



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**Σχεδίαση και μελέτη κυκλωμάτων σειριακών-  
παράλληλων πολλαπλασιαστών με κωδικοποίηση  
Modified-Booth**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**ΝΙΚΟΛΑΟΥ ΑΠΟΣΤΟΛΟΥ**

**Επιβλέπων :** Κιαμάλ Πεκμεστζή

Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2012





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

**Σχεδίαση και μελέτη κυκλωμάτων σειριακών-  
παράλληλων πολλαπλασιαστών με κωδικοποίηση  
Modified-Booth**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**ΝΙΚΟΛΑΟΥ ΑΠΟΣΤΟΛΟΥ**

**Επιβλέπων :** Κιαμάλ Πεκμεστζή

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 23<sup>η</sup> Ιουλίου 2012.

.....  
Κ. Πεκμεστζή  
Καθηγητής Ε.Μ.Π.

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

.....  
Γ. Οικονομάκος  
Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2012



.....

**ΝΙΚΟΛΑΟΣ ΑΠΟΣΤΟΛΟΥ**

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

Copyright © Αποστόλου Νικόλαος 2012

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

Ο σκοπός της παρούσας διπλωματικής εργασίας ήταν η σχεδίαση διαφόρων τοπολογιών σειριακών-παράλληλων πολλαπλασιαστών και η διερεύνησή τους ως προς τις επιδόσεις τους και ειδικότερα ως προς τις δυνατότητές τους στην επίτευξη λειτουργίας με χαμηλή κατανάλωση.

Υλοποιήθηκαν δύο νέες σχεδιαστικές ιδέες. Η πρώτη αφορά την χρήση μιας γενικευμένης μορφής αναπαράστασης «σωσίματος – κρατουμένου» (carry – save) για τα ενδιάμεσα στάδια των υπολογισμών, με σκοπό την ελάττωση των απαιτούμενων καταχωρητών, ενώ η δεύτερη αφορά την χρήση της κανονικής αναπαράστασης προσημασμένου ψηφίου (canonical signed digit representation) σε συνδυασμό με την κωδικοποίηση Modified-Booth, με σκοπό την παραγωγή όσο το δυνατόν λιγότερων μη μηδενικών μερικών γινομένων.

Σχεδιάστηκαν κυκλώματα πολλαπλασιασμού, τα οποία λαμβάνουν δεδομένα σε μορφή συμπληρώματος του δύο, με την μια είσοδο παράλληλα (πολλαπλασιαστέος) και την άλλη σειριακά (πολλαπλασιαστής), και μετασχηματίζουν την μία είσοδό τους (σειριακή) σύμφωνα με την μέθοδο MacSorley-Booth (Modified Booth) για την ελάττωση των προς άθροιση μερικών γινομένων. Μελετήθηκαν αρχιτεκτονικές με διαφορετικό μήκος ενδιάμεσης λέξης (radix-2 ως και radix-8) σκοπεύοντας πρωτίστως στην ελάττωση της δυναμικής κατανάλωσης. Επίσης, διερευνήθηκαν διαφορετικές κωδικοποιήσεις Modified Booth και οι επιπτώσεις τους στην δυναμική κατανάλωση του κυκλώματος.

Πιο συγκεκριμένα, υλοποιήθηκε σε γλώσσα Verilog μια σειρά σειριακών-παράλληλων πολλαπλασιαστών Modified Booth σε radix-2, radix-4 και radix-8 και σε μήκη λέξεως 8 ως 64 bits. Η λειτουργία τους προσομοιώθηκε με βάση μια standard cell CMOS βιβλιοθήκη 90nm της Artisan. Τα αποτελέσματα καθυστέρησης (critical path), κατανάλωσης και επιφανείας συγκρίθηκαν με έναν απλό σειριακό-παράλληλο πολλαπλασιαστή και με έναν, παρόμοιας λογικής, δενδρικό παράλληλο πολλαπλασιαστή της Designware (tree multiplier).

### Λέξεις Κλειδιά:

Σειριακοί παράλληλοι πολλαπλασιαστές, Modified Booth, Canonic Modified Booth, Verilog, χαμηλή κατανάλωση, CMOS 90nm





## **Abstract**

The scope of this thesis was the design of various serial-parallel multiplier schemes, as well as the investigation of their performance, especially as far as their energy consumption is concerned.

The designs created, were based on two novelties. The first one is the use of a generalized “carry-save” scheme, which is used in the intermediate steps of the calculations in order to reduce the number of registers needed, while the second is the use of the “canonical signed digit representation” applied on the Modified-Booth encoding scheme, in order to minimize the number of the non-zero partial products created.

The designed multiplier schemes receive data in two’s complement form. The multiplier is in parallel series form while the multiplicand is in bit-series form. In all designs the serial input is transformed according to the MacSoreley-Booth method (Modified Booth) in order to reduce the number of partial products which will be added. Various radix schemes were studied (radix-2 to radix-8), targeting mainly to the reduction of their dynamic power consumption.

More specifically, we designed a series of Modified-Booth serial-parallel multipliers using Verilog HDL. The designs include radix-2, radix-4 and radix-8 schemes, while the inputs can be 8, 16, 24, 32, 48, or 64 bits wide. The designs were synthesized and simulated using a 90nm standard cell CMOS library by Artisan. The energy, delay and area results are presented, compared to a simple (radix-1) serial-parallel multiplier as well as a parallel tree-multiplier synthesized using the Designware library.

### **Keywords:**

Serial parallel multipliers, Modified Booth, Canonic Modified Booth, Verilog, low power, CMOS 90nm



## Πίνακας περιεχομένων

<b>1</b>	<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
1.1	Ψηφιακή σχεδίαση χαμηλής κατανάλωσης ( <i>Low power design</i> ) .....	1
1.2	Αντικείμενο διπλωματικής.....	2
1.2.1	Συνεισφορά .....	3
1.3	Οργάνωση κειμένου .....	4
<b>2</b>	<b>ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ</b> .....	<b>5</b>
2.1	Αριθμητικά συστήματα .....	6
2.1.1	Δυαδικό σύστημα.....	6
2.1.2	Αναπαράσταση σε μορφή συμπληρώματος ως προς δύο .....	7
2.1.3	Αναπαράσταση σε μορφή Σωσίματος-Κρατουμένου ( <i>Carry-Save</i> ) .....	11
2.1.4	Αναπαράσταση σε μορφή Προσημασμένου ψηφίου ( <i>Signed Digit</i> ).....	12
2.2	Κωδικοποιήσεις <i>Booth</i> και <i>Modified Booth</i> .....	13
2.2.1	Κωδικοποίηση <i>Booth</i> .....	13
2.2.2	Κωδικοποίηση <i>Modified Booth</i> .....	15
2.3	Είδη αθροιστών και πολλαπλασιαστών .....	18
2.3.1	Δομικά στοιχεία.....	18
2.3.2	Αθροιστές .....	22
2.3.3	Πολλαπλασιαστές .....	25
<b>3</b>	<b>ΣΧΕΔΙΑΣΜΟΣ ΣΕΙΡΙΑΚΟΥ ΚΥΚΛΩΜΑΤΟΣ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΥ ΜΕ ΚΩΔΙΚΟΠΟΙΗΣΗ MODIFIED BOOTH</b> .....	<b>33</b>
3.1	Στόχοι εργασίας .....	33
3.2	Χρήση διπλής διοχέτευσης ( <i>double pipeline</i> ) .....	34
3.3	Κωδικοποίηση <i>Modified Booth</i> .....	37
3.3.1	Κύκλωμα κωδικοποίησης.....	38
3.3.2	Κύκλωμα δημιουργίας μερικών γινομένων.....	39
3.4	Επέκταση προσήμου .....	40
3.5	Κωδικοποίηση <i>Canonic Modified Booth</i> .....	43
3.5.1	Μετατροπή σε κανονική αναπαράσταση .....	44
3.5.2	Μετατροπή σε <i>Canonic Modified Booth</i> αναπαράσταση .....	45
3.5.3	Κυκλωματική υλοποίηση .....	48

3.6	<i>Radix-2 Modified Booth</i> σειριακός πολλαπλασιαστής.....	50
3.6.1	Γενικευμένη <i>Carry-Save</i> αναπαράσταση .....	50
3.6.2	Βασική μονάδα άθροισης .....	51
3.6.3	Λιγότερο σημαντική μονάδα άθροισης.....	52
3.6.4	Χειρισμός διορθωτικών όρων .....	53
3.6.5	Κύκλωμα <i>radix-2 Modified Booth</i> σειριακού πολλαπλασιαστή .....	54
3.7	<i>Radix-4</i> σειριακός πολλαπλασιαστής.....	56
3.7.1	Γενικευμένη <i>Carry-Save</i> αναπαράσταση .....	56
3.7.2	Βασική μονάδα άθροισης .....	57
3.7.3	Χειρισμός διορθωτικών όρων – περισσότερο σημαντικό στάδιο άθροισης.....	58
3.7.4	Λιγότερο σημαντική μονάδα άθροισης.....	59
3.7.5	Κύκλωμα <i>radix-4 Modified Booth</i> σειριακού πολλαπλασιαστή .....	61
3.8	<i>Radix-8 Modified Booth</i> σειριακός πολλαπλασιαστής.....	63
3.8.1	Γενικευμένη <i>Carry-Save</i> αναπαράσταση .....	63
3.8.2	Βασική μονάδα άθροισης .....	64
3.8.3	Χειρισμός διορθωτικών όρων – περισσότερο σημαντικό στάδιο άθροισης.....	65
3.8.4	Λιγότερο σημαντική μονάδα άθροισης .....	66
3.8.5	Κύκλωμα <i>radix-8 Modified Booth</i> σειριακού πολλαπλασιαστή .....	68
<b>4</b>	<b>ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ .....</b>	<b>72</b>
4.1	Οργάνωση πειραμάτων .....	72
4.2	Σειριακός παράλληλος πολλαπλασιαστής με/χωρίς διπλή διοχέτευση και <i>radix-2 Modified Booth</i> σειριακός πολλαπλασιαστής.....	73
4.3	<i>Modified Booth</i> και <i>Canonic Modified Booth</i> σειριακοί και παράλληλοι πολλαπλασιαστές .....	85
<b>5</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ .....</b>	<b>103</b>
5.1	Σύνοψη και συμπεράσματα.....	103
5.2	Μελλοντικές επεκτάσεις .....	106
<b>6</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>108</b>

# 1

## ***ΕΙΣΑΓΩΓΗ***

### ***1.1 Ψηφιακή σχεδίαση χαμηλής κατανάλωσης (Low power design)***

Στο κατώφλι της τρίτης χιλιετίας, η αλματώδης διείσδυση της τεχνολογίας σε όλους τους τομείς της ανθρώπινης καθημερινότητας, είναι γεγονός. Αποτέλεσμα αυτής, είναι η συνεχής εξέλιξη των ψηφιακών συστημάτων ώστε να καλύψουν τις ολοένα αυξανόμενες ανάγκες των εφαρμογών.

Μέχρι πρόσφατα υπήρχε η επιτακτική ανάγκη για συνεχώς μεγαλύτερες συχνότητες λειτουργίας, κάτι που οδήγησε την σχεδίαση των ψηφιακών συστημάτων στην εύρεση αλγορίθμων, τεχνολογιών και τοπολογιών κυκλωμάτων τα οποία θα προσέφεραν την ελάχιστη καθυστέρηση (critical path). Ωστόσο, η ποικιλία των εφαρμογών που αναπτύχθηκαν, καθώς και οι περιορισμοί που προέκυψαν εξερευνώντας τα όρια της υπάρχουσας τεχνολογίας έθεσαν νέες παραμέτρους στην σχεδίαση των ψηφιακών συστημάτων. Η πιο σημαντική από αυτές, σε ένα πλαίσιο αυξανόμενου ενδιαφέροντος για την οικολογία, τα ενεργειακά αποθέματα και το περιβάλλον, δεν θα μπορούσε παρά να είναι η ενεργειακή κατανάλωση. Άλλωστε, οι σχεδιαστές των ψηφιακών συστημάτων πάντα διερευνούσαν την *απόδοση* ενός συστήματος, την βελτίωση δηλαδή ενός χαρακτηριστικού χωρίς την ανάλογη επιβάρυνση κάποιου άλλου (πχ ταχύτητα έναντι κόστους, επιφάνειας ολοκληρωμένου, αξιοπιστίας κτλ).

Στα ψηφιακά συστήματα, οι περιορισμοί που προέκυψαν όσον αφορά την κατανάλωση ενός κυκλώματος διακρίνονται σε δύο βασικές αιτίες. Πρώτον, η ανάπτυξη ποικίλων ενσωματωμένων συστημάτων καθώς και η διάδοση και εξέλιξη των φορητών συσκευών, απαιτούν πολλές φορές την δυνατότητα εκτέλεσης αυξημένου υπολογιστικού φόρτου εργασίας από μια συσκευή η οποία τροφοδοτείται από μπαταρία. Η σχετικά μικρή όμως διάρκεια ζωής μιας μπαταρίας, όπως και η περιορισμένη πυκνότητα αποθηκευμένης ενέργειάς της, καθιστούν σε αυτές τις περιπτώσεις την κατανάλωση λειτουργίας ως το μεγαλύτερο πρόβλημα προς επίλυση. Συνήθως αυτή η απαίτηση συνάδει και με περιορισμούς που αφορούν το μέγεθος του κυκλώματος (επιφάνεια ολοκληρωμένου). Δεύτερον, η ολοένα αυξανόμενη κλίμακα ολοκλήρωσης και η εξερεύνηση όλο και μικρότερων τεχνολογιών τρανζίστορ έχουν οδηγήσει σε ολοκληρωμένα κυκλώματα με τεράστια πυκνότητα, με επακόλουθο να ανακύψουν προβλήματα στην απαγωγή των αντίστοιχων ποσών θερμότητας από το εσωτερικό τους.

Βασικό συστατικό κάθε ψηφιακού συστήματος είναι ο μικροεπεξεργαστής, ο οποίος είναι υπεύθυνος για την εκτέλεση των επιμέρους λειτουργιών και προγραμμάτων. Η «καρδιά» κάθε μικροεπεξεργαστή είναι η Αριθμητική και Λογική Μονάδα (ALU) στην οποία εκτελείται όλο το υπολογιστικό φορτίο. Μια από της επιμέρους λειτουργίες της ALU που έχει ιδιαίτερο ενδιαφέρον είναι η πράξη του πολλαπλασιασμού.

Ο πολλαπλασιασμός, όντας η πιο απαιτητική στοιχειώδης πράξη έχει τύχει ιδιαίτερης διερεύνησης στον χώρο των ψηφιακών συστημάτων. Έχει γίνει εκτεταμένη έρευνα τόσο σε επίπεδο αλγορίθμου όσο και σε τεχνολογικό επίπεδο για την εύρεση αποδοτικότερων κυκλωμάτων υλοποίησής του, κάτι που είναι απόλυτα δικαιολογημένο αν αναλογιστούμε ότι η συγκεκριμένη λειτουργία βρίσκεται στην βάση κάθε υπολογιστικού συστήματος, αλλά και ότι υπάρχουν πλείστες εφαρμογές, κυρίως αλγόριθμοι επεξεργασίας ήχου, γραφικών και κρυπτογραφίας που κάνουν σχεδόν αποκλειστική χρήση της συγκεκριμένης λειτουργίας.

## ***1.2 Αντικείμενο διπλωματικής***

Η παρούσα διπλωματική έχει ως αντικείμενο την διερεύνηση μερικών συναφών υλοποιήσεων σειριακών-παράλληλων πολλαπλασιαστών ως προς την επιφάνεια, την κατανάλωση και την ταχύτητα λειτουργίας, με απώτερο σκοπό την εξαγωγή συμπερασμάτων που θα οδηγήσουν σε οικονομικότερες ενεργειακά υλοποιήσεις.

Οι σειριακοί πολλαπλασιαστές καταλαμβάνουν πολύ μικρότερη επιφάνεια από τους αντίστοιχους παράλληλους και μπορούν να λειτουργήσουν σε αρκετά μεγαλύτερες συχνότητες. Ωστόσο, είναι πιο αργοί από τους παράλληλους πολλαπλασιαστές επειδή απαιτούν πολλούς κύκλους για να εξάγουν το τελικό αποτέλεσμα. Επίσης, οι σειριακοί πολλαπλασιαστές αν και κάνουν πιο αποδοτική χρήση του υπάρχοντος κυκλώματος μέσω

pipeline, αποτελούνται από πολλούς ενδιάμεσους καταχωρητές που συμβάλλουν στην αύξηση των ενεργειακών απαιτήσεων του κυκλώματος.

Για την δεδομένη τεχνολογία που χρησιμοποιήσαμε (90nm), αλλά και δεδομένου ότι υποθέσαμε πως τα υπό εξέταση κυκλώματα θα έβρισκαν εν δυνάμει εφαρμογή σε μια μονάδα που θα είχε να διεκπεραιώσει ένα μεγάλο υπολογιστικό φορτίο, οπότε θα βρίσκονταν σε συνεχή λειτουργία, μελετήσαμε την δυναμική κατανάλωση των κυκλωμάτων, η οποία σε αυτήν την περίπτωση είναι πολύ πιο σημαντική έναντι της στατικής.

Μελετήθηκαν σειριακοί-παράλληλοι πολλαπλασιαστές, που εφαρμόζουν κωδικοποίηση Modified Booth στα δεδομένα της σειριακής εισόδου για την παραγωγή λιγότερων μερικών γινομένων. Οι πολλαπλασιαστές αυτοί είναι κατάλληλοι για το χειρισμό αριθμών στην μορφή συμπληρώματος του δύο.

Στα πλαίσια της εργασίας μας, υλοποιήθηκαν σε γλώσσα Verilog μια σειρά από μοντέλα πολλαπλασιαστών, σε διάφορα radices (1 ως 8) και σε διάφορα μήκη λέξεως (8 ως 64). Βασική ιδέα ήταν η ελάττωση της δυναμικής κατανάλωσης μέσω της μείωσης των ενδιάμεσων καταχωρητών ενός αντίστοιχου σειριακού-παράλληλου πολλαπλασιαστή πλήρους pipeline. Υιοθετήσαμε structural περιγραφή των κυκλωμάτων ώστε να μελετήσουμε το πρόβλημα σε επίπεδο αρχιτεκτονικής και να εξάγουμε συμπεράσματα τα οποία θα είναι χρήσιμα και στην αυτοματοποιημένη σχεδίαση.

Η λειτουργία των κυκλωμάτων αυτών προσομοιώθηκε και επαληθεύτηκε με το περιβάλλον Modelsim. Κατόπιν, πραγματοποιήθηκε η σύνθεση των παραπάνω κυκλωμάτων από τον Synopsys Design Compiler, με χρήση μιας standard cell βιβλιοθήκης της Artisan. Τα αποτελέσματα καθυστέρησης, επιφάνειας και κατανάλωσης εξαχθήκαν από το Synopsys Design Compiler και το Synopsys Primepower.

Επίσης, διερευνήσαμε μια παραλλαγή της κωδικοποίησης Modified Booth, η οποία στόχευε στην μεγιστοποίηση των παραγόμενων μηδενικών μερικών γινομένων για περαιτέρω δυνατότητες εξοικονόμησης ενέργειας.

### **1.2.1 Συνεισφορά**

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελετήσαμε τα προβλήματα της σχεδίασης αριθμητικών κυκλωμάτων.
2. Υλοποιήσαμε μια βασική τοπολογία σειριακού παράλληλου πολλαπλασιαστή για διάφορα radices, η οποία μπορεί να παραχθεί παραμετρικά για διάφορα μήκη λέξης.
3. Αξιολογήσαμε την απόδοση και τα πλεονεκτήματα/μειονεκτήματα κάθε σχήματος.
4. Παρουσιάσαμε τα συγκριτικά αποτελέσματα μιας οικογένειας πολλαπλασιαστών στα 90nm ως εγχειρίδιο αναφοράς.

5. Εκτιμήσαμε το κόστος των καταχωρητών στην δυναμική κατανάλωση ενός CMOS κυκλώματος στα 90nm.
6. Διερευνήσαμε την Canonic Modified Booth κωδικοποίηση για την ελάττωση της δυναμικής κατανάλωσης.

### ***1.3 Οργάνωση κειμένου***

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

Στο κεφάλαιο 3 αναλύεται συνοπτικά η πορεία του κατασκευαστικού μέρους της εργασίας με αρκετές αναφορές σε σχετικές εργασίες που αφορούν πιο εξειδικευμένα θέματα σχεδίασης.

Στο κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα των προσομοιώσεων καθώς και μια αναλυτική συγκριτική παρουσίαση των επιμέρους σχημάτων.

Στο κεφάλαιο 5 γίνεται μια συνόψιση των συμπερασμάτων που προέκυψαν, και προτείνονται πιθανές εφαρμογές και μελλοντικές επεκτάσεις της εργασίας.

Στο κεφάλαιο 6 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 ώστε να γίνει κατανοητή η παρούσα διπλωματική εργασία από τον αναγνώστη.

Αρχικά, παρουσιάζεται το δυαδικό σύστημα αρίθμησης, και πιο συγκεκριμένα η αναπαράσταση αριθμών στη μορφή συμπληρώματος του δύο, που είναι και η πιο διαδεδομένη στα αριθμητικά κυκλώματα.

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

Στο ίδιο πλαίσιο, αναλύονται οι Canonic Signed Digit αναπαραστάσεις, οι οποίες εγγυώνται τον ελάχιστο αριθμό μη μηδενικών μερικών γινομένων, με προφανή οφέλη στη μείωση της κατανάλωσης ενός κυκλώματος πολλαπλασιασμού, καθώς η πρόσθεση ενός μηδενικού μερικού γινομένου είναι μια λειτουργία η οποία μπορεί να «παραλειφθεί».

Τέλος, παρατίθενται μερικές βασικές τοπολογίες αριθμητικών κυκλωμάτων για την εκτέλεση των πράξεων της πρόσθεσης και του πολλαπλασιασμού.

## 2.1 Αριθμητικά συστήματα

### 2.1.1 Δυαδικό σύστημα

Το δυαδικό σύστημα 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:

$$\begin{array}{r} 154 = 10011010 \\ +185 = +10111001 \\ \hline 339 = 101010011 \end{array}$$

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) 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:

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

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:

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

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

συνεχία 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:

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

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 ταχύτατη 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  $\{-a, \dots, -1, 0, 1, \dots, a\}$ .

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 κρατούμενου ή την μεγιστοποίηση του αριθμού των μηδενικών ψηφίων της αναπαράστασης.

#### 2.1.4.1 Κανονική αναπαράσταση προσημασμένου ψηφίου (Canonic Signed Digit)

Όπως αναφέρθηκε και στην προηγούμενη παράγραφο, στις αναπαραστάσεις προσημασμένου ψηφίου υπάρχουν πολλοί διαφορετικοί συνδυασμοί ψηφίων που αντιπροσωπεύουν έναν συγκεκριμένο δεκαδικό αριθμό.

Σε μια τέτοια αναπαράσταση, το πλήθος των μη μηδενικών ψηφίων ονομάζεται βάρος Hamming. Το διάνυσμα με το μικρότερο δυνατό βάρος Hamming ονομάζεται κανονικό.

Το γεγονός ότι ένα διάνυσμα είναι κανονικό δεν συνεπάγεται ότι είναι και το μοναδικό κανονικό σε δεδομένη αναπαράσταση, ωστόσο αποδεικνύεται ότι στην κανονική αναπαράσταση δεν υπάρχουν διαδοχικά μη μηδενικά ψηφία.

Το γεγονός αυτό υποδεικνύει την δυνατότητα περαιτέρω μείωσης του υλικού ή της ενέργειας που απαιτείται για την εκτέλεση των πράξεων, καθώς αναμένουμε τουλάχιστον τα μισά ψηφία της αναπαράστασης να είναι μηδενικά. Για την ακρίβεια ο θεωρητικά αναμενόμενος μέσος όρος μη μηδενικών ψηφίων είναι το 1/3 του αριθμού των ψηφίων (bits) της απλής αναπαράστασης.



## 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 0 1 0 0 1 0 1 1 0	-2 Πρόσθεσε -2A (Πρώτο μερικό γινόμενο)
1 0 1 1 0 1 0 1	+1 Πρόσθεσε A (Δεύτερο μερικό γινόμενο)
<u>1</u> <u>1</u> 1 1 0 1 1 0 1 0 1 0	Άθροισμα των δύο πρώτων μερικών γινομένων
0 1 0 0 1 0 1 1	-1 Πρόσθεσε -A (Τρίτο μερικό γινόμενο)
<u>0</u> <u>0</u> <u>0</u> 0 1 0 0 0 0 0 1 1 0 1 0	Άθροισμα των τριών μερικών γινομένων
1 0 1 1 0 1 0 1 0	+2 Πρόσθεσε +2A (Τέταρτο μερικό γινόμενο)
1 0 1 1 1 1 0 1 0 0 1 1 0 1 0	Τελικό αποτέλεσμα = -8550

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

Ο τροποποιημένος αλγόριθμος του 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.

## ***2.3 Είδη αθροιστών και πολλαπλασιαστών***

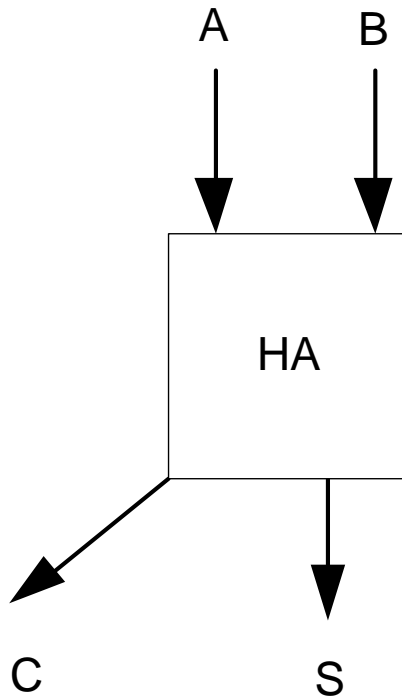
### ***2.3.1 Δομικά στοιχεία***

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:

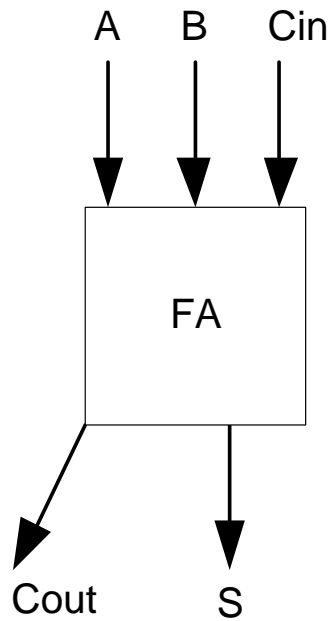
Πίνακας 2.5 Πίνακας αληθείας ημιαθροιστή

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 <sub>in</sub>	C <sub>out</sub>	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

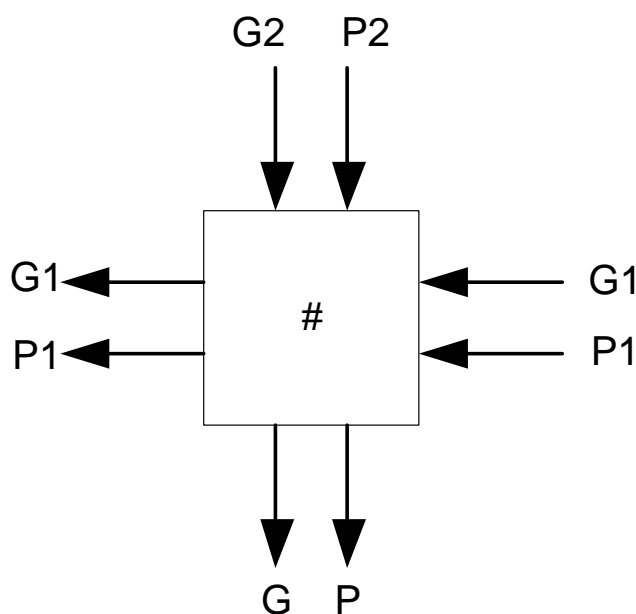


### 2.3.1.3 Κύκλωμα πρόβλεψης κρατουμένου

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 γέννησεως 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 κρατουμένου είναι οι εξής:

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

$$P = P1 \cdot P2$$

## 2.3.2 Αθροιστές

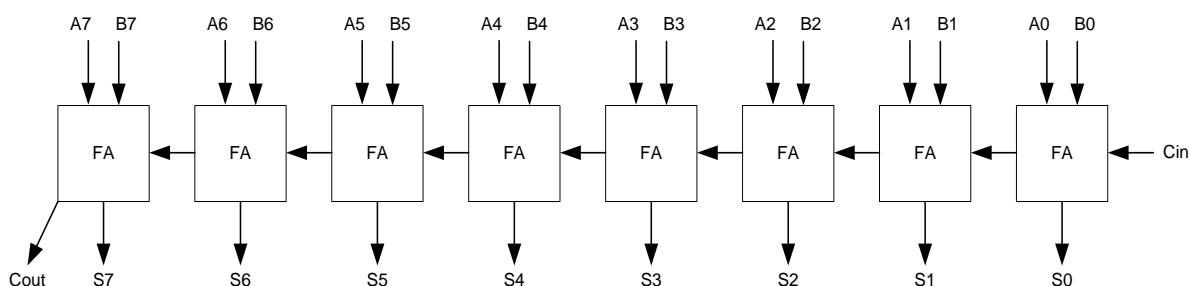
### 2.3.2.1 Αθροιστής διάδοσης κρατουμένου (Carry Propagate Adder)

Ο αθροιστής διάδοσης κρατουμένου (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 κρατουμένου 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 κρατουμένου

Το παραπάνω κύκλωμα μπορεί να λειτουργήσει και για δυαδικούς αριθμούς σε μορφή συμπληρώματος του δύο, αν προστεθεί έλεγχος για τις περιπτώσεις υπερχείλισης, 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.

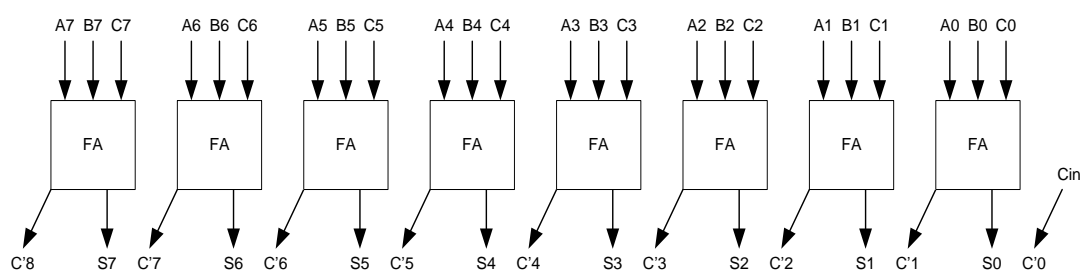
### 2.3.2.2 Αθροιστής σωσίματος κρατουμένου (Carry Save Adder)

Ο αθροιστής σωσίματος κρατουμένου (Carry Save Adder) 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 έξοδος 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 αθροιστή είναι μόνο 1 επίπεδο πλήρους αθροιστή. Το βασικό μειονέκτημα του είναι ότι το αποτέλεσμα δεν βρίσκεται σε δυαδική μορφή, οπότε θα χρειαστεί περαιτέρω επεξεργασία.

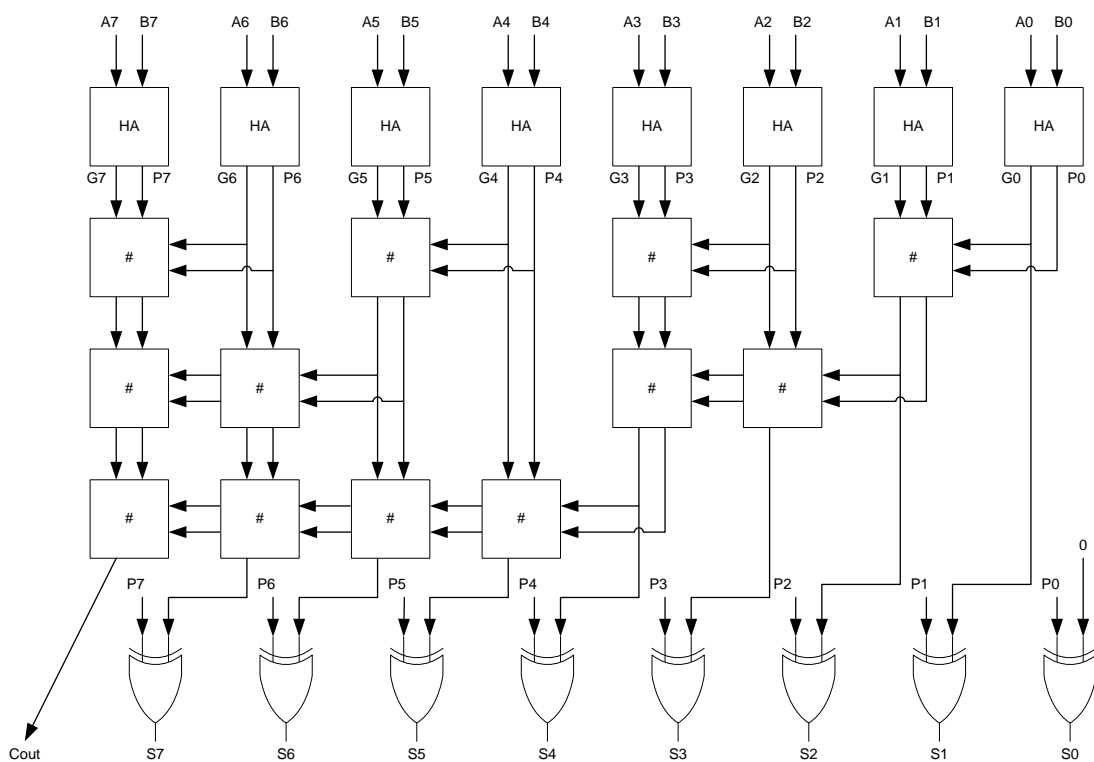
### 2.3.2.3 Αθροιστής πρόβλεψης κρατουμένου (Carry Look ahead Adder)

Ο αθροιστής πρόβλεψης κρατουμένου (Carry Look ahead Adder) 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 προσηταιριστικότητα 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  $\{A7, B7, Cout\}$ .

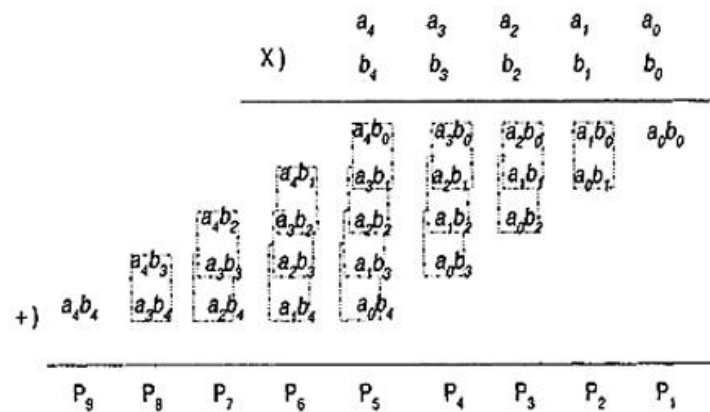
Ο αθροιστής πρόβλεψης κρατουμένου έχει πιο πολύπλοκη δομή, αλλά είναι πιο γρήγορος από τον αθροιστή διάδοσης κρατουμένου, καθώς αποφεύγει την διάδοση κρατουμένου και παράγει τα αποτελέσματά του με καθυστέρηση ενός HA συν  $\log N$  κυκλωμάτων # συν μιας πύλης XOR. Το μειονέκτημα του αθροιστή πρόβλεψης 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.

