσιάζονται τα πειραματικά αποτελέσματα από την απεικόνιση επιλεγμένων DSP αλγορίθμων στην προτεινόμενη αρχιτεκτονική. Η σύγκριση των αποτελεσμάτων σε επίπεδο απόδοσης μονάδας και συνολικής απεικόνισης των DSP αλγορίθμων γίνεται με την μονάδα FCU και κατ'επέκταση με την FCU based αρχιτεκτονική, που είναι και αυτή που έχει παρουσιάσει τα βέλτιστα αποτελέσματα σε σχέση με μια σειρά διάφορων προτεινόμενων μονάδων/αρχιτεκτονικών (FCC, RAU) για την απεικόνιση DSPαλγορίθμων. Αυτή χρησιμοποιεί παρομοίως 4 μονάδες FCU αλλά και μια μονάδα CStoBIN για την μετατροπή των αποτελεσμάτων από carry save μορφή σε binary.

Αρχικά γίνεται η σύγκριση κατανάλωσης επιφάνειας και ενεργειακής απόδοσης σε επίπεδο μονάδας, και συγκεκριμένα ως προς:

- επιφάνεια-καθυστέρηση απόκρισης (area- latency curve)
- ενεργειακή σύγκριση βάσει της μετρικής MOPS/W (Mega Operations per Second/Watt), δηλαδή πράξεις/λειτουργίες που δύναται να εκτελέσει η μονάδα κάθε δευτερόλεπτο ανά Watt

Στη συνέχεια παρουσιάζονται τα αποτελέσματα των απεικονίσεων στις συγκρινόμενες αρχιτεκτονικές. Για κάθε απεικόνιση ενός αλγορίθμου στην προτεινόμενη αρχιτεκτονική δίνονται οι γράφοι των στοιχειωδών λειτουργιών που επιτελούν οι UDMA σε κάθε κύκλο εκτέλεσης και στο τέλος παρουσιάζονται τα συγκριτικά αποτελέσματα στην ταχύτητα εκτέλεσης έκαστης απεικόνισης, την επιφάνεια πυριτίου και την ενεργειακή κατανάλωση. Όλες οι μετρήσεις πραγματοποιήθηκαν με τη βοήθεια του εργαλείου Synopsys.

5.2 Πειραματικά αποτελέσματα για τις μονάδες UDMA- Σύγκριση με Flexible Computational Unit (FCU)

Με το εργαλείο Synopsys μετρήθηκε η καταλαμβανόμενη από τις προτεινόμενες μονάδες UDMA επιφάνεια που αντιστοιχεί σε δεδομένη περίοδο λειτουργίας. Η ελάχιστη περίοδος λειτουργίας για την μονάδα UDMA μετρήθηκε στα $1.17ns$ ενώ για την μονάδα UDMA που χρησιμοποιεί pipeline είναι $0.83ns$. Στους πίνακες 5.1 και 5.2 παρουσιάζονται οι μετρήσεις επιφάνειας και ισχύος για τις δυο μονάδες (συμπεριλαμβάνοντας και τους εσωτερικούς registers), οι οποίες καλύπτουν το εύρος περιόδων λειτουργίας από την ελάχιστη περίοδο λειτουργίας (αφού η σύνθεση με το Synopsys για μικρότερη περίοδο οδηγεί σε timing violation) έως την περίοδο για την οποία το κύκλωμα έχει απλουστευτεί και δεν παρουσιάζει πλέον μείωση στην καταλαμβανόμενη επιφάνεια.

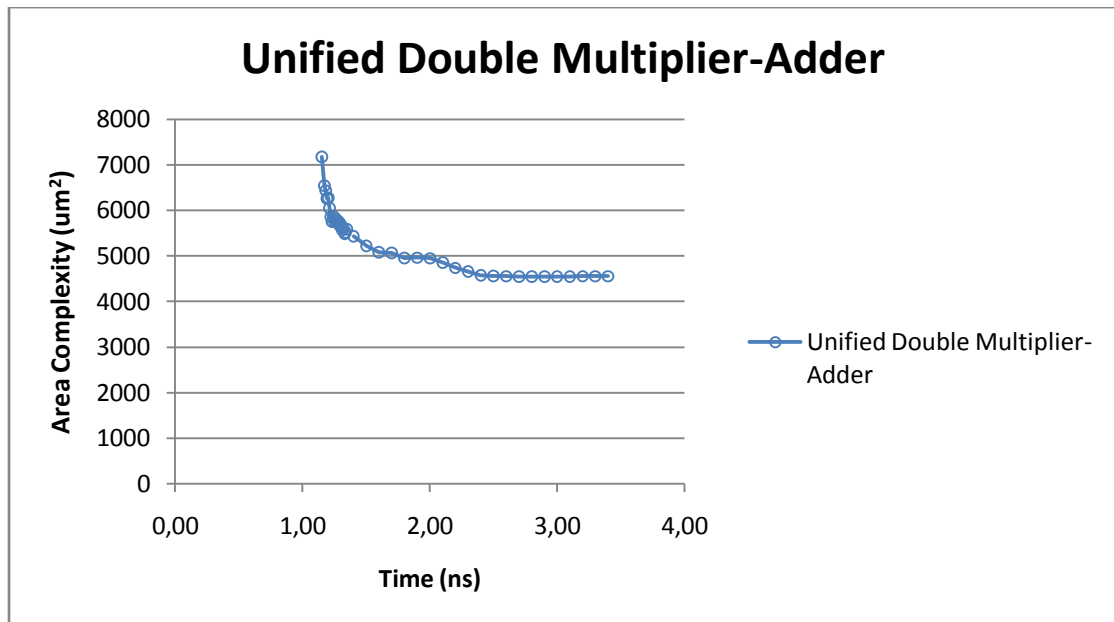
Πίνακας 5.1 Πειραματικά αποτελέσματα επιφάνειας-ισχύος-καθυστέρησης απόκρισης της μονάδας UDMA

<i>T(ns)</i>	<i>Area(um²)</i>	<i>Power(mW)</i>
1.17	6540	12.41
1.18	6435	12.31
1.19	6259	11.79
1.20	6276	11.79
1.21	6052	10.67
1.22	5855	10.47
1.23	5746	10.04
1.24	5865	10.57
1.25	5752	9.98
1.26	5813	9.93
1.27	5742	10.26
1.28	5759	10.66
1.29	5672	9.82
1.30	5695	9.44
1.31	5563	9.64
1.32	5618	9.74
1.33	5482	9.48
1.34	5517	9.33
1.35	5583	9.54
1.40	5430	9.05
1.50	5219	8.20
1.60	5077	7.43
1.70	5063	6.97
1.80	4953	6.45
1.90	4960	6.31
2.00	4947	6.06
2.10	4853	5.65
2.20	4736	6.07
2.30	4655	6.00
2.40	4576	5.74
2.50	4564	5.60
2.60	4556	5.40
2.70	4549	5.27

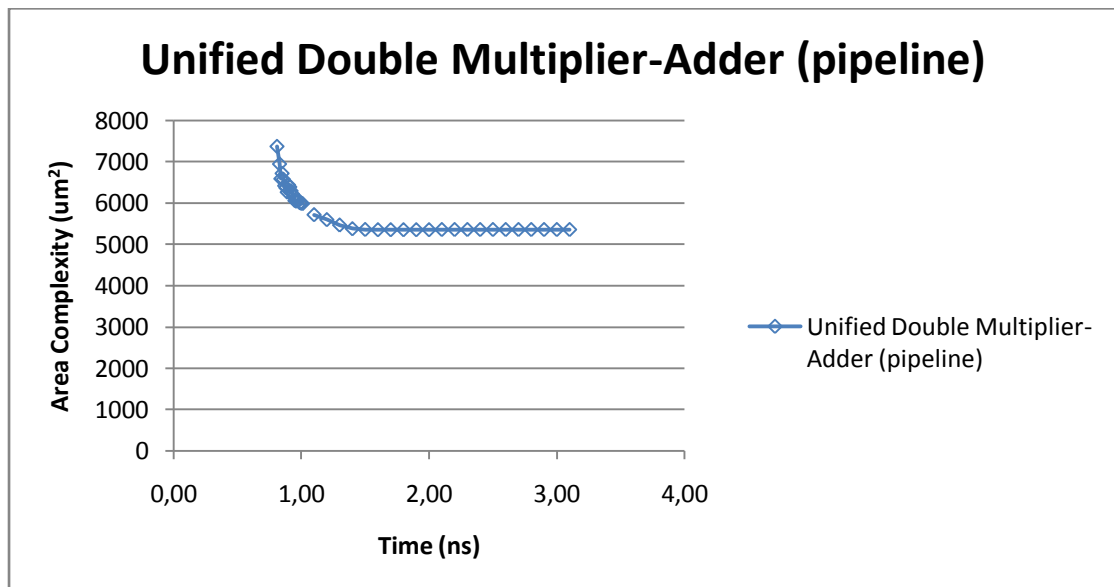
Πίνακας 5.2 Πειραματικά αποτελέσματα επιφάνειας-ισχύος-καθυστέρησης απόκρισης της μονάδας UDMA pipeline