### 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 και (AND).

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

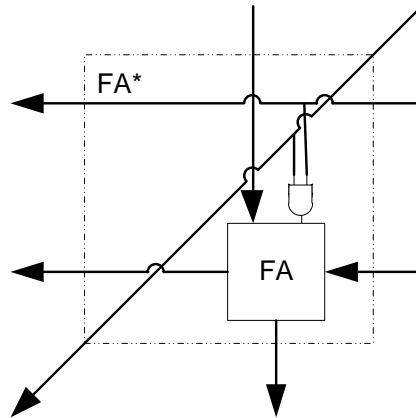


Σχήμα 2.7 Πράξεις πολλαπλασιασμού

#### 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.

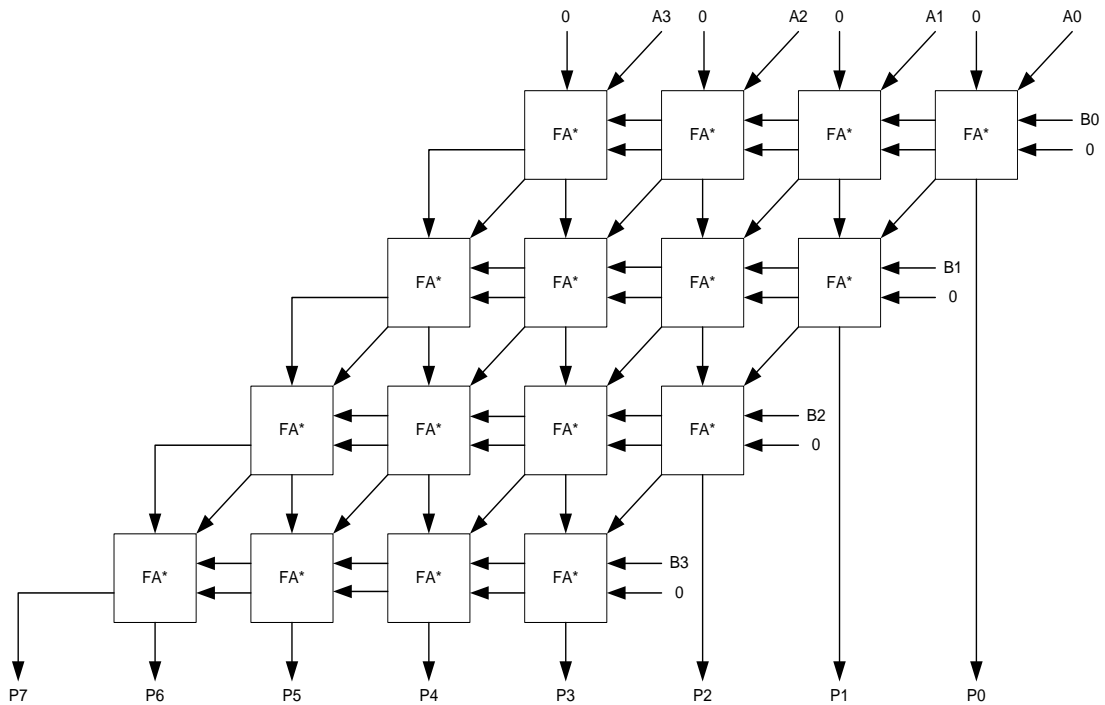


Σχήμα 2.8 Κύκλωμα FA\*

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 (η 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.

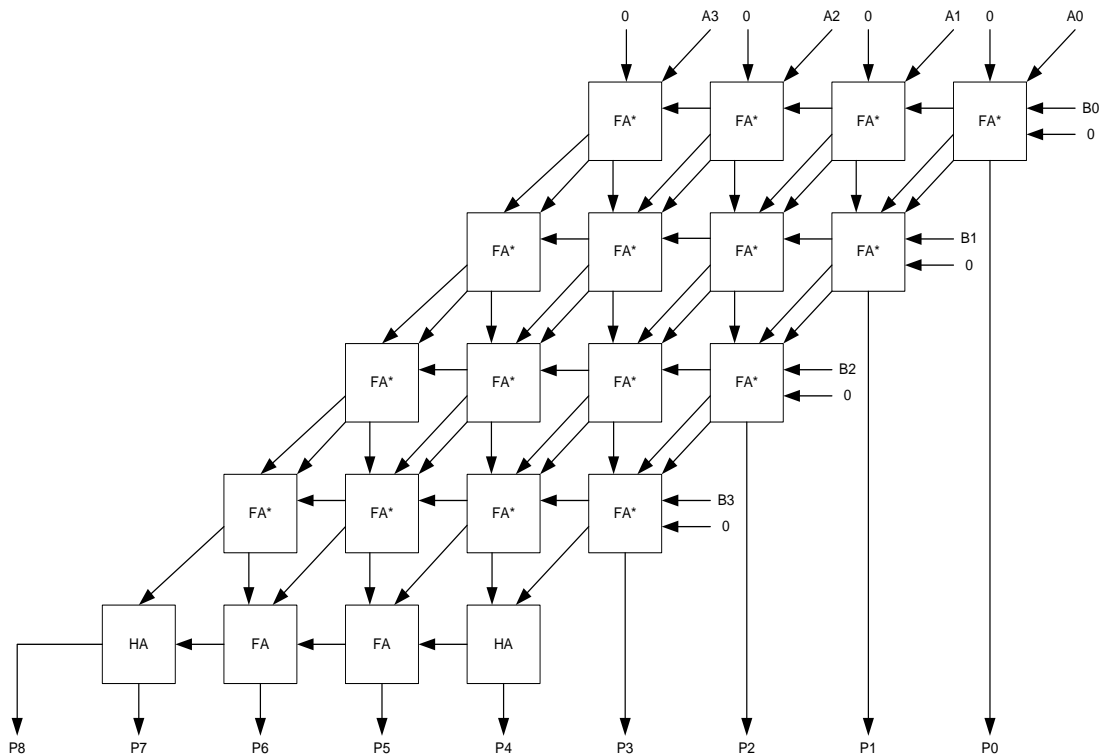


**Σχήμα 2.9** Κύκλωμα παράλληλου πολλαπλασιαστή με διάδοση κρατουμένου

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 καθυστέρηση 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 (η 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 Σειριακός Παράλληλος Πολλαπλασιαστής (Serial Parallel Multiplier)

Οι σειριακοί πολλαπλασιαστές είναι κυκλώματα συνεχούς διοχέτευσης ή συστολικά, που προκύπτουν από το χρονικό «δίπλωμα» των αντίστοιχων παράλληλων πολλαπλασιαστών. Οι παράλληλοι πολλαπλασιαστές τύπου πίνακα μετατρέπονται εύκολα σε σειριακούς, λόγω της επαναληπτικής δομής τους, και λόγω του ότι αποτελούνται από όμοια στοιχεία.

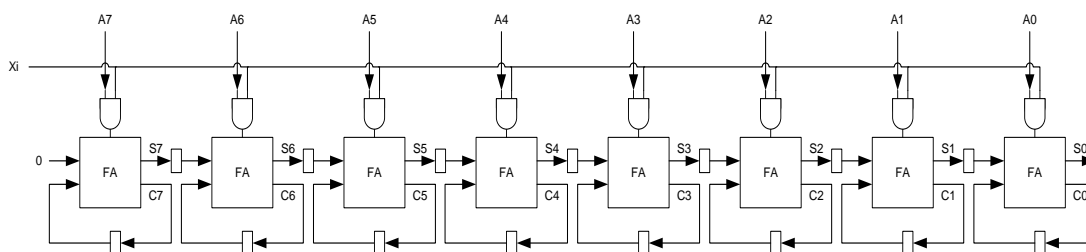
Στην παράγραφο αυτή θα παρουσιάσουμε τον σειριακό-παράλληλο πολλαπλασιαστή, ο οποίος έχει όμοια λειτουργία με ένα συστολικό κύκλωμα που προκύπτει από τον παράλληλο πολλαπλασιαστή με σώσιμο κρατουμένου.



Ο σειριακός παράλληλος πολλαπλασιαστής είναι ένα συστολικό κύκλωμα, δηλαδή υποστηρίζει την λογική της συνεχούς διοχέτευσης, ενώ ταυτόχρονα αποτελείται από πανομοιότυπα στοιχεία.

Η λογική που ακολουθεί, είναι παρόμοια με αυτή του παράλληλου πολλαπλασιαστή σωσίματος κρατουμένου, μόνο που χρησιμοποιεί ένα μόνο επίπεδο αθροιστών, οι οποίοι σε συνδυασμό με ένα σύνολο καταχωρητών επιτελούν τις λειτουργίες της πρόσθεσης και της ολίσθησης, ώστε το τελικό αποτέλεσμα να είναι το ίδιο και στις δύο περιπτώσεις.

Για το πολλαπλασιασμό δύο N-bit αριθμών A και X ακολουθείται η εξής διαδικασία, τα ψηφία του A εισέρχονται παράλληλα στο κύκλωμα άθροισης, ενώ τα ψηφία του X εισάγονται σειριακά, ένα σε κάθε κύκλο ρολογιού, ξεκινώντας από το λιγότερο σημαντικό και καταλήγοντας στο περισσότερο σημαντικό. Παράγεται ένα bit του αποτελέσματος σε κάθε κύκλο ρολογιού, ξεκινώντας από το λιγότερο σημαντικό. Αμέσως παρακάτω παρουσιάζεται ένας σειριακός παράλληλος πολλαπλασιαστής οκτώ bit.



Σχήμα 2.11 Κύκλωμα σειριακού παράλληλου πολλαπλασιαστή 8 bit

Ο σειριακός παράλληλος πολλαπλασιαστής παράγει τα αποτελέσματά του με συνολική καθυστέρηση N πύλες AND συν N FA συν N καταχωρητές, ενώ έχει συνολική επιφάνεια ίση με  $N \text{ FA}^*$  συν N καταχωρητές, όπου N το μήκος λέξης του όρου A. Είναι δηλαδή αρκετά πιο αργός από τον παράλληλο πολλαπλασιαστή σωσίματος κρατουμένου, ωστόσο καταλαμβάνει σημαντικά μικρότερη επιφάνεια. Επίσης, έχει την δυνατότητα να υποστηρίξει διάφορα μήκη λέξης για τον παράγοντα X.

### 2.3.3.3 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.

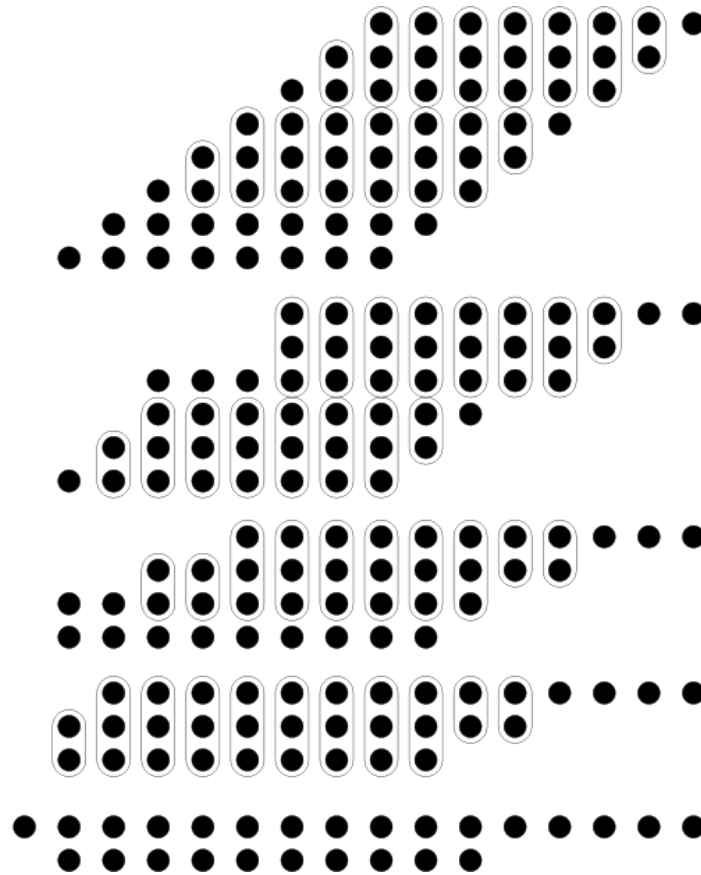
Θα παρουσιάσουμε δύο τύπους δενδρικών πολλαπλασιαστών, που χρησιμοποιούν διαφορετικό αλγόριθμο για την δημιουργία του δέντρου συμπίεσης, τους πολλαπλασιαστές Wallace και τους πολλαπλασιαστές Dadda.

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).

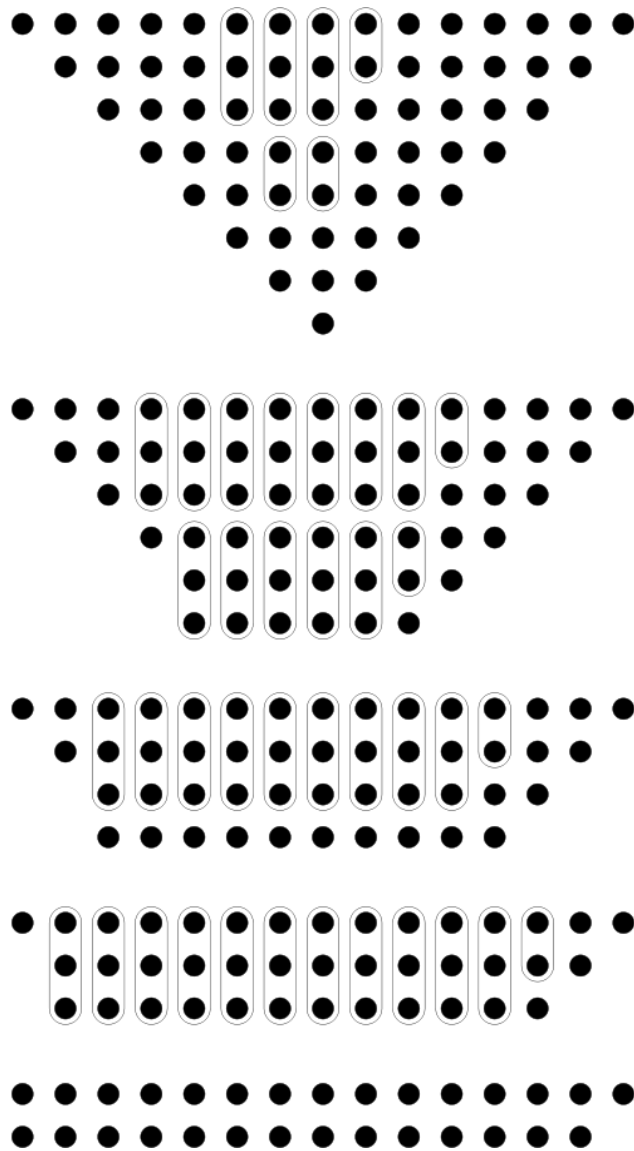


Σχήμα 2.12 Wallace δενδρικός συμπιεστής

### ***B. Δενδρικός πολλαπλασιαστής Dadda***

Ο δενδρικός συμπιεστής 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 πιο «επιθετικά» τα κυκλώματα των FA. 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, νεώτερες μελέτες έδειξαν ότι επιτυγχάνει και μικρότερη καθυστέρηση έναντι του δενδρικού συμπιεστή 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 συμπίεστης

Συνοψίζοντας, έχοντας δει τις βασικές μορφές πολλαπλασιαστών, συμπεραίνουμε ότι οι δενδρικοί πολλαπλασιαστές έχουν τα καλύτερα αποτελέσματα στον τομέα της ταχύτητας, οι πολλαπλασιαστές τύπου πίνακα προσφέρουν μια κανονική δομή, η οποία μπορεί να οδηγήσει σε σειριακούς πολλαπλασιαστές, οι οποίοι έχουν σημαντικά μικρότερη επιφάνεια και μπορούν να λειτουργήσουν σε μεγαλύτερες συχνότητες, λόγω του μικρότερου κρίσιμου μονοπατιού (critical path).

# 3

## ΣΧΕΔΙΑΣΜΟΣ ΣΕΙΡΙΑΚΟΥ ΚΥΚΛΩΜΑΤΟΣ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΥ ΜΕ ΚΩΔΙΚΟΠΟΙΗΣΗ MODIFIED BOOTH

Σε αυτό το κεφάλαιο περιγράφεται αναλυτικά η υλοποίηση των αλγορίθμων που εξετάστηκαν, με έμφαση στα δομικά στοιχεία που χρησιμοποιήθηκαν, και τεκμηριώνονται οι επιλογές και οι παραδοχές που έγιναν.

### *3.1 Στόχοι εργασίας*

Στόχος της παρούσας εργασίας ήταν η σχεδίαση ενός κατά το δυνατόν οικονομικού ενεργειακά σειριακού πολλαπλασιαστή, καταβάλλοντας παράλληλα προσπάθεια να επιτευχθεί η ελάχιστη δυνατή καθυστέρηση. Όσον αφορά την επιφάνεια, υπήρξε μια μεγαλύτερη ευελιξία, καθώς η επιφάνεια ενός σειριακού πολλαπλασιαστή αυξάνει γραμμικά με το μήκος λέξης, ενώ αυτή ενός παράλληλου δενδρικού πολλαπλασιαστή αυξάνει τετραγωνικά, οπότε υπάρχει σαφές πλεονέκτημα σε αυτόν τον τομέα.

Ως μέτρο σύγκρισης χρησιμοποιήθηκε ένας δενδρικός παράλληλος πολλαπλασιαστής, με όμοιο κύκλωμα παραγωγής μερικών γινομένων, ο οποίος κάνει χρήση ενός δένδρου Wallace για την μετατροπή του αθροίσματος των μερικών γινομένων σε Carry-Save μορφή και ενός

αθροιστή πρόβλεψης κρατούμενου για την παραγωγή του αποτελέσματος σε δυαδική μορφή συμπληρώματος του δύο. Στόχος αυτής της τοπολογίας πολλαπλασιαστή είναι η μέγιστη ταχύτητα για την παραγωγή του τελικού αποτελέσματος.

Επίσης, τα αποτελέσματα συγκρίνονται ενδεικτικά με την τοπολογία του απλού σειριακού-παράλληλου πολλαπλασιαστή, ο οποίος αποτελεί την απλούστερη σχεδιαστικά υλοποίηση συστολικού κυκλώματος σειριακού πολλαπλασιαστή.

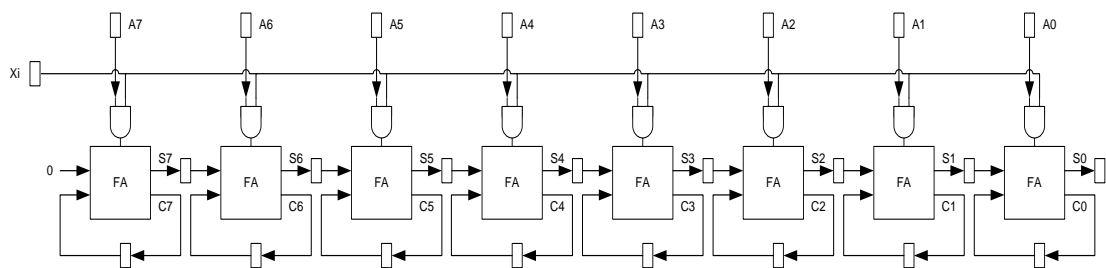
Σε όλα τα κυκλώματα που εξετάζονται συμπεριλαμβάνονται οι καταχωρητές εισόδου και εξόδου, κάτι το οποίο είναι άλλωστε και σημαντικό στη μελέτη της κατανάλωσης των κυκλωμάτων αυτών.

Όλα τα κυκλώματα συντέθηκαν και προσομοιώθηκαν με κελιά της βιβλιοθήκης TSMC 90nm της Artisan, με χρήση της εντολής `-compile_ultra`, και με ιδιαίτερη έμφαση στην επίτευξη ελάχιστης καθυστέρησης.

### 3.2 Χρήση διπλής διοχέτευσης (*double pipeline*)

Ο σειριακός-παράλληλος πολλαπλασιαστής, που παρουσιάστηκε στο προηγούμενο κεφάλαιο, αποτελεί την απλούστερη λύση σειριακού πολλαπλασιαστή, ωστόσο χρησιμοποιεί μη αποδοτικά το υπάρχον υλικό και ιδίως τους καταχωρητές.

Από το παρακάτω σχήμα, και γενικεύοντας για οποιοδήποτε μήκος λέξης  $N$ , βλέπουμε ότι ο σειριακός-παράλληλος πολλαπλασιαστής αποτελείται από  $3N+1$  καταχωρητές και πρέπει να λειτουργήσει για  $2N$  κύκλους ρολογιού ώστε να παράγει το τελικό αποτέλεσμά του. Μια πιο προσεκτική ανάλυση βέβαια, δείχνει ότι τελικά περιλαμβάνει μόνο  $3N$  καταχωρητές, καθώς ο πρώτος πλήρης αθροιστής αθροίζει συνεχώς δύο μηδενικά ψηφία, αφού το  $C7$ , όπως και όλοι οι υπόλοιποι ενδιάμεσοι καταχωρητές περιέχουν αρχικά μηδενική τιμή, οπότε δεν παράγει ποτέ κρατούμενο, απλώς μεταφέρει την τιμή του ( $X_i \text{ AND } A7$ ) στον  $S7$ , οπότε γενικά δεν χρειάζεται ούτε ο καταχωρητής  $C7$  ούτε ο πρώτος πλήρης αθροιστής,

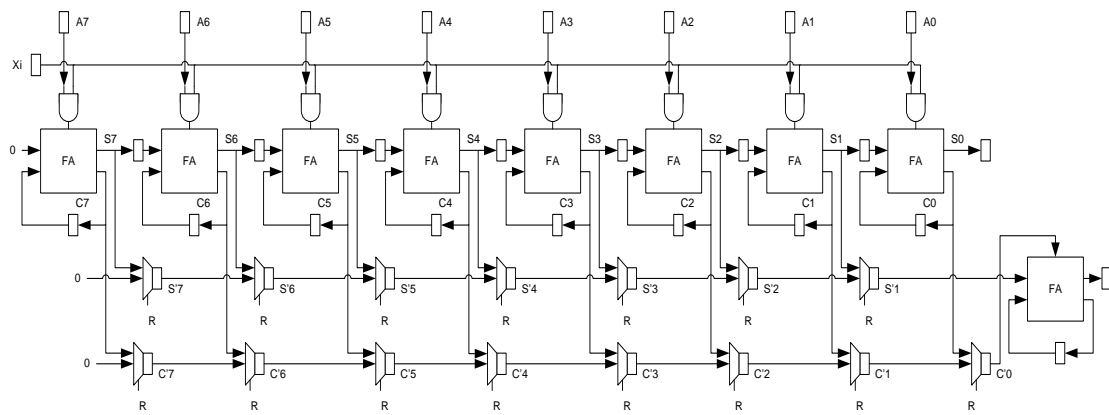


**Σχήμα 3.1** Κύκλωμα σειριακού παράλληλου πολλαπλασιαστή με καταχωρητές εισόδου/εξόδου

Σε αυτή την περίπτωση, για την παραγωγή του τελικού αποτελέσματος πρέπει να εργαστούν  $3N$  καταχωρητές,  $N-1$  πλήρεις αθροιστές και  $N$  πύλες AND επί  $2N$  κύκλους ρολογιού.

Επίσης, για την σωστή λειτουργία του κυκλώματος, μετά τον N-οστό κύκλο θα πρέπει οι καταχωρητές εισόδου να τεθούν στο 0, ώστε να αθροιστούν μόνο τα ενδιάμεσα αποτελέσματα που είναι σε μορφή Carry-Save. Αυτό αποτελεί μια μη αποδοτική χρήση του υπάρχοντος υλικού, καθώς οι N πύλες AND θα παράγουν μηδενικά αποτελέσματα και οι N-1 πλήρεις αθροιστές θα εκτελέσουν την λειτουργία του ημιαθροιστή, καθώς θα έχουν πάντα μια μηδενική είσοδο.

Ένας πιο αποδοτικός τρόπος υλοποίησης του σειριακού πολλαπλασιασμού είναι η διατήρηση του παραπάνω σχήματος για την εκτέλεση των πρώτων N κύκλων του πολλαπλασιασμού, και η χρήση ενός δεύτερου κυκλώματος, παράλληλα με το πρώτο, το οποίο θα λαμβάνει το Carry-Save ενδιάμεσο αποτέλεσμα, και απλώς θα αθροίζει τα ψηφία ίσου βάρους, κάνοντας χρήση ενός μόνο κυκλώματος πλήρους αθροιστή. Η τοπολογία που χρησιμοποιήσαμε φαίνεται στο αμέσως επόμενο σχήμα:



**Σχήμα 3.2** Κύκλωμα σειριακού παράλληλου πολλαπλασιαστή διπλής διοχέτευσης με καταχωρητές εισόδου/εξόδου

Το σήμα επιλογής R, ενεργοποιείται κάθε N-οστό κύκλο ρολογιού, και φροντίζει για την φόρτωση του Carry-Save αποτελέσματος στο δεύτερο στάδιο, καθώς και τον μηδενισμό των καταχωρητών του πρώτου σταδίου.

Ακολουθώντας την προηγούμενη ανάλυση, το νέο κύκλωμα αποτελείται από  $5N+2$  καταχωρητές και παράγει το τελικό του αποτέλεσμα σε  $2N$  κύκλους ρολογιού. Ωστόσο, επειδή τα δύο τμήματα του πολλαπλασιαστή εργάζονται παράλληλα, σε συνθήκες συνεχούς λειτουργίας μπορούμε να θεωρήσουμε ότι απαιτούνται μόνο N κύκλοι για την εξαγωγή του αποτελέσματος, καθώς σε κάθε κύκλο ρολογιού έχουμε δύο έγκυρα bit αποτελέσματος, ένα για την τρέχουσα πράξη και ένα για την αμέσως προηγούμενη.

Επίσης, όπως και στην παραπάνω ανάλυση, οι καταχωρητές C7 και C'7 μπορούν να απαλειφθούν, όπως και ο πρώτος πλήρης αθροιστής.

Οπότε, σε αυτή την περίπτωση, για την παραγωγή του τελικού αποτελέσματος πρέπει να εργαστούν  $5N$  καταχωρητές,  $N$  πλήρεις αθροιστές,  $N$  πύλες AND και  $2(N-1)$  πολυπλέκτες επί  $N$  κύκλους ρολογιού.

Αναλυτικά η σύγκριση των δύο σχημάτων φαίνεται στον παρακάτω πίνακα:

**Πίνακας 3.1** Σύγκριση στοιχείων σειριακού πολλαπλασιαστή με/χωρίς διπλή διοχέτευση

Multiplier	registers	FA	AND	MUX	clock cycles
Serial parallel	$3N$	$N-1$	$N$	0	$2N$
Serial parallel (Double Pipeline)	$5N$	$N$	$N$	$2N-2$	$N$

Από την παραπάνω σύγκριση, θεωρώντας ως μονάδα μέτρησης τα χαρακτηριστικά της πύλης αναφοράς OXI-KAI (NAND), και με τις παραδοχές του επόμενου πίνακα, προκύπτει ότι:

**Πίνακας 3.2** Επιφάνεια και καθυστέρηση βασικών στοιχείων, ως προς την πύλη αναφοράς NAND

Component	area	delay
NAND NOR	1	1
AND OR	1	1.5
XOR XNOR	2	2
Half Adder	3	2
Full Adder	7	4
2-1 MUX	3	2
#	3	2
register	8	2

- Η επιφάνεια της προτεινόμενης τοπολογίας, για μεγάλα μήκη λέξης, θα είναι περίπου 70% μεγαλύτερη από αυτή του απλού σειριακού-παράλληλου πολλαπλασιαστή

**Πίνακας 3.3** Σύγκριση επιφάνειας σειριακού πολλαπλασιαστή με/χωρίς διπλή διοχέτευση

Multiplier	registers	FA	AND	MUX	total
Serial parallel	$24N$	$7N-7$	$N$	0	$32N-7$
Serial parallel (Double Pipeline)	$40N$	$7N$	$N$	$6N-6$	$54N-6$

- Η καθυστέρηση της προτεινόμενης τοπολογίας είναι κατά 2 πύλες NAND μεγαλύτερη από αυτή του σειριακού παράλληλου πολλαπλασιαστή. Στην βιβλιοθήκη που χρησιμοποιήσαμε η καθυστέρηση μιας πύλης NAND είναι περίπου 0.020ns.



**Πίνακας 3.4** Σύγκριση καθυστέρησης σειριακού πολλαπλασιαστή με/χωρίς διπλή διοχέτευση

Multiplier	AND	FA	MUX	registers	total
Serial parallel	1.5	4	0	2	7.5
Serial parallel (Double Pipeline)	1.5	4	2	2	9.5

- Η προτεινόμενη τοπολογία φαίνεται να είναι έχει ελαφρώς χειρότερες επιδόσεις με βάση το κριτήριο  $area*delay$ . Ωστόσο, αυτό δεν επιβεβαιώνεται από τα πειραματικά αποτελέσματα.

**Πίνακας 3.5** Σύγκριση σειριακού πολλαπλασιαστή με/χωρίς διπλή διοχέτευση με βάση το κριτήριο  $area*delay$

Multiplier	area	delay	clock cycles	$area*delay$
Serial parallel	$32N-7$	7.5	$2N$	$480N^2-105N$
Serial parallel (Double Pipeline)	$54N-6$	9.5	$N$	$513N^2-57N$

Το κέρδος σε κύκλους λειτουργίας και πιθανόν στην ενεργειακή κατανάλωση έχει το κόστος της επιπλέον επιφάνειας και της χρήσης περισσότερων καταχωρητών (κατά 67% !), επομένως και της μεγαλύτερης δυναμικής κατανάλωσης. Επίσης, υπάρχει μια μικρή επιβάρυνση στην καθυστέρηση του κυκλώματος, καθώς προστίθεται ένας πολυπλέκτης στο κρίσιμο μονοπάτι. Γενικά, καθώς εφαρμόζουμε την παραπάνω υλοποίηση σε μεγαλύτερα radices, αναμένουμε μικρότερο κέρδος από την μείωση των καταχωρητών, αλλά μεγαλύτερο κέρδος από την εξοικονόμηση του σταδίου άθροισης το οποίο γίνεται πιο πολύπλοκο. Σε κάθε περίπτωση το κύκλωμα της δεύτερης διοχέτευσης, φαίνεται να επιδρά θετικά στην μείωση της ενεργειακής κατανάλωσης του κυκλώματος. Τέλος, πρέπει να αναφέρουμε ότι το κύκλωμα της δεύτερης διοχέτευσης αποτελεί ουσιαστικά έναν καταχωρητή ολίσθησης με δυνατότητα παράλληλης φόρτωσης, κάτι που ίσως επιτρέπει την περαιτέρω βελτιστοποίησή του.

### 3.3 Κωδικοποίηση Modified Booth

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, η χρήση της κωδικοποίησης Modified Booth έχει θετικά αποτελέσματα όσον αφορά την μείωση της ενεργειακής κατανάλωσης. Τόσο ο παράλληλος πολλαπλασιαστής, όσο και οι σειριακοί πολλαπλασιαστές που υλοποιήσαμε κάνουν χρήση της κωδικοποίησης αυτής. Στον μεν παράλληλο πολλαπλασιαστή έχουμε μείωση της επιφάνειας, καθώς παράγονται τα μισά μερικά γινόμενα, όπως επίσης απαιτείται και μικρότερο δένδρο αθροιστών για τον υπολογισμό του Carry-Save αποτελέσματος. Επιπλέον, έχουμε πιο γρήγορο κύκλωμα αφού το δένδρο άθροισης έχει

μικρότερο βάθος. Στους δε σειριακούς πολλαπλασιαστές έχουμε εξοικονόμηση κύκλων ρολογιού, καθώς το αποτέλεσμα παράγεται πλέον σε λιγότερους κύκλους.

### 3.3.1 Κύκλωμα κωδικοποίησης

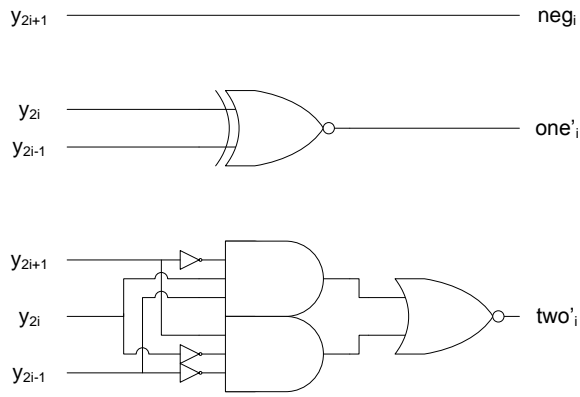
Το σχήμα κωδικοποίησης που θα χρησιμοποιήσουμε τόσο στον παράλληλο, όσο και στους σειριακούς πολλαπλασιαστές προτάθηκε το 2003 από τον Huang, ο οποίος μελέτησε διάφορα σχήματα κωδικοποίησης Modified Booth, τα οποία προσομοιώθηκαν με την βιβλιοθήκη TSMC 180nm της Artisan. Το σχήμα που εμφάνισε τα περισσότερα πλεονεκτήματα, ιδίως ελάχιστη κατανάλωση, αλλά και ελάχιστη καθυστέρηση, όπως επίσης και τον ελάχιστο αριθμό σημάτων κωδικοποίησης, φαίνεται αμέσως παρακάτω

**Πίνακας 3.6** 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

Ο παραπάνω πίνακας δείχνει τις πράξεις που πρέπει να εκτελεστούν σε κάθε περίπτωση. Ο πολλαπλασιασμός με το 0 ή με το -0 παράγει ένα μηδενικό μερικό γινόμενο, ο πολλαπλασιασμός με το +1 παράγει ένα μερικό γινόμενο ίσο με τον παράλληλο όρο ενώ με το -1 το συμπλήρωμα ως προς δύο του παράλληλου όρου, και τέλος ο πολλαπλασιασμός με το +2 ολισθαίνει κατά μια θέση αριστερά τον παράλληλο όρο ενώ με το -2 παράγει το συμπλήρωμα ως προς δύο του ολισθημένου όρου. Υπενθυμίζουμε ότι για τον υπολογισμό του συμπληρώματος ως προς δύο ενός αριθμού πρέπει να αντιστρέψουμε όλα τα ψηφία του, και κατόπιν να τους προσθέσουμε μια μονάδα. Η επιπλέον αυτή μονάδα, είναι ένα επιπλέον σήμα, απαραίτητο για την μετατροπή σε μορφή συμπληρώματος του δύο, οποίο έχει την τιμή  $neg_i$ .

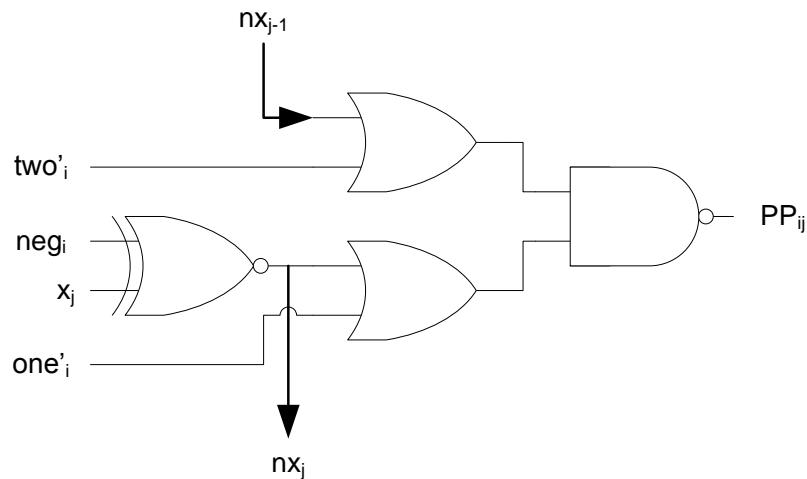
Η υλοποίηση της παραπάνω κωδικοποίησης φαίνεται στο αμέσως επόμενο σχήμα:



**Σχήμα 3.3** Κύκλωμα κωδικοποίησης Modified Booth

### 3.3.2 Κύκλωμα δημιουργίας μερικών γινομένων

Το βέλτιστο κύκλωμα δημιουργίας των μερικών γινομένων με βάση την παραπάνω κωδικοποίηση είναι το εξής:

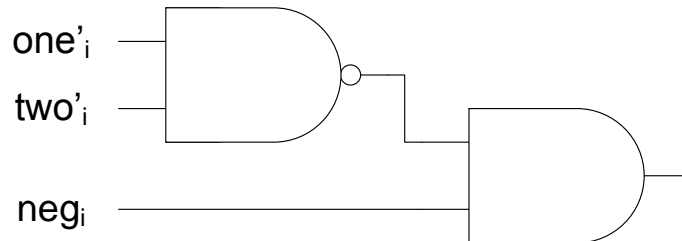


**Σχήμα 3.4** Κύκλωμα παραγωγής μερικών γινομένων

Τόσο στο κύκλωμα κωδικοποίησης, όσο και στο κύκλωμα παραγωγής μερικών γινομένων χρησιμοποιείται αρνητική λογική (δηλαδή τα σήματα  $one'$  και  $two'$ ) διότι στην παρούσα βιβλιοθήκη οι αντίστοιχες πύλες είναι ταχύτερες και καταναλώνουν λιγότερη ενέργεια.

Ένα άλλο πλεονέκτημα αυτής της κωδικοποίησης, είναι το ότι στην περίπτωση του αρνητικού μηδενικού, τα σήματα  $one'$  και  $two'$  θα προκαλέσουν αμέσως το μηδενισμό του αντίστοιχου μερικού γινομένου, παρακάμπτοντας δηλαδή τον υπολογισμό  $111..11 + 1$ , ο οποίος είναι μια άσκοπη δαπάνη ενέργειας.

Λόγω του παραπάνω, ο όρος που πρέπει να προστεθεί στο κάθε μερικό γινόμενο πρέπει να διορθωθεί ελαφρώς, ώστε να μην προσθέτει μια μονάδα στην περίπτωση του αρνητικού μηδενικού. Σε αυτή την περίπτωση ο όρος που απαιτείται για την σωστή μετατροπή σε μορφή συμπληρώματος του δύο, θα είναι:

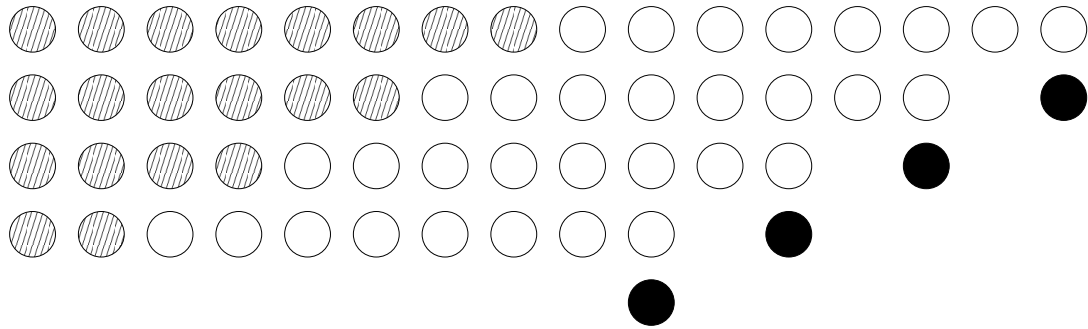


**Σχήμα 3.5** Κύκλωμα παραγωγής της μονάδας που απαιτείται για την μετατροπή σε μορφή συμπληρώματος του δύο

Ο επιπλέον όρος αυτός, παράγεται παράλληλα με τα ψηφία του μερικού γινομένου και δεν συνεισφέρει στην συνολική καθυστέρηση του κυκλώματος, καθώς το κύκλωμά του είναι πιο απλό και με μικρότερη καθυστέρηση από την γεννήτρια των μερικών γινομένων. Ωστόσο, είναι ένα επιπλέον κύκλωμα, το οποίο όμως παρουσιάζεται μόνο μια φορά μέσα στο κύκλωμά μας, οπότε η επιβάρυνση που επιφέρει σε επιφάνεια είναι αμελητέα σε σχέση με το ενεργειακό όφελος που προκύπτει.

### 3.4 Επέκταση προσήμου

Επειδή τα κυκλώματα που θα υλοποιήσουμε χειρίζονται αριθμούς στην μορφή συμπληρώματος του δύο, απαιτείται η επέκταση προσήμου μεταξύ αριθμών διαφορετικού μήκους, ώστε να παραχθεί σωστό αποτέλεσμα. Οι διορθωτικοί όροι που πρέπει να προστεθούν στα μερικά γινόμενα φαίνονται στο παρακάτω παράδειγμα για έναν πολλαπλασιασμό 8x8 bit.



**Σχήμα 3.6** Μερικά γινόμενα και διορθωτικοί όροι για πολλαπλασιασμό 8x8bit

Οι λευκές κουκκίδες παριστάνουν τα bits των μερικών γινομένων, οι μαύρες κουκκίδες τους όρους που πρέπει να προστεθούν για την μετατροπή κάθε μερικού γινομένου σε μορφή συμπληρώματος του δύο, και οι οποίοι προφανώς ισούνται με ένα αν το αντίστοιχο μερικό γινόμενο προκύπτει από αντιστροφή του παράλληλου όρου, και οι σκιασμένες κουκκίδες παριστάνουν τα bits της επέκτασης προσήμου. Τα ψηφία της επέκτασης προσήμου πρέπει να καλύψουν μέχρι και τη βαθμίδα βάρους 15, καθώς το αποτέλεσμα ενός 8bit x 8bit πολλαπλασιασμού έχει 16 ψηφία στα βάρη 15 – 0. Τα ψηφία αυτά λαμβάνουν την τιμή του προσήμου του αντίστοιχου μερικού γινομένου.

Η παραπάνω μορφή, παρόλο που είναι κατάλληλη για υλοποίηση σε μια παράλληλη τοπολογία πολλαπλασιαστή, είτε δενδρικού είτε και πίνακα, δεν προσφέρεται για ένα κύκλωμα συνεχούς διοχέτευσης. Επίσης, είναι μη αποδοτική, καθώς χρησιμοποιεί ένα μεγάλο μέρος του υλικού ώστε να υπολογίσει και να προσθέσει έναν αριθμό από άσσους ή μηδενικά τα οποία θα μπορούσαν να υπολογιστούν εκ των προτέρων, αφού λαμβάνουν λίγες προκαθορισμένες μορφές. Ένας τέτοιος τρόπος υπολογισμού των ψηφίων επέκτασης προσήμου αναλύεται αμέσως παρακάτω:

Στην περίπτωση μας έχουμε τέσσερεις όρους, ο καθένας από τους οποίους λαμβάνει την τιμή του προσήμου για τα βάρη ως το 15. Αυτή η σχέση γράφεται αναλυτικά:

$$ct = \sum_{k=0}^3 s_k \cdot \sum_{i=8+2k}^{15} 2^i$$

Στην γενική περίπτωση, η παραπάνω σχέση γίνεται:

$$ct = \sum_{k=0}^{N/2} s_k \cdot \sum_{i=N+2k}^{2N} 2^i$$

Όπου N το μήκος της λέξης.

Χρησιμοποιώντας την σχέση:

$$\sum_{q=j}^k 2^q = 2^{k+1} - 2^j$$

Η παραπάνω σχέση γίνεται:

$$ct = \sum_{k=0}^3 s_k \cdot \{2^{16} - 2^{8+2k}\} = \sum_{k=0}^3 -s_k \cdot 2^{8+2k}$$

Και στη γενική περίπτωση:

$$ct = \sum_{k=0}^{N/2} s_k \cdot \{2^{2N} - 2^{N+2k}\} = \sum_{k=0}^{N/2} -s_k \cdot 2^{N+2k}$$

Τέλος, χρησιμοποιώντας την σχέση:

$$-s_k = \bar{s}_k - 1$$

Έχουμε ότι:

$$ct = \sum_{k=0}^3 \bar{s}_k \cdot 2^{8+2k} - \{2^{14} + 2^{12} + 2^{10} + 2^8\}$$

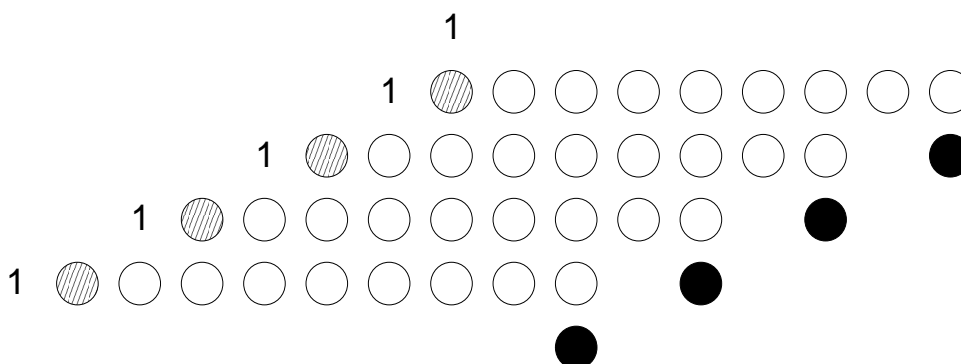
Και λαμβάνοντας το συμπλήρωμα ως προς δύο του δεύτερου όρου, έχουμε τελικά:

$$ct = \sum_{k=0}^3 \bar{s}_k \cdot 2^{8+2k} + \{2^{15} + 2^{13} + 2^{11} + 2^9 + 2^8\}$$

Στην γενική περίπτωση, η παραπάνω σχέση γίνεται:

$$ct = \sum_{k=0}^{\frac{N}{2}-1} s_k \cdot 2^{N+2k} + \{2^{2N-1} + \dots + 2^{N+3} + 2^{N+1} + 2^N\}$$

Δηλαδή, πρέπει να προστεθεί το συμπλήρωμα του αντίστοιχου bit προσήμου σε κάθε αντίστοιχο άρτιο βάρος που είναι μεγαλύτερο ή ίσο του μήκους λέξης N, και μια μονάδα στο βάρος N και σε κάθε περιττό βάρος μεγαλύτερο του N. Το παραπάνω εξηγείται στο επόμενο σχήμα, το οποίο απλοποιεί τα μερικά γινόμενα του πολλαπλασιασμού 8x8 bit.



Σχήμα 3.7 Μερικά γινόμενα και απλοποιημένοι διορθωτικοί όροι για πολλαπλασιασμό 8x8bit

Η μορφή αυτή μπορεί να μετατραπεί πλέον εύκολα σε κύκλωμα συνεχούς διοχέτευσης και επιπλέον, εξαλείφει τις περιττές πράξεις της επέκτασης προσήμου.

### 3.5 Κωδικοποίηση *Canonic Modified Booth*

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, στην αριθμητική προσημασμένου ψηφίου, μπορούν να υπάρχουν διαφορετικά διανύσματα που αναπαριστούν έναν δυαδικό αριθμό. Προφανώς, μεταξύ αυτών είναι δυνατόν να υπάρξει εξοικονόμηση ενέργειας από τις αναπαραστάσεις που μεγιστοποιούν τον αριθμό των μηδενικών ψηφίων. Σε αυτή την παράγραφο διερευνούμε μια δυνατότητα τροποποίησης της κωδικοποίησης *Modified Booth*, ώστε να επιτύχουμε περισσότερα μηδενικά μερικά γινόμενα.

### 3.5.1 Μετατροπή σε κανονική αναπαράσταση

Το πρώτο στάδιο της νέας κωδικοποίησης Modified Booth βασίζεται στον αλγόριθμο που προτάθηκε από τον G.Reitweiser το 1960 και μετατρέπει έναν αριθμό B από δυαδική αναπαράσταση σε κανονικό διάνυσμα αναπαράστασης προσημασμένου ψηφίου D, με στοιχεία από το σύνολο  $\{-1, 0, +1\}$ .

Ο αλγόριθμος του Reitweiser έχει ως εξής:

1. Θέτουμε τον μετρητή i στην τιμή μηδέν ( $i=0$ ). Ο μετρητής δείχνει το τρέχον βάρος του ψηφίου που θα παραχθεί. Επίσης, επειδή ο αλγόριθμος χρησιμοποιεί μια επαναληπτική άθροιση, θέτουμε αρχικά την τιμή του κρατούμενου εισόδου στο μηδέν ( $C_0=0$ ).
2. Εξετάζουμε δύο διαδοχικά ψηφία του αριθμού B, με βάρη i και i+1, μαζί με την τιμή του κρατούμενου εισόδου, και παράγουμε την τιμή του κρατούμενου εξόδου  $C_{i+1}$  σύμφωνα με τη συμβατική δυαδική αριθμητική, δηλαδή το κρατούμενο εξόδου  $C_{i+1}$  έχει τιμή 1 αν δύο τουλάχιστον εκ των  $B_i, B_{i+1}, C_i$  έχουν τιμή 1.
3. Υπολογίζουμε το ψηφίο βάρους i της κανονικής αναπαράστασης σύμφωνα με την σχέση:  $D_i = B_i + C_i - 2C_{i+1}$
4. Το κρατούμενο εξόδου γίνεται το νέο κρατούμενο εισόδου. Αυξάνουμε το i κατά ένα και ελέγχουμε αν  $i=N$ , αν η ισότητα δεν ισχύει πηγαίνουμε στο βήμα 2, αλλιώς σταματάμε.

Αξίζει να σημειώσουμε ότι, εάν έχουμε αριθμούς σε μορφή συμπληρώματος του δύο, το ψηφίο βάρους N+1 που πρέπει να χρησιμοποιηθεί για παραχθούν σωστά αποτελέσματα από τον παραπάνω αλγόριθμο, πρέπει να λάβει την τιμή του MSB του αρχικού αριθμού.

Τα παραπάνω παρουσιάζονται αναλυτικά στο επόμενο παράδειγμα:

Έστω ο αριθμός  $B = -41$ , ο οποίος σε δυαδική αναπαράσταση συμπληρώματος του δύο είναι  $B = 110101111$ .

Στον επόμενα πίνακα φαίνονται αναλυτικά όλα τα βήματα του αλγορίθμου του Reitweiser:

**Πίνακας 3.7** Παράδειγμα μετατροπής 8-bit δυαδικού αριθμού σε μορφή συμπληρώματος του δύο, σε κανονική αναπαράσταση προσημασμένου ψηφίου

i	$B_i$	$B_{i+1}$	$C_i$	$C_{i+1}$	$D_i$
i=0	1	1	0	1	-1
i=1	1	1	1	1	0
i=2	1	0	1	1	0
i=3	0	1	1	1	-1
i=4	1	0	1	1	0
i=5	0	1	1	1	-1
i=6	1	1	1	1	0
i=7	1	1	1	1	0



Το διάνυσμα που προκύπτει είναι το  $D = \{0, 0, -1, 0, -1, 0, 0, -1\}$  και η τιμή του είναι πράγματι  $D = -32 - 8 - 1 = -41$ . Παρατηρούμε ότι η νέα αναπαράσταση έχει πέντε μηδενικά ψηφία, ενώ η απλή δυαδική είχε μόλις τρία.

Στον παρακάτω πίνακα, παρουσιάζονται αναλυτικά όλες οι περιπτώσεις αντιστοίχισης δυαδικών ψηφίων με τα ψηφία της κανονικής αναπαράστασης προσημασμένου ψηφίου, όπως προκύπτουν από τα βήματα 2 και 3 του αλγορίθμου του Reitweisner:

**Πίνακας 3.8** Αντιστοίχιση δυαδικών ψηφίων με κανονική αναπαράσταση προσημασμένου ψηφίου

$C_i$	$B_i$	$B_{i+1}$	$C_{i+1}$	$D_i$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	+1
0	1	1	1	-1
1	0	0	0	+1
1	0	1	1	-1
1	1	0	1	0
1	1	1	1	0

Αξίζει να σημειώσουμε ότι, παρόλο που ο παραπάνω πίνακας υποδεικνύει μηδενικό ψηφίο κωδικοποίησης στις μισές περιπτώσεις (όπως πχ και ο απλός αλγόριθμος του Booth), στην πραγματικότητα το ποσοστό αυτό είναι ακόμα μεγαλύτερο. Όπως αποδείχθη από τον Reitweisner, σε αυτή την αναπαράσταση δεν υπάρχουν διαδοχικά μη μηδενικά ψηφία, δηλαδή όλες οι μονάδες του διανύσματος χωρίζονται από τουλάχιστον ένα μηδενικό, κάτι το οποίο συνεπάγεται ότι σε κάθε περίπτωση τουλάχιστον τα μισά ψηφία της αναπαράστασης θα είναι μηδενικά. Για την ακρίβεια, αποδείχθηκε ότι ο αναμενόμενος μέσος όρος μη μηδενικών ψηφίων είναι το  $1/3$  των ψηφίων της απλής αναπαράστασης.

### 3.5.2 Μετατροπή σε *Canonic Modified Booth* αναπαράσταση

Το δεύτερο στάδιο της νέας κωδικοποίησης συνίσταται στην χρήση του κανονικού διανύσματος που προέκυψε από τον αλγόριθμο του Reitweisner, και στην συμπίεσή του στο μισό μέγεθος με χρήση της κωδικοποίησης Modified Booth.

Στόχος της τεχνικής αυτής είναι να εκμεταλλευτούμε την αυξημένη πιθανότητα εμφάνισης μηδενικού ψηφίου ώστε να παράγουμε περισσότερα μηδενικά μερικά γινόμενα, διατηρώντας το προηγούμενο κύκλωμα παραγωγής μερικών γινομένων.

Για την παραγωγή των νέων ψηφίων της κωδικοποίησης πρέπει να εξετάσουμε τρία διαδοχικά ψηφία κάθε φορά, τα οποία θα μας δώσουν δύο διαδοχικά ψηφία κανονικής αναπαράστασης  $D_i$  και  $D_{i+1}$ , τα οποία θα αθροίσουμε σύμφωνα με την σχέση  $D = D_i + 2D_{i+1}$ . Επειδή ένα τουλάχιστον από τα διαδοχικά ψηφία της κανονικής αναπαράστασης είναι

μηδενικό, το ψηφίο  $D$  της κωδικοποίησης που θα προκύψει μπορεί να πάρει τιμές από το σύνολο  $\{-2, -1, 0, +1, +2\}$ , όπως ακριβώς και στην Modified Booth κωδικοποίηση.

Με βάση τα παραπάνω, ο αλγόριθμος του Reitweiser τροποποιείται στον αλγόριθμο μετατροπής σε Canonic Modified Booth αναπαράσταση, ως εξής:

Έστω αριθμός  $B$  μήκους  $N$  bits σε δυαδική αναπαράσταση συμπληρώματος του δύο. Θεωρούμε τον αριθμό  $B$  με  $N+1$  bits, όπου το bit βάρους  $N+1$  είναι η επέκταση προσήμου του αριθμού  $B$ .

1. Θέτουμε τον μετρητή  $i$  στην τιμή μηδέν ( $i=0$ ). Επίσης, θεωρούμε τον μετρητή  $j$ , ο οποίος έχει πάντα την τιμή  $2i$  ( $j=2i$ ). Ο μετρητής  $i$  δείχνει το τρέχον βάρος του ψηφίου που θα παραχθεί, ενώ ο  $j$  τα αντίστοιχα ψηφία της δυαδικής κωδικοποίησης. Επίσης, επειδή ο αλγόριθμος χρησιμοποιεί μια επαναληπτική άθροιση, θέτουμε αρχικά την τιμή του κρατουμένου εισόδου στο μηδέν ( $C_0=0$ ).
2. Εξετάζουμε τρία διαδοχικά ψηφία του αριθμού  $B$ , με βάρη  $j$ ,  $j+1$  και  $j+2$ . Τα ψηφία  $B_j$  και  $B_{j+1}$  μαζί με την τιμή του κρατουμένου εισόδου παράγουν την τιμή του ενδιάμεσου κρατουμένου  $C_{j+1}$ , ενώ τα ψηφία  $B_{j+1}$  και  $B_{j+2}$  μαζί με την τιμή του ενδιάμεσου κρατουμένου παράγουν την τιμή του κρατουμένου εξόδου, σύμφωνα με τη συμβατική δυαδική αριθμητική.
3. Υπολογίζουμε το ψηφίο βάρους  $i$  της κανονικής αναπαράστασης σύμφωνα με την σχέση:  $CMB_i = D_j + 2D_{j+1} = B_i + C_i - 2C_{i+1} + 2(B_{i+1} + C_{i+1} - 2C_{i+2}) = B_i + C_i - 2C_{i+1} + 2B_{i+1} + 2C_{i+1} - 4C_{i+2} = B_i + C_i + 2B_{i+1} - 4C_{i+2}$
4. Το κρατούμενο εξόδου γίνεται το νέο κρατούμενο εισόδου. Αυξάνουμε το  $i$  κατά ένα και ελέγχουμε αν  $i=N/2$ , αν η ισότητα δεν ισχύει πηγαίνουμε στο βήμα 2, αλλιώς σταματάμε.

Τα παραπάνω παρουσιάζονται αναλυτικά στο επόμενο παράδειγμα:

Έστω ο αριθμός  $B = -41$ , ο οποίος σε δυαδική αναπαράσταση συμπληρώματος του δύο είναι ο  $B = 110101111$ .

Στον επόμενο πίνακα φαίνονται αναλυτικά όλα τα βήματα του παραπάνω αλγορίθμου:

**Πίνακας 3.9** Παράδειγμα μετατροπής 8-bit δυαδικού αριθμού σε μορφή συμπληρώματος του δύο, σε Canonic Modified Booth αναπαράσταση

$i$	$j$	$B_j$	$B_{j+1}$	$B_{j+2}$	$C_j$	$C_{j+1}$	$C_{j+2}$	$D_i$
$i=0$	$j=0$	1	1	1	0	1	1	-1
$i=1$	$j=2$	1	0	1	1	1	1	-2
$i=2$	$j=4$	1	0	1	1	1	1	-2
$i=3$	$j=6$	1	1	1	1	1	1	0

Το διάνυσμα που προκύπτει είναι το  $D=\{0, -2, -2, -1\}$  και η τιμή του είναι πράγματι  $D = -32 - 8 - 1 = -41$ . Αν χρησιμοποιούσαμε την απλή κωδικοποίηση Modified Booth το αντίστοιχο διάνυσμα θα ήταν το  $D=\{-1, +1, +2, -1\} = -64 + 16 + 8 - 1 = -41$  το οποίο δεν έχει κανένα μηδενικό ψηφίο, ενώ το νέο διάνυσμα έχει ένα μηδενικό ψηφίο.

Στον παρακάτω πίνακα, παρουσιάζονται αναλυτικά όλες οι περιπτώσεις αντιστοίχισης δυαδικών ψηφίων με τα ψηφία της κανονικής αναπαράστασης προσημασμένου ψηφίου, όπως προκύπτουν από τα βήματα 2 και 3 του παραπάνω αλγορίθμου:

**Πίνακας 3.10** Αντιστοίχιση δυαδικών ψηφίων με την αναπαράσταση Canonic Modified Booth

$C_i$	$B_i$	$B_{i+1}$	$B_{i+2}$	$C_{i+1}$	$C_{i+2}$	$D_i$	$D_{i+1}$	$CMB_i$
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	+1	+2
0	0	1	1	0	1	0	-1	-2
0	1	0	0	0	0	+1	0	+1
0	1	0	1	0	0	+1	0	+1
0	1	1	0	1	1	-1	0	-1
0	1	1	1	1	1	-1	0	-1
1	0	0	0	0	0	+1	0	+1
1	0	0	1	0	0	+1	0	+1
1	0	1	0	1	1	-1	0	-1
1	0	1	1	1	1	-1	0	-1
1	1	0	0	1	0	0	+1	+2
1	1	0	1	1	1	0	-1	-2
1	1	1	0	1	1	0	0	0
1	1	1	1	1	1	0	0	0

Αξίζει να σημειώσουμε ότι, παρόλο που ο παραπάνω πίνακας υποδεικνύει μηδενικό ψηφίο κωδικοποίησης στο  $\frac{1}{4}$  των περιπτώσεων (όπως και ο απλός αλγόριθμος Modified Booth), στην πραγματικότητα το ποσοστό αυτό είναι μεγαλύτερο. Το προηγούμενο επιβεβαιώνεται αν πραγματοποιήσουμε μια εξαντλητική διερεύνηση των δύο κωδικοποιήσεων για διάφορα μήκη λέξεως. Μερικά ενδεικτικά αποτελέσματα φαίνονται στον παρακάτω πίνακα.

**Πίνακας 3.11** Σύγκριση μηδενικών ψηφίων στις αναπαραστάσεις Modified Booth και Canonic Modified Booth

Bit length	Modified Booth	Canonic Modified Booth	$\Delta P\%$
8	25%	30.57%	5.57%
16	25%	31.94%	6.94%
32	25%	32.64%	7.64%

### 3.5.3 Κυκλωματική υλοποίηση

Κατά τη διάρκεια σχεδιασμού της κυκλωματικής υλοποίησης, είχαμε ως βασικό άξονα την διατήρηση του σχήματος κωδικοποίησης που μελετήσαμε στην παράγραφο 3.3, ώστε να εκμεταλλευτούμε το βέλτιστο κύκλωμα παραγωγής μερικών γινομένων, το οποίο χρησιμοποιείται  $N$  φορές στο κύκλωμα, αλλάζοντας μόνο το κύκλωμα κωδικοποίησης, το οποίο χρησιμοποιείται μόνο μια φορά, οπότε δεν θα υπάρχει ουσιαστική διαφορά στην επιφάνεια και την δυναμική κατανάλωση του κυκλώματος. Επίσης, το κύκλωμα κωδικοποίησης δεν συμμετέχει στο κρίσιμο μονοπάτι του κυκλώματος, καθώς παρεμβάλλονται ενδιάμεσοι καταχωρητές που αποθηκεύουν τα σήματα κωδικοποίησης, για να οδηγήσουν πιο αποτελεσματικά όλα τα στοιχεία παραγωγής μερικών γινομένων. Αυτό προφανώς ισχύει όσο το στάδιο κωδικοποίησης είναι πιο απλό από το κύκλωμα παραγωγής μερικών γινομένων και του 2-bit αθροιστή διάδοσης κρατουμένου.

Με βάση τον πίνακα 3.8 εξετάζουμε τα σήματα τα οποία προσφέρονται για την επιθυμητή υλοποίηση.

Αμέσως παρατηρούμε ότι το σήμα  $C_{i+2}$  είναι κατάλληλο για χρήση ως σήμα  $neg_i$ , καθώς λαμβάνει την τιμή 1, κάθε φορά που το ψηφίο της Canonic Modified Booth κωδικοποίησης είναι αρνητικό, εκτός από τις δύο περιπτώσεις του αρνητικού μηδενικού (που αντιστοιχούν στις δύο τελευταίες γραμμές του πίνακα). Αυτό δεν αποτελεί πρόβλημα αν φροντίσουμε στις αντίστοιχες περιπτώσεις να μηδενίζονται τόσο το σήμα  $one_i$  όσο και το σήμα  $two_i$ , ώστε η μονάδα παραγωγής μερικών γινομένων αλλά και το κύκλωμα του συμπληρωματικού όρου να παράγουν σωστά αποτελέσματα.

Για τις περιπτώσεις που το ψηφίο της Canonic Modified Booth κωδικοποίησης έχει απόλυτη τιμή 1, άρα και το σήμα  $one_i$  πρέπει να λάβει την τιμή 1, έχουμε τον παρακάτω πίνακα αληθείας

Πίνακας 3.12 Πίνακας αληθείας για το σήμα  $one_i$

	$B'_{i+1} B'_{i+2}$	$B'_{i+1} B_{i+2}$	$B_{i+1} B_{i+2}$	$B_{i+1} B'_{i+2}$
$C'_i B'_i$	0	0	0	0
$C'_i B_i$	1	1	1	1
$C_i B_i$	0	0	0	0
$C_i B'_i$	1	1	1	1

Από τον οποίο προκύπτει αμέσως ότι:

$$one_i = C'_i B_i + C_i B'_i = B_i \oplus C_i$$

Τέλος, για τις περιπτώσεις που το ψηφίο της Canonic Modified Booth κωδικοποίησης έχει απόλυτη τιμή 2, άρα και το σήμα  $two_i$  πρέπει να λάβει την τιμή 1, έχουμε τον παρακάτω πίνακα αληθείας:

**Πίνακας 3.13** Πίνακας αληθείας για το σήμα  $two_i$

	$B'_{i+1}B'_{i+2}$	$B'_{i+1}B_{i+2}$	$B_{i+1}B_{i+2}$	$B_{i+1}B'_{i+2}$
$C'_iB'_i$	0	0	1	1
$C'_iB_i$	0	0	0	0
$C_iB_i$	1	1	0	0
$C_iB'_i$	0	0	0	0

Από τον οποίο προκύπτει ότι:

$$two_i = C'_i B'_i B_{i+1} + C_i B_i B'_{i+1}$$

Η παραπάνω σχέση μπορεί να απλοποιηθεί περαιτέρω, αν παρατηρήσουμε στον πίνακα 3.7 ότι, όπως το σήμα  $one_i$  έχει όμοια συμπεριφορά με το αποτέλεσμα  $D_i$  έτσι και το σήμα  $two_i$  έχει την ίδια συμπεριφορά με το  $D_{i+1}$ , οπότε μπορούμε να δοκιμάσουμε την σχέση

$$two_i = B_{i+1} \oplus C_{i+1}$$

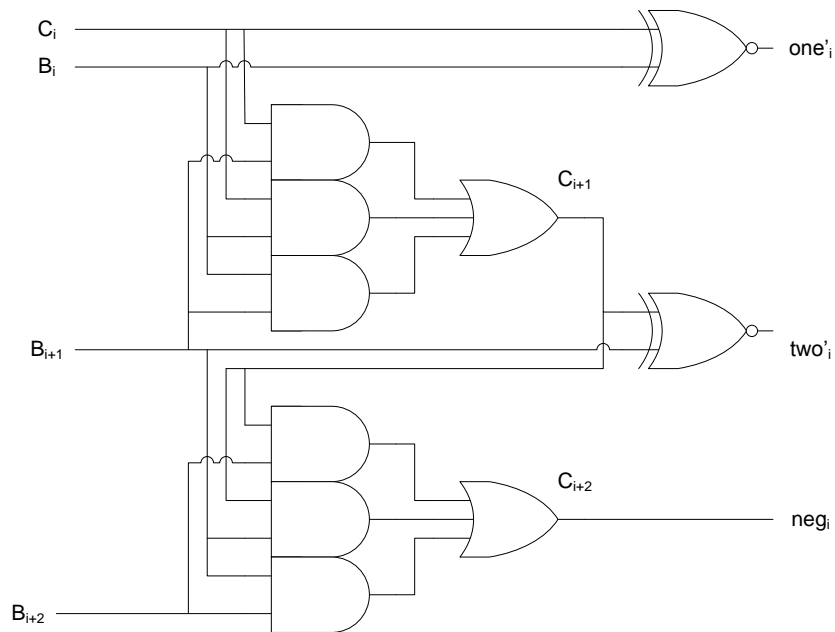
η οποία είναι ανάλογη με αυτή του σήματος  $one_i$ .

Πράγματι, όπως μπορούμε να δούμε και από τον παραπάνω πίνακα, οι δύο αυτές σχέσεις είναι ισοδύναμες, οπότε θα χρησιμοποιήσουμε την δεύτερη που είναι απλούστερη.

Επανεξετάζοντας την εκκρεμότητα που αφήσαμε με το σήμα  $neg_i$ , βλέπουμε αμέσως από τις δύο τελευταίες γραμμές του πίνακα ότι στις δύο περιπτώσεις που κάναμε λάθος παραδοχή για την τιμή του σήματος αυτού, τόσο η τιμή του σήματος  $one_i$  όσο και η τιμή του σήματος  $two_i$  λαμβάνουν την τιμή 1, οπότε το αντίστοιχο μερικό γινόμενο θα λάβει την τιμή 0 όπως και ο αντίστοιχος συμπληρωματικός όρος. Επομένως, έχουμε σωστό αποτέλεσμα και στις δύο αυτές περιπτώσεις.

Με βάση την παραπάνω ανάλυση, το κύκλωμα κωδικοποίησης αποτελείται από δύο πλήρεις αθροιστές, σε διάταξη διάδοσης κρατουμένου και δύο πύλες ΑΠΟΚΛΕΙΣΤΙΚΟΥ-Η (XOR). Για την ακρίβεια, το κύκλωμα δεν χρειάζεται ολόκληρους τους πλήρεις αθροιστές, αλλά μόνο την έξοδο κρατουμένου τους.

Το τελικό κύκλωμα κωδικοποίησης για την Canonic Modified Booth αναπαράσταση φαίνεται αμέσως παρακάτω:



Σχήμα 3.8 Κύκλωμα κωδικοποίησης Canonic Modified Booth

Τα κυκλώματα παραγωγής μερικών γινομένων και διορθωτικών όρων, παραμένουν ακριβώς ίδια με αυτά που χρησιμοποιήθηκαν και στην απλή Modified Booth κωδικοποίηση.

### 3.6 Radix-2 Modified Booth σειριακός πολλαπλασιαστής

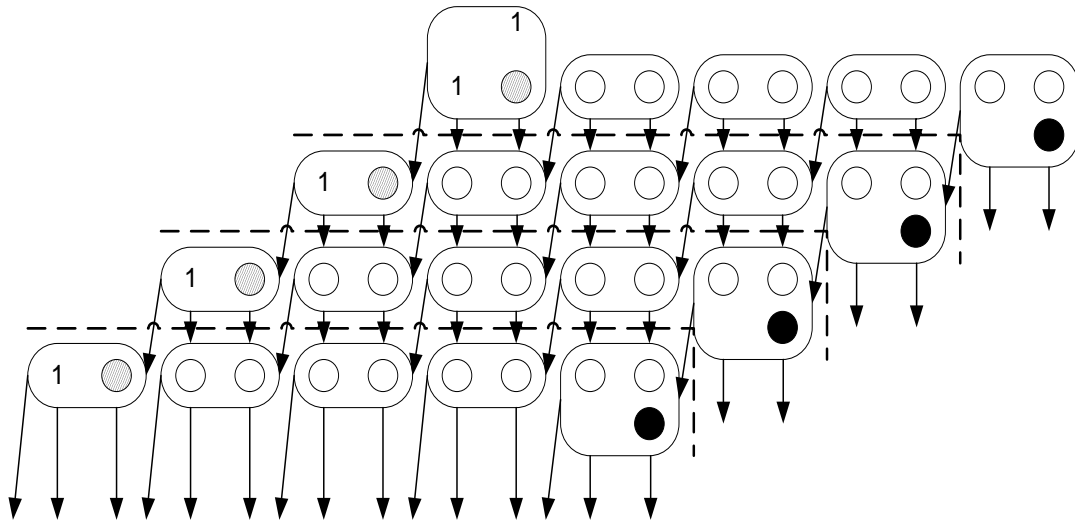
#### 3.6.1 Γενικευμένη Carry-Save αναπαράσταση

Η διάταξη των μερικών γινομένων, όπως παρουσιάστηκε στο σχήμα 3.7, υποδεικνύει ένα κύκλωμα το οποίο θα αθροίζει τα μερικά γινόμενα ανά δύο bits και θα παράγει δύο bits αποτελέσματος σε κάθε κύκλο.

Η παρατήρηση αυτή, σε συνδυασμό με το γεγονός ότι η ελάχιστη καθυστέρηση είναι πάντα ένας πρωταρχικός στόχος, μας οδηγεί στο να χρησιμοποιήσουμε την Carry-Save αναπαράσταση που παρουσιάσαμε σε προηγούμενο κεφάλαιο, καθώς αυτή προσφέρει την ταχύτερη υλοποίηση κυκλώματος αθροιστή, τροποποιώντας την ώστε να αποκομίσουμε περισσότερα οφέλη, κυρίως όσον αφορά την ενεργειακή κατανάλωση.

Συγκεκριμένα, ενώ η Carry-Save αναπαράσταση χρησιμοποιεί δύο διανύσματα C,S μήκους N, όπως προκύπτουν από μια σειρά N πλήρων αθροιστών, στην γενικευμένη Carry-Save αναπαράσταση θα χρησιμοποιηθούν N/2 αθροιστές των 2 bit, επομένως το διάνυσμα S θα έχει και πάλι N στοιχεία ενώ το C μόνο N/2. Αυτό έχει ως αποτέλεσμα την μείωση των καταχωρητών που απαιτούνται για την αποθήκευση του ενδιάμεσου αποτελέσματος για τον επόμενο κύκλο ρολογιού. Το κόστος αυτής της εξοικονόμησης είναι μια επιβάρυνση στην

καθυστέρηση του κυκλώματος, καθώς απαιτείται ένα πιο πολύπλοκο κύκλωμα για την άθροιση των δύο bit. Η ομαδοποίηση των bit των μερικών γινομένων που πρέπει να αθροιστούν φαίνεται στο παρακάτω σχήμα, για τον πολλαπλασιασμό 8x8 bits. Οι διακεκομμένες γραμμές υποδεικνύουν το χρονικό «κόψιμο» του κυκλώματος, ώστε να μετατραπεί σε συνεχούς διοχέτευσης.

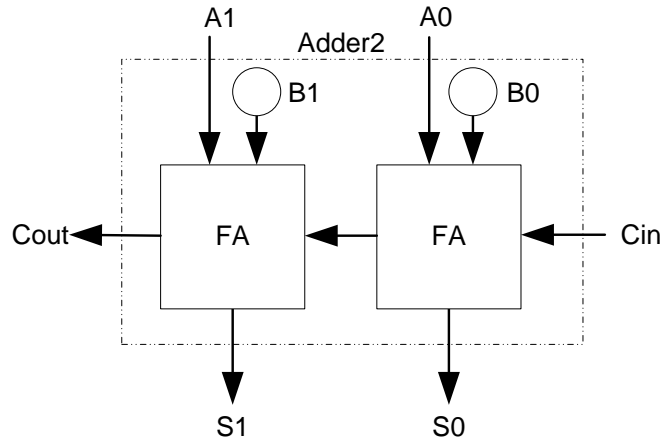


Σχήμα 3.9 Χρονική ανάλυση του radix-2 Modified Booth πολλαπλασιασμού 8x8bit (α)

Προφανώς τα αποτελέσματα του τελευταίου σταδίου, θα οδηγηθούν σε ένα δεύτερο κύκλωμα καταχωρητών ολίσθησης, όπου και θα αθροιστούν από έναν επιπλέον 2-bit αθροιστή. Επίσης, το παραπάνω κύκλωμα επιδέχεται ορισμένες βελτιστοποιήσεις.

### 3.6.2 Βασική μονάδα άθροισης

Η βασική δομική μονάδα, η οποία φαίνεται να ταιριάζει στις ανάγκες του κυκλώματός μας είναι ένας αθροιστής διάδοσης κρατουμένου 2 bit. Το κύκλωμα αυτό έχει πέντε εισόδους και τρεις εξόδους, όσα ακριβώς και τα ψηφία που θέλουμε να αθροίσουμε, είναι απλούστατο και καταλαμβάνει όση επιφάνεια θα καταλάμβανε και ένας συμβατικός αθροιστής σωσίματος-κρατουμένου. Επίσης, αν και πιο αργός από τον συμβατικό αθροιστή σωσίματος-κρατουμένου, έχει σταθερή καθυστέρηση ανεξάρτητη του μήκους λέξης και επιπλέον η καθυστέρηση αυτή μπορεί να ελαχιστοποιηθεί με τη χρήση πλήρων αθροιστών με γρήγορο μονοπάτι μεταξύ κρατουμένου εισόδου και κρατουμένου εξόδου.