<i>T(ns)</i>	<i>Area(um²)</i>	<i>Power(mW)</i>
0.83	6944	15.77
0.84	6587	15.15
0.85	6716	14.94
0.86	6572	14.43
0.87	6417	13.91
0.88	6367	13.50
0.89	6261	13.19
0.90	6436	13.70
0.91	6387	13.80
0.92	6292	12.87
0.93	6220	12.67
0.94	6157	12.26
0.95	6057	11.95
0.96	6040	11.84
0.97	6048	11.74
0.98	6061	11.73
0.99	6002	11.73
1.00	5991	11.53
1.01	5984	11.42
1.10	5715	10.39
1.20	5595	9.61
1.30	5465	8.95
1.40	5375	8.75
1.50	5357	8.24
1.60	5351	7.74

Τα αποτελέσματα επιφάνειας-καθυστέρησης για κάθε μονάδα φαίνονται στη χαρακτηριστική καμπύλη στα σχήματα 5.1 και 5.2.

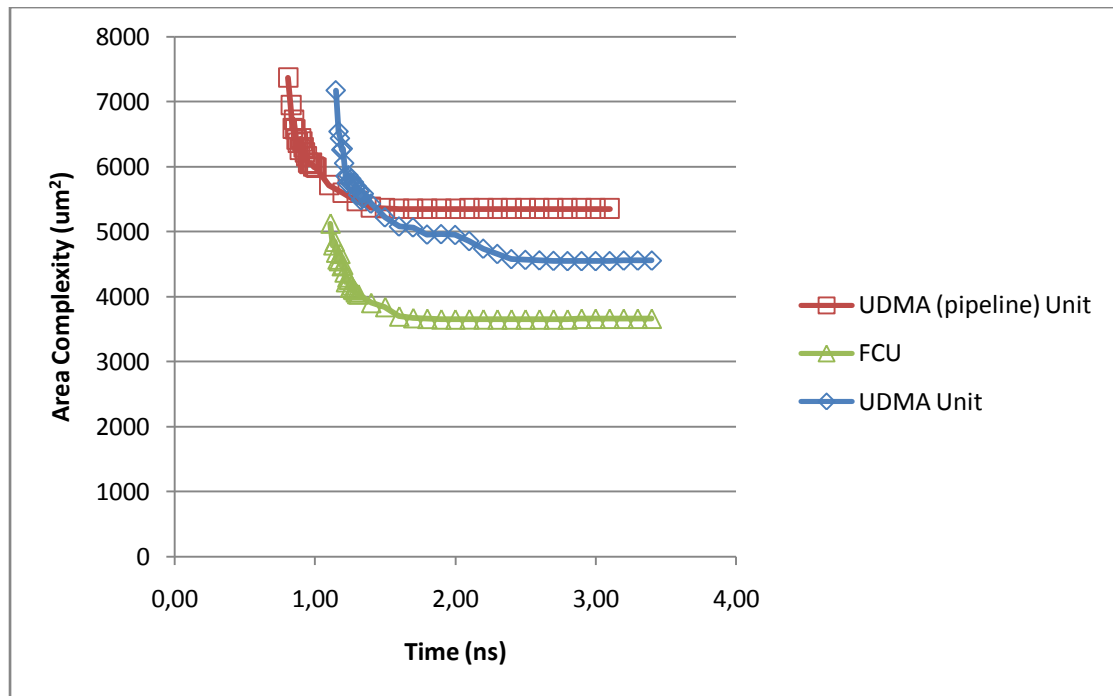


Σχήμα 5.1 Καμπύλη Επιφάνειας-Καθυστέρησης απόκρισης της μονάδας UDMA



Σχήμα 5.2 Καμπύλη Επιφάνειας-Καθυστέρησης απόκρισης της μονάδας UDMA pipeline

Η αντίστοιχη καμπύλη σε σύγκριση με την μονάδα FCU φαίνεται στο σχήμα 5.3, όπου διαφαίνεται ήδη το κέρδος που μπορεί να έχει μια UDMA-based αρχιτεκτονική σε αυτόν τον τομέα αφού η επιφάνεια που καταλαμβάνει το κύκλωμα είναι αρκετά μικρότερη από το διπλάσιο της αντίστοιχης επιφάνειας της FCU, ενώ όμως δύναται να εκτελέσει τους διπλάσιους πολλαπλασιασμούς/προσθέσεις.



Σχήμα 5.3 Συγκριτική καμπύλη Επιφάνειας-Καθυστέρησης απόκρισης

Η σύγκριση της ενεργειακής απόδοσης των μονάδων UDMA με την FCU γίνεται με βάση την μετρική MOPS/W, που ορίζεται:

$$MOPS/W \text{ (Mega Operations Per Second/Watt)} = \left(\frac{\#ops}{\#cycles} \times ClkFreq \right) / P$$

όπου:

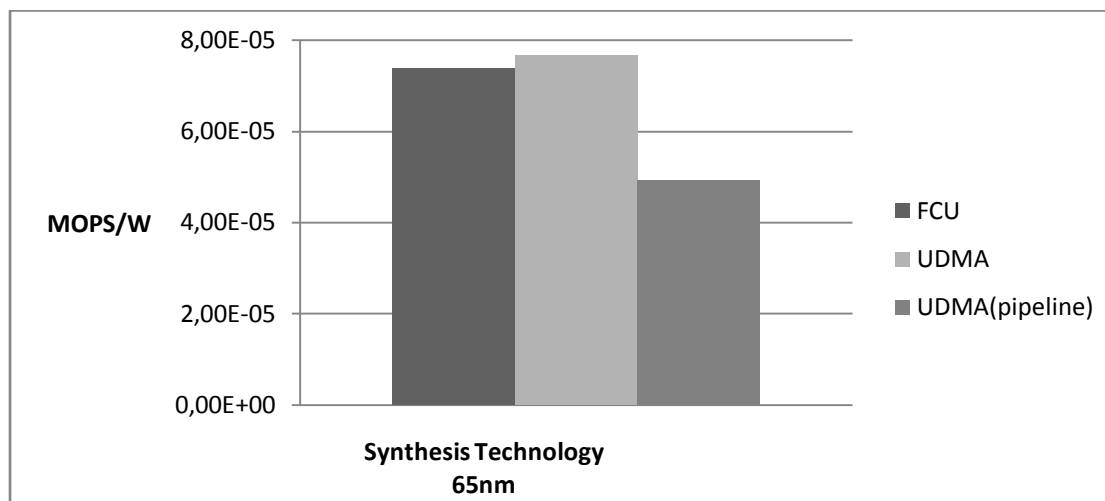
$\frac{\#ops}{\#cycles}$ είναι το μέγιστο πλήθος των λειτουργιών που μπορεί να εκτελέσει η κάθε μονάδα ανά κύκλο. Για τις μονάδες UDMA, όπως φαίνεται και στο σχήμα 3.1, έχουμε 7 λειτουργίες (UDMA) και 3.5 λειτουργίες (UDMA με pipeline) αντίστοιχα ανά κύκλο, δηλαδή $\left(\frac{\#ops}{\#cycles}\right)_{UDMA} = 7$ και $\left(\frac{\#ops}{\#cycles}\right)_{UDMA,pipeline} = \frac{7}{2}$.

Η συχνότητα του ρολογιού ($ClkFreq$) είναι η μέγιστη δυνατή, δηλαδή $(ClkFreq)_{UDMA} = 1/1.17$ και $(ClkFreq)_{UDMA,pipeline} = 1/0.83$.

Οι τιμές της ισχύος P , εξήχθησαν με τη χρήση του Synopsys PrimeTime-PX για 2^{16} τυχαίες εισόδους και ενεργοποίηση της ρύθμισης υπολογισμού μέσου όρου ισχύος (average calculation mode).