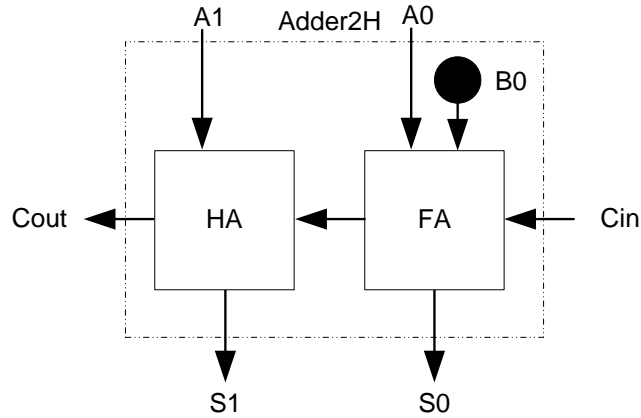


Σχήμα 3.10 Βασική δομική μονάδα του radix-2 Modified Booth πολλαπλασιαστή

### 3.6.3 Λιγότερο σημαντική μονάδα άθροισης

Ωστόσο, η βασική αυτή μονάδα δεν ταιριάζει στα δύο λιγότερο σημαντικά ψηφία του πολλαπλασιαστή. Όπως μπορούμε να δούμε και από το προηγούμενο σχήμα, στα βάρη 0,2,4,6 πρέπει να προστεθεί ένα μερικό γινόμενο, αλλά και ο όρος για την σωστή μετατροπή σε μορφή συμπληρώματος του δύο. Αυτό θα απαιτούσε την χρήση ενός διαφορετικού αθροιστή για το λιγότερο σημαντικό τμήμα του πολλαπλασιασμού. Το βασικό πρόβλημα από μια τέτοια επιλογή, θα ήταν μια σημαντική επιβάρυνση στο κρίσιμο μονοπάτι του κυκλώματος. Κάτι τέτοιο όμως δεν είναι καθόλου συμφέρον, διότι δεν έχουμε κανένα ουσιαστικό όφελος από αυτή τη σχεδίαση. Μια πολύ καλύτερη λύση είναι να χρησιμοποιήσουμε το κύκλωμα του αθροιστή διάδοσης κρατουμένου 2-bit για την πρόσθεση των ψηφίων των μερικών γινομένων, και να χρησιμοποιήσουμε μια επιπλέον μονάδα αθροιστή 2-bit για την πρόσθεση του όρου για την σωστή μετατροπή σε μορφή συμπληρώματος του δύο. Με αυτόν τον τρόπο δεν επιβαρύνουμε το κρίσιμο μονοπάτι του κυκλώματος και επίσης διατηρούμε μια κανονική δομή. Επιπλέον, η λιγότερο σημαντική μονάδα άθροισης, επειδή προσθέτει έναν αριθμό ενός bit και έναν δύο bit δεν είναι απαραίτητο να αποτελείται από δύο πλήρεις αθροιστές, αλλά μπορεί να αποτελείται από έναν πλήρη αθροιστή και έναν ημιαθροιστή.

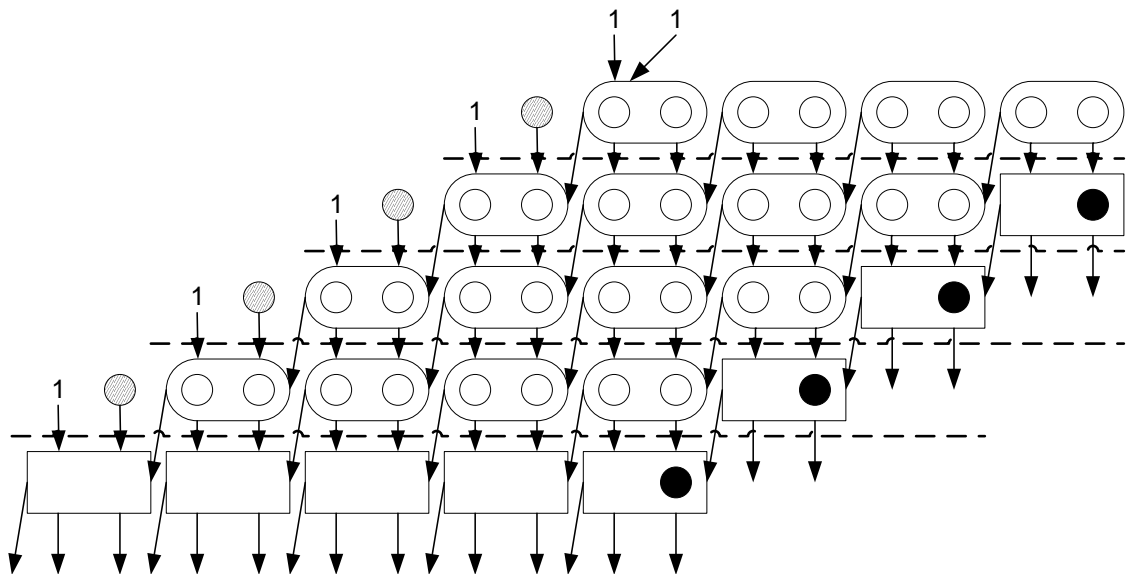




Σχήμα 3.11 Λιγότερο σημαντική μονάδα άθροισης του radix-2 Modified Booth πολλαπλασιαστή

### 3.6.4 Χειρισμός διορθωτικών όρων

Όσον αφορά το πιο σημαντικό τμήμα του πολλαπλασιαστή, παρατηρούμε ότι δεν χρειάζεται ένας επιπλέον αθροιστής για να προσθέσει τους διορθωτικούς όρους, αφού αυτοί απλώς μεταφέρονται στο επόμενο στάδιο, οπότε αρκεί να παρεμβληθεί ένας καταχωρητής ώστε να διατηρήσει το συμπλήρωμα του κρατουμένου μέχρι τον επόμενο κύκλο ρολογιού. Η διορθωτική μονάδα δεν χρειάζεται να αποθηκευτεί, αφού έχει πάντα τιμή 1 και έτσι μπορεί να χρησιμοποιηθεί απ' ευθείας στο επόμενο στάδιο. Οπότε το σχήμα 3.8 θα πρέπει να τροποποιηθεί ως εξής:

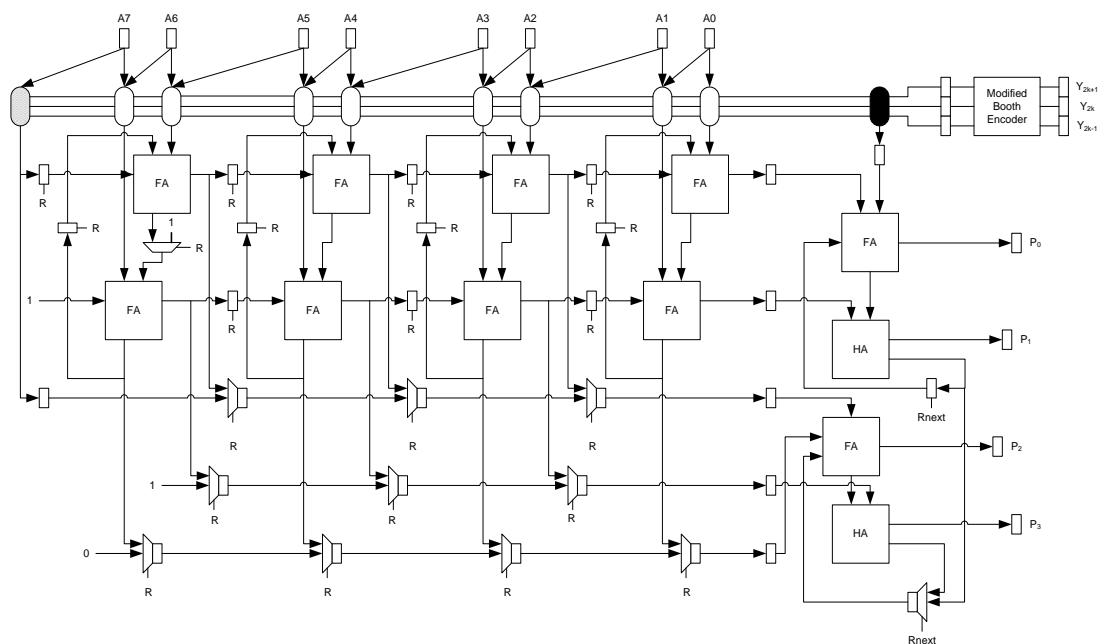


Σχήμα 3.12 Χρονική ανάλυση του radix-2 Modified Booth πολλαπλασιασμού 8x8bit ( $\beta$ )

Η μονάδα άθροισης της δεύτερης διοχέτευσης είναι ακριβώς όμοια με την less significant μονάδα άθροισης του πρώτου σταδίου, καθώς έχει να προσθέσει έναν αριθμό δύο bit (S) και έναν αριθμό ενός bit (C). Επίσης, για να διατηρηθεί η κανονικότητα του πρώτου σταδίου, αλλά και επειδή η διορθωτική μονάδα βάρους 8, δεν μπορεί να προστεθεί στο υπάρχον σχήμα, προστίθενται δύο μονάδες στο βάρος 7, η μια όπως και στα επόμενα στάδια, η δεύτερη μέσω του ενδιάμεσου κρατούμενου του 2-bit αθροιστή. Αυτό είναι εφικτό, διότι στον πρώτο κύκλο οι καταχωρητές έχουν μηδενιστεί, επομένως στον πρώτο πλήρη αθροιστή προστίθεται μόνο το ψηφίο του παραγόμενου μερικού γινομένου και ένα μηδενικό ψηφίο, οπότε δεν παράγεται κρατούμενο.

### 3.6.5 Κύκλωμα radix-2 Modified Booth σειριακού πολλαπλασιαστή

Με βάση τα παραπάνω, το τελικό κύκλωμα του 8bit radix-2 σειριακού πολλαπλασιαστή φαίνεται αμέσως παρακάτω:



**Σχήμα 3.13** Κύκλωμα σειριακού πολλαπλασιαστή Modified Booth 8bit radix-2

Το παραπάνω κύκλωμα γενικεύεται για οποιοδήποτε μήκος λέξης, παρεμβάλλοντας τον κατάλληλο αριθμό αθροιστών 2-bit στο πρώτο στάδιο, και καταχωρητών στο δεύτερο στάδιο. Η βασική δομή του κυκλώματος παραμένει σχετικά απλή, ωστόσο η ανάλυσή του παρουσιάζει κάποιες λεπτομέρειες που χρήζουν προσοχής, ιδίως όσον αφορά τον συγχρονισμό των δεδομένων. Η πιο σημαντική από αυτές είναι η χρήση δύο σημάτων ελέγχου R και Rnext, τα οποία ενεργοποιούνται ανά N/2 κύκλους και με έναν κύκλο

ρολογιού διαφορά μεταξύ τους, αποτέλεσμα της χρήσης ενός επιπλέον σταδίου άθροισης στο λιγότερο σημαντικό τμήμα του αθροιστή.

Ακολουθώντας την ανάλυση που εφαρμόσαμε και προηγουμένως, το τελικό κύκλωμα αποτελείται από  $4N+16$  καταχωρητές και παράγει το τελικό του αποτέλεσμα σε  $N$  κύκλους ρολογιού. Ωστόσο, επειδή τα δύο τμήματα του πολλαπλασιαστή εργάζονται παράλληλα, σε συνθήκες συνεχούς λειτουργίας μπορούμε να θεωρήσουμε ότι απαιτούνται μόνο  $N/2$  κύκλοι για την εξαγωγή του αποτελέσματος, καθώς σε κάθε κύκλο ρολογιού έχουμε τέσσερα έγκυρα bit αποτελέσματος, δύο για την τρέχουσα πράξη και δύο για την αμέσως προηγούμενη.

Οπότε, σε αυτή την περίπτωση, για την παραγωγή του τελικού αποτελέσματος πρέπει να εργαστούν  $4N+16$  καταχωρητές,  $N+2$  πλήρεις αθροιστές, 2 ημιαθροιστές,  $3N/2$  πολυπλέκτες,  $N$  κυκλώματα δημιουργίας μερικών γινομένων, ένα κύκλωμα κωδικοποίησης Booth, ένα κύκλωμα παραγωγής του διορθωτικού όρου και ένα κύκλωμα δημιουργίας του προσθετικού όρου επί  $N/2$  κύκλους ρολογιού.

Σύμφωνα με την παραπάνω ανάλυση, και κάνοντας τις παραδοχές του πίνακα 3.2, έχουμε ότι:

- Η επιφάνεια του radix-2 Modified Booth σειριακού πολλαπλασιαστή είναι  $48.5N+160$  πύλες NAND, περίπου ίση με του σειριακού παράλληλου πολλαπλασιαστή διπλής διοχέτευσης για μήκη λέξεως  $N$  από 8 ως 64.

**Πίνακας 3.14** Σύγκριση επιφάνειας σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2 modified booth πολλαπλασιαστή

Multiplier	registers	FA	HA	MUX	PP gen	total
Serial parallel (Double Pipeline)	40N	7N	0	6N-6	N	54N-6
Modified Booth (r2)	32N+128	7N+14	6	4.5N	5N+12	48.5N+160

- Η καθυστέρηση του radix-2 Modified Booth σειριακού πολλαπλασιαστή είναι κατά 7 πύλες NAND μεγαλύτερη από αυτή του σειριακού παράλληλου πολλαπλασιαστή διπλής διοχέτευσης.

**Πίνακας 3.15** Σύγκριση καθυστέρησης σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2 modified booth πολλαπλασιαστή

Multiplier	PP gen	FA	MUX	registers	total
Serial parallel (Double Pipeline)	1.5	4	2	2	9.5
Modified Booth (r2)	4.5	8	2	2	16.5

- Ο radix-2 Modified Booth σειριακός παράλληλος πολλαπλασιαστής φαίνεται να έχει σημαντικά καλύτερη επίδοση με βάση το κριτήριο  $area*delay$ .

**Πίνακας 3.16** Σύγκριση σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2 modified booth πολλαπλασιαστή με βάση το κριτήριο  $area \cdot delay$

Multiplier	area	delay	clock cycles	area*delay
Serial parallel (Double Pipeline)	$54N-6$	9.5	N	$513N^2-57N$
Modified Booth (r2)	$48.5N+160$	16.5	$0.5N$	$400N^2+1320N$

Το κέρδος σε κύκλους λειτουργίας και πιθανόν στην ενεργειακή κατανάλωση έχει το κόστος της μεγαλύτερης καθυστέρησης του κυκλώματος, χωρίς παράλληλα την επιβάρυνση σε επιφάνεια ολοκληρωμένου.

Γενικά, καθώς εφαρμόζουμε την παραπάνω υλοποίηση σε μεγαλύτερα radices, αναμένουμε μεγαλύτερο κέρδος από την περαιτέρω εξοικονόμηση καταχωρητών, αλλά και του σταδίου άθροισης το οποίο γίνεται πιο πολύπλοκο.

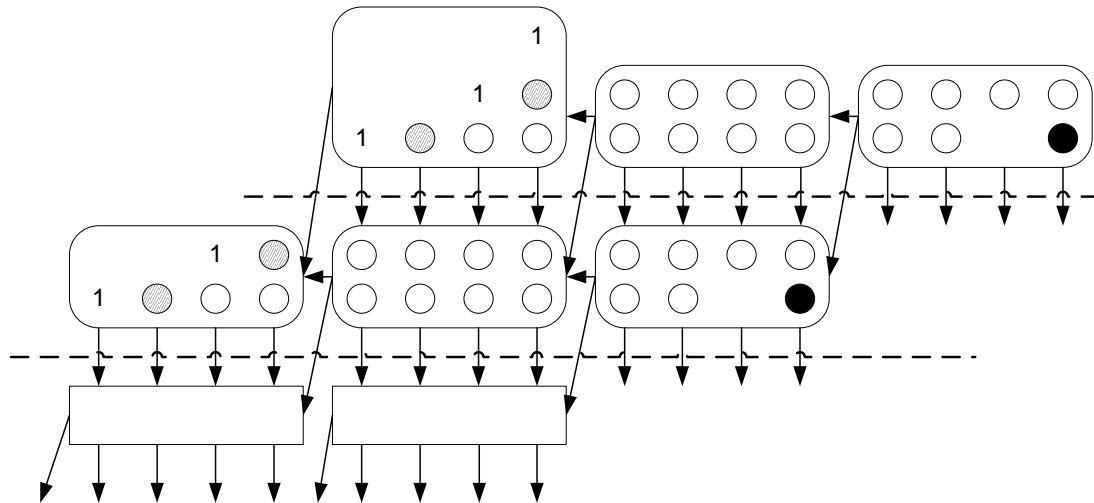
### 3.7 Radix-4 σειριακός πολλαπλασιαστής

#### 3.7.1 Γενικευμένη Carry-Save αναπαράσταση

Η γενικευμένη Carry-Save αναπαράσταση που χρησιμοποιήθηκε στην προηγούμενη παράγραφο, μπορεί να γενικευτεί περαιτέρω ώστε να επιτευχθεί μεγαλύτερη μείωση καταχωρητών. Το κόστος σε αυτή την περίπτωση θα είναι ένα μεγαλύτερο και πιο αργό κύκλωμα. Τελικός στόχος μας είναι να διερευνήσουμε ποιά μορφή έχει τα καλύτερα αποτελέσματα, ιδίως στον τομέα της ενεργειακής κατανάλωσης, χωρίς παράλληλα να θυσιάζεται η καθυστέρηση του κυκλώματος.

Το επόμενο στάδιο γενίκευσης, με βάση την διάταξη των μερικών γινομένων του σχήματος 3.7, αφορά την ταυτόχρονη παραγωγή δύο μερικών γινομένων και την ομαδοποίησή τους ανά 4 bits, ώστε να παράγονται 4 bits αποτελέσματος σε κάθε κύκλο ρολογιού.

Σε αυτή την περίπτωση, η γενικευμένη Carry-Save αναπαράσταση θα αποτελείται από ένα διάνυσμα S μήκους N και ένα διάνυσμα C μήκους  $N/4$ , όπως προκύπτουν από μια σειρά  $N/4$  αθροιστών 4 bit. Η ομαδοποίηση των bit των μερικών γινομένων που πρέπει να αθροιστούν φαίνεται στο παρακάτω σχήμα, για τον πολλαπλασιασμό 8x8 bits. Οι διακεκομμένες γραμμές υποδεικνύουν το χρονικό «κόψιμο» του κυκλώματος, ώστε να μετατραπεί σε συνεχούς διοχέτευσης.



Σχήμα 3.14 Χρονική ανάλυση του radix-4 Modified Booth πολλαπλασιασμού 8x8bit

Προφανώς τα αποτελέσματα του τελευταίου σταδίου, θα οδηγηθούν σε ένα δεύτερο κύκλωμα καταχωρητών ολίσθησης, όπου και θα αθροιστούν από έναν επιπλέον 4-bit αθροιστή.

### 3.7.2 Βασική μονάδα άθροισης

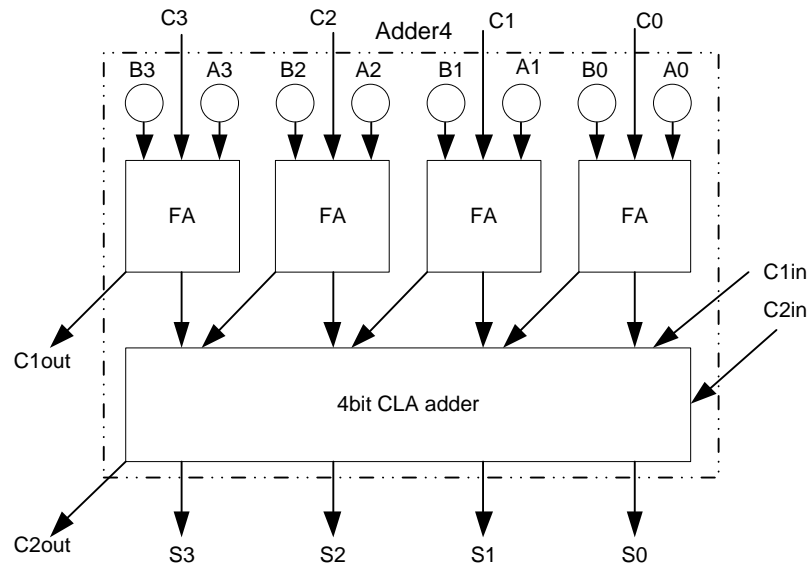
Με βάση το παραπάνω σχήμα, η βασική μονάδα άθροισης του κυκλώματος πρέπει να εκτελεί την πρόσθεση τριών 4bit αριθμών, δύο μερικών γινομένων που παράγονται στον τρέχοντα κύκλο ρολογιού και του αποτελέσματος του προηγούμενου κύκλου.

Σε αυτή την περίπτωση, η επιλογή του αθροιστή διάδοσης κρατούμενου δεν ενδείκνυται διότι θα χρειαζούν δύο τέτοια κυκλώματα συνδεδεμένα σε σειρά, αφού ο αθροιστής διάδοσης κρατούμενου μπορεί να προσθέσει μόνο δύο δυαδικούς αριθμούς, οπότε η καθυστέρηση του κυκλώματος θα γίνει απαγορευτικά μεγάλη. Γι' αυτό, θα χρησιμοποιηθεί μια τοπολογία που στοχεύει στην μικρότερη καθυστέρηση. Το κύκλωμα που θα χρησιμοποιήσουμε, αποτελείται από έναν αθροιστή σωσίματος-κρατούμενου 4bit, ο οποίος συμπιέζει τους τρεις παραπάνω αριθμούς σε Carry-Save μορφή, ώστε να προστεθούν από έναν αθροιστή πρόβλεψης κρατούμενου 4bit.

Επειδή το άθροισμα των τριών 4bit αριθμών παράγει δύο bits κρατούμενου, θα πρέπει το κύκλωμά μας να μπορεί να αθροίσει και αυτά όταν θα προέρχονται από τον προηγούμενο κύκλο. Οπότε, στην χειρότερη περίπτωση, θα πρέπει να εκτελεστεί η πράξη :

$$1111 + 1111 + 1111 + 1 + 1 = 10\ 1111$$

Η οποία απαιτεί 4bit αποτελέσματος και δύο κρατούμενα εξόδου, οπότε το κύκλωμά μας λειτουργεί ορθά με δύο κρατούμενα εισόδου και δύο κρατούμενα εξόδου.



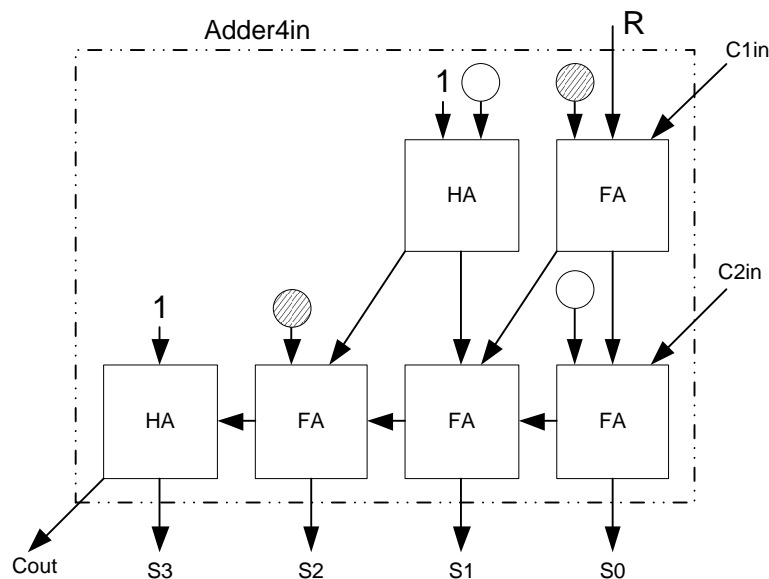
**Σχήμα 3.15** Βασική δομική μονάδα του radix-4 Modified Booth πολλαπλασιαστή

Η παραπάνω μορφή όμως δεν ανταποκρίνεται πλήρως στην επιθυμητή αναπαράσταση που θέλουμε να πετύχουμε, καθώς παράγει τέσσερα ψηφία Save και δύο ψηφία Carry, κάτι το οποίο δεν αποτελεί εξοικονόμηση σε σχέση με την radix-2 αναπαράσταση.

Παρατηρούμε όμως, ότι το πρώτο από τα δύο κρατούμενα εξόδου παράγεται με καθυστέρηση ενός μόνο πλήρους αθροιστή, οπότε μπορούμε να το μεταβιβάσουμε στο αμέσως επόμενο πιο σημαντικό στάδιο, το οποίο άλλωστε λαμβάνει το ένα του κρατούμενο αμέσως μετά το στάδιο του αθροιστή σωσίματος κρατουμένου, αποθηκεύοντας έτσι μόνο το δεύτερο κρατούμενο, το οποίο θα χρησιμοποιηθεί στον επόμενο κύκλο ρολογιού. Η λεπτομέρεια αυτή φαίνεται και στο σχήμα 3.14.

### 3.7.3 Χειρισμός διορθωτικών όρων – περισσότερο σημαντικό στάδιο άθροισης

Το πιο σημαντικό τμήμα του πολλαπλασιαστή πρέπει να αθροίσει δυο ψηφία μερικών γινομένων, καθώς και δύο bits επέκτασης προσήμου, μαζί με κάποιες διορθωτικές μονάδες. Θέλουμε αυτό το στάδιο να παραμείνει όσο το δυνατόν απλό, χωρίς να επιβαρύνει το κρίσιμο μονοπάτι του κυκλώματος. Το κύκλωμα που χρησιμοποιήσαμε για αυτό το κύκλωμα φαίνεται αμέσως παρακάτω:



Σχήμα 3.16 Περισσότερο σημαντική μονάδα άθροισης του radix-4 Modified Booth πολλαπλασιαστή

Το παραπάνω κύκλωμα είναι το απλούστερο δυνατό για την πράξη που θέλουμε να εκτελέσουμε. Το πρώτο κρατούμενο εισόδου είναι απαραίτητο για το κρατούμενο που προκύπτει από τον προηγούμενο κύκλο ρολογιού, ενώ το δεύτερο υπάρχει για να δεχτεί το πρώτο από τα δύο κρατούμενα του προηγούμενου σταδίου, το οποίο θα παραχθεί από τον αντίστοιχο αθροιστή σωσίματος-κρατούμενου.

Στην χειρότερη περίπτωση, θα πρέπει να εκτελεστεί η πράξη:

$$1111 + 11 + 1 + 1 + 1 = 10101$$

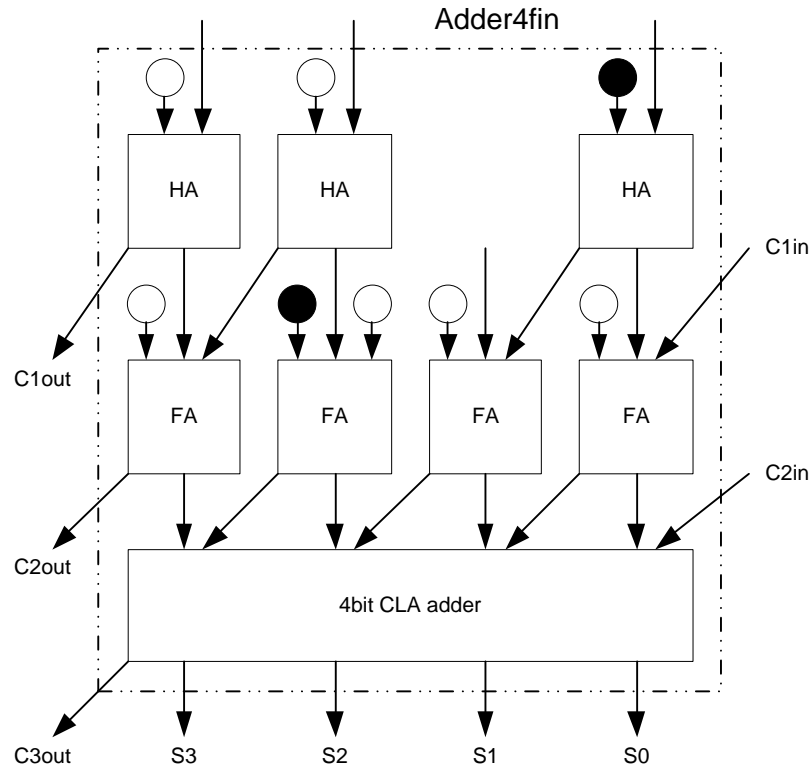
Από την οποία φαίνεται ότι χρειαζόμαστε ένα μόνο κρατούμενο εξόδου.

Επίσης, παρόλο που το παραπάνω κύκλωμα ουσιαστικά αποτελεί έναν αθροιστή διάδοσης κρατούμενου 5bit, η καθυστέρησή του στην πραγματικότητα είναι πολύ μικρότερη, καθώς μπορούν να γίνουν πολλές βελτιστοποιήσεις κατά την υλοποίηση. Για παράδειγμα, ο τελευταίος ημιαθροιστής που απλά προσθέτει πάντα μια μονάδα στο τελευταίο bit του αποτελέσματος, μπορεί να αντικατασταθεί από έναν αντιστροφέα. Επίσης, η χρήση πλήρων αθροιστών με γρήγορο μονοπάτι κρατούμενου μειώνει το κρίσιμο μονοπάτι του κυκλώματος. η επιλογή του κυκλώματος αυτού τεκμηριώνεται από το γεγονός ότι το κρίσιμο μονοπάτι του κυκλώματος όπως προκύπτει από τις προσομοιώσεις, δεν βρίσκεται σε αυτό το στάδιο.

### 3.7.4 Λιγότερο σημαντική μονάδα άθροισης

Οι δύο προηγούμενες μονάδες, δεν είναι κατάλληλες για το λιγότερο σημαντικό τμήμα του πολλαπλασιαστή. Σε αυτή την τοπολογία δεν θα χρησιμοποιήσουμε επιπλέον κύκλωμα άθροισης για να προσθέσουμε τους διορθωτικούς όρους, αλλά θα ενσωματώσουμε τον υπολογισμό τους στο ίδιο κύκλωμα, διότι θα είναι αρκετά μεγάλη σπατάλη πόρων η χρήση

ενός επιπλέον σταδίου άθροισης, ενώ μπορούμε να πετύχουμε το ίδιο αποτέλεσμα με πολύ μικρή ή και καθόλου επιβάρυνση στο κρίσιμο μονοπάτι του κυκλώματος. Το κύκλωμα που χρησιμοποιήσαμε σε αυτό το στάδιο φαίνεται αμέσως παρακάτω:



**Σχήμα 3.17** Λιγότερο σημαντική μονάδα άθροισης του radix-4 Modified Booth πολλαπλασιαστή

Το παραπάνω κύκλωμα, όπως και η βασική μονάδα αυτής της τοπολογίας, χρησιμοποιεί ένα δίκτυο ημιαθροιστών και πλήρων αθροιστών για την γρήγορη παραγωγή ενός Carry-Save αποτελέσματος το οποίο θα αθροιστεί από έναν αθροιστή πρόβλεψης κρατούμενου.

Στην χειρότερη περίπτωση, θα πρέπει να εκτελεστεί η πράξη:

$$1111 + 1111 + 1100 + 0101 + 1 = 11\ 0001$$

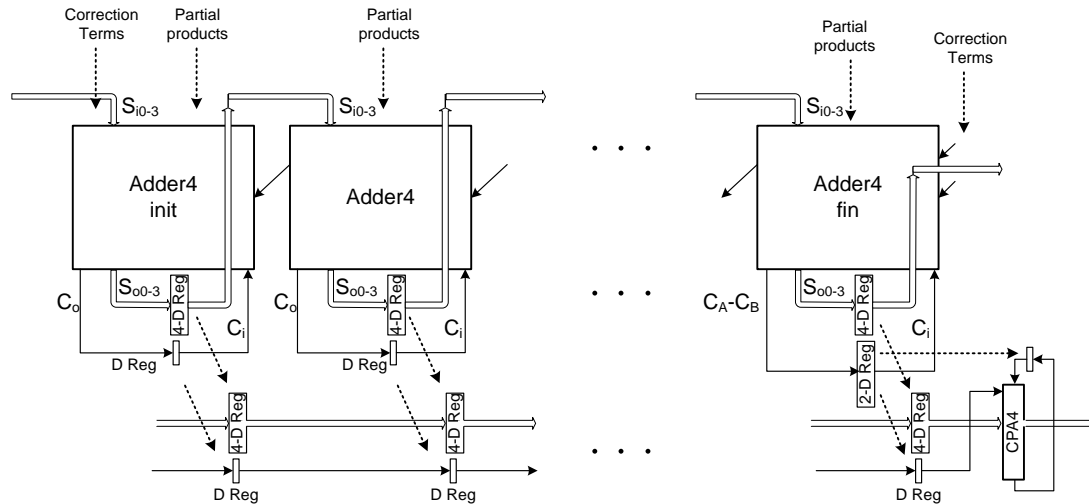
Από την οποία βλέπουμε ότι ακόμα και δύο κρατούμενα εξόδου δεν είναι αρκετά για την ορθή απεικόνιση του αποτελέσματος. Οπότε, θα χρειαστούμε τρία κρατούμενα εξόδου, και αφού το πιο γρήγορο από αυτά θα μεταβιβαστεί στο επόμενο πιο σημαντικό στάδιο, μένουν δύο επιπλέον, τα οποία πρέπει να αποθηκευτούν και να χρησιμοποιηθούν στον επόμενο κύκλο, όπως φαίνεται στο παραπάνω σχήμα.

Παρόλο που το κύκλωμα αυτό φαίνεται να έχει μεγαλύτερη καθυστέρηση από την βασική μονάδα, αυτό δεν είναι τελείως αληθές, διότι ο τελικός αθροιστής πρόβλεψης κρατούμενου δεν έχει κρατούμενο εισόδου, οπότε είναι κατά ένα στάδιο πιο γρήγορος από αυτόν του βασικού κυκλώματος, οπότε ισοφαρίζει περίπου την καθυστέρηση της μιας επιπλέον πύλης του ημιαθροιστή.



### 3.7.5 Κύκλωμα radix-4 Modified Booth σειριακού πολλαπλασιαστή

Με βάση τα παραπάνω, το τελικό κύκλωμα του radix-4 Modified Booth σειριακού πολλαπλασιαστή φαίνεται αμέσως παρακάτω:



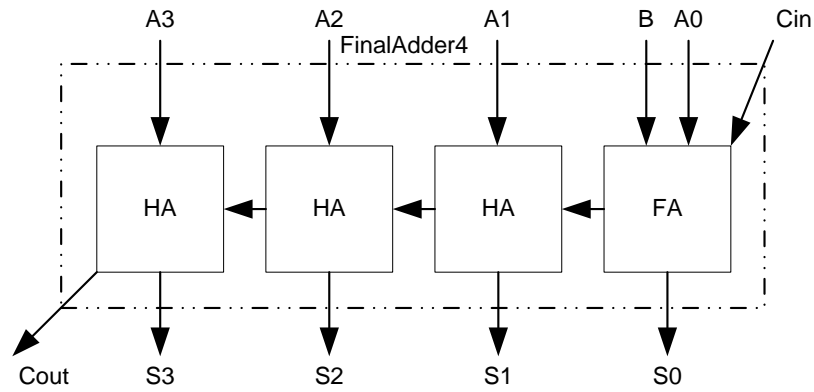
Σχήμα 3.18 Κύκλωμα σειριακού πολλαπλασιαστή Modified Booth radix-4

Το παραπάνω κύκλωμα ισχύει για οποιοδήποτε μήκος λέξης  $N$ , με  $N$  πολλαπλάσιο του τέσσερα, παρεμβάλλοντας ενδιάμεσα τις κατάλληλες βασικές μονάδες άθροισης. Για την ορθή λειτουργία του απαιτείται μόνο ένα σήμα ελέγχου  $R$ , το οποίο ενεργοποιείται ανά  $N/4$  κύκλους ρολογιού, ώστε να μεταφερθούν τα ενδιάμεσα αποτελέσματα στους καταχωρητές ολίσθησης.

Τα μερικά γινόμενα παράγονται από δύο κυκλώματα κωδικοποίησης Modified Booth και δύο σειρές  $N$  κυκλωμάτων παραγωγής μερικών γινομένων.

Για την παραγωγή του τελικού αποτελέσματος, το κύκλωμα πρέπει να εργαστεί επί  $N/2$  κύκλους ρολογιού. Ωστόσο, σε συνθήκες συνεχούς λειτουργίας, θεωρούμε ότι το κύκλωμα παράγει 8 bit αποτελέσματος ανά κύκλο ρολογιού οπότε ολοκληρώνει μια πράξη σε  $N/4$  κύκλους.

Η μονάδα άθροισης της δεύτερης διοχέτευσης είναι ένας αθροιστής διάδοσης κρατουμένου, ο οποίος όμως χρησιμοποιεί ημιαθροιστές αντί για πλήρεις αθροιστές, διότι έχει να προσθέσει έναν 4bit αριθμό και δύο bits κρατουμένου. Ως εκ τούτου η καθυστέρησή του είναι μικρότερη από το κρίσιμο μονοπάτι του κυκλώματος.



**Σχήμα 3.19** Μονάδα άθροισης του δεύτερου σταδίου του radix-4 Modified Booth πολλαπλασιαστή

Ακολουθώντας την ανάλυση που εφαρμόσαμε και προηγουμένως, το τελικό κύκλωμα αποτελείται από  $3.5N+22$  καταχωρητές και παράγει το τελικό του αποτέλεσμα σε  $N/2$  κύκλους ρολογιού. Ωστόσο, επειδή τα δύο τμήματα του πολλαπλασιαστή εργάζονται παράλληλα, σε συνθήκες συνεχούς λειτουργίας μπορούμε να θεωρήσουμε ότι απαιτούνται μόνο  $N/4$  κύκλοι για την εξαγωγή του αποτελέσματος, καθώς σε κάθε κύκλο ρολογιού έχουμε οκτώ έγκυρα bit αποτελέσματος, τέσσερα για την τρέχουσα πράξη και τέσσερα για την αμέσως προηγούμενη.

Οπότε, σε αυτή την περίπτωση, για την παραγωγή του τελικού αποτελέσματος πρέπει να εργαστούν  $3.5N+22$  καταχωρητές,  $N+4$  πλήρεις αθροιστές, 5 ημιαθροιστές,  $N/4$  αθροιστές πρόβλεψης κρατούμενου 4-bit,  $1.25N+1$  πολυπλέκτες,  $2N$  κυκλώματα δημιουργίας μερικών γινομένων, δύο κυκλώματα κωδικοποίησης Booth, δύο κυκλώματα παραγωγής διορθωτικών όρων και δύο κυκλώματα δημιουργίας των προσθετικών όρων επί  $N/4$  κύκλους ρολογιού.

Σύμφωνα με την παραπάνω ανάλυση, και κάνοντας τις παραδοχές του πίνακα 3.2, έχουμε ότι:

- Η επιφάνεια του radix-4 Modified Booth σειριακού πολλαπλασιαστή είναι  $58.75N+246$  πύλες NAND, περίπου 20% μεγαλύτερη από αυτή του radix-2 Modified Booth σειριακού πολλαπλασιαστή.

**Πίνακας 3.17** Σύγκριση επιφάνειας σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2/radix-4 modified booth πολλαπλασιαστών

Multiplier	registers	MUX	FA/HA	CLA	PP gen	total
Serial parallel (Double Pipeline)	$40N$	$6N-6$	$7N$	0	$N$	$54N-6$
Modified Booth (r2)	$32N+128$	$4.5N$	$7N+20$	0	$5N+12$	$48.5N+160$
Modified Booth (r4)	$28N+176$	$3.75N+3$	$7N+43$	$10N$	$10N+24$	$58.75N+246$

- Η καθυστέρηση του radix-4 Modified Booth σειριακού πολλαπλασιαστή είναι κατά 6 πύλες NAND μεγαλύτερη από αυτή του radix-2 Modified Booth σειριακού πολλαπλασιαστή.

**Πίνακας 3.18** Σύγκριση καθυστέρησης σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2/radix-4 modified booth πολλαπλασιαστών

Multiplier	PP gen	FA	CLA	MUX	registers	total
Serial parallel (Double Pipeline)	1.5	4	0	2	2	9.5
Modified Booth (r2)	4.5	8	0	2	2	16.5
Modified Booth (r4)	4.5	4	10	2	2	22.5

- Ο radix-4 σειριακός Modified Booth παράλληλος πολλαπλασιαστής φαίνεται να έχει σημαντικά καλύτερη επίδοση με βάση το κριτήριο  $area*delay$ .

**Πίνακας 3.19** Σύγκριση σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2/radix-4 modified booth πολλαπλασιαστών με βάση το κριτήριο  $area*delay$

Multiplier	area	delay	clock cycles	$area*delay$
Serial parallel (Double Pipeline)	$54N-6$	9.5	N	$513N^2-57N$
Modified Booth (r2)	$48.5N+160$	16.5	0.5N	$400N^2+1320N$
Modified Booth (r4)	$58.75N+246$	22.5	0.25N	$331N^2+1384N$

Το κέρδος σε κύκλους λειτουργίας και πιθανόν στην ενεργειακή κατανάλωση έχει το κόστος της μεγαλύτερης καθυστέρησης του κυκλώματος και της μεγαλύτερης επιφάνειας ολοκληρωμένου.

Γενικά, καθώς εφαρμόζουμε την παραπάνω υλοποίηση σε μεγαλύτερα radices, αναμένουμε μεγαλύτερο κέρδος από την περαιτέρω εξοικονόμηση καταχωρητών, αλλά και του σταδίου άθροισης το οποίο γίνεται πιο πολύπλοκο.

## 3.8 Radix-8 Modified Booth σειριακός πολλαπλασιαστής

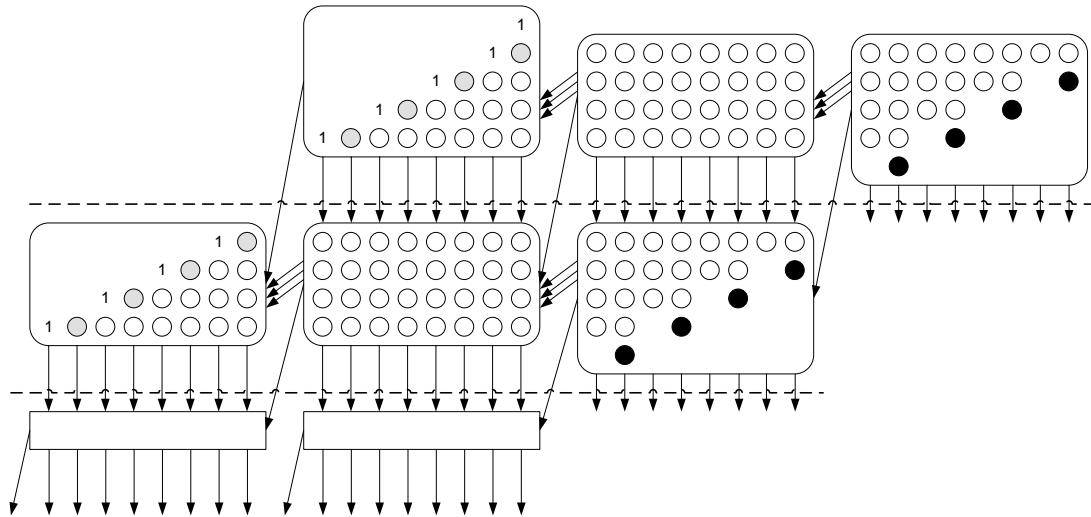
### 3.8.1 Γενικευμένη Carry-Save αναπαράσταση

Τα αποτελέσματα της προηγούμενης παραγράφου μπορούν να γενικευτούν με ανάλογο τρόπο και για υψηλότερα radices.

Το επόμενο στάδιο γενίκευσης αφορά την ταυτόχρονη παραγωγή τεσσάρων μερικών γινομένων και την ομαδοποίησή τους ανά 8 bits, ώστε να παράγονται 8 bits αποτελέσματος σε κάθε κύκλο ρολογιού. Προφανώς, μια τέτοια τοπολογία έχει νόημα για μήκη λέξεως μεγαλύτερα των 16 bits.

Σε αυτή την περίπτωση, η γενικευμένη Carry-Save αναπαράσταση θα αποτελείται από ένα διάνυσμα S μήκους N και ένα διάνυσμα C μήκους N/8, όπως προκύπτουν από μια σειρά N/8

αθροιστών 8 bit. Η ομαδοποίηση των bit των μερικών γινομένων που πρέπει να αθροιστούν φαίνεται στο παρακάτω σχήμα, για την απλούστερη περίπτωση του πολλαπλασιασμού 16x16 bits. Οι διακεκομμένες γραμμές υποδεικνύουν το χρονικό «κόψιμο» του κυκλώματος, ώστε να μετατραπεί σε συνεχούς διοχέτευσης.



Σχήμα 3.19 Χρονική ανάλυση του radix-8 Modified Booth πολλαπλασιασμού 16x16bit

Προφανώς τα αποτελέσματα του τελευταίου σταδίου, θα οδηγηθούν σε ένα δεύτερο κύκλωμα καταχωρητών ολίσθησης, όπου και θα αθροιστούν από έναν επιπλέον 8-bit αθροιστή.

### 3.8.2 Βασική μονάδα άθροισης

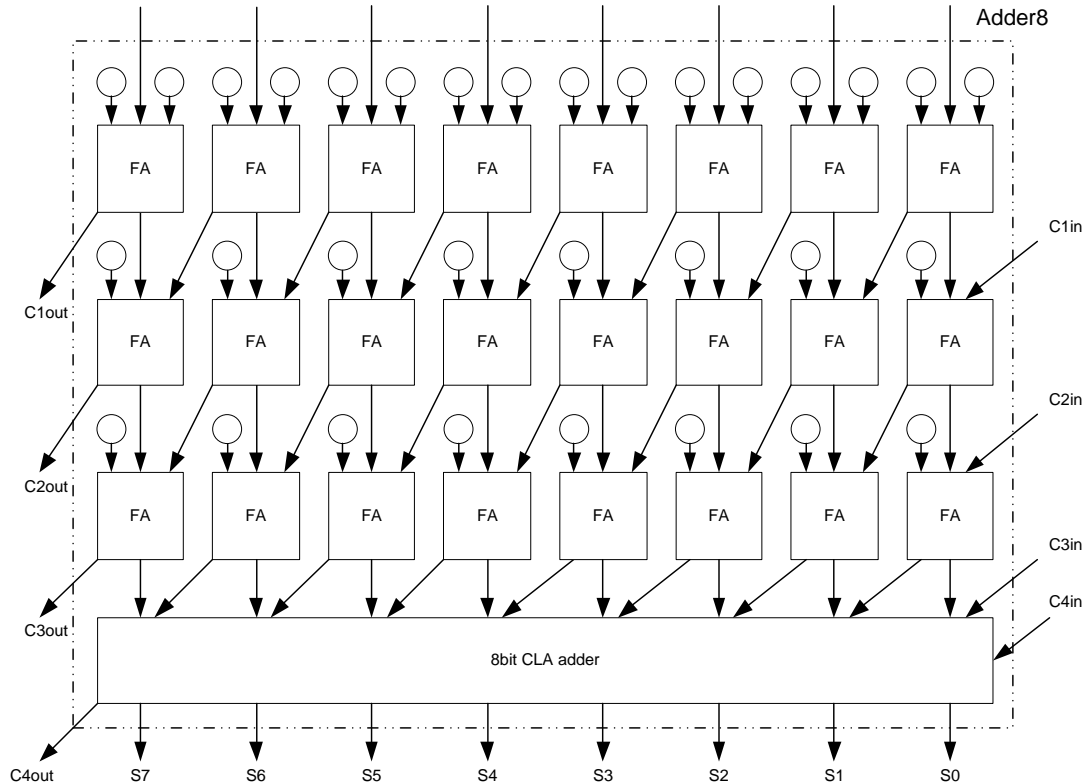
Με βάση το παραπάνω σχήμα, η βασική μονάδα άθροισης του κυκλώματος πρέπει να εκτελεί την πρόσθεση πέντε 8bit αριθμών, τεσσάρων μερικών γινομένων που παράγονται στον τρέχοντα κύκλο ρολογιού και του αποτελέσματος του προηγούμενου κύκλου.

Το κύκλωμα που θα χρησιμοποιήσουμε, αποτελείται από ένα δίκτυο τριών σταδίων πλήρων αθροιστών σε διάταξη αθροιστών σωσίματος-κρατούμενου 4bit, ο οποίος συμπιέζει τους πέντε παραπάνω αριθμούς σε Carry-Save μορφή, ώστε να προστεθούν από έναν αθροιστή πρόβλεψης κρατούμενου 8bit.

Επειδή το άθροισμα των πέντε 8bit αριθμών παράγει τέσσερα bits κρατούμενου, θα πρέπει το κύκλωμά μας να μπορεί να αθροίσει και αυτά όταν θα προέρχονται από τον προηγούμενο κύκλο. Οπότε, στην χειρότερη περίπτωση, θα πρέπει να εκτελεστεί η πράξη :

$$11111111 + 11111111 + 11111111 + 11111111 + 11111111 + 1 + 1 + 1 + 1 = 100\ 11111110$$

Η οποία απαιτεί 8bit αποτελέσματος και τέσσερα κρατούμενα εξόδου, οπότε το κύκλωμά μας λειτουργεί ορθά με τέσσερα κρατούμενα εισόδου και τέσσερα κρατούμενα εξόδου.



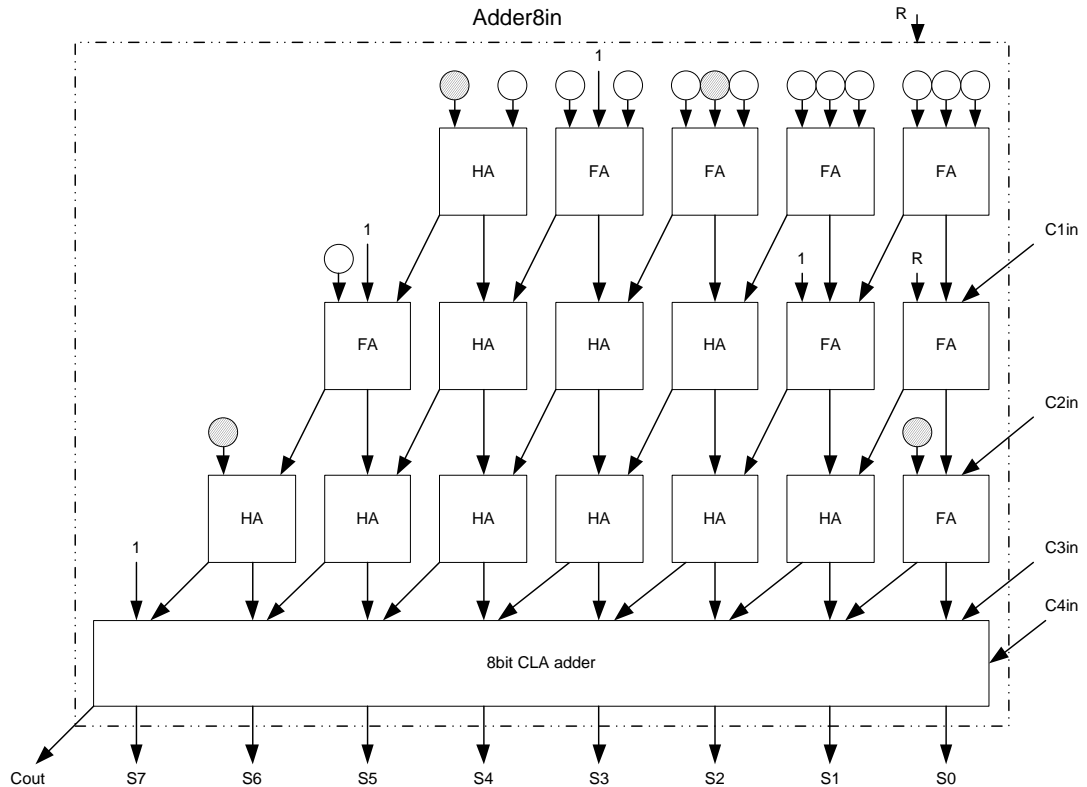
**Σχήμα 3.20** Βασική δομική μονάδα του radix-8 Modified Booth πολλαπλασιαστή

Η παραπάνω μορφή όμως δεν ανταποκρίνεται πλήρως στην επιθυμητή αναπαράσταση που θέλουμε να πετύχουμε, καθώς παράγει οκτώ ψηφία Save και τέσσερα ψηφία Carry, κάτι το οποίο δεν αποτελεί εξοικονόμηση σε σχέση με την radix-2 αναπαράσταση.

Παρατηρούμε όμως, ότι κάθε ένα από τα τρία πρώτα κρατούμενα εξόδου παράγεται με καθυστέρηση ενός πλήρους αθροιστή, σε σχέση με το προηγούμενο, οπότε μπορούμε να το μεταβιβάσουμε στο αμέσως επόμενο πιο σημαντικό στάδιο, στην αντίστοιχη βαθμίδα, αποθηκεύοντας έτσι μόνο το τελευταίο κρατούμενο, το οποίο θα χρησιμοποιηθεί στον επόμενο κύκλο ρολογιού. Η λεπτομέρεια αυτή φαίνεται και στο σχήμα 3.19.

### 3.8.3 Χειρισμός διορθωτικών όρων – περισσότερο σημαντικό στάδιο άθροισης

Το πιο σημαντικό τμήμα του πολλαπλασιαστή πρέπει να αθροίσει τρία σύνολα μερικών γινομένων, τέσσερα bits επέκτασης προσήμου, καθώς και τις αντίστοιχες διορθωτικές μονάδες. Το κύκλωμα που χρησιμοποιήσαμε είναι όμοιο με αυτό της βασικής δομικής μονάδας, μόνο που χρησιμοποιεί ένα μικρότερο σε επιφάνεια δένδρο Carry-Save άθροισης. Επειδή το βάθος του δένδρου αυτού είναι το ίδιο με της βασικής μονάδας, δεν έχουμε επιβάρυνση στο κρίσιμο μονοπάτι του κυκλώματος. Το κύκλωμα που υλοποιήσαμε φαίνεται αμέσως παρακάτω:



**Σχήμα 3.21** Περισσότερο σημαντική μονάδα άθροισης του radix-8 Modified Booth πολλαπλασιαστή

Το παραπάνω κύκλωμα είναι παρόμοιας δομής με τα υπόλοιπα στάδια του πολλαπλασιαστή. Το πρώτο κρατούμενο εισόδου είναι απαραίτητο για το κρατούμενο που προκύπτει από τον προηγούμενο κύκλο ρολογιού, ενώ τα τρία επόμενα υπάρχουν για να δεχτεί τα τρία πρώτα από τα τέσσερα κρατούμενα του προηγούμενου σταδίου, τα οποία θα παραχθούν από τον αντίστοιχο αθροιστή σωσίματος-κρατουμένου.

Στην χειρότερη περίπτωση, θα πρέπει να εκτελεστεί η πράξη:

$$111111 + 1111 + 11 + 11111111 + 1 + 1 + 1 + 1 + 1 = 1\ 01010101$$

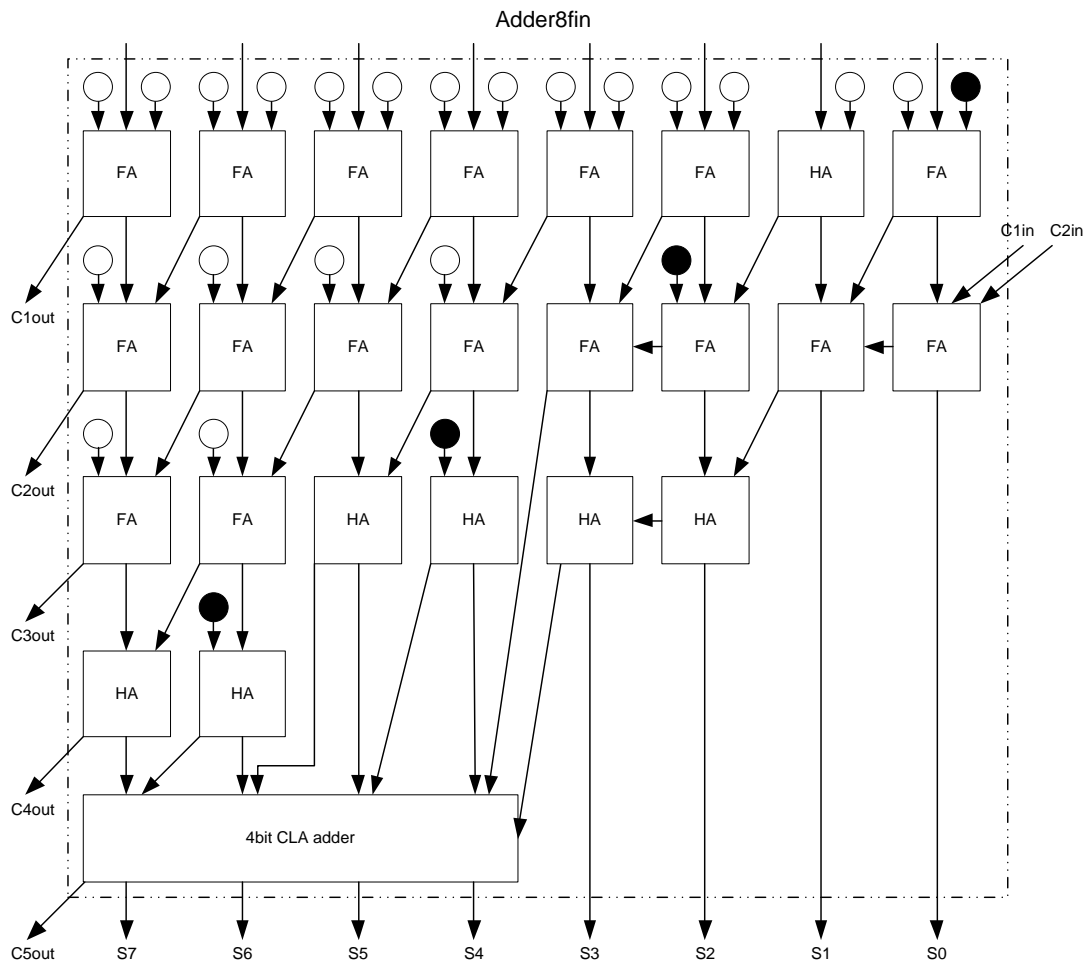
Από την οποία φαίνεται ότι χρειαζόμαστε ένα μόνο κρατούμενο εξόδου.

Κατά την υλοποίηση του παραπάνω κυκλώματος μπορούν να γίνουν επιπλέον βελτιστοποιήσεις. Για παράδειγμα, οι πλήρεις αθροιστές που έχουν πάντα μια μονάδα ως είσοδο, μπορούν να απλοποιηθούν λογικά σε ισοδύναμους ημιαθροιστές.

### 3.8.4 Λιγότερο σημαντική μονάδα άθροισης

Το λιγότερο σημαντικό τμήμα του πολλαπλασιαστή είναι ανάλογο με αντίστοιχο του radix-4 πολλαπλασιαστή. Το κύκλωμα αυτό κάνει χρήση ενός δικτύου αθροιστών και ημιαθροιστών για την γρήγορη παραγωγή ενός Carry-Save αποτελέσματος το οποίο θα αθροιστεί από έναν αθροιστή πρόβλεψης κρατουμένου.

Οι δύο προηγούμενες μονάδες, δεν είναι κατάλληλες για το λιγότερο σημαντικό τμήμα του πολλαπλασιαστή. Σε αυτή την τοπολογία δεν θα χρησιμοποιήσουμε επιπλέον κύκλωμα άθροισης για να προσθέσουμε τους διορθωτικούς όρους, αλλά θα ενσωματώσουμε τον υπολογισμό τους στο ίδιο κύκλωμα, διότι θα είναι αρκετά μεγάλη σπατάλη πόρων η χρήση ενός επιπλέον σταδίου άθροισης, ενώ μπορούμε να πετύχουμε το ίδιο αποτέλεσμα με πολύ μικρή ή και καθόλου επιβάρυνση στο κρίσιμο μονοπάτι του κυκλώματος. Το κύκλωμα που χρησιμοποιήσαμε σε αυτό το στάδιο είναι ανάλογο με αντίστοιχο του radix-4 πολλαπλασιαστή και φαίνεται αμέσως παρακάτω:



**Σχήμα 3.22** Λιγότερο σημαντική μονάδα άθροισης του radix-8 Modified Booth πολλαπλασιαστή

Το παραπάνω κύκλωμα, όπως και η βασική μονάδα αυτής της τοπολογίας, χρησιμοποιεί ένα δίτυο ημιαθροιστών και πλήρων αθροιστών για την γρήγορη παραγωγή ενός Carry-Save αποτελέσματος το οποίο θα αθροιστεί από έναν αθροιστή πρόβλεψης κρατούμενου. Ωστόσο, η ιδιαίτερη μορφή των επιμέρους όρων μας επιτρέπει να τροποποιήσουμε την λογική του κυκλώματος, εκμεταλλευόμενοι το γεγονός ότι οι λιγότερο σημαντικοί όροι μπορούν να παραχθούν πολύ γρήγορα σε δυαδική μορφή, οπότε να χρησιμοποιήσουμε έναν μικρότερο

αθροιστή πρόβλεψης κρατούμενου για τον υπολογισμό του περισσότερου σημαντικού τμήματος, το οποίο βρίσκεται σε Carry-Save μορφή.

Στην χειρότερη περίπτωση, θα πρέπει να εκτελεστεί η πράξη:

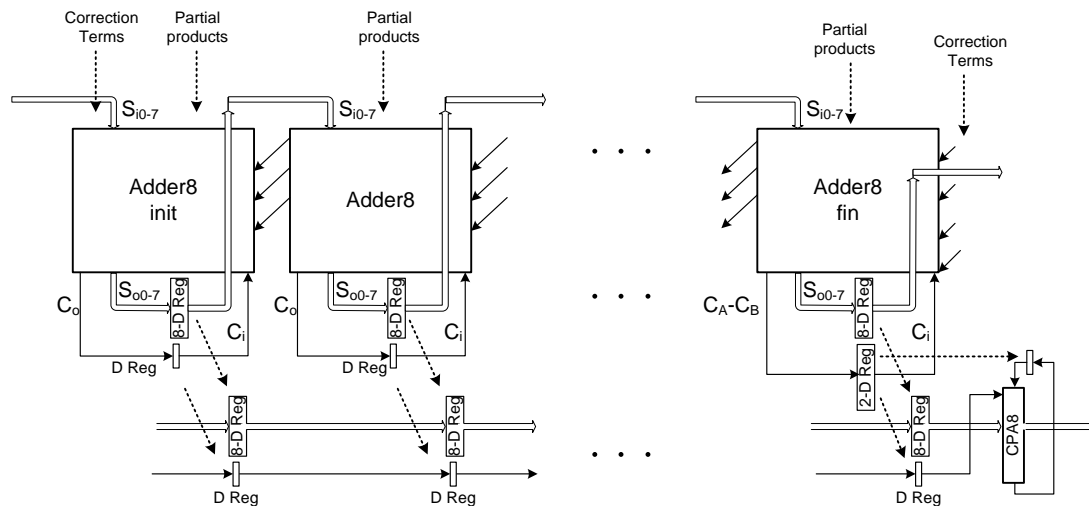
$$11111111 + 11111111 + 11111100 + 11110000 + 11000000 + 01010101 + 1 = 1010000000$$

Από την οποία βλέπουμε ότι χρειάζονται πέντε κρατούμενα για την ορθή απεικόνιση του αποτελέσματος. Οπότε, τα τρία γρήγορα κρατούμενα θα μεταβιβαστούν στα αντίστοιχα στάδια του αμέσως πιο σημαντικού τμήματος, ενώ τα δύο τελευταία θα αποθηκευτούν για να χρησιμοποιηθούν στον επόμενο κύκλο, όπως φαίνεται στο παραπάνω σχήμα.

Παρόλο που το κύκλωμα αυτό φαίνεται να έχει μεγαλύτερη καθυστέρηση από την βασική μονάδα, αυτό δεν είναι τελειώς αληθές, διότι ο τελικός αθροιστής πρόβλεψης κρατούμενου έχει το μισό μήκος από αυτόν του βασικού σταδίου, οπότε είναι κατά ένα στάδιο πιο γρήγορος από αυτόν, οπότε ισοφαρίζει περίπου την καθυστέρηση της μιας πύλης του επιπλέον ημιαθροιστή.

### 3.8.5 Κύκλωμα radix-8 Modified Booth σειριακού πολλαπλασιαστή

Με βάση τα παραπάνω, το τελικό κύκλωμα του radix-8 σειριακού πολλαπλασιαστή φαίνεται αμέσως παρακάτω:



Σχήμα 3.23 Κύκλωμα σειριακού πολλαπλασιαστή Modified Booth radix-8

Το παραπάνω κύκλωμα ισχύει για οποιοδήποτε μήκος λέξης N, με N πολλαπλάσιο του οκτώ, παρεμβάλλοντας ενδιάμεσα τις κατάλληλες βασικές μονάδες άθροισης. Για την ορθή λειτουργία του απαιτείται μόνο ένα σήμα ελέγχου R, το οποίο ενεργοποιείται ανά N/8

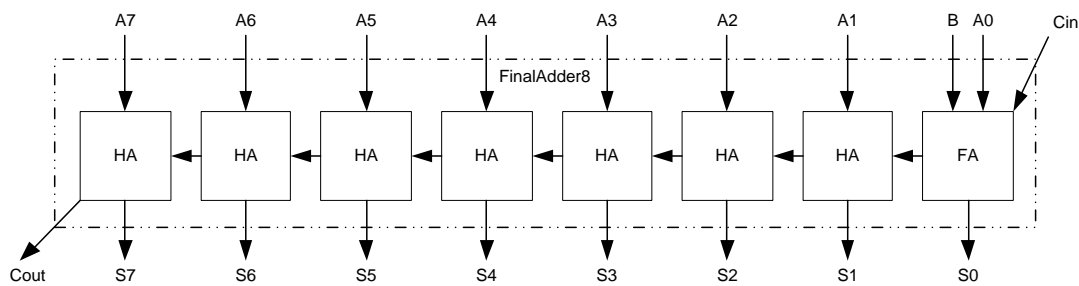


κύκλους ρολογιού, ώστε να μεταφερθούν τα ενδιάμεσα αποτελέσματα στους καταχωρητές ολίσθησης.

Τα μερικά γινόμενα παράγονται από τέσσερα κυκλώματα κωδικοποίησης Modified Booth και τέσσερις σειρές  $N$  κυκλωμάτων παραγωγής μερικών γινομένων.

Για την παραγωγή του τελικού αποτελέσματος, το κύκλωμα πρέπει να εργαστεί επί  $N/4$  κύκλους ρολογιού. Ωστόσο, σε συνθήκες συνεχούς λειτουργίας, θεωρούμε ότι το κύκλωμα παράγει 16 bit αποτελέσματος ανά κύκλο ρολογιού οπότε ολοκληρώνει μια πράξη σε  $N/8$  κύκλους.

Η μονάδα άθροισης της δεύτερης διοχέτευσης είναι ένας αθροιστής διάδοσης κρατουμένου, ο οποίος όμως χρησιμοποιεί ημιαθροιστές αντί για πλήρεις αθροιστές, διότι έχει να προσθέσει έναν 8bit αριθμό και δύο bits κρατουμένου. Ως εκ τούτου η καθυστέρησή του είναι μικρότερη από το κρίσιμο μονοπάτι του κυκλώματος.



**Σχήμα 3.24** Μονάδα άθροισης του δεύτερου σταδίου του radix-8 πολλαπλασιαστή

Ακολουθώντας την ανάλυση που εφαρμόσαμε και προηγουμένως, το τελικό κύκλωμα αποτελείται από  $3.25N+37$  καταχωρητές και παράγει το τελικό του αποτέλεσμα σε  $N/4$  κύκλους ρολογιού. Ωστόσο, επειδή τα δύο τμήματα του πολλαπλασιαστή εργάζονται παράλληλα, σε συνθήκες συνεχούς λειτουργίας μπορούμε να θεωρήσουμε ότι απαιτούνται μόνο  $N/8$  κύκλοι για την εξαγωγή του αποτελέσματος, καθώς σε κάθε κύκλο ρολογιού έχουμε δεκαέξι έγκυρα bit αποτελέσματος, οκτώ για την τρέχουσα πράξη και οκτώ για την αμέσως προηγούμενη.

Οπότε, σε αυτή την περίπτωση, για την παραγωγή του τελικού αποτελέσματος πρέπει να εργαστούν  $3.25N+37$  καταχωρητές,  $3N+2$  πλήρεις αθροιστές, 24 ημιαθροιστές, ένας αθροιστής πρόβλεψης κρατουμένου 4-bit,  $N/8$  αθροιστές πρόβλεψης κρατουμένου 8-bit  $1.125N+1$  πολυπλέκτες,  $4N$  κυκλώματα δημιουργίας μερικών γινομένων, τέσσερα κυκλώματα κωδικοποίησης Booth, τέσσερα κυκλώματα παραγωγής διορθωτικών όρων και τέσσερα κυκλώματα δημιουργίας των προσθετικών όρων επί  $N/8$  κύκλους ρολογιού.

Σύμφωνα με την παραπάνω ανάλυση, και κάνοντας τις παραδοχές του πίνακα 3.2, έχουμε ότι:

- Η επιφάνεια του radix-8 Modified Booth σειριακού πολλαπλασιαστή είναι  $81.88N+473$  πύλες NAND, περίπου 40% μεγαλύτερη από αυτή του radix-4 σειριακού πολλαπλασιαστή.

**Πίνακας 3.20** Σύγκριση επιφάνειας σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2/radix-4/radix-8 modified booth πολλαπλασιαστών

Multiplier	registers	MUX	FA/HA	CLA	PP gen	total
Serial parallel (Double Pipeline)	40N	6N-6	7N	0	N	54N-6
Modified Booth (r2)	32N+128	4.5N	7N+20	0	5N+12	48.5N+160
Modified Booth (r4)	28N+176	3.75N+3	7N+43	10N	10N+24	58.75N+246
Modified Booth (r8)	26N+296	3.375N+3	21N+86	11.5N+40	20N+48	81.88N+473

- Η καθυστέρηση του radix-8 Modified Booth σειριακού πολλαπλασιαστή είναι κατά 10 πύλες NAND μεγαλύτερη από αυτή του radix-4 σειριακού πολλαπλασιαστή.

**Πίνακας 3.21** Σύγκριση καθυστέρησης σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2/radix-4/ radix-8 modified booth πολλαπλασιαστών

Multiplier	PP gen	FA	CLA	MUX	registers	total
Serial parallel (Double Pipeline)	1.5	4	0	2	2	9.5
Modified Booth (r2)	4.5	8	0	2	2	16.5
Modified Booth (r4)	4.5	4	10	2	2	22.5
Modified Booth (r8)	4.5	12	12	2	2	32.5

- Ο radix-8 Modified Booth σειριακός παράλληλος πολλαπλασιαστής φαίνεται να έχει ελαφρώς χειρότερες επιδόσεις από τον radix-4 σειριακό παράλληλο πολλαπλασιαστή, με βάση το κριτήριο  $area*delay$ . Ωστόσο, αυτό δεν επιβεβαιώνεται από τα πειραματικά αποτελέσματα.

**Πίνακας 3.22** Σύγκριση σειριακού πολλαπλασιαστή με διπλή διοχέτευση και radix-2/radix-4/radix-8 modified booth πολλαπλασιαστών με βάση το κριτήριο  $area*delay$

Multiplier	area	delay	clock cycles	area*delay
Serial parallel (Double Pipeline)	54N-6	9.5	N	513N <sup>2</sup> -57N
Modified Booth (r2)	48.5N+160	16.5	0.5N	400N <sup>2</sup> +1320N
Modified Booth (r4)	58.75N+246	22.5	0.25N	331N <sup>2</sup> +1384N
Modified Booth (r8)	81.88N+473	32.5	0.125N	333N <sup>2</sup> +1775N

Το κέρδος σε κύκλους λειτουργίας και πιθανόν στην ενεργειακή κατανάλωση έχει το κόστος της μεγαλύτερης καθυστέρησης του κυκλώματος και της μεγαλύτερης επιφάνειας ολοκληρωμένου.

Γενικά, καθώς εφαρμόζουμε την παραπάνω υλοποίηση σε μεγαλύτερα radices, αναμένουμε μεγαλύτερο κέρδος από την περαιτέρω εξοικονόμηση καταχωρητών, αλλά και του σταδίου άθροισης το οποίο γίνεται πιο πολύπλοκο.

# 4

## ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

### *4.1 Οργάνωση πειραμάτων*

Οι τοπολογίες που παρουσιάστηκαν στο προηγούμενο κεφάλαιο, περιγράφηκαν σε γλώσσα Verilog και συντέθηκαν από το Synopsys Design Compiler, κάνοντας χρήση της βιβλιοθήκης κελιών TSMC 90nm της Artisan, και της εντολής `-compile_ultra` για βέλτιστα αποτελέσματα. Τα κυκλώματα που παρήχθησαν, προσομοιώθηκαν από το περιβάλλον Modelsim, για επιβεβαίωση της ορθής λειτουργίας τους.