Τα αποτελέσματα φαίνονται στο σχήμα 5.4 όπου παρατηρούμε πλέον πως με βάση αυτή την μετρική η μονάδα UDMA εμφανίζει την καλύτερη απόδοση, με μικρή διαφορά, έναντι της FCU. Αναδεικνύεται η αποδοτικότητα της ενσωμάτωσης περισσότερων πράξεων ανά μονάδα χρόνου στη σχεδίαση, σε σχέση με την FCU. Αναμενόμενα, η μονάδα που χρησιμοποιεί pipeline παρουσιάζει τα χειρότερα αποτελέσματα από τις τρεις. Σημειώνεται ωστόσο πως είναι αποδοτικότερη ως προς

αυτή την μετρική συγκριτικά με άλλες μονάδες που έχουν προταθεί στο παρελθόν όπως η FCC και η RAU.



Σχήμα 5.4 Σύγκριση των μονάδων UDMA και FCU με βάση την μετρική MOPS/W

5.3 Πειραματικά αποτελέσματα απεικόνισης DSP αλγορίθμων- Σύγκριση με Flexible Accelerator Architecture

Στην παράγραφο αυτή παρουσιάζονται αρχικά οι αλγόριθμοι που απεικονίστηκαν στην προτεινόμενη αρχιτεκτονική και στη συνέχεια δίνονται σχηματικά (με βάση τους γράφους του Κεφ. 3.6) οι λειτουργίες που εκτελούν οι μονάδες σε κάθε κύκλο κατά την απεικόνιση κάθε αλγόριθμου. Τέλος γίνεται η σύγκριση με την αντίστοιχη απεικόνιση κάθε αλγόριθμου στην FCU based αρχιτεκτονική.

Οι αλγόριθμοι που απεικονίστηκαν είναι οι εξής:

- Finite Impulse Response Filter 16-tap. Πρόκειται για ένα πεπερασμένο ψηφιακό φίλτρο 16 βημάτων, κάθε ένα από τα οποία αντιστοιχεί σε ένα συντελεστή. Οι 16 συντελεστές του αλγορίθμου πολλαπλασιάζονται με μια διαφορετική είσοδο ο καθένας και τα γινόμενα προστίθενται για να δώσουν το τελικό αποτέλεσμα εξόδου.
- Non linear Volterra Infinite Impulse Response (IIR) Filter. Ο αλγόριθμος αυτός είναι ένα μη πεπερασμένο ψηφιακό φίλτρο. Αποτελείται από μια αλληλουχία προσθέσεων και πολλαπλασιασμών. Το αποτέλεσμα εξόδου που

προκύπτει από μια εκτέλεση του αλγορίθμου χρησιμοποιείται ως είσοδος στην επόμενη εκτέλεση, επηρεάζοντας έτσι το επόμενο αποτέλεσμα εξόδου.

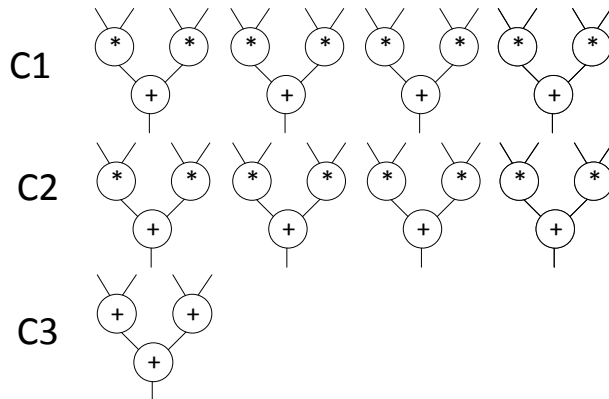
- Elliptic Filter 6th Grade. Ανήκει σε μια ιδιαίτερη κατηγορία αλγορίθμων ΨΕΣ, τα ελλειπτικά φίλτρα. Σε κάθε εκτέλεση, μια είσοδος υπεισέρχεται σε μια αλληλουχία προσθέσεων και πολλαπλασιασμών με τους συντελεστές του φίλτρου διαμορφώνοντας παράλληλα τόσο την τελική έξοδο όσο και τους νέους συντελεστές. Έτσι, κάθε νέα εκτέλεση του αλγορίθμου πραγματοποιείται με διαφορετικούς συντελεστές.
- 1D Unrolled Discrete Cosine Transformation (UDCT). Ο Μονοδιάστατος Διακριτός Μετασχηματισμός Συνημιτόνου εκτελεί το ένα από τα 2 σκέλη ενός 2D DCT, που είναι ο πολλαπλασιασμός ενός πίνακα δεδομένων εισόδου 8×8 με ένα πίνακα συντελεστών $\cos 8 \times 8$.
- Inverse Discrete Cosine Transformation (IDCT) in MPEG. Ο αλγόριθμος αυτός αποτελεί τον κύριο υπολογιστικό πυρήνα (kernel) του αντίστροφου DCT σε μια MPEG κωδικοποίηση.
- 2D Discrete Cosine Transformation (DCT) in JPEG. Ο αλγόριθμος αυτός αποτελεί τον κύριο υπολογιστικό πυρήνα του DCT σε μια JPEG κωδικοποίηση.

Οι απεικονίσεις των 6 αυτών αλγορίθμων στις προτεινόμενες αρχιτεκτονικές περιγράφηκαν σε γλώσσα Verilog και στη συνέχεια συντέθηκαν με το εργαλείο Synopsys. Για τη σύνθεση χρησιμοποιήθηκαν οι ακόλουθες ρυθμίσεις:

- Τεχνολογία TSMC στα 65nm
- Μέγιστη προσπάθεια βελτιστοποίησης της απεικόνισης, χωρίς να καταρρίπτεται η ιεραρχία (compile ultra –no autoungroup)
- Σύνθεση των απεικονίσεων για χρόνους μεγαλύτερους κατά 0.5ns συγκριτικά με το critical path της κάθε μονάδας, ώστε να μην υπάρχει timing violation λόγω των καθυστερήσεων που εισάγουν οι registers και το δίκτυο διασύνδεσης (interconnection network της απεικόνισης)
- Enclosed τύπος καλωδίωσης
- Διατήρηση του αρχικού design για κάθε υπομονάδα που χρησιμοποιείται από την αρχιτεκτονική
- Αποτροπή της διάδοσης των σταθερών τιμών (constant propagation)/ χρήση καταχωρητών για την αποθήκευσή τους
- Αποτροπή συγχώνευσης καταχωρητών (register merging)
- Τυπικές συνθήκες λειτουργίας (typical operating conditions)

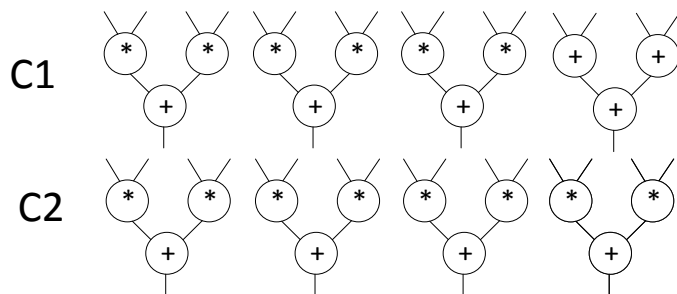
UDMA-based architecture: Γράφοι Ροής Δεδομένων

FIR 16-tap



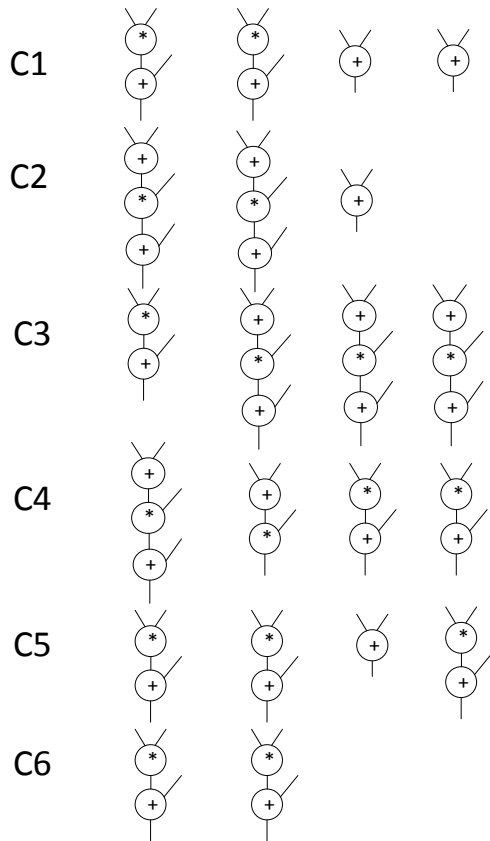
Κάθε εκτέλεση του αλγορίθμου FIR16 διαρκεί συνολικά 4 κύκλους (+1 κύκλος αρχικοποιήσεων). Στον γράφο φαίνονται οι λειτουργίες που εκτελούν οι UDMA σε κάθε κύκλο εκτέλεσης.