Από το περιβάλλον του Design Compiler, εξήχθησαν οι τιμές καθυστέρησης και επιφάνειας κάθε κυκλώματος, ενώ από το περιβάλλον του Synopsys Primepower (με βάση το .vcd αρχείο που προέκυπτε από κάθε προσομοίωση του Modelsim) υπολογίστηκε η μέση δυναμική κατανάλωση κάθε κυκλώματος.

Για την προσομοίωση της λειτουργίας κάθε κυκλώματος χρησιμοποιήθηκαν 65536 ζεύγη τυχαίων δυαδικών αριθμών, μήκους λέξης N.

Όλα τα κυκλώματα υλοποιήθηκαν για μήκη λέξης  $N = 8, 16, 24, 32, 48, 64$ .

Η ίδια διαδικασία ακολουθήθηκε και για τα αντίστοιχα κυκλώματα που κάνουν χρήση της Canonic Modified Booth κωδικοποίησης. Σε αυτή την περίπτωση δεν αλλάζει η καθυστέρηση του κυκλώματος, όπως παρουσιάζεται στο προηγούμενο κεφάλαιο, αλλά υπάρχει μια μικρή επιβάρυνση σε επιφάνεια, καθώς το νέο κύκλωμα κωδικοποίησης είναι μεγαλύτερο. Η επιβάρυνση αυτή είναι της τάξης των 15 πυλών NAND, για κάθε κύκλωμα κωδικοποίησης,

καθώς απαιτείται και ένας επιπλέον καταχωρητής σε αυτή την τοπολογία. Επιπλέον, απαιτείται και ένα ακόμα σήμα ελέγχου για τον συγχρονισμό των δεδομένων.

Τέλος, ακολουθήσαμε την ίδια διαδικασία, για έναν παράλληλο πολλαπλασιαστή, για κάθε μήκος λέξης. Στο κύκλωμα αυτό, χρησιμοποιήσαμε το κύκλωμα παραγωγής μερικών γινομένων του σχήματος 3.4, ένα δένδρο άθροισης Wallace (που δημιουργήθηκε από την συνθετική βιβλιοθήκη DW02\_tree της Desinware) για την μετατροπή του αθροίσματος σε Carry-Save μορφή, και έναν αθροιστή πρόβλεψης κρατουμένου (που δημιουργήθηκε από την συνθετική βιβλιοθήκη DW01\_add της Desinware) για την παραγωγή του τελικού αποτελέσματος σε δυαδική μορφή συμπληρώματος του δύο.

Ο παράλληλος πολλαπλασιαστής Modified Booth αναμένουμε να έχει επιφάνεια  $6N^2 - 3.5N + 3N \log_2 N$  πύλες NAND, και καθυστέρηση  $4 + 9 \log_2 N$  πύλες NAND.

Για όλα τα κυκλώματα, λάβαμε μετρήσεις στην high performance συχνότητα, δηλαδή στην μέγιστη συχνότητα που μπορέσαμε να προσομοιώσουμε το εκάστοτε κύκλωμα. Η περίοδος του ρολογιού για αυτό το σκοπό μεταβάλλονταν με ακρίβεια 0.01ns. Επίσης, λάβαμε μετρήσεις στο 110% και στο 130% της high performance περιόδου. Επιπλέον, λάβαμε μετρήσεις για όλα τα κυκλώματα, στην high performance περίοδο του αντίστοιχου παράλληλου πολλαπλασιαστή.

Τα κυκλώματα συγκρίθηκαν ως προς την επιφάνεια, την καθυστέρηση και την ισχύ που καταναλώνουν. Επίσης συγκρίθηκαν ως προς τις επιδόσεις με βάση τα κριτήρια  $area * delay$ ,  $area * delay^2$  και  $power * delay$ .

## 4.2 Σειριακός παράλληλος πολλαπλασιαστής με/χωρίς διπλή

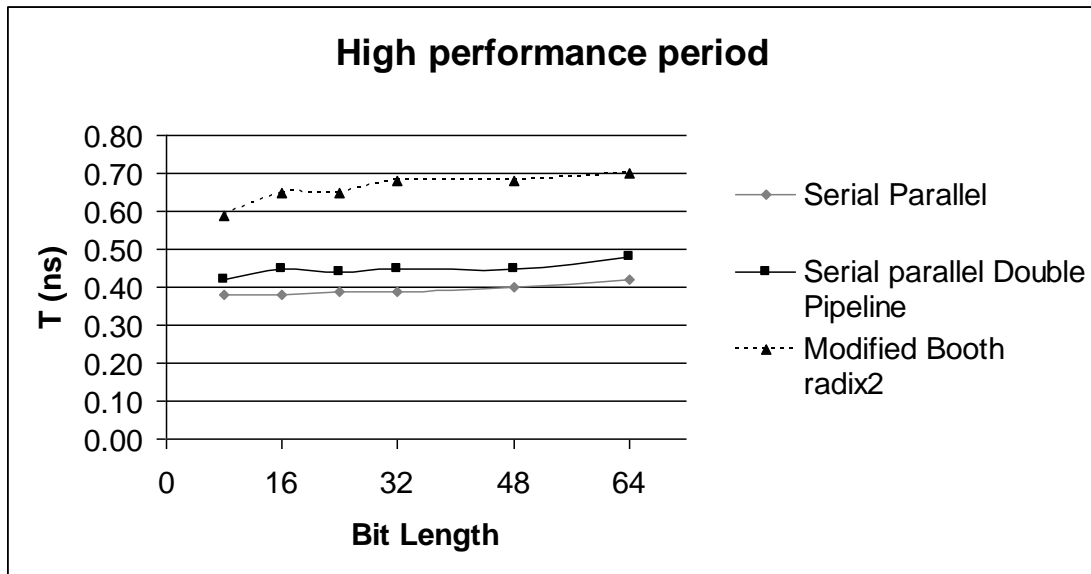
### διοχέτευση και radix-2 Modified Booth σειριακός

#### πολλαπλασιαστής

Οι high performance περίοδοι λειτουργίας των τριών κυκλωμάτων, για τα διάφορα μήκη λέξης, φαίνονται στον παρακάτω πίνακα και το αντίστοιχο διάγραμμα:

**Πίνακας 4.1** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την high performance περίοδο (ns)

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel	0.38	0.38	0.39	0.39	0.40	0.42
Serial parallel (Double Pipeline)	0.42	0.45	0.44	0.45	0.45	0.48
Modified Booth (r2)	0.59	0.65	0.65	0.68	0.68	0.70



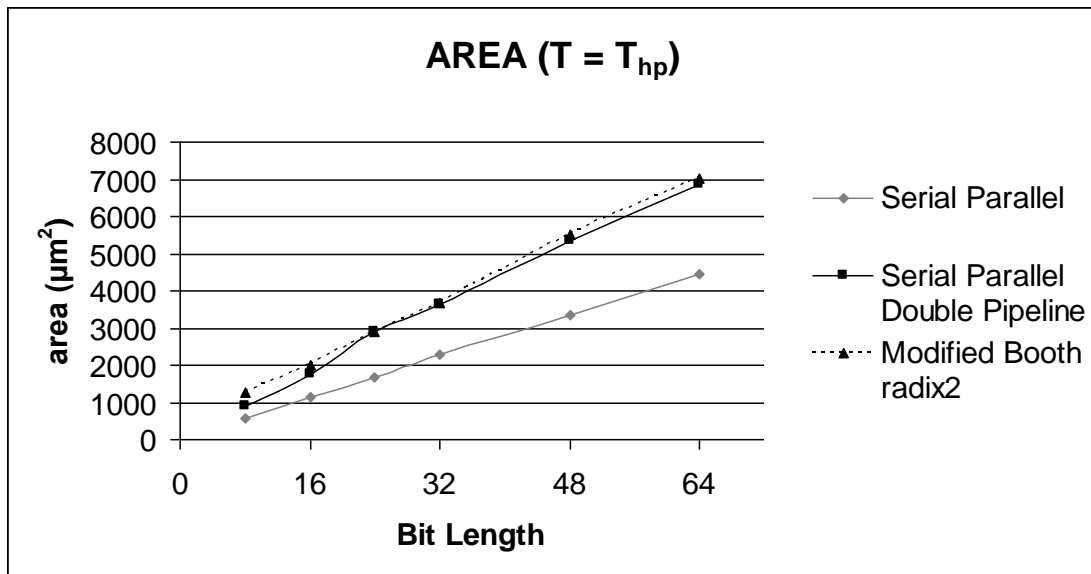
**Σχήμα 4.1** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την high performance περίοδο

Αν και θεωρητικά αναμέναμε ότι η επιβάρυνση του κρίσιμου μονοπατιού που προκύπτει λόγω της δεύτερης διοχέτευσης, θα ήταν της τάξης του 26%, αυτή στην πραγματικότητα είναι πολύ μικρότερη, περίπου 10%-15%. Το γεγονός αυτό συνεπάγεται ότι τελικά θα έχουμε αρκετά καλύτερα αποτελέσματα με τη χρήση της δεύτερης διοχέτευσης. Επίσης, αναμέναμε θεωρητικά ότι το κρίσιμο μονοπάτι του radix-2 Modified Booth σειριακού πολλαπλασιαστή θα έπρεπε να ήταν 73% μεγαλύτερο από το κρίσιμο μονοπάτι του σειριακού παράλληλου πολλαπλασιαστή διπλής διοχέτευσης, ενώ η διαφορά που μετρήσαμε ήταν της τάξης του 45%. Αυτό συμβαίνει διότι πιθανότατα υποεκτιμήσαμε την καθυστέρηση που εισάγουν στο κύκλωμα οι καταχωρητές. Επίσης, η διαφορά των κρίσιμων μονοπατιών είναι κοντά στις θεωρητικά αναμενόμενες τιμές, οι οποίες είναι 2 και 7 πύλες NAND, δεδομένου ότι μια πύλη NAND έχει καθυστέρηση περίπου 0.02ns σε αυτή την βιβλιοθήκη.

Οι επιφάνειες (σε  $\mu\text{m}^2$ ) που καταλαμβάνουν τα παραπάνω κυκλώματα, όταν συντεθούν στην συχνότητα high performance, φαίνονται στον επόμενο πίνακα και το αντίστοιχο διάγραμμα:

**Πίνακας 4.2** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την επιφάνεια ( $\mu\text{m}^2$ ) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel	554	1158	1691	2273	3367	4432
Serial parallel (Double Pipeline)	912	1736	2908	3613	5343	6852
Modified Booth (r2)	1284	2005	2913	3680	5503	7013



**Σχήμα 4.2** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την επιφάνεια ( $T = T_{hp}$ )

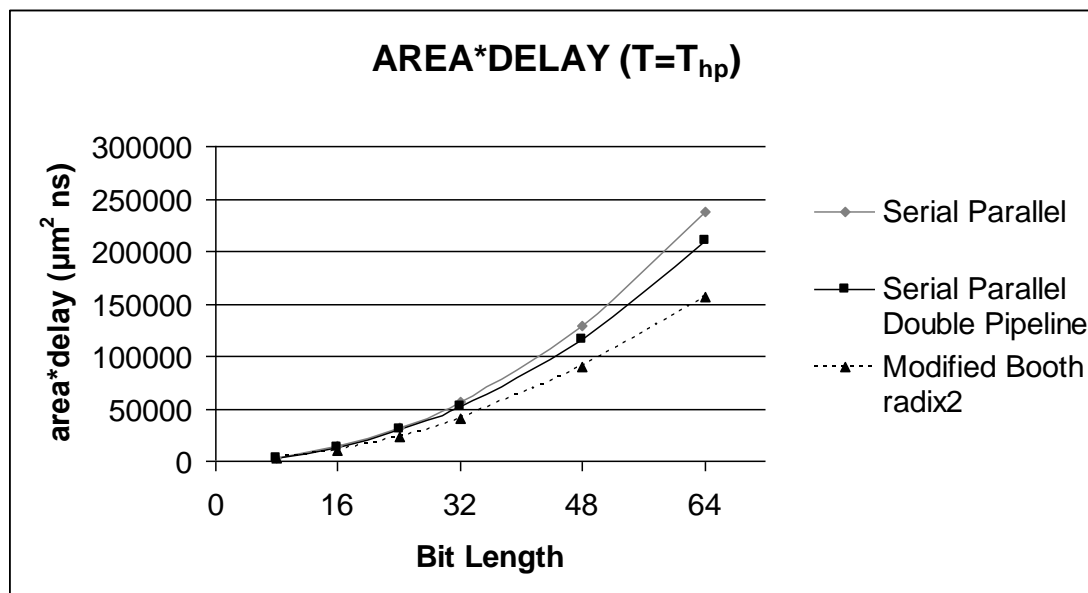
Οι επιφάνειες των σειριακών παράλληλων πολλαπλασιαστών με διπλή διοχέτευση είναι περίπου 55%-70% μεγαλύτερες από των αντίστοιχων σειριακών παράλληλων πολλαπλασιαστών χωρίς διπλή διοχέτευση. Το αποτέλεσμα αυτό είναι λίγο καλύτερο από το θεωρητικά αναμενόμενο, το οποίο ήταν της τάξης του 70%. Επίσης, αναμέναμε οι επιφάνειες των σειριακών πολλαπλασιαστών Modified Booth radix-2 να είναι περίπου ίσες με αυτές των σειριακών παράλληλων πολλαπλασιαστών με διπλή διοχέτευση, κάτι το οποίο επιβεβαιώνεται και από τα πειραματικά αποτελέσματα. Πρέπει να τονιστεί ωστόσο ότι τα παραπάνω αποτελέσματα δεν αφορούν τις ίδιες συχνότητες λειτουργίας, αλλά την εκάστοτε high performance συχνότητα.

Μια πιο ακριβής σύγκριση των δύο σχημάτων, που λαμβάνει υπόψη τα κριτήρια επίδοσης, φαίνεται στον επόμενο πίνακα και τα αντίστοιχα διαγράμματα:

**Πίνακας 4.3** Συνολική σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ( $T = T_{hp}$ )

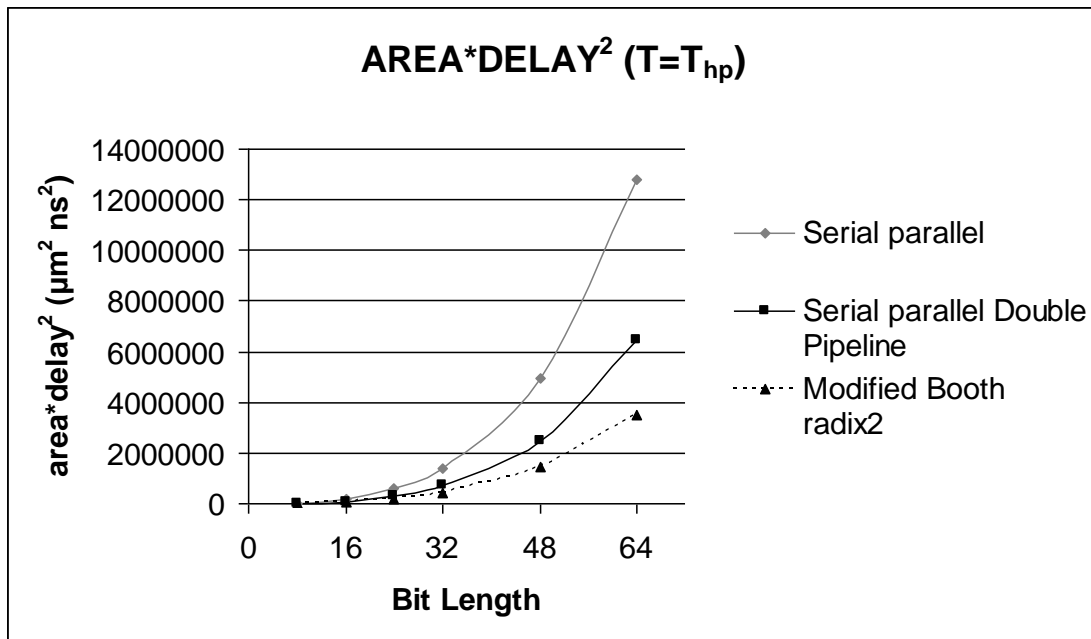
Serial parallel	N=8	N=16	N=24	N=32	N=48	N=64
Area (μm <sup>2</sup> )	554	1158	1691	2273	3367	4432
Delay (ns)	0.38	0.38	0.39	0.39	0.40	0.42
Clock cycles	16	32	48	64	96	128
Area*delay (μm <sup>2</sup> ns)	3368	14080	31662	56745	129298	238258
Area* delay <sup>2</sup> (μm <sup>2</sup> ns <sup>2</sup> )	20476	171212	592705	1416359	4965025	12808725

Serial parallel (Double Pipeline)	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	912	1736	2908	3613	5343	6852
Delay (ns)	0.42	0.45	0.44	0.45	0.45	0.48
Clock cycles	8	16	24	32	48	64
Area* delay ( $\mu\text{m}^2 \text{ ns}$ )	3063	12518	30706	52022	115405	210496
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	10292	90129	324257	749124	2492738	6466436
Modified Booth (r2)	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	1284	2005	2913	3680	5503	7013
Delay (ns)	0.59	0.65	0.65	0.68	0.68	0.70
Clock cycles	4	8	12	16	24	32
Area* delay ( $\mu\text{m}^2 \text{ ns}$ )	3031	10428	22719	40043	89809	157090
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	7152	54224	177210	435666	1465675	3518822



**Σχήμα 4.3** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς το κριτήριο  $area*delay$  ( $T = T_{hp}$ )





**Σχήμα 4.4** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς το κριτήριο  $area*delay^2$  ( $T = T_{hp}$ )

Τα πειραματικά αποτελέσματα δείχνουν ότι ο σειριακός παράλληλος πολλαπλασιαστής διπλής διοχέτευσης παρουσιάζει ελαφρώς καλύτερα αποτελέσματα ως προς το κριτήριο  $area*delay$ , της τάξης του 10%, ενώ παρουσιάζει σαφώς καλύτερα αποτελέσματα ως προς το κριτήριο  $area*delay^2$ , της τάξης του 50%. Οπότε, πράγματι η δεύτερη διοχέτευση, επιδρά θετικά στην απόδοση του κυκλώματος.

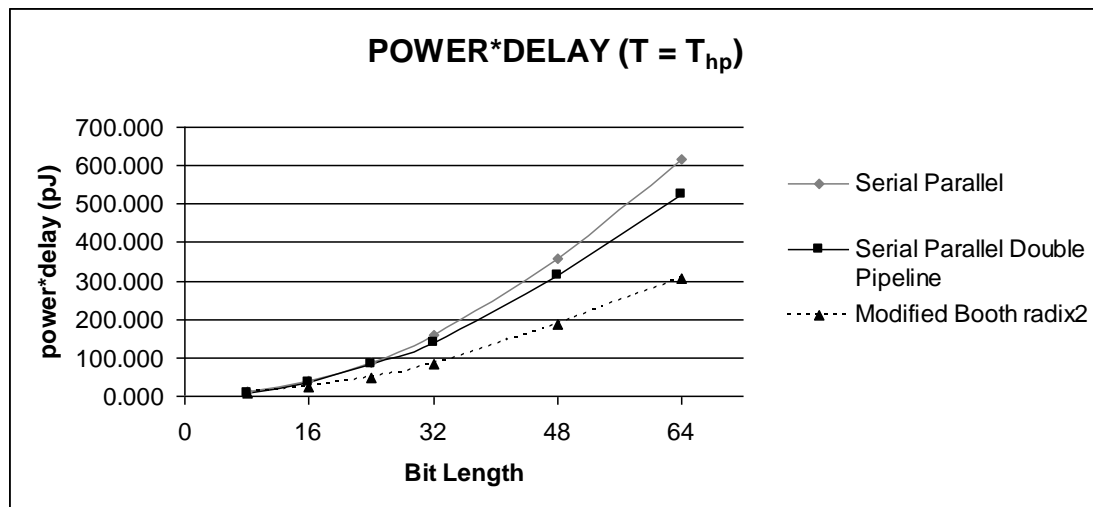
Επίσης, οι radix-2 Modified Booth πολλαπλασιαστές παρουσιάζουν πολύ καλύτερες επιδόσεις από τους σειριακούς παράλληλους πολλαπλασιαστές διπλής διοχέτευσης όσον αφορά τα κριτήρια  $area*delay$  (περίπου 25% μικρότερο) και  $area*delay^2$  (περίπου 40% μικρότερο), ενώ παράλληλα παρουσιάζουν ενεργειακή κατανάλωση της τάξης του 60% του σειριακού παράλληλου πολλαπλασιαστή διπλής διοχέτευσης.

Οι καταναλώσεις των τριών κυκλωμάτων φαίνονται στον επόμενο πίνακα και το αντίστοιχο διάγραμμα:

**Πίνακας 4.4** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την κατανάλωση ( $T = T_{hp}$ )

Serial parallel	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	1.644	3.237	4.637	6.382	9.065	11.500
Delay (ns)	0.38	0.38	0.39	0.39	0.40	0.42
Clock cycles	16	32	48	64	96	128
Power* delay (pJ)	9.996	39.362	89.030	159.295	356.798	618.240

Serial parallel (Double Pipeline)	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	2.764	4.893	7.909	9.566	14.300	17.100
Delay (ns)	0.42	0.45	0.44	0.45	0.45	0.48
Clock cycles	8	16	24	32	48	64
Power* delay (pJ)	9.287	35.230	83.519	140.812	315.744	525.312
Modified Booth (r2)	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	3.207	4.475	6.195	7.427	11.100	13.600
Delay (ns)	0.59	0.65	0.65	0.68	0.68	0.70
Clock cycles	4	8	12	16	24	32
Power* delay (pJ)	7.825	23.628	49.064	83.182	186.480	304.640



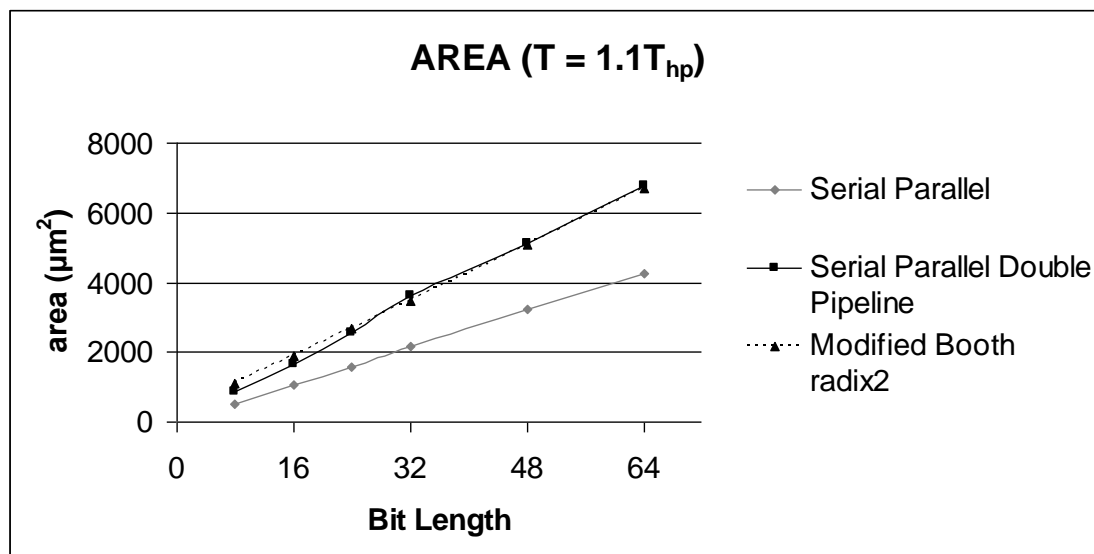
**Σχήμα 4.5** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την κατανάλωση ( $T = T_{hp}$ )

Το κύκλωμα της δεύτερης διοχέτευσης επιδρά θετικά και στην ενεργειακή κατανάλωση του κυκλώματος, καθώς το κύκλωμα του σειριακού παράλληλου πολλαπλασιαστή με διπλή διοχέτευση απαιτεί λιγότερη ενέργεια κατά 10% και πλέον. Προφανώς, η ισχύς της δυναμικής κατανάλωσης του σειριακού παράλληλου πολλαπλασιαστή με διπλή διοχέτευση είναι σημαντικά μεγαλύτερη, λόγω του μεγαλύτερου κυκλώματος και του μεγαλύτερου αριθμού καταχωρητών. Επίσης, οι radix-2 Modified Booth σειριακοί πολλαπλασιαστές παρουσιάζουν ενεργειακή κατανάλωση της τάξης του 60% των σειριακών παράλληλων πολλαπλασιαστών διπλής διοχέτευσης.

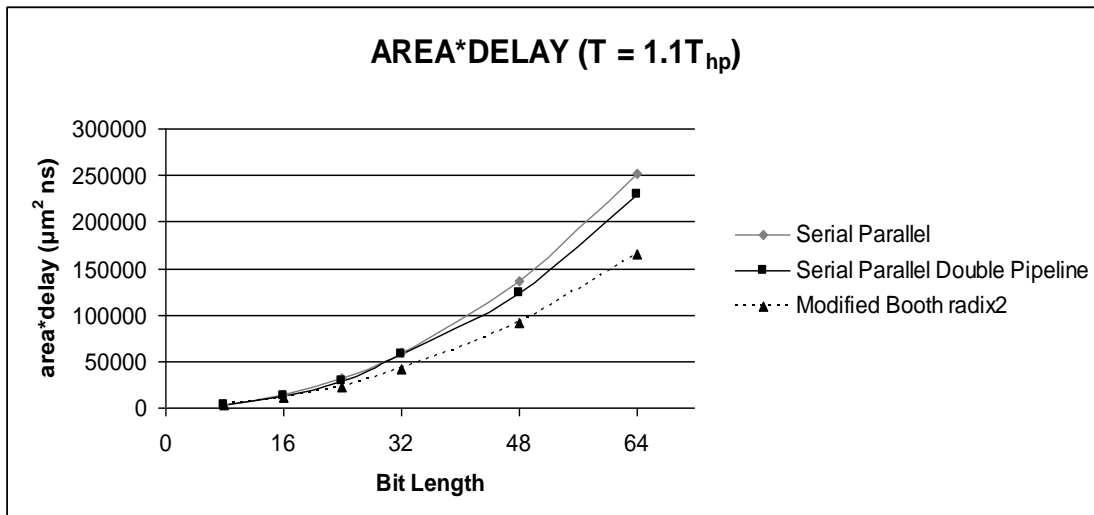
Παρόμοια αποτελέσματα προκύπτουν και από την μελέτη των παραπάνω κυκλωμάτων, όταν συντεθούν σε περίοδο μεγαλύτερη της high performance περιόδου.

**Πίνακας 4.5** Συνολική σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ( $T = 1.1T_{hp}$ )

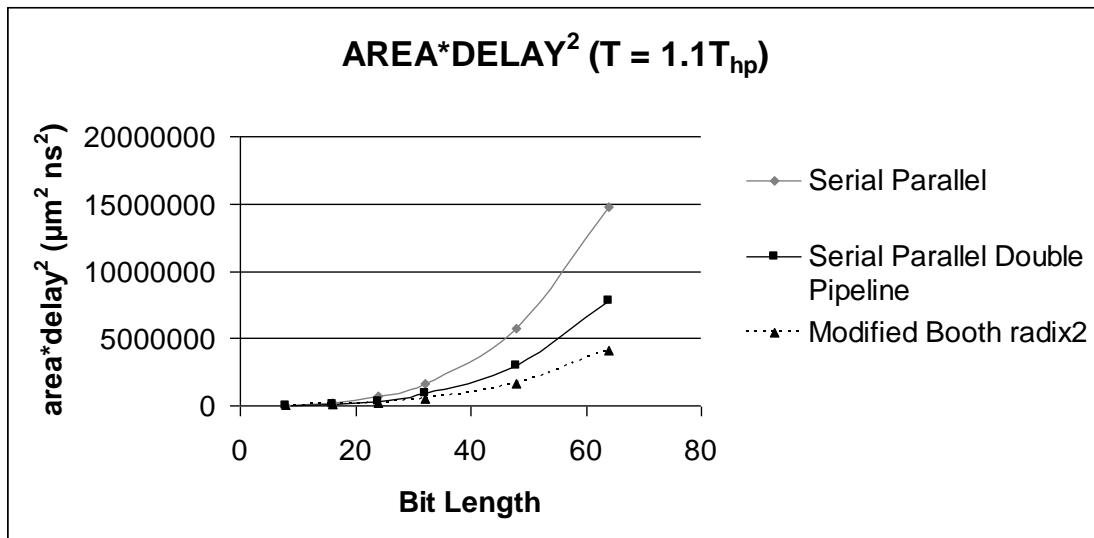
Serial parallel	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	519	1058	1586	2158	3227	4269
Delay (ns)	0.42	0.42	0.43	0.43	0.44	0.46
Clock cycles	16	32	48	64	96	128
Area*delay ( $\mu\text{m}^2 \text{ ns}$ )	3490	14215	32739	59400	136326	251352
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	23452	191055	675732	1634688	5758410	14799586
Serial parallel (Double Pipeline)	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	868	1646	2553	3608	5121	6781
Delay (ns)	0.46	0.50	0.48	0.50	0.50	0.53
Clock cycles	8	16	24	32	48	64
Area* delay ( $\mu\text{m}^2 \text{ ns}$ )	3194	13169	29409	57724	122893	230005
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	11753	105355	338791	923580	2949431	7801779
Modified Booth (r2)	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	1109	1897	2677	3453	5092	6686
Delay (ns)	0.65	0.72	0.72	0.75	0.75	0.77
Clock cycles	4	8	12	16	24	32
Area* delay ( $\mu\text{m}^2 \text{ ns}$ )	2884	10925	23130	41430	91662	164750
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	7498	62926	199840	497160	1649910	4059430



**Σχήμα 4.6** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την επιφάνεια ( $T = 1.1T_{hp}$ )



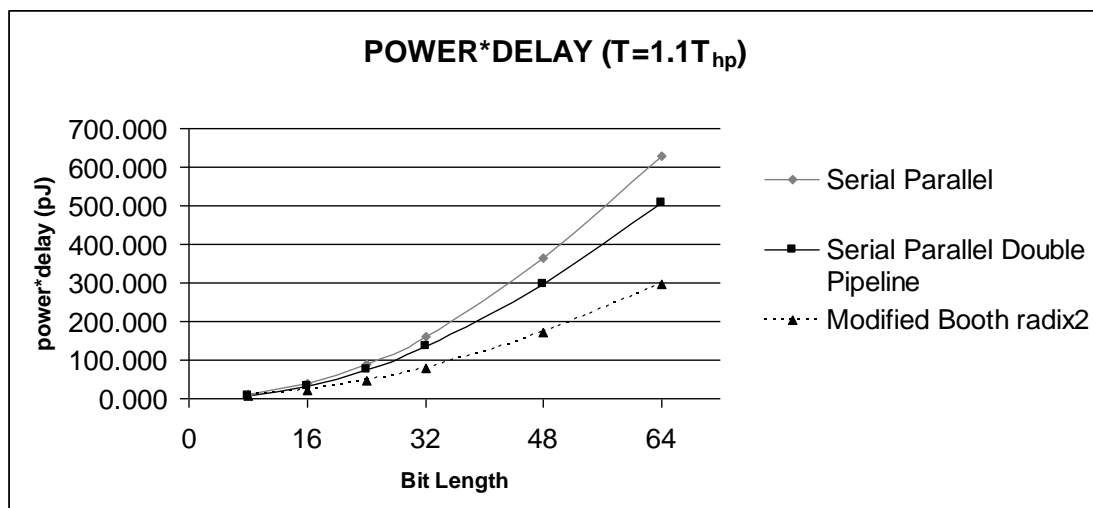
**Σχήμα 4.7** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς το κριτήριο  $area*delay$  ( $T = 1.1T_{hp}$ )



**Σχήμα 4.8** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς το κριτήριο  $area*delay^2$  ( $T = 1.1T_{hp}$ )

**Πίνακας 4.6** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την κατανάλωση ( $T = 1.1T_{hp}$ )

Serial parallel	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	1.443	2.918	4.373	5.816	8.588	10.700
Delay (ns)	0.42	0.42	0.43	0.43	0.44	0.46
Clock cycles	16	32	48	64	96	128
Power* delay (pJ)	9.697	39.218	88.160	160.056	362.757	630.016
Serial parallel (Double Pipeline)	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	2.272	3.886	6.379	8.589	12.300	14.900
Delay (ns)	0.46	0.50	0.48	0.50	0.50	0.53
Clock cycles	8	16	24	32	48	64
Power* delay (pJ)	8.361	31.710	75.017	137.424	295.200	505.408
Modified Booth (r2)	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	2.654	3.765	5.371	6.605	9.306	12.100
Delay (ns)	0.65	0.72	0.72	0.75	0.75	0.77
Clock cycles	4	8	12	16	24	32
Power* delay (pJ)	6.900	21.686	46.405	79.260	169.741	298.144

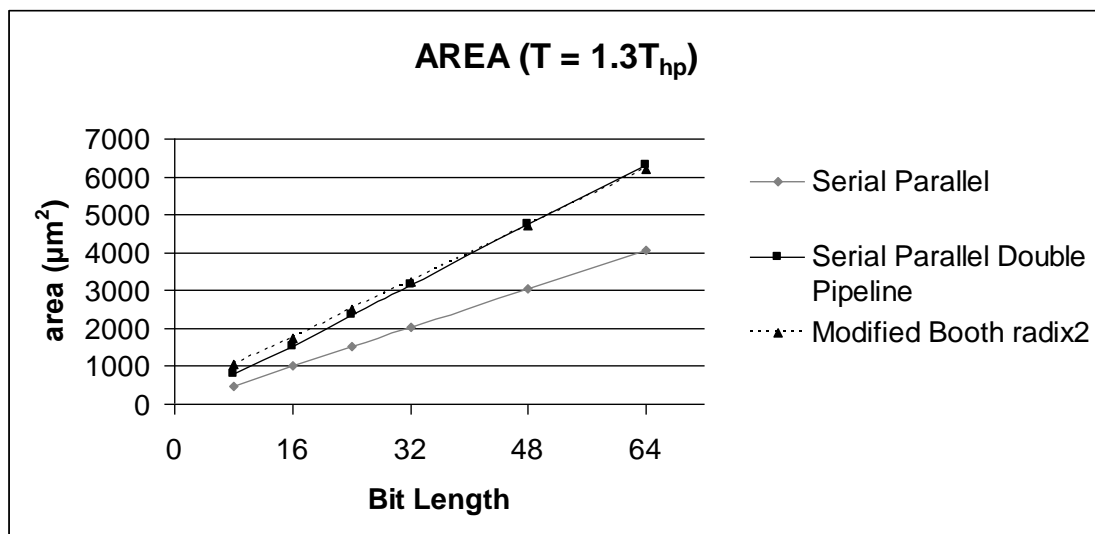


**Σχήμα 4.9** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την κατανάλωση ( $T = 1.1 T_{hp}$ )

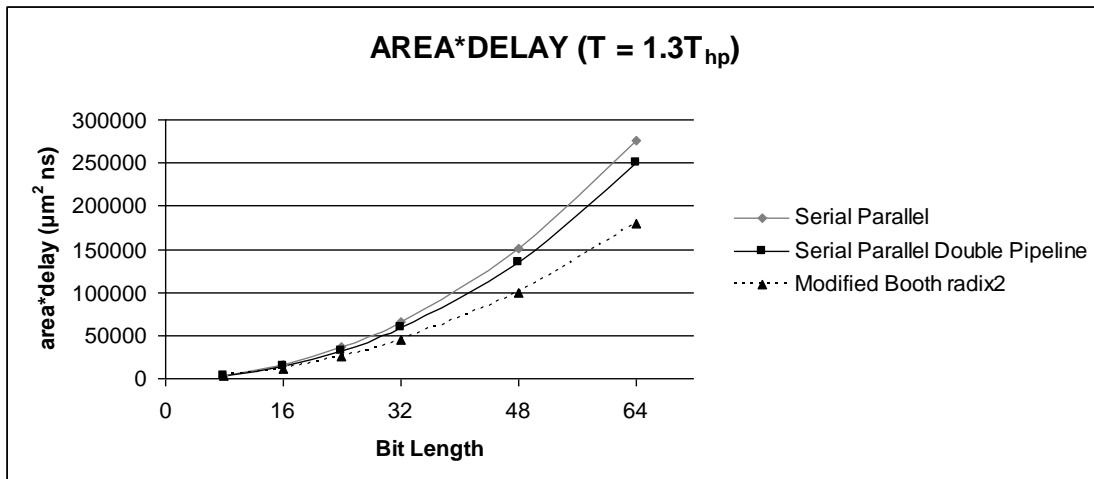
Παρατηρούμε επίσης ότι τα κριτήρια  $area*delay$  και  $area*delay^2$  γίνονται μεγαλύτερα (άρα χειρότερα) όσο απομακρυνόμαστε από την συχνότητα high performance, ενώ η ενεργειακή κατανάλωση γίνεται γενικά μικρότερη, κάτι που είναι αναμενόμενο άλλωστε αφού είναι ανάλογη της συχνότητας λειτουργίας.

**Πίνακας 4.7** Συνολική σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ( $T = 1.3T_{hp}$ )

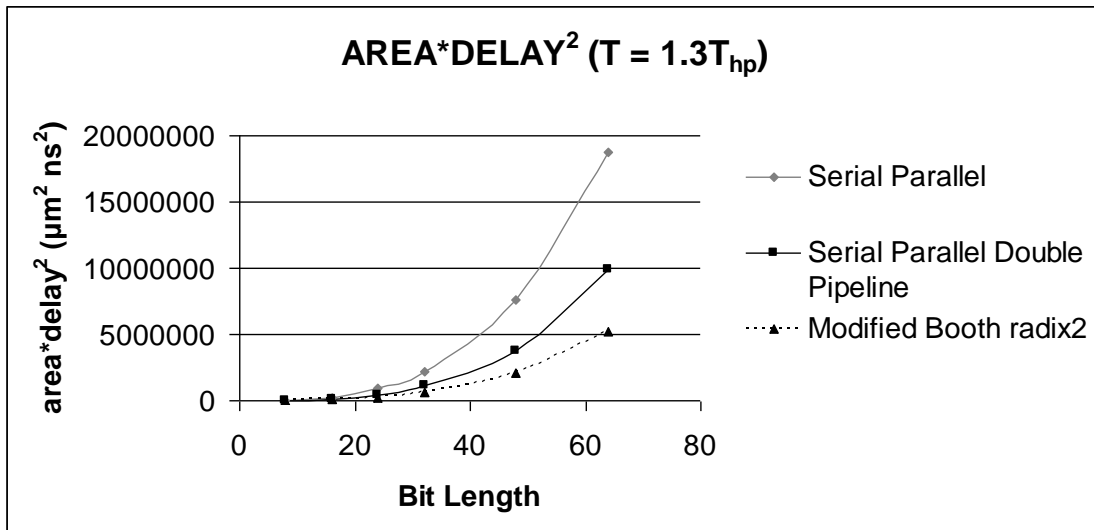
Serial parallel	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	470	1003	1511	2021	3037	4070
Delay (ns)	0.49	0.49	0.51	0.51	0.52	0.53
Clock cycles	16	32	48	64	96	128
Area*delay ( $\mu\text{m}^2 \text{ ns}$ )	3684	15722	36982	65960	151602	276102
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	28885	246516	905312	2152940	7567980	18730765
Serial parallel (Double Pipeline)	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	783	1526	2367	3156	4740	6317
Delay (ns)	0.55	0.59	0.57	0.59	0.59	0.62
Clock cycles	8	16	24	32	48	64
Area* delay ( $\mu\text{m}^2 \text{ ns}$ )	3446	14407	32385	59588	134243	250668
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	15163	136006	443020	1125023	3801763	9946505
Modified Booth (r2)	N=8	N=16	N=24	N=32	N=48	N=64
Area ( $\mu\text{m}^2$ )	1048	1749	2487	3238	4708	6194
Delay (ns)	0.77	0.85	0.85	0.88	0.88	0.91
Clock cycles	4	8	12	16	24	32
Area* delay ( $\mu\text{m}^2 \text{ ns}$ )	3227	11894	25370	45591	99443	180383
Area* delay <sup>2</sup> ( $\mu\text{m}^2 \text{ ns}^2$ )	9940	80882	258772	641922	2100233	5252746



**Σχήμα 4.10** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την επιφάνεια ( $T = 1.3T_{hp}$ )



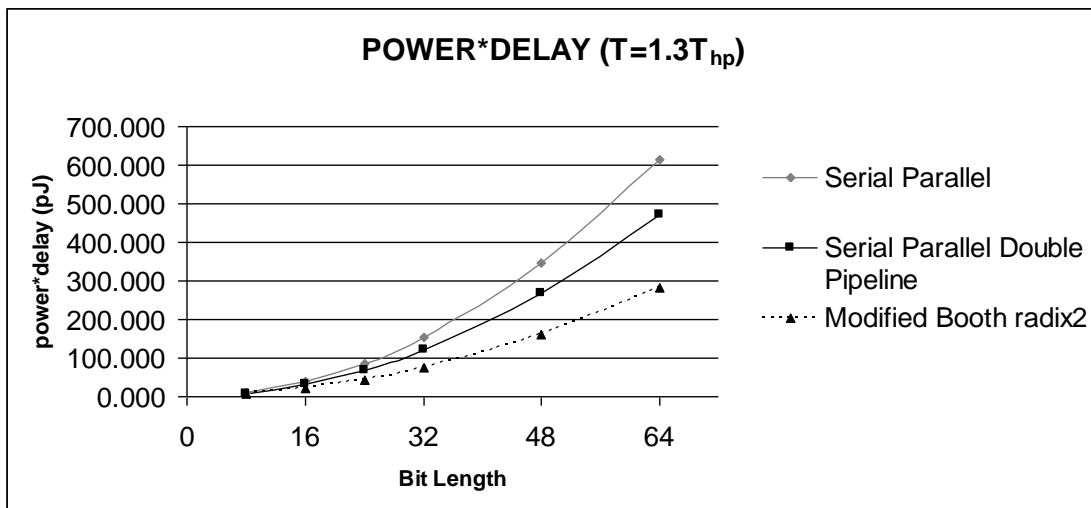
**Σχήμα 4.11** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς το κριτήριο  $area*delay$  ( $T = 1.3T_{hp}$ )



**Σχήμα 4.12** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς το κριτήριο  $area*delay^2$  ( $T = 1.3T_{hp}$ )

**Πίνακας 4.8** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση ως προς την κατανάλωση ( $T = 1.3T_{hp}$ )

Serial parallel	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	1.206	2.442	3.505	4.730	6.974	9.081
Delay (ns)	0.49	0.49	0.51	0.51	0.52	0.53
Clock cycles	16	32	48	64	96	128
Power* delay (pJ)	9.455	38.291	85.802	154.387	348.142	616.055
Serial parallel (Double Pipeline)	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	1.763	3.229	4.987	6.408	9.513	11.900
Delay (ns)	0.55	0.59	0.57	0.59	0.59	0.62
Clock cycles	8	16	24	32	48	64
Power* delay (pJ)	7.757	30.482	68.222	120.983	269.408	472.192
Modified Booth (r2)	N=8	N=16	N=24	N=32	N=48	N=64
Power (mW)	2.048	3.960	4.143	5.343	7.638	9.708
Delay (ns)	0.77	0.85	0.85	0.88	0.88	0.91
Clock cycles	4	8	12	16	24	32
Power* delay (pJ)	6.308	20.128	42.259	75.229	161.315	282.697



**Σχήμα 4.13** Σύγκριση σειριακού παράλληλου πολλαπλασιαστή με/χωρίς διπλή διοχέτευση και Modified Booth radix2 πολλαπλασιαστή ως προς την κατανάλωση ( $T = 1.3 T_{hp}$ )

Παρατηρούμε και πάλι ότι τα κριτήρια  $area*delay$  και  $area*delay^2$  γίνονται μεγαλύτερα (άρα χειρότερα) όσο απομακρυνόμαστε από την συχνότητα high performance, ενώ η ενεργειακή κατανάλωση γίνεται γενικά μικρότερη, κάτι που είναι αναμενόμενο άλλωστε αφού είναι ανάλογη της συχνότητας λειτουργίας.

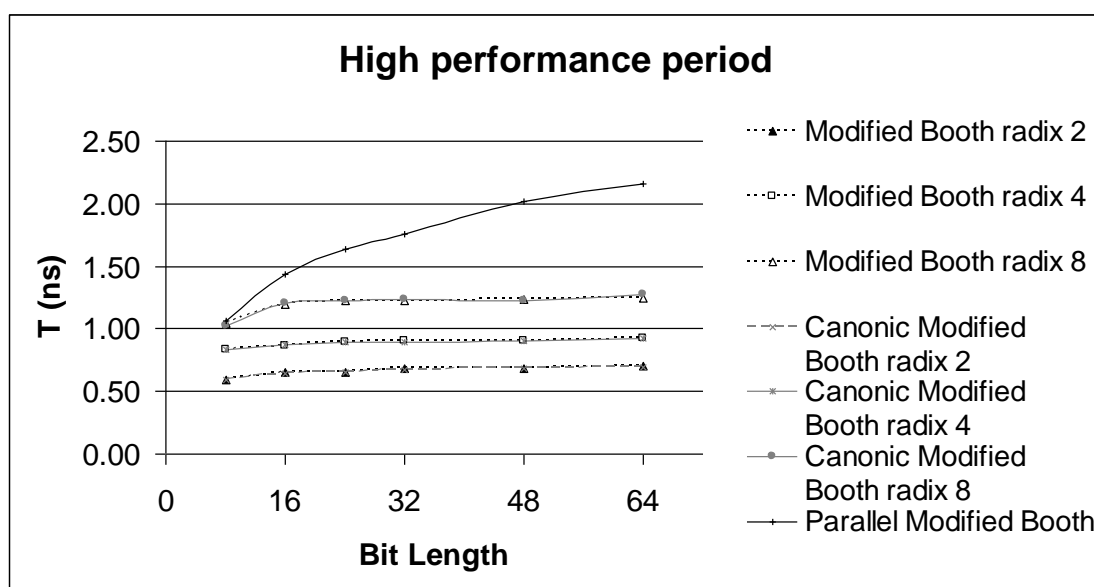


### 4.3 Modified Booth και Canonic Modified Booth σειριακοί και παράλληλοι πολλαπλασιαστές

Οι high performance περίοδοι λειτουργίας των κυκλωμάτων radix-2, radix-4 και radix-8 σειριακού πολλαπλασιαστή, με κωδικοποίηση Modified Booth και Canonic Modified Booth, καθώς και του παράλληλου Modified Booth πολλαπλασιαστή, για τα διάφορα μήκη λέξης, φαίνονται στον παρακάτω πίνακα και το αντίστοιχο διάγραμμα:

**Πίνακας 4.9** High performance περίοδος για τα κυκλώματα σειριακών και παράλληλων πολλαπλασιαστών Modified Booth (σε ns)

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	0.59	0.65	0.65	0.68	0.68	0.70
Modified Booth (r4)	0.83	0.86	0.89	0.90	0.90	0.92
Modified Booth (r8)	1.04	1.19	1.22	1.22	1.23	1.25
Canonic Modified Booth (r2)	0.59	0.64	0.65	0.67	0.68	0.69
Canonic Modified Booth (r4)	0.83	0.87	0.89	0.89	0.90	0.92
Canonic Modified Booth (r8)	1.02	1.20	1.22	1.23	1.22	1.28
Parallel (MB)	1.06	1.44	1.64	1.76	2.02	2.16



**Σχήμα 4.14** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την high performance περίοδο

Το κρίσιμο μονοπάτι του radix-4 Modified Booth σειριακού πολλαπλασιαστή είναι περίπου 34%-42% μεγαλύτερο από αυτό του radix-2 Modified Booth σειριακού πολλαπλασιαστή, μια τιμή που είναι πολύ κοντά στο θεωρητικά αναμενόμενο ποσοστό, το οποίο είναι 37%.

Το κρίσιμο μονοπάτι του radix-8 Modified Booth σειριακού πολλαπλασιαστή είναι περίπου 35% μεγαλύτερο από αυτό του radix-4 Modified Booth σειριακού πολλαπλασιαστή και περίπου 80% μεγαλύτερο από αυτό του radix-2 Modified Booth σειριακού πολλαπλασιαστή, τιμές που είναι κοντά στα θεωρητικά αναμενόμενα ποσοστά, τα οποία είναι 44% και 96% αντίστοιχα.

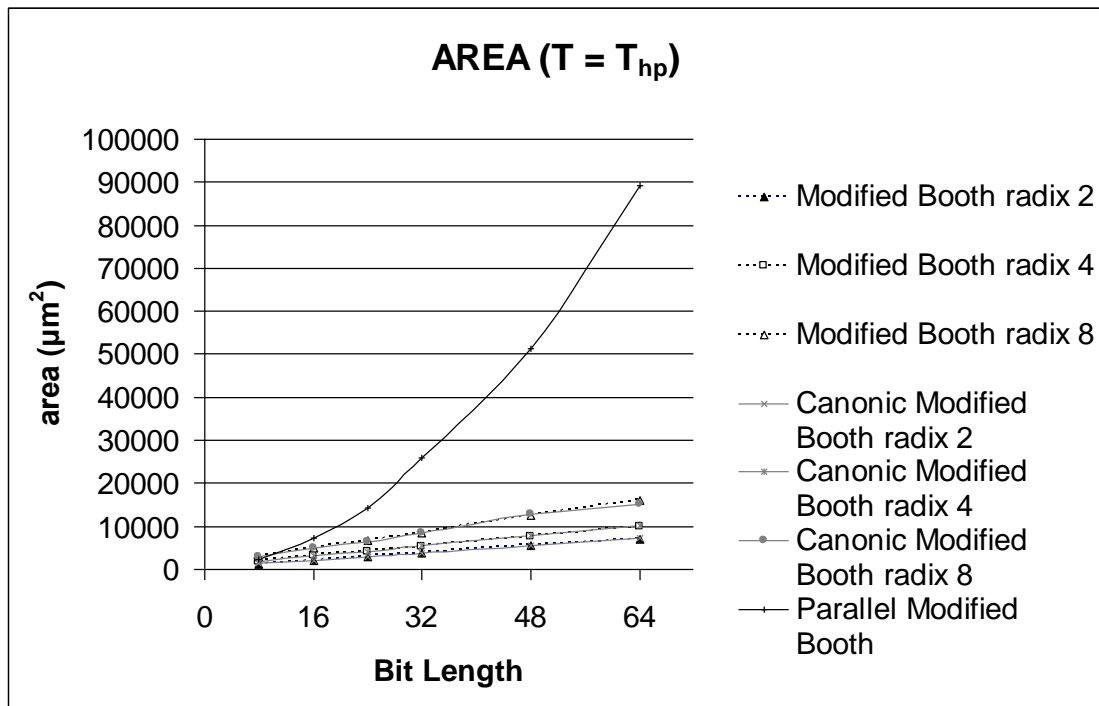
Το κρίσιμο μονοπάτι του παράλληλου Modified Booth πολλαπλασιαστή αυξάνει λογαριθμικά, όπως ήταν αναμενόμενο, και επίσης η αναλογία του προς τα κρίσιμα μονοπάτια των σειριακών Modified Booth πολλαπλασιαστών είναι αρκετά κοντά στην θεωρητικά αναμενόμενη τιμή.

Επιπλέον, το κρίσιμο μονοπάτι των δύο πολλαπλασιαστών με διαφορετική κωδικοποίηση δεν παρουσιάζει ουσιαστικές διαφορές σε όλα τα μήκη λέξης και σε όλα τα radices, κάτι αναμενόμενο, αφού το κύκλωμα κωδικοποίησης δεν βρίσκεται στο κρίσιμο μονοπάτι.

Οι επιφάνειες (σε  $\mu\text{m}^2$ ) που καταλαμβάνουν τα κυκλώματα radix-2, radix-4 και radix-8 σειριακού πολλαπλασιαστή, με κωδικοποίηση Modified Booth και Canonic Modified Booth, καθώς και του παράλληλου Modified Booth πολλαπλασιαστή, όταν συντεθούν στην συχνότητα high performance, φαίνονται στον επόμενο πίνακα και το αντίστοιχο διάγραμμα:

**Πίνακας 4.10** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $\mu\text{m}^2$ ) ( $T = T_{\text{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	1284	2005	2913	3680	5503	7013
Modified Booth (r4)	1796	3082	4085	5318	7691	10008
Modified Booth (r8)	2790	4933	6608	8544	12578	15927
Canonic Modified Booth (r2)	1312	2085	3061	3775	5527	7290
Canonic Modified Booth (r4)	1758	3071	4028	5603	7790	10168
Canonic Modified Booth (r8)	2851	5010	6487	8361	12718	15107
Parallel (MB)	2393	7430	14427	26043	51209	89193



**Σχήμα 4.15** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $T = T_{hp}$ )

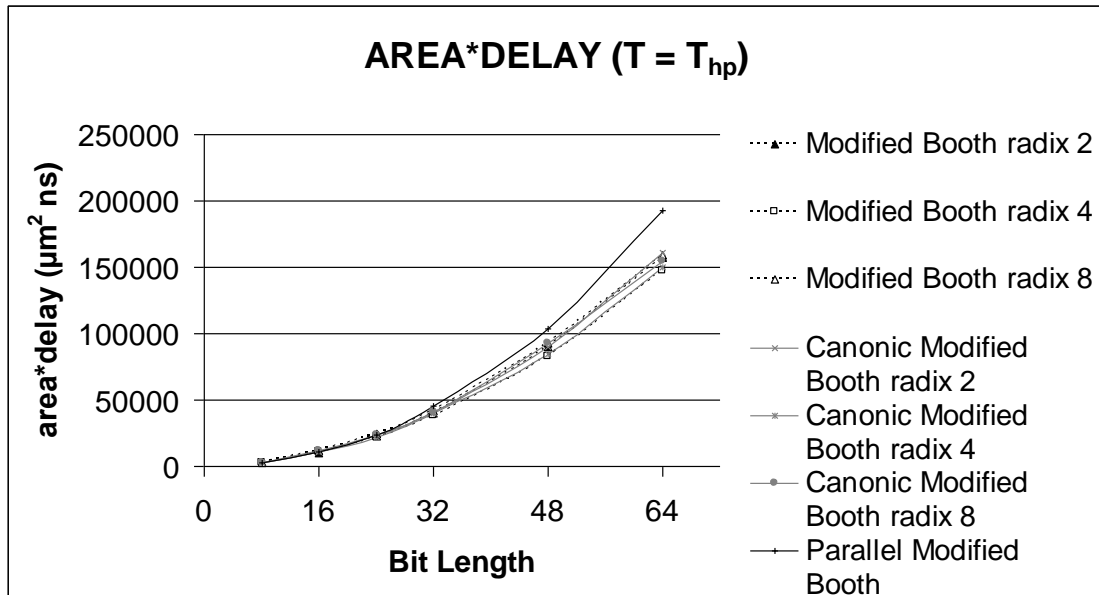
Οι επιφάνειες των σειριακών πολλαπλασιαστών radix-4 Modified Booth αναμέναμε να είναι περίπου 20%-30% μεγαλύτερες από αυτές των radix-2 Modified Booth σειριακών πολλαπλασιαστών, ωστόσο τα πειράματα έδειξαν ότι αυτό το ποσοστό είναι της τάξης του 40%-50%. Οι επιφάνειες των σειριακών πολλαπλασιαστών radix-8 Modified Booth αναμέναμε να είναι περίπου 40%-50% μεγαλύτερες από αυτές των radix-4 Modified Booth σειριακών πολλαπλασιαστών και 70% μεγαλύτερες από αυτές των radix-2 Modified Booth σειριακών πολλαπλασιαστών, ωστόσο τα πειράματα έδειξαν ότι αυτό το ποσοστό είναι της τάξης του 60% και 125% αντίστοιχα. Πρέπει να τονιστεί επίσης ότι τα παραπάνω αποτελέσματα δεν αφορούν τις ίδιες συχνότητες λειτουργίας, αλλά την εκάστοτε high performance συχνότητα.

Η επιφάνεια του παράλληλου Modified Booth πολλαπλασιαστή αυξάνει τετραγωνικά, όπως ήταν αναμενόμενο, ωστόσο ο λόγος της επιφάνειας του παράλληλου πολλαπλασιαστή φαίνεται να αυξάνει κάπως πιο γρήγορα από το θεωρητικώς αναμενόμενο.

Μια πιο ακριβής σύγκριση των παραπάνω σχημάτων, που λαμβάνει υπόψη τα κριτήρια επίδοσης, φαίνεται στα επόμενα διαγράμματα.

**Πίνακας 4.11** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $\mu m^2 ns$ ) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	3031	10428	22719	40043	89809	157090
Modified Booth (r4)	2908	10602	21816	38290	83063	147321
Modified Booth (r8)	2902	11740	24185	41695	92826	159268
Canonic Modified Booth (r2)	3097	10675	23875	40468	90200	160969
Canonic Modified Booth (r4)	2919	10686	21507	39895	84130	149668
Canonic Modified Booth (r8)	2908	12025	23743	41138	93099	154695
Parallel (MB)	2536	10699	23661	45836	103442	192658

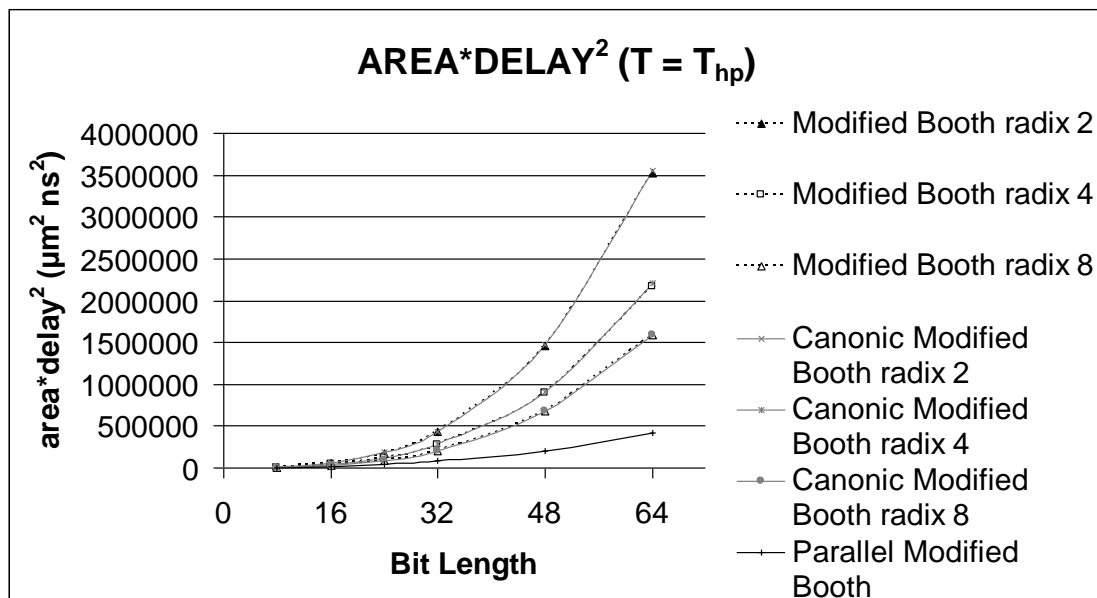


**Σχήμα 4.16** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $T = T_{hp}$ )

Τα πειραματικά αποτελέσματα δείχνουν ότι όλα τα κυκλώματα σειριακών πολλαπλασιαστών έχουν παρόμοιες επιδόσεις όσον αφορά το κριτήριο  $area*delay$ , και μάλιστα είναι κάπως καλύτερα από τους αντίστοιχους παράλληλους πολλαπλασιαστές. Τις καλύτερες επιδόσεις μεταξύ αυτών παρουσιάζουν τα κυκλώματα των radix-4 Modified Booth πολλαπλασιαστών.

**Πίνακας 4.12** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $\mu m^2 ns^2$ ) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	7152	54224	177210	435666	1465675	3518822
Modified Booth (r4)	4950	36472	116498	275691	897083	2168567
Modified Booth (r8)	3018	27942	88517	203473	685055	1592680
Canonic Modified Booth (r2)	7310	54658	186225	433812	1472065	3554194
Canonic Modified Booth (r4)	4845	37188	114848	284049	908605	2203120
Canonic Modified Booth (r8)	2966	28860	86901	202398	681485	1584073
Parallel (MB)	2688	15407	38804	80671	208953	416141



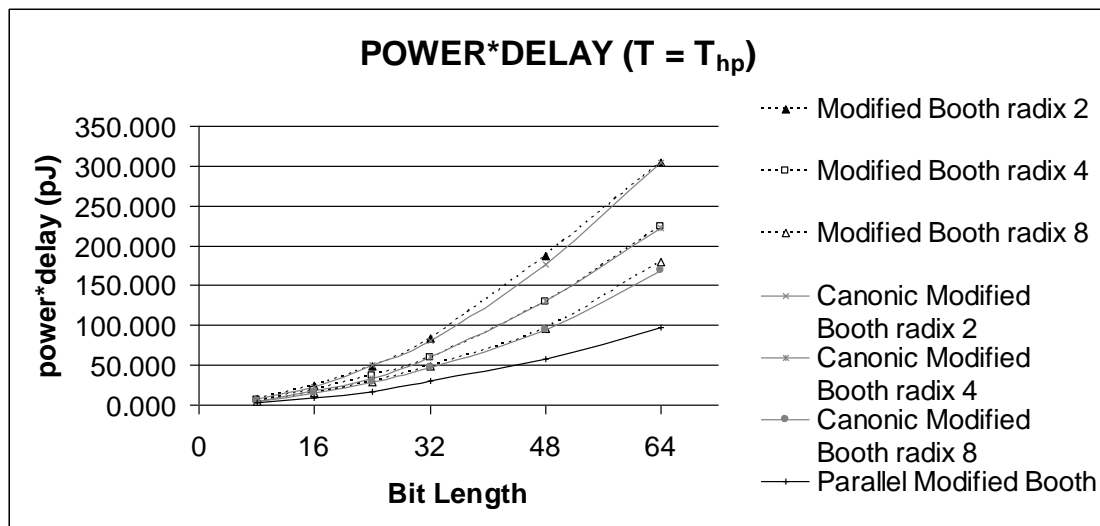
**Σχήμα 4.17** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $T = T_{hp}$ )

Τα πειραματικά αποτελέσματα δείχνουν ότι τα κυκλώματα με μεγαλύτερα radices έχουν καλύτερες επιδόσεις όσον αφορά το κριτήριο  $area*delay^2$ , κάτι απολύτως αναμενόμενο αφού αυτό κριτήριο ευνοεί τα κυκλώματα με μικρότερη καθυστέρηση. Για άλλη μια φορά παρατηρούμε ότι τα σειριακά κυκλώματα πολλαπλασιαστών έχουν ταυτόσημες επιδόσεις και για τις δύο κωδικοποιήσεις σε όλα τα radices.

Οι καταναλώσεις όλων των κυκλωμάτων φαίνονται στον επόμενο πίνακα και το αντίστοιχο διάγραμμα:

**Πίνακας 4.13** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση (pJ) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	7.825	23.628	49.064	83.182	186.480	304.640
Modified Booth (r4)	5.465	17.852	35.938	59.795	129.168	224.096
Modified Booth (r8)	4.022	15.150	29.347	49.362	96.012	180.224
Canonic Modified Booth (r2)	7.740	23.046	49.492	81.056	176.400	304.128
Canonic Modified Booth (r4)	5.297	17.970	34.042	61.268	130.272	222.592
Canonic Modified Booth (r8)	4.082	15.023	29.001	47.830	93.744	168.728
Parallel (MB)	3.033	9.210	17.430	30.967	57.936	97.228



**Σχήμα 4.18** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση ( $T = T_{hp}$ )

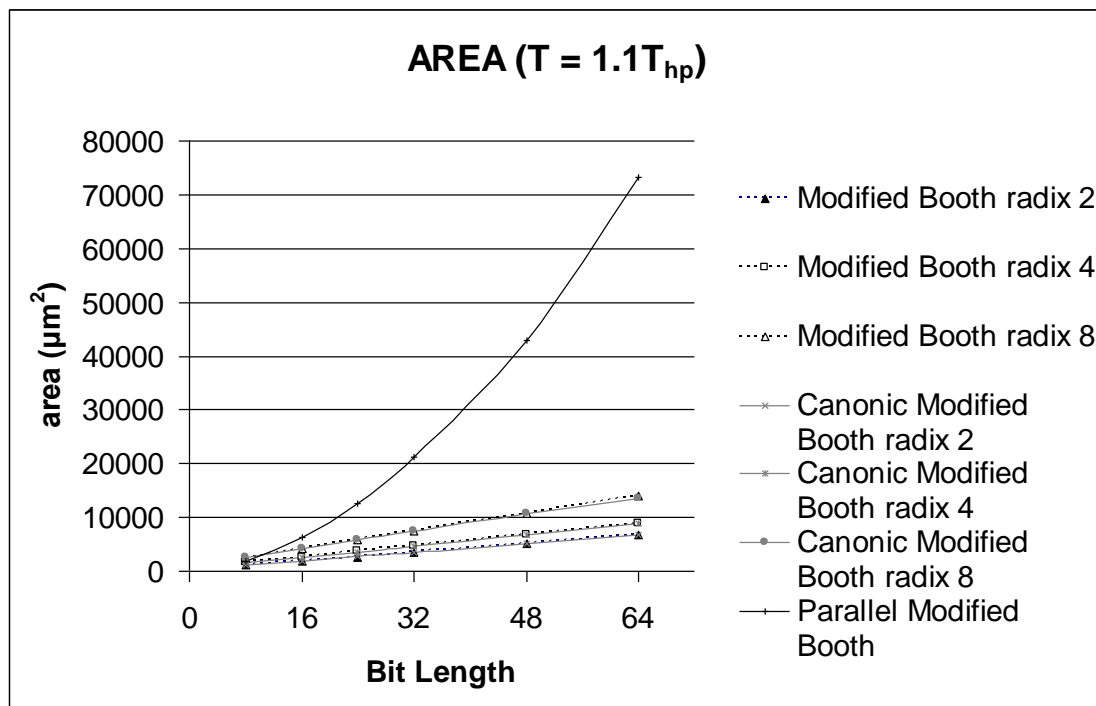
Τα πειραματικά αποτελέσματα δείχνουν ότι τα μεγαλύτερα radices παρουσιάζουν χαμηλότερη κατανάλωση ενέργειας. Πιο συγκεκριμένα, οι radix-2 Modified Booth σειριακοί πολλαπλασιαστές παρουσιάζουν 2.5-3 φορές μεγαλύτερη κατανάλωση από αυτή του παράλληλου Modified Booth πολλαπλασιαστή, ενώ οι radix-4 Modified Booth σειριακοί πολλαπλασιαστές παρουσιάζουν 1.8-2.3 φορές μεγαλύτερη κατανάλωση, ενώ τέλος οι radix-8 Modified Booth σειριακοί πολλαπλασιαστές παρουσιάζουν 1.3-1.8 φορές μεγαλύτερη κατανάλωση από αυτή του παράλληλου Modified Booth πολλαπλασιαστή, ο οποίος είναι ο πιο οικονομικός ενεργειακά.

Επίσης, μπορούμε να δούμε ότι οι δύο διαφορετικές κωδικοποιήσεις δεν παρουσιάζουν διαφοροποίηση ούτε ως προς την ενεργειακή κατανάλωση, οπότε η δεδομένη υλοποίηση δεν καταφέρνει τελικά να επιτύχει κάποιο ενεργειακό όφελος.

Παρόμοια αποτελέσματα προκύπτουν και από την μελέτη των παραπάνω κυκλωμάτων, όταν συντεθούν σε περίοδο μεγαλύτερη της high performance περιόδου.

**Πίνακας 4.14** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $\mu\text{m}^2$ ) ( $T = 1.1T_{hp}$ )

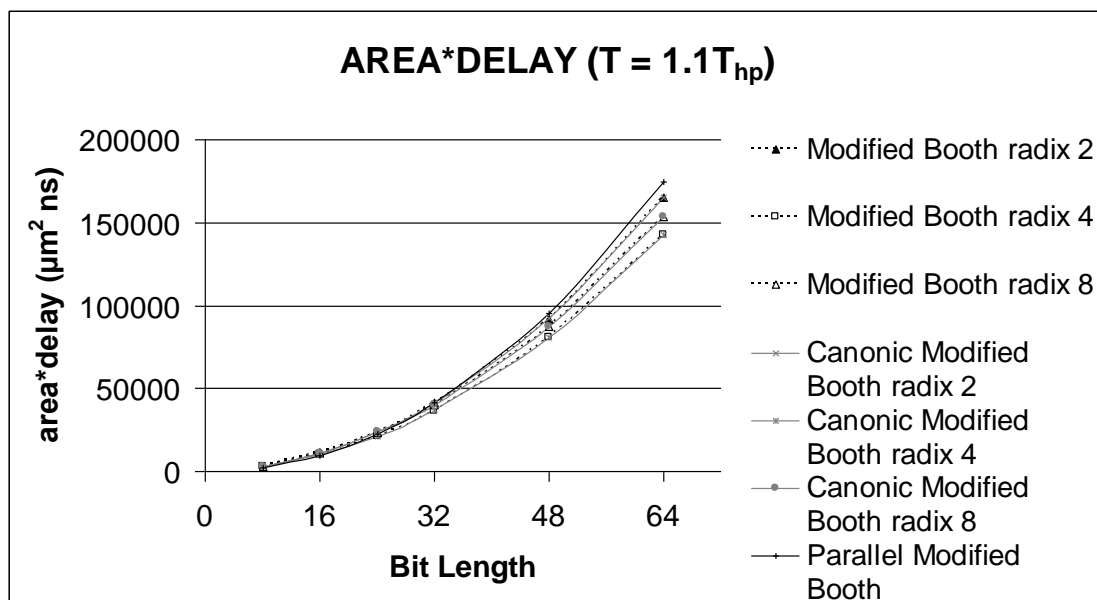
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	1109	1897	2677	3453	5092	6686
Modified Booth (r4)	1575	2586	3635	4611	6870	8811
Modified Booth (r8)	2324	4304	5878	7403	10808	13879
Canonic Modified Booth (r2)	1137	1919	1762	3521	5142	6795
Canonic Modified Booth (r4)	1605	2614	3578	4701	6766	8803
Canonic Modified Booth (r8)	2480	4242	5892	7361	10810	13600
Parallel (MB)	1935	6214	12524	21277	42921	73244



**Σχήμα 4.19** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $T = 1.1T_{hp}$ )

**Πίνακας 4.15** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $\mu m^2 ns$ ) ( $T = 1.1T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	2884	10925	23130	41430	91662	164750
Modified Booth (r4)	2866	9827	21375	36520	80548	142383
Modified Booth (r8)	2650	11277	23631	39681	87548	153226
Canonic Modified Booth (r2)	2957	10748	23861	41688	92551	165253
Canonic Modified Booth (r4)	2920	10039	21039	36854	80380	142258
Canonic Modified Booth (r8)	2778	11199	23685	39748	86911	153405
Parallel (MB)	2264	9817	22543	41278	95284	174321

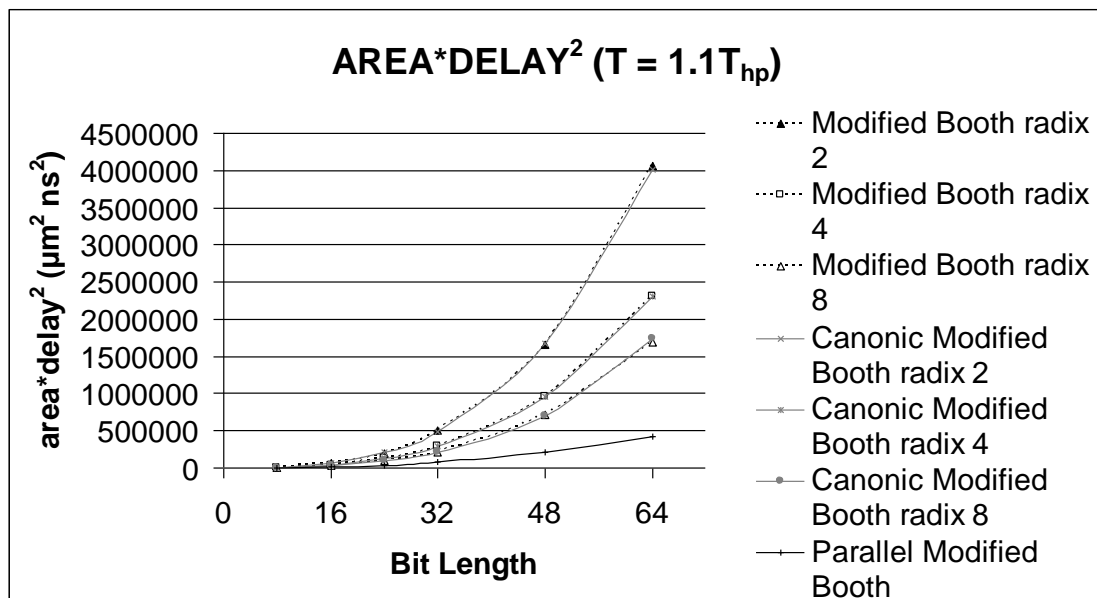


**Σχήμα 4.20** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $T = 1.1T_{hp}$ )



**Πίνακας 4.16** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $\mu m^2 ns^2$ ) ( $T = 1.1T_{hp}$ )

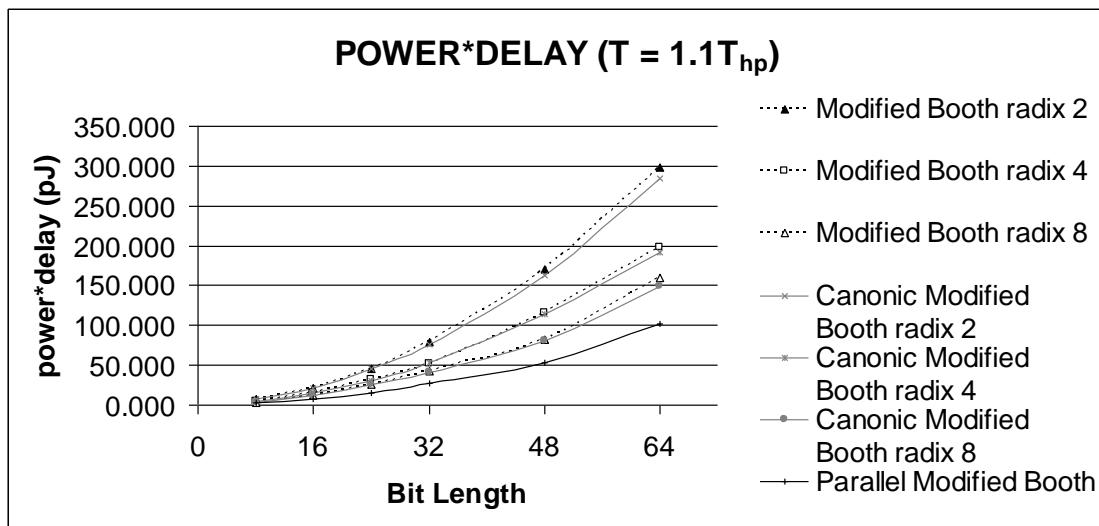
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	7498	62926	199840	497160	1649910	4059430
Modified Booth (r4)	5217	37342	125687	289237	956907	2300909
Modified Booth (r8)	3021	29545	94997	212690	709138	1691613
Canonic Modified Booth (r2)	7689	60187	206161	493586	1665913	4018944
Canonic Modified Booth (r4)	5315	38549	123711	288932	954915	2298882
Canonic Modified Booth (r8)	3111	29566	95213	214641	698762	1730408
Parallel (MB)	2648	15511	40577	80080	211532	414884