Volterra



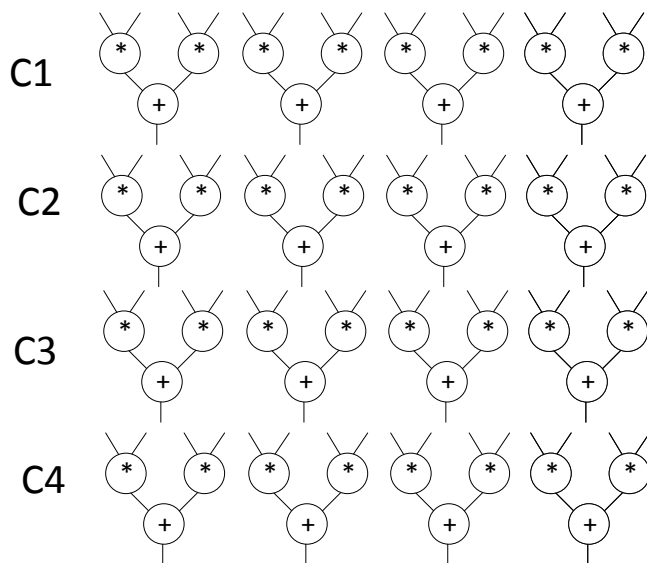
Κάθε εκτέλεση του αλγορίθμου Volterra διαρκεί συνολικά 3 κύκλους (+1 κύκλος αρχικοποιήσεων).

Elliptic



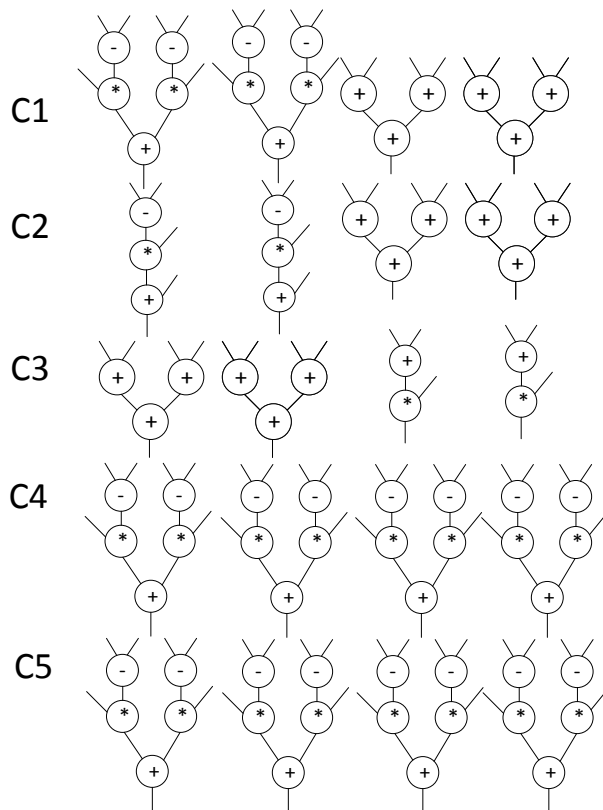
Κάθε εκτέλεση του αλγορίθμου Elliptic διαρκεί συνολικά 6 κύκλους.

UDCT



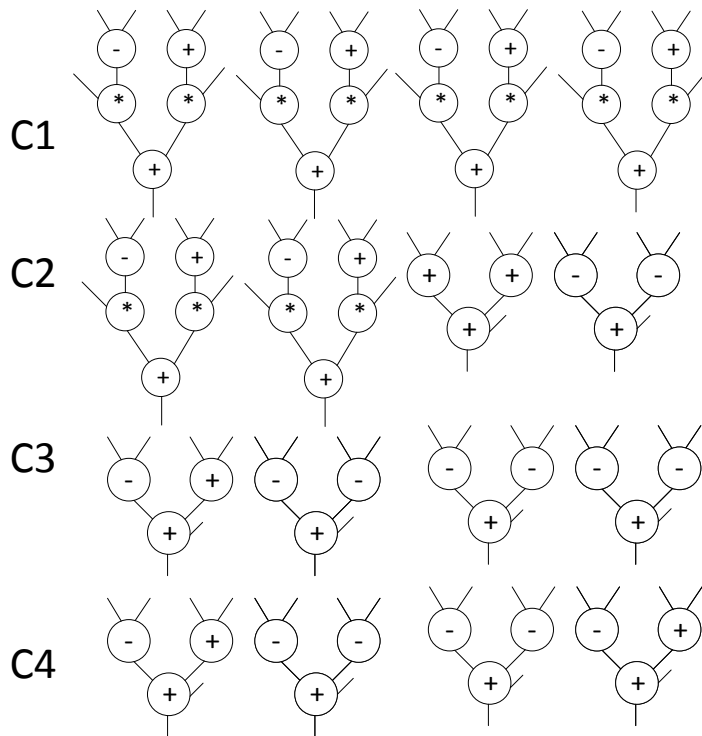
Κάθε εκτέλεση του αλγορίθμου UDCT διαρκεί συνολικά 68 κύκλους. Η αλληλουχία που φαίνεται στον γράφο εκτελείται συνολικά 16 φορές, ενώ απαιτούνται και 4 κύκλοι για την εξαγωγή του τελικού αποτελέσματος.

JPEG DCT



Κάθε εκτέλεση του αλγορίθμου JPEG διαρκεί συνολικά 82 κύκλους. Η αλληλουχία που φαίνεται στον γράφο εκτελείται συνολικά 16 φορές, ενώ απαιτούνται και 2 κύκλοι για την εξαγωγή του τελικού αποτελέσματος.

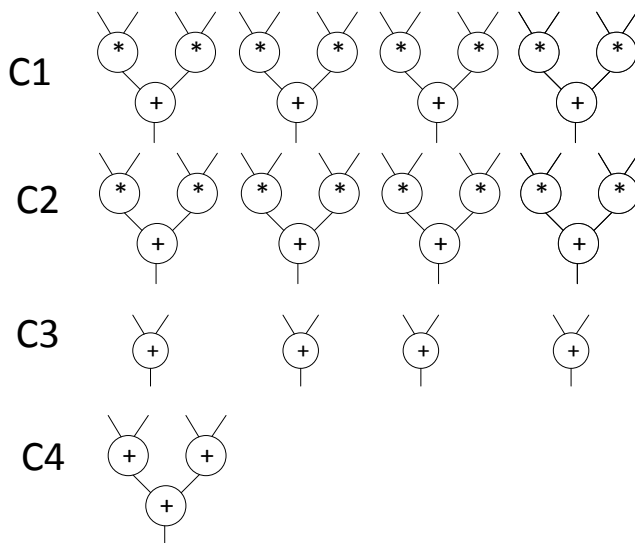
MPEG IDCT



Κάθε εκτέλεση του αλγορίθμου MPEG διαρκεί συνολικά 66 κύκλους. Η αλληλουχία που φαίνεται στον γράφο εκτελείται συνολικά 16 φορές, ενώ απαιτούνται και 2 κύκλοι για την εξαγωγή του τελικού αποτελέσματος.

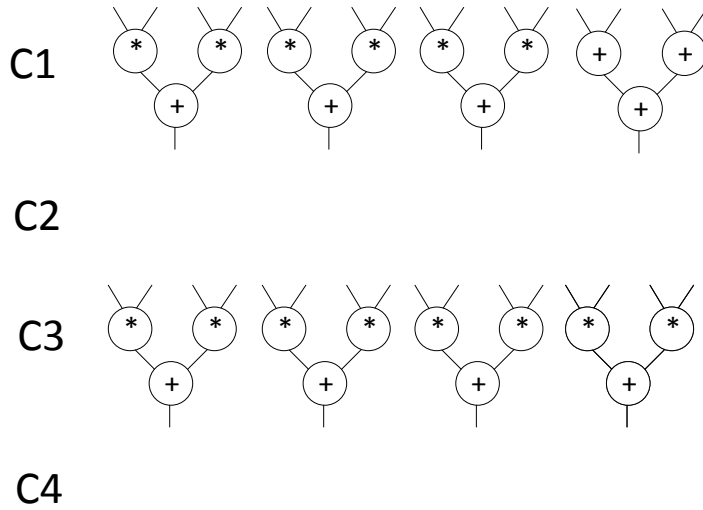
UDMA pipeline-based architecture: Γράφοι Ροής δεδομένων

FIR 16-tap



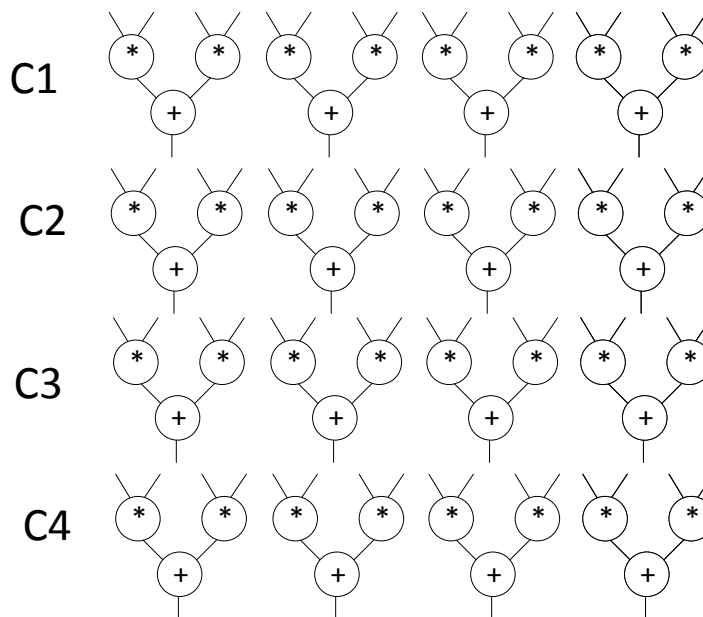
Κάθε εκτέλεση του αλγορίθμου FIR16 διαρκεί συνολικά 4 κύκλους. Στον γράφο φαίνονται οι λειτουργίες που εκτελούν οι UDMA σε κάθε κύκλο εκτέλεσης.

Volterra



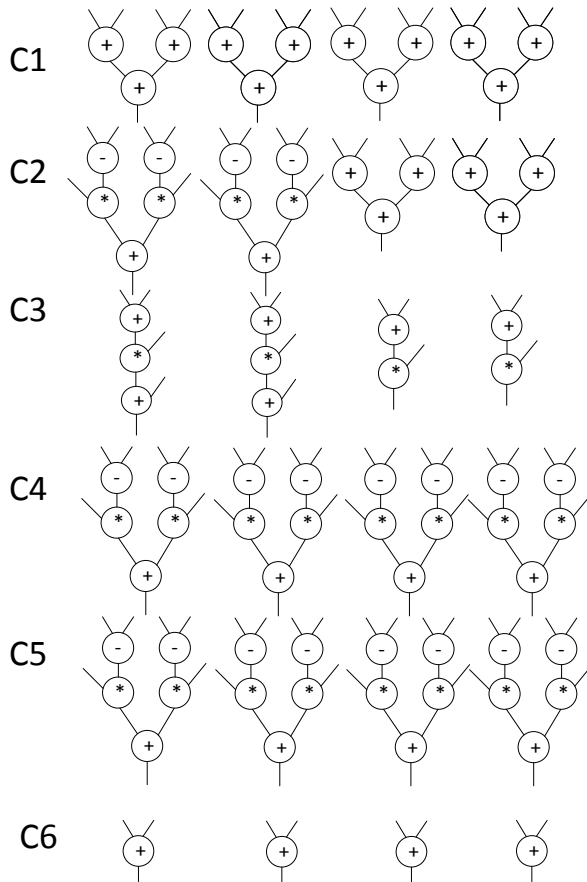
Κάθε εκτέλεση του αλγορίθμου Volterra διαρκεί συνολικά 5 κύκλους (+1 κύκλος αρχικοποιήσεων). Στον γράφο φαίνονται οι λειτουργίες που εκτελούν οι UDMA σε κάθε κύκλο εκτέλεσης.

UDCT



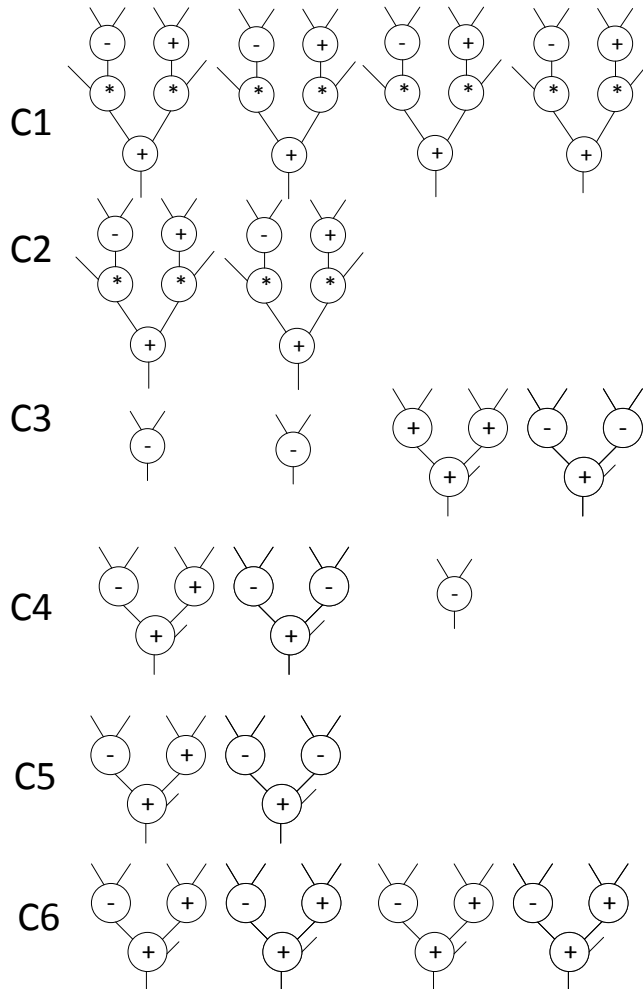
Κάθε εκτέλεση του αλγορίθμου UDCT διαρκεί συνολικά 69 κύκλους. Η αλληλουχία που φαίνεται στον γράφο εκτελείται συνολικά 16 φορές, ενώ απαιτούνται και 5 κύκλοι για την εξαγωγή του τελικού αποτελέσματος.

JPEG DCT



Κάθε εκτέλεση του αλγορίθμου JPEG διαρκεί συνολικά 98 κύκλους. Η αλληλουχία που φαίνεται στον γράφο εκτελείται συνολικά 16 φορές, ενώ απαιτούνται και 2 κύκλοι για την εξαγωγή του τελικού αποτελέσματος.

MPEG IDCT



Κάθε εκτέλεση του αλγορίθμου MPEG διαρκεί συνολικά 98 κύκλους. Η αλληλουχία που φαίνεται στον γράφο εκτελείται συνολικά 16 φορές, ενώ απαιτούνται και 2 κύκλοι για την εξαγωγή του τελικού αποτελέσματος.

Στον πίνακα 5.3 φαίνονται τα συγκριτικά αποτελέσματα μεταξύ των αρχιτεκτονικών, όσον αφορά την συνολική καθυστέρηση, την καταλαμβανόμενη επιφάνεια και την κατανάλωση ισχύος. Η συνολική καθυστέρηση λαμβάνεται ως ο συνολικός αριθμός κύκλων εκτέλεσης που απαιτούνται επί την περίοδο του ρολογιού (clock period), για κάθε αρχιτεκτονική.

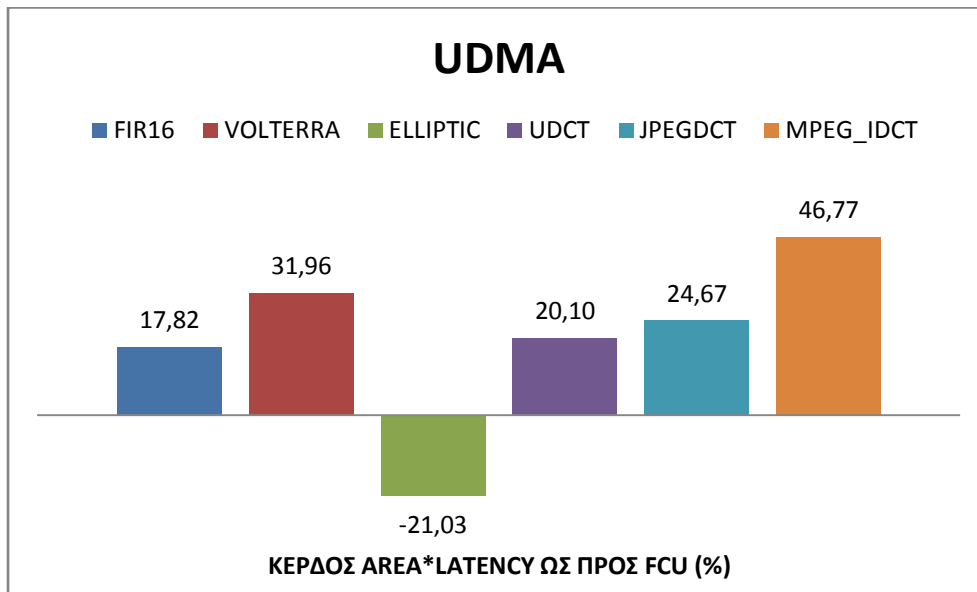
Πίνακας 5.3 Καθυστέρηση απόκρισης, Επιφάνεια και Ενέργεια κατά την απεικόνιση των DSP αλγορίθμων στις μονάδες UDMA και FCU

Kernel	Proposed UDMA				Proposed UDMA (pipeline)				FCU			
	<i>Cycles</i>	<i>Latency</i>	<i>Area</i>	<i>Energy</i>	<i>Cycles</i>	<i>Latency</i>	<i>Area</i>	<i>Energy</i>	<i>Cycles</i>	<i>Latency</i>	<i>Area</i>	<i>Energy</i>
FIR16	4	6.8	24539.4	100.64	4	5.2	25845.12	133.64	6	9.6	21150	182.4
VOLTERRA	3	5.1	20689.56	102	5	6.5	21894.84	128.7	5	8	19385	144
ELLIPTIC	6	10.2	24646.68	163.2	-	-	-	-	6	9.6	21636	198.72
UDCT	68	115.6	25573.32	2520.08	69	89.7	26887.32	2646.15	133	212.8	17387	3553.76
JPEGDCT	82	139.4	48477.96	4934.76	98	127.4	48758.4	5554.64	131	209.6	42803	5240
MPEG_IDCT	66	112.2	42552.72	2344.98	98	127.4	42905.52	3783.78	135	216	41525	3218.4

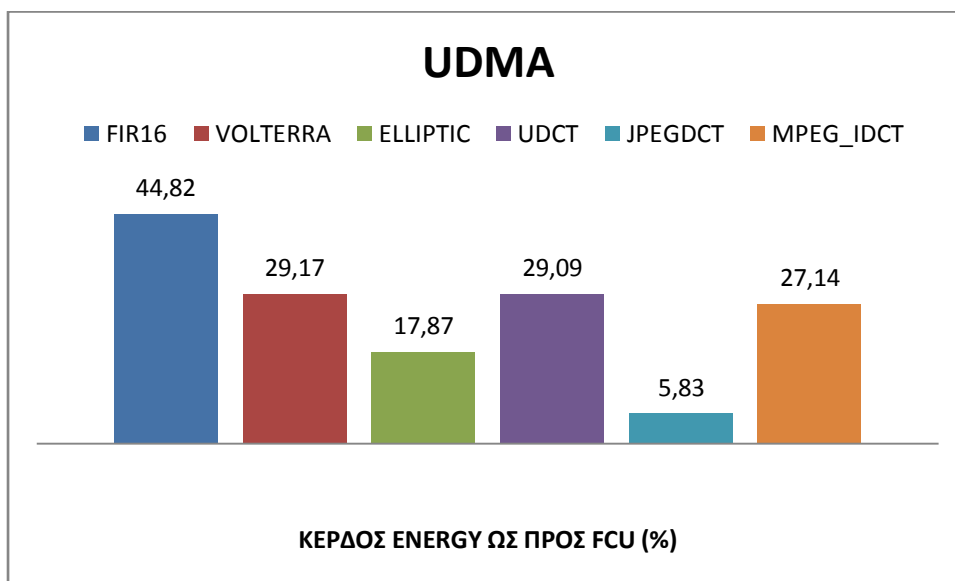
Στον πίνακα 5.4 παρουσιάζονται τα ποσοστιαία κέρδη σχετικά με την γινόμενο επιφάνειας-καθυστέρησης ($A \times L$) και την καταναλισκόμενη ενέργεια (E), όπου δίνεται μια πλέον ξεκάθαρη εικόνα των εμφανώς καλύτερων συνολικά αποτελεσμάτων που παρουσιάζει η παρούσα προσέγγιση για την επιτάχυνση DSP αλγορίθμων, των σημείων όπου υστερεί, αλλά και των διαφορών μεταξύ των δυο UDMA-based αρχιτεκτονικών, οι οποίες εξηγούνται στη συνέχεια. Στα σχήματα 5.5-5.8 φαίνονται τα αντίστοιχα συγκριτικά διαγράμματα.

Πίνακας 5.4 Κέρδη των προτεινόμενων μονάδων έναντι της FCU

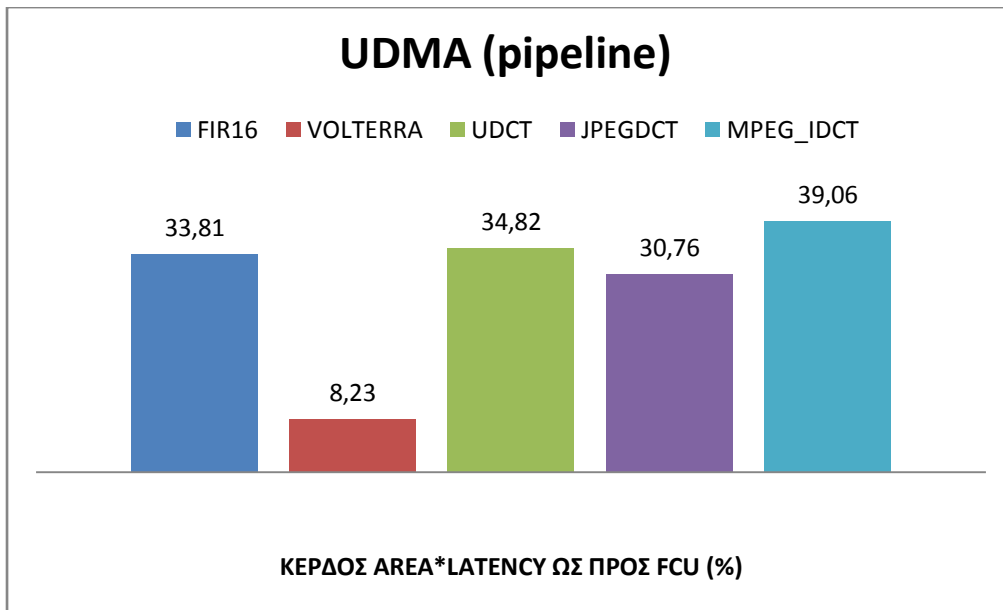
Kernel	Proposed UDMA		Proposed UDMA (pipeline)	
	<i>A*L Gain over FCU(%)</i>	<i>E Gain over FCU(%)</i>	<i>A*L Gain over FCU(%)</i>	<i>E Gain over FCU(%)</i>
FIR16	17.82	44.82	33.81	26.73
VOLTERRA	31.96	29.17	8.23	10.63
ELLIPTIC	-21.03	17.87	-	-
UDCT	20.10	29.09	34.82	25.54
JPEGDCT	24.67	5.83	30.76	-6.00
MPEG_IDCT	46.77	27.14	39.06	-17.57



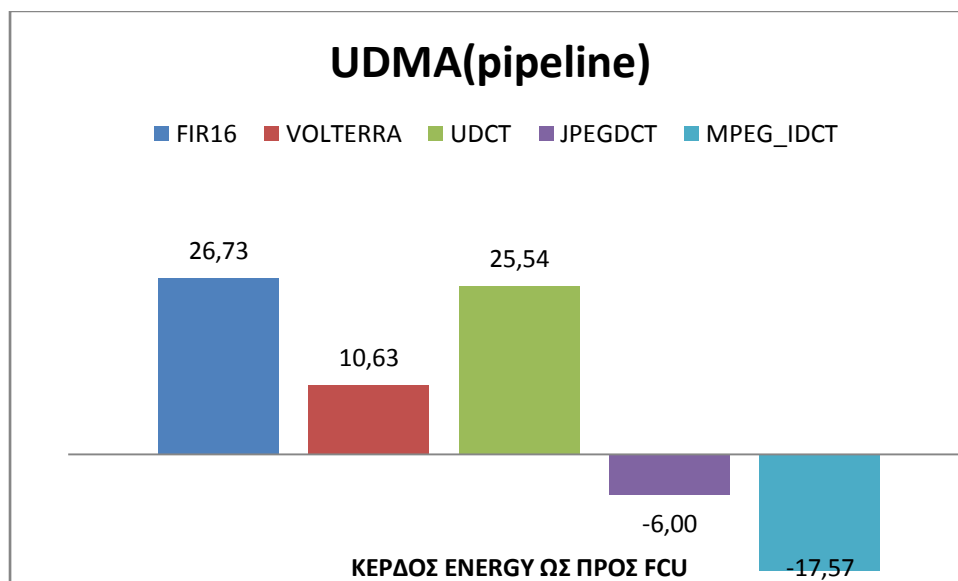
Σχήμα 5.5 Διάγραμμα κέρδους $A \times L$ της μονάδας UDMA ως προς FCU



Σχήμα 5.6 Διάγραμμα κέρδους Energy της μονάδας UDMA ως προς FCU



Σχήμα 5.7 Διάγραμμα κέρδους $A \times L$ της μονάδας UDMA pipeline ως προς FCU



Σχήμα 5.8 Διάγραμμα κέρδους Energy της μονάδας UDMA pipeline ως προς FCU

6

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ

6.1 Σύνοψη και συμπεράσματα

Συνοψίζοντας την μελέτη της παρούσας διπλωματικής εργασίας, στόχος ήταν η παρουσίαση δυο νέων αρχιτεκτονικών που στοχεύουν στην επιτάχυνση DSP αλγορίθμων. Για το σκοπό αυτό, χρησιμοποιούν αντιστοίχως δυο ειδικά σχεδιασμένα αριθμητικά κυκλώματα, τις μονάδες Unified Double Multiplier-Adder και Unified Double Multiplier-Adder με pipeline, συστατικά τα οποία τις διαφοροποιούν από τις έως τώρα προταθείσες αρχιτεκτονικές. Προτεραιότητα της σχεδίασης, πέρα από την ελαχιστοποίηση της καθυστέρησης απόκρισης, αποτέλεσε η επίτευξη της ελάχιστης δυνατής καταλαμβανόμενης επιφάνειας και κατανάλωσης ισχύος από το κύκλωμα. Στις αρχιτεκτονικές αυτές απεικονίστηκε ένα πλήθος αλγορίθμων (DSP kernels) με περιγραφή σε γλώσσα Verilog, αφού πρώτα οι υπολογιστικοί βρόχοι των αλγορίθμων είχαν αναλυθεί από υψηλό επίπεδο.

Κεντρική ιδέα στις προτεινόμενες μονάδες που απαρτίζουν την αρχιτεκτονική, αποτέλεσε η αποδοτική υλοποίηση ενοποιημένου διπλού πολλαπλασιαστή-αθροιστή, αφού οι διαδοχικοί πολλαπλασιασμοί και προσθαφαιρέσεις είναι οι πιο συχνά εμφανιζόμενες ακολουθίες στους DSP αλγορίθμους και άλλες υπολογιστικά βαριές εφαρμογές, καθώς και η διερεύνηση του πως η χρήση διοχέτευσης μπορεί να επιταχύνει περαιτέρω την διαδικασία εκτέλεσης των αλγορίθμων.

Τα τελικά κυκλώματα συντέθηκαν με το εργαλείο Synopsys για την εξαγωγή των πειραματικών αποτελεσμάτων, και η σύγκριση πραγματοποιήθηκε σε σχέση με την FCU-based αρχιτεκτονική.

Τα συμπεράσματα που εξήχθησαν από την μελέτη των συγκριτικών αποτελεσμάτων είναι τα εξής:

- Αμφότερες οι UDMA-based αρχιτεκτονικές υπερτερούν σαφώς έναντι της αρχιτεκτονικής που χρησιμοποιεί την FCU ως κύριο επεξεργαστικό στοιχείο. Ο βασικός λόγος για τα σημεία όπου εμφανίζονται κέρδη, είναι ότι η UDMA παρόλο που είναι μια πιο σύνθετη και ενεργοβόρα μονάδα σε σχέση με την FCU, εμφανίζει μόνο 37% αύξηση στην καταλαμβανόμενη επιφάνεια και 22.8% αύξηση στην κατανάλωση ισχύος, ενώ κατορθώνει να επιταχύνει την εκτέλεση των αλγορίθμων σε μεγαλύτερο ποσοστό (εκτός του elliptic filter). Αυτό επιτυγχάνεται λόγω του ενοποιημένου διπλού πολλαπλασιαστή κάθε μονάδας και της αποδοτικής απεικόνισης. Αυτό φαίνεται και σε επίπεδο μονάδας όπου με βάση την μετρική MOPS/W η μονάδα UDMA εμφανίζει καλύτερη απόδοση. Συγκεκριμένα, η ποσοστιαία ελάττωση σε κύκλους εκτέλεσης και συνολική καθυστέρηση για την μονάδα UDMA είναι:

Πίνακας 6.1 Κέρδος της UDMA σε κύκλους εκτέλεσης και συνολική καθυστέρηση

Kernel	Cycles(%)	Latency(%)
Fir16	-33.3	-29.2
Volterra	-40	-36.3
Elliptic	0	+6.2
Udct	-48.8	-45.7
Jpeg	-37.4	-33.5
Mpeg	-51.1	-48

Και για την μονάδα που χρησιμοποιεί pipeline, όπου κάθε κύκλος εκτέλεσης έχει πολύ μικρότερη κρίσιμη καθυστέρηση:

Πίνακας 6.2 Κέρδος της UDMA pipeline σε κύκλους εκτέλεσης και συνολική καθυστέρηση

Kernel	Cycles(%)	Latency(%)
Fir16	-33.3	-45.8
Volterra	0	-18.8
Udct	-48.1	-57.8
Jpeg	-25.2	-39.2
Mpeg	-27.4	-41

- Στην περίπτωση του elliptic filter δεν μπορούμε να επιτύχουμε καλύτερο latency, γιατί οι πολλαπλασιασμοί εκτελούνται πάντα σε αλυσίδα οπότε δεν μπορεί να βελτιστοποιηθεί σε σχέση με την FCU. Κατά την απεικόνιση του elliptic filter απαιτείται η χρήση μόνο ενός από τους πολλαπλασιαστές κάθε UDMA, οπότε σε αυτή την περίπτωση η μονάδα κρίνεται μη αποδοτική, τουλάχιστον όσον αφορά τη μετρική $area \times latency$.

- Λόγω της μεγάλης μείωσης στην καθυστέρηση εκτέλεσης των αλγορίθμων (latency), εμφανίζονται αναμενόμενα κέρδη τόσο στη μετρική $area \times latency$ όσο και $energy (= power \times execution\ latency)$, που αποτυπώνουν τις επιδόσεις και την αποδοτικότητα ενός συστήματος.
- Ωστόσο εμφανίζεται κέρδος στην ενεργειακή κατανάλωση και έναντι του elliptic filter. Αυτό οφείλεται στο ότι με την χρήση δεδομένων (είσοδοι-έξοδοι) σε binary μορφή χρειαζόμαστε λιγότερες πράξεις (CS προσθέσεις) και μικρότερο register file για την αποθήκευση των δεδομένων. Αυτό συμβάλλει και στην αύξηση του αναμενόμενου κέρδους σε $area/energy$ και για τους άλλους αλγόριθμους, όπως φαίνεται στους πίνακες 5.3 και 5.4.
- Η αρχιτεκτονική που βασίζεται στην UDMA που χρησιμοποιεί διοχέτευση, εμφανίζει για τους ίδιους λόγους κέρδη στις περισσότερες περιπτώσεις έναντι της FCU, που είναι όμως μικρότερα, όσον αφορά την ενεργειακή κατανάλωση, λόγω των επιπλέον τεσσάρων 20-bit registers που χρησιμοποιεί η βασική μονάδα. Άρα η UDMA-based αρχιτεκτονική κρίνεται συνολικά αποδοτικότερη. Οι αλγόριθμοι Jpeg και Mpeg, όπου οι μονάδες χρησιμοποιούνται πλήρως και το register file είναι μεγάλο, αποτελούν τις μοναδικές περιπτώσεις που η αρχιτεκτονική που βασίζεται στην FCU εμφανίζει καλύτερη ενεργειακή απόδοση.
- Το πλεονέκτημα όμως που προσφέρει είναι η ταχύτερη εκτέλεση των αλγορίθμων που απαιτούν συνεχές accumulation και/η δεν έχουν data dependencies μεταξύ των αποτελεσμάτων κάθε κύκλου. Στην περίπτωση αυτή έχει όπως δείξαμε πολύ μικρότερο latency. Αυτό είναι εμφανές στα Fir16, Udct, Jpeg όπου το κέρδος σε latency και $A * L$ είναι μεγαλύτερο από την UDMA-based αρχιτεκτονική ενώ υστερεί στα Volterra και Mpeg που έχουμε data dependencies, δηλαδή υπάρχει πάντα κάποιος πολλαπλασιασμός που αναμένει το αποτέλεσμα μιας προηγούμενης εκτέλεσης. Συνεπώς, ανάλογα με τον αλγόριθμο και τις απαιτήσεις του συστήματος για ενεργειακή κατανάλωση, η μονάδα με διοχέτευση μπορεί να αποτελέσει καλύτερη λύση ως τμήμα ενός accelerator.
- Η αρχιτεκτονική αυτή δεν προσφέρεται καθόλου για αλγόριθμους όπου οι πολλαπλασιασμοί εκτελούνται σε αλυσίδα όπως στο elliptic filter αφού θα χρειαζόμαστε πάντα τους διπλάσιους κύκλους εκτέλεσης και ενδιάμεσα οι μονάδες θα μένουν συχνά αχρησιμοποίητες. Για αυτό το λόγο δεν πραγματοποιήθηκε πειραματική αξιολόγηση για τον αλγόριθμο αυτό.

6.2 Επεκτάσεις

Η μονάδα UDMA που αναπτύχθηκε στα πλαίσια της διπλωματικής εμφανίζει πολύ καλά αποτελέσματα στους 6 αλγόριθμους που μελετήθηκαν. Θα μπορούσε να εξεταστεί η αποδοτικότητα της και σε μια ευρύτερη γκάμα αλγορίθμων ώστε να αξιολογηθεί καλύτερα η υπεροχή της στην απεικόνιση DSP αλγορίθμων, και να

αναζητηθούν ενδεχομένως περιπτώσεις όπου δεν αποδίδει τόσο καλά, όπως είδαμε στην περίπτωση των αλυσιδωτών πολλαπλασιασμών στο ελλειπτικό φίλτρο. Επίσης, ερευνώντας περισσότερους αλγόριθμους θα μπορούσε να επαναξιολογηθεί η αναγκαιότητα της ύπαρξης της δεύτερης εισόδου από το register file (B_2). Η είσοδος αυτή εξασφαλίζει πιο αποδοτική απεικόνιση (και συνεπώς ταχύτερη εκτέλεση) μόνο για τον JPEG υπολογισμό, που είναι αρκετά σημαντικό στην παρούσα μελέτη, ωστόσο ίσως εάν δεν είναι απαραίτητη για περισσότερες περιπτώσεις αλγορίθμων τότε θα μπορούσαμε να επιτύχουμε μικρότερη κρίσιμη καθυστέρηση για την μονάδα αφού η ελάττωση των εισόδων του τελικού δένδρου από 20 σε 19 οδηγεί σε άμεση ελάττωση του βάθους του (πίνακας 3.5).

Εφόσον η βασική ιδέα χρησιμοποίησης δυο πολλαπλασιαστών από την μονάδα αποδεικνύεται κομβική με βάση και τη μορφή που έχουν οι DSP αλγόριθμοι, θα μπορούσε να μελετηθεί κατά πόσο είναι εφικτή η αποδοτική σχεδίαση μιας μονάδας με περισσότερους (τουλάχιστον 4) πολλαπλασιαστές και να διερευνηθούν πειραματικά τα αποτελέσματα της αρχιτεκτονικής που χρησιμοποιεί μια τέτοια μονάδα.

Μπορούν να εξεταστούν εναλλακτικές, πιο ευέλικτες ή επαναδιατάξιμες μορφές σχεδίασης για τη βασική μονάδα της αρχιτεκτονικής. Αυτό θα μπορούσε να αφορά και πλήρη ευελιξία, δηλαδή χωρίς κάποιο περιορισμό για την βασική μονάδα, για παράδειγμα να εξεταστεί μια πλήρως ευέλικτη αρχιτεκτονική η οποία διαθέτει όλα τα απαραίτητα επεξεργαστικά στοιχεία (αθροιστές, πολλαπλασιαστές) και για κάθε αλγόριθμο προς απεικόνιση να ορίζεται κάθε φορά η βέλτιστη διασύνδεση μεταξύ τους. Αν η ελάχιστη κατανάλωση ισχύος είναι το ζητούμενο, μια εναλλακτική low power σχεδίαση θα μπορούσε να αναζητηθεί, πιθανώς εξοικονομώντας ενέργεια σε κύκλους όπου απαιτείται μόνο ο ένας (ή κανένας) πολλαπλασιαστής.

Για την περίπτωση της μονάδας UDMA που χρησιμοποιεί διοχέτευση, θα μπορούσε να αναζητηθεί η ικανότητα καλύτερου 'μοιράσματος' της καθυστέρησης ώστε να αυξηθεί η διεκπεραιωτική ικανότητα (throughput) του κυκλώματος, κάτι που θα οδηγούσε σε βελτίωση σε αμφότερες τις μετρικές $A \times L$ και Energy. Αυτό δεν βελτιστοποιείται περαιτέρω με βάση την παρούσα σχεδίαση, ωστόσο είναι δυνατό να διερευνηθεί πως θα επηρέαζε η τοποθέτηση καταχωρητών ένα επίπεδο πιο πάνω στους δενδρικούς αθροιστές των πολλαπλασιαστών. Η κρίσιμη καθυστέρηση θα γινόταν η ελάχιστη δυνατή, αλλά θα υπήρχε tradeoff με την επιφάνεια και την κατανάλωση ισχύος λόγω της περισσότερης απαιτούμενης χωρητικότητας, ενώ πλέον απαιτείται και η ξεχωριστή κατασκευή των δενδρικών αθροιστών (της DesignWare δεν προσφέρονται για ένα τέτοιο εγχείρημα).

Τέλος, μπορεί να εξεταστεί κατά πόσο είναι αποδοτική η τροποποίηση του κυκλώματος ώστε να παρακάμπτονται οι πολλαπλασιαστές όταν πραγματοποιούνται μόνο προσθέσεις ώστε να μην καθυστερεί η εξαγωγή του αθροίσματος ένα κύκλο λόγω των registers.

7

Βιβλιογραφία

- [1] Neil H. Weste, Kamran Eshraghian, **“Principles of CMOS VLSI Design: A Systems Perspective”**, Second Edition
- [2] Kiamal Pekmestzi, **“DIGITAL VLSI SYSTEMS”**, NTUA Lectures Notes, Athens 2003.
- [3] Kostas Tsoumanis, Sotirios Xydis, Georgios Zervakis, Kiamal Pekmestzi, **“Flexible DSP Accelerator Architecture Exploiting Carry-Save Arithmetic”**
- [4] Y. H. Chen, T. Y. Chang, **“A High-Accuracy Adaptive Conditional-Probability Estimator for Fixed-Width Booth Multipliers”**
- [5] Kostas Tsoumanis, Sotirios Xydis, Constantinos Efstathiou, Nikos Moschopoulos, Kiamal Pekmestzi, **“An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator”**
- [6] Nikolaos Eftaxiopoulos, Georgios Zervakis, Kostas Tsoumanis, Kiamal Pekmestzi, **“High Performance MAC Architectures”**
- [7] Xilinx DSP slices, www.xilinx.com.
- [8] Synopsys Design Compiler, www.synopsys.com.
- [9] ModelSim Corporation, www.mentor.com.
- [10] Synopsys Primetime PX, www.synopsys.com.
- [11] DW01_add, www.synopsys.com/dw/ipdir.php?c=DW01_add .
- [12] DW02_tree, www.synopsys.com/dw/ipdir.php?c=DW02_tree .