**Σχήμα 4.21** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $T = 1.1T_{hp}$ )

**Πίνακας 4.17** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση (pJ) ( $T = 1.1T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	6.900	21.686	46.405	79.260	169.741	298.144
Modified Booth (r4)	4.765	15.249	31.523	51.784	115.272	198.016
Modified Booth (r8)	3.379	12.888	25.869	42.180	82.416	159.040
Canonic Modified Booth (r2)	7.208	21.510	45.429	76.190	163.206	284.544
Canonic Modified Booth (r4)	4.819	15.107	30.704	52.912	114.389	192.304
Canonic Modified Booth (r8)	3.557	12.582	25.325	41.699	80.182	149.864
Parallel (MB)	2.457	7.820	15.098	26.852	53.520	102.565

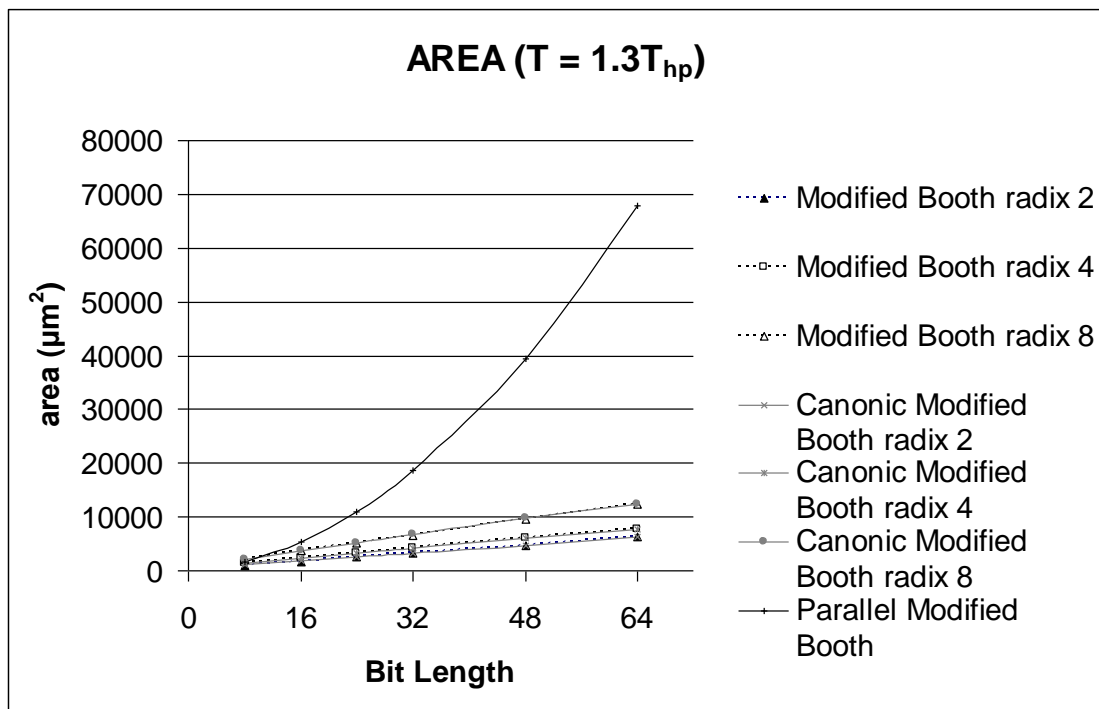


**Σχήμα 4.22** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση ( $T = 1.1T_{hp}$ )

Παρατηρούμε ότι τα κριτήρια  $area*delay$  και  $area*delay^2$  γίνονται μεγαλύτερα (άρα χειρότερα) όσο απομακρυνόμαστε από την συχνότητα high performance, ενώ η ενεργειακή κατανάλωση γίνεται γενικά μικρότερη, κάτι που είναι αναμενόμενο άλλωστε αφού είναι ανάλογη της συχνότητας λειτουργίας.

**Πίνακας 4.18** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $\mu\text{m}^2$ ) ( $T = 1.3T_{hp}$ )

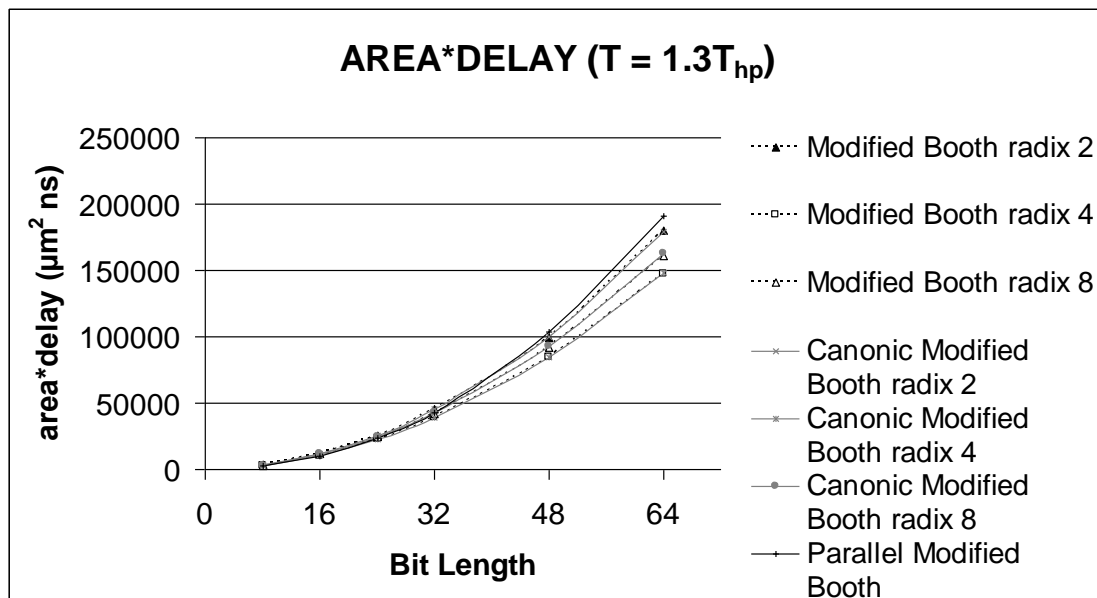
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	1048	1749	2487	3238	4708	6194
Modified Booth (r4)	1417	2312	3270	4181	5991	7661
Modified Booth (r8)	2023	3776	5160	6638	9523	12347
Canonic Modified Booth (r2)	1076	1783	2515	3263	4745	6218
Canonic Modified Booth (r4)	1461	2424	3306	4192	6034	7737
Canonic Modified Booth (r8)	2101	3800	5233	6655	9679	12253
Parallel (MB)	1747	5375	11039	18649	39442	67874



**Σχήμα 4.23** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $T = 1.3T_{hp}$ )

**Πίνακας 4.19** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $\mu m^2 ns$ ) ( $T = 1.3T_{hp}$ )

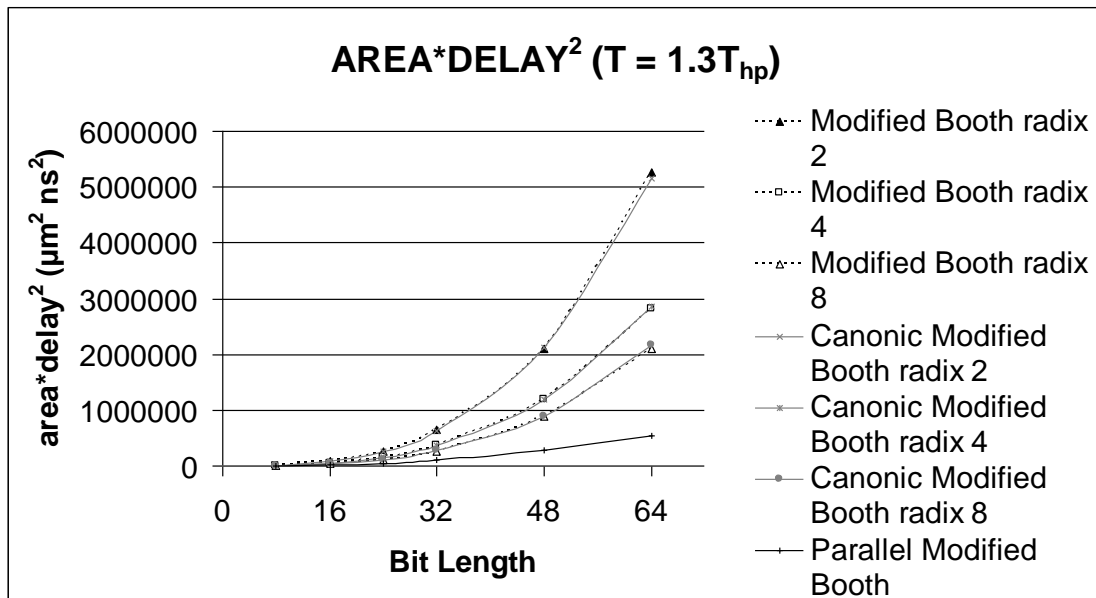
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	3227	11894	25370	45591	99443	180383
Modified Booth (r4)	3060	10356	22757	39131	84117	147085
Modified Booth (r8)	2731	11705	24613	42215	91419	161000
Canonic Modified Booth (r2)	3314	11839	25658	45427	100218	179071
Canonic Modified Booth (r4)	3156	10958	23013	38901	84712	148549
Canonic Modified Booth (r8)	2795	11857	24960	42593	92342	162726
Parallel (MB)	2411	10050	23513	42706	103733	190727



**Σχήμα 4.24** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $T = 1.3T_{hp}$ )

**Πίνακας 4.20** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $\mu m^2 ns^2$ ) ( $T = 1.3T_{hp}$ )

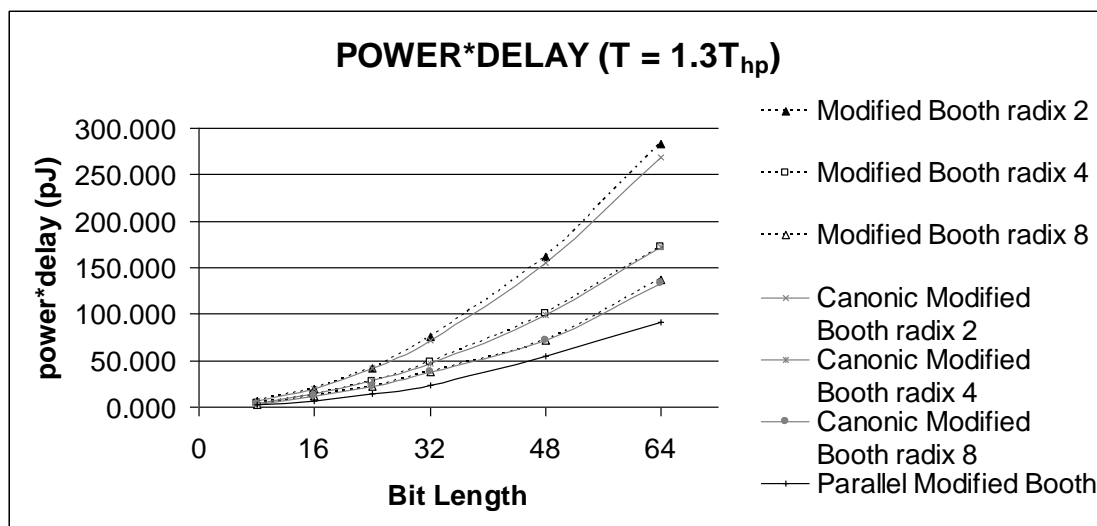
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	9940	80882	258772	641922	2100233	5252746
Modified Booth (r4)	6610	46394	158392	366268	1181005	2824040
Modified Booth (r8)	3687	36284	117406	268487	877619	2099434
Canonic Modified Booth (r2)	10208	78614	261709	632337	2116600	5157248
Canonic Modified Booth (r4)	6818	49532	160169	361006	1189350	2852132
Canonic Modified Booth (r8)	3717	36994	119060	272598	880940	2160999
Parallel (MB)	3327	18794	50083	97797	272819	535944



**Σχήμα 4.25** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $T = 1.3T_{hp}$ )

**Πίνακας 4.21** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση (pJ) ( $T = 1.3T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	6.308	20.128	42.259	75.229	161.315	282.697
Modified Booth (r4)	4.357	13.346	28.038	47.936	100.210	172.474
Modified Booth (r8)	3.021	11.207	22.448	37.823	71.232	136.920
Canonic Modified Booth (r2)	6.339	19.714	41.412	72.161	155.316	269.280
Canonic Modified Booth (r4)	4.445	13.763	28.118	46.950	99.191	172.246
Canonic Modified Booth (r8)	3.122	11.223	22.510	37.293	71.273	132.800
Parallel (MB)	2.271	7.084	14.169	24.045	54.704	90.763



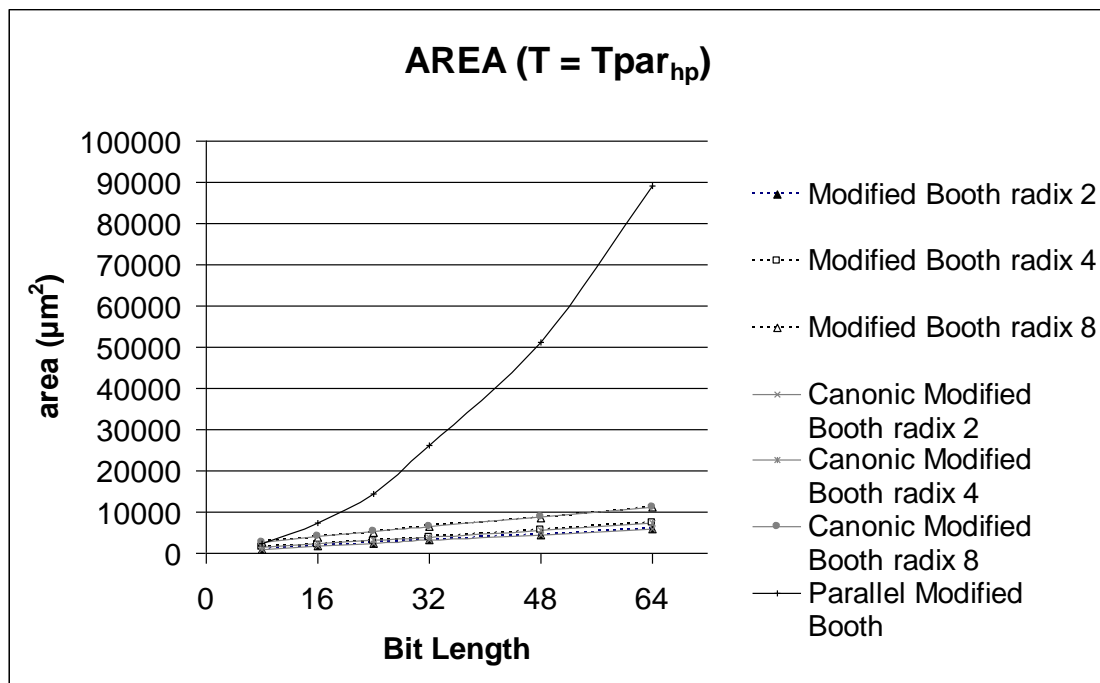
**Σχήμα 4.26** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση ( $T = 1.3T_{hp}$ )

Παρατηρούμε και πάλι ότι τα κριτήρια  $area*delay$  και  $area*delay^2$  γίνονται μεγαλύτερα (άρα χειρότερα) όσο απομακρυνόμαστε από την συχνότητα high performance, ενώ η ενεργειακή κατανάλωση γίνεται γενικά μικρότερη, κάτι που είναι αναμενόμενο άλλωστε αφού είναι ανάλογη της συχνότητας λειτουργίας.

Τέλος, επειδή παρατηρούμε ότι όσο απομακρυνόμαστε από τις high performance συχνότητες έχουμε μικρότερες ενεργειακές καταναλώσεις, δοκιμάσαμε να μετρήσουμε την απόδοση και κατανάλωση των κυκλωμάτων των σειριακών πολλαπλασιαστών όταν συντεθούν στις high performance συχνότητες των αντίστοιχων παράλληλων Modified Booth πολλαπλασιαστών.

**Πίνακας 4.22** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $\mu\text{m}^2$ ) ( $T = T_{\text{par}_{\text{hp}}}$ )

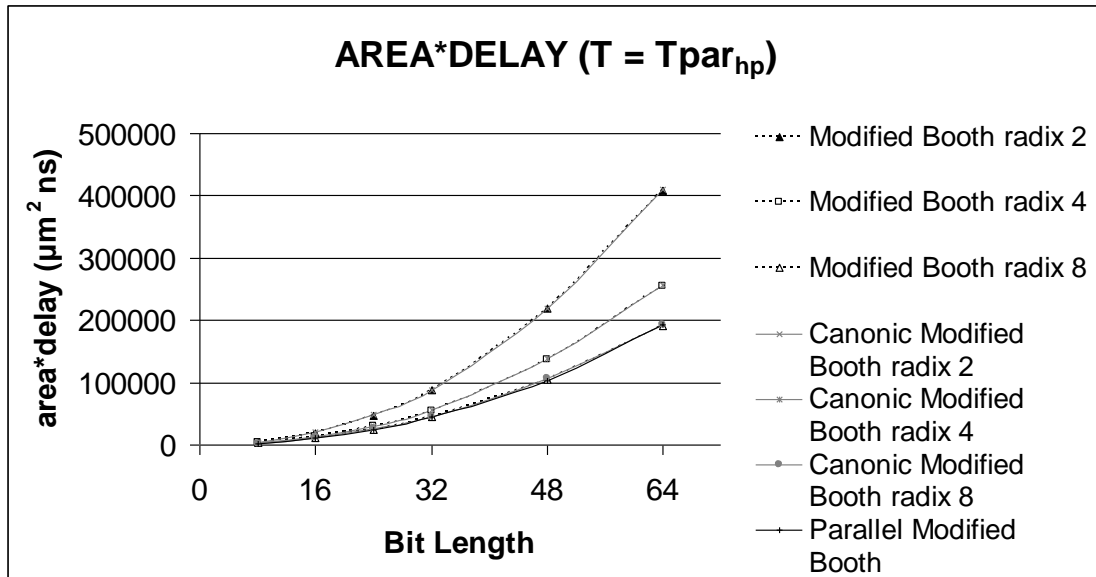
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	994	1697	2400	3101	4508	5915
Modified Booth (r4)	1420	2194	3056	3918	5643	7371
Modified Booth (r8)	2652	3922	5129	6405	8668	11089
Canonic Modified Booth (r2)	1022	1725	2428	3131	4536	5943
Canonic Modified Booth (r4)	1468	2237	3099	3961	5686	7414
Canonic Modified Booth (r8)	2605	3991	5164	6483	8741	1161
Parallel (MB)	2393	7430	14427	26043	51209	89193



**Σχήμα 4.27** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια ( $T = T_{\text{par}_{\text{hp}}}$ )

**Πίνακας 4.23** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $\mu m^2 ns$ ) ( $T = T_{par_{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	4215	19549	47227	87367	218552	408848
Modified Booth (r4)	3010	12636	30071	55168	136779	254731
Modified Booth (r8)	2811	11297	25235	45089	105060	191609
Canonic Modified Booth (r2)	4335	19874	47782	88162	219920	410799
Canonic Modified Booth (r4)	3111	12884	30494	55774	137822	256219
Canonic Modified Booth (r8)	2761	11494	25405	45641	105941	192865
Parallel (MB)	2536	10699	23661	45836	103442	192658

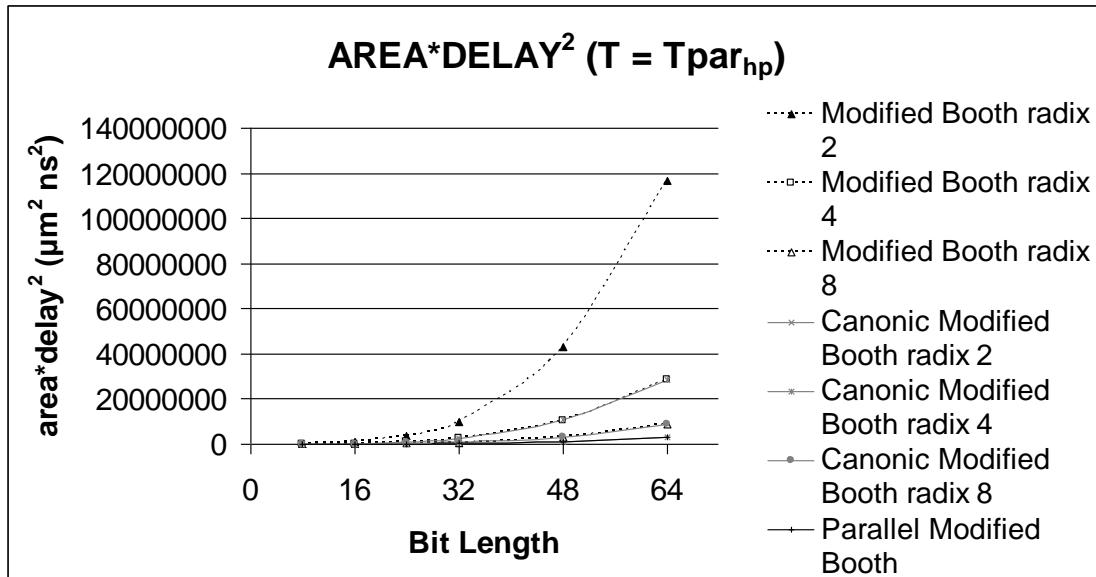


**Σχήμα 4.28** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  ( $T = T_{par_{hp}}$ )



**Πίνακας 4.24** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $\mu m^2 ns^2$ ) ( $T = T_{par_{hp}}$ )

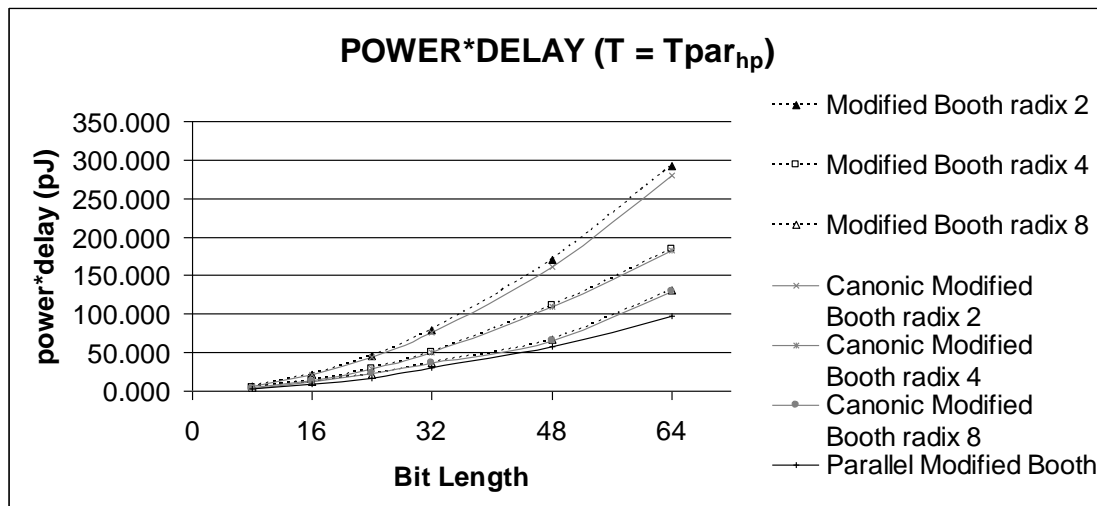
Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	54647	807930	3538437	9661975	42965284	116463261
Modified Booth (r4)	17873	225205	929427	2460256	10595384	28259567
Modified Booth (r8)	6381	72782	295895	776768	3315514	8803514
Canonic Modified Booth (r2)	2979	32534	124155	317429	1273324	3311010
Canonic Modified Booth (r4)	18381	228951	940358	2482637	10661719	28394409
Canonic Modified Booth (r8)	6596	74210	300062	785301	3340805	8854923
Parallel (MB)	2927	33102	124992	321310	1284000	3332711



**Σχήμα 4.29** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  ( $T = T_{par_{hp}}$ )

**Πίνακας 4.25** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση (pJ) ( $T = T_{par_{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Modified Booth (r2)	6.517	21.612	46.169	79.665	169.728	292.170
Modified Booth (r4)	4.348	13.743	28.752	50.153	110.971	184.516
Modified Booth (r8)	3.831	11.583	21.855	36.348	66.830	130.738
Canonic Modified Booth (r2)	6.525	21.116	44.378	76.004	161.972	279.452
Canonic Modified Booth (r4)	4.458	13.795	28.625	49.801	109.589	182.719
Canonic Modified Booth (r8)	3.739	11.701	22.273	36.397	65.860	129.662
Parallel (MB)	3.033	9.210	17.430	30.967	57.936	97.228



**Σχήμα 4.30** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση ( $T = T_{par_{hp}}$ )

Παρατηρούμε ότι το κριτήριο  $area*delay$  γίνεται πλέον μεγαλύτερο (άρα χειρότερο) από τον παράλληλο πολλαπλασιαστή, ωστόσο η ενεργειακή κατανάλωση γίνεται αρκετά μικρότερη και σε κάποιες περιπτώσεις πλησιάζει αρκετά αυτή του παράλληλου πολλαπλασιαστή.

# 5

## ***ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ***

Σε αυτό το κεφάλαιο θα συνοψίσουμε την παρουσίαση της διπλωματικής εργασίας μας.

### ***5.1 Σύνοψη και συμπεράσματα***

Συνοψίζοντας την μελέτη της παρούσας διπλωματικής εργασίας, διερευνήσαμε τις επιδόσεις ενός συνόλου συγγενών τοπολογιών σειριακών-παράλληλων πολλαπλασιαστών, με κωδικοποίηση Modified Booth σε διάφορα radices.

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

- Οι Modified Booth πολλαπλασιαστές υπερτερούν σαφώς έναντι των σειριακών-παράλληλων πολλαπλασιαστών. Με ένα μικρό κόστος στην συχνότητα λειτουργίας και μια λογική θυσία επιφάνειας ολοκληρωμένου, πετυχαίνουν χαμηλότερη κατανάλωση λειτουργίας, ταχύτερη εξαγωγή αποτελεσμάτων, μεγαλύτερο throughput, και καλύτερες επιδόσεις σύμφωνα με τα κριτήρια  $area*time$  και  $area*time^2$ .
- Όσο αυξάνεται το radix ενός modified Booth πολλαπλασιαστή, έχουμε εξοικονόμηση ενέργειας και ταχύτερη παραγωγή αποτελεσμάτων, με το κόστος μιας μικρής επιβάρυνσης σε καθυστέρηση και επιφάνεια ολοκληρωμένου. Όσον αφορά την επίδοση, το κύκλωμα του radix-4 πολλαπλασιαστή φαίνεται να έχει τα καλύτερα αποτελέσματα.

Μια συγκριτική παρουσίαση των επιμέρους μεγεθών, χρησιμοποιώντας ως σημείο αναφοράς τα μεγέθη του παράλληλου Modified Booth πολλαπλασιαστή, φαίνεται

στους επόμενους πίνακες, ώστε να τονιστεί το σχετικό κέρδος ή απώλεια που προκύπτει από την κάθε επιλογή. Όλα τα μεγέθη αναφέρονται στις συχνότητες high performance.

**Πίνακας 5.1** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια (%) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	38.10	23.40	20.15	13.87	10.43	7.68
Modified Booth (r2)	53.67	26.99	20.19	14.13	10.75	7.86
Modified Booth (r4)	75.08	41.48	28.32	20.42	15.02	11.22
Modified Booth (r8)	116.60	66.39	45.80	32.81	24.56	17.86
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.2** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το critical path (%) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	39.62	31.25	26.83	25.57	22.28	22.22
Modified Booth (r2)	55.66	45.14	39.63	38.64	33.66	32.41
Modified Booth (r4)	78.30	59.72	54.27	51.14	44.55	42.59
Modified Booth (r8)	98.11	82.64	74.39	69.32	60.89	57.87
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.3** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay$  (%) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	120.77	117.00	129.78	113.50	111.56	109.26
Modified Booth (r2)	119.49	97.46	96.02	87.36	86.82	81.54
Modified Booth (r4)	117.58	99.09	92.20	83.54	80.30	76.47
Modified Booth (r8)	114.40	109.73	102.22	90.97	89.74	82.67
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.4** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area*delay^2$  (%) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	382.83	585.00	835.63	928.62	1192.97	1553.90
Modified Booth (r2)	266.05	351.95	456.68	540.05	701.44	845.58
Modified Booth (r4)	184.13	236.73	300.22	341.75	429.32	521.11
Modified Booth (r8)	112.24	181.36	228.11	252.23	327.85	382.73
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.5** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση (%) ( $T = T_{hp}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	306.15	382.53	479.17	454.71	544.99	540.29
Modified Booth (r2)	257.96	256.56	281.49	268.62	321.87	313.33
Modified Booth (r4)	180.16	193.84	206.18	193.09	222.95	230.49
Modified Booth (r8)	132.58	164.50	168.37	159.40	165.72	185.36
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

Αν οι βέλτιστες επιδόσεις ή η μέγιστη συχνότητα λειτουργίας δεν είναι το ζητούμενο της σχεδίασης, τότε μπορούμε να συνθέσουμε τα κυκλώματα των σειριακών πολλαπλασιαστών Modified Booth με πιο χαλαρούς χρονικούς περιορισμούς. Σε αυτή την περίπτωση θα πετύχουμε χαμηλότερη κατανάλωση λειτουργίας, με κόστος την μικρότερη συχνότητα λειτουργίας και την χειροτέρευση των κριτηρίων επίδοσης. Το όριο αυτής της περίπτωσης είναι η high performance συχνότητα του παράλληλου πολλαπλασιαστή Modified Booth, στην οποία τα παραπάνω μεγέθη γίνονται:

**Πίνακας 5.6** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την επιφάνεια (%) ( $T = T_{par_{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	31.76	20.48	15.83	11.70	8.92	6.83
Modified Booth (r2)	41.55	22.84	16.63	11.91	8.80	6.63
Modified Booth (r4)	59.33	29.53	21.18	15.05	11.02	8.26
Modified Booth (r8)	110.82	52.79	35.55	24.59	16.93	12.43
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.7** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area \cdot delay$  (%) ( $T = T_{par_{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	254.08	327.75	379.95	374.28	428.38	437.29
Modified Booth (r2)	166.20	182.72	199.60	190.61	211.28	212.21
Modified Booth (r4)	118.67	118.10	127.09	120.36	132.23	132.22
Modified Booth (r8)	110.82	105.58	106.65	98.37	101.56	99.46
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.8** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς το κριτήριο  $area \cdot delay^2$  (%) ( $T = T_{par_{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	2032.67	5243.99	9118.76	11977.05	20562.19	27986.48
Modified Booth (r2)	664.82	1461.73	2395.19	3049.75	5070.70	6790.86
Modified Booth (r4)	237.33	472.40	762.54	962.89	1586.73	2115.51
Modified Booth (r8)	110.82	211.17	319.95	393.49	609.38	795.65
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

**Πίνακας 5.9** Σύγκριση σειριακών και παράλληλων πολλαπλασιαστών Modified Booth ως προς την κατανάλωση (%) ( $T = T_{par_{hp}}$ )

Multiplier	N=8	N=16	N=24	N=32	N=48	N=64
Serial parallel (Double Pipeline)	262.78	344.49	407.37	407.03	487.68	505.17
Modified Booth (r2)	214.83	234.66	264.88	257.26	292.96	300.50
Modified Booth (r4)	143.34	149.23	164.96	161.96	191.54	189.78
Modified Booth (r8)	126.28	125.77	125.39	117.38	115.35	134.47
Parallel (MB)	100.00	100.00	100.00	100.00	100.00	100.00

Βλέπουμε λοιπόν ότι, στην οριακή αυτή περίπτωση, οι radix-8 Modified Booth σειριακοί πολλαπλασιαστές πετυχαίνουν αρκετά χαμηλές καταναλώσεις λειτουργίας, συγκρίσιμες με αυτές των παράλληλων πολλαπλασιαστών Modified Booth.

- Όσον αφορά την κωδικοποίηση Canonic Modified Booth, αυτή δεν έδωσε τα αναμενόμενα αποτελέσματα, δηλαδή δεν παρατηρήθηκε κάποια εξοικονόμηση ενέργειας σε σχέση με την απλή κωδικοποίηση Modified Booth. Αυτό πιθανότατα οφείλεται στο γεγονός ότι ο πλήρης άθροιστής που είναι η βασική μονάδα όλων των κυκλωμάτων αποτελείται από δύο διαδοχικές πύλες XOR οπότε ακόμα και στην περίπτωση του μηδενικού μερικού γινομένου εκτελεί δυο μεταβάσεις για να διατηρήσει την προηγούμενη τιμή του, οπότε δεν συνεισφέρει στην ενεργειακή εξοικονόμηση που αναμέναμε.

## 5.2 Μελλοντικές επεκτάσεις

- Τα κυκλώματα που κάνουν χρήση της κωδικοποίησης Canonic Modified Booth θα μπορούσαν να τροποποιηθούν, ώστε να απομονώνουν πλήρως το εκάστοτε κύκλωμα άθροισης σε περίπτωση μηδενικού μερικού γινομένου, σε αυτή την περίπτωση θα πρέπει και οι καταχωρητές του πρώτου σταδίου να μπορούν να εκτελέσουν την λειτουργία της ολίσθησης. Με αυτό τον τρόπο πιθανόν θα γινόταν εμφανές το πλεονέκτημα της κωδικοποίησης αυτής.

- Το δεύτερο στάδιο των σειριακών πολλαπλασιαστών Modified Booth αποτελείται ουσιαστικά από καταχωρητές ολίσθησης, αυτό σημαίνει ότι μπορούν να διερευνηθούν τρόποι χειρισμού των δεδομένων τους ώστε να επιτευχθεί επιπλέον εξοικονόμηση ενέργειας. Πχ η μεταφορά μεγαλύτερων «πακέτων» bits αντί η ολίσθηση όλων των bit κατά μια θέση κάθε φορά, καθώς αυτή η λειτουργία παράγει πολύ μεγάλη μεταγωγική δραστηριότητα.
- Στο ίδιο πνεύμα με την προηγούμενη παρατήρηση, θα μπορούσαν να διερευνηθούν αρχιτεκτονικές FPGA, οι οποίες διαθέτουν έτοιμα κυκλώματα καταχωρητών ολίσθησης, οπότε μια τοπολογία σειριακού πολλαπλασιαστή θα ήταν πιο εύκολα υλοποιήσιμη και περισσότερο ενεργειακά αποδοτική.
- Η μελέτη των κυκλωμάτων των σειριακών πολλαπλασιαστών θα μπορούσε να επεκταθεί και σε μεγαλύτερα radices και μήκη λέξης, δεδομένου ότι σε μεγαλύτερα radices θα κάναμε χρήση δενδρικών αθροιστών ως βασικών δομικών στοιχείων. Αυτό πιθανώς θα είχε καλύτερα αποτελέσματα στην καθυστέρηση των κυκλωμάτων, επομένως και στα επιμέρους κριτήρια.
- Τέλος, για να αναδειχτούν τα ενεργειακά πλεονεκτήματα των παραπάνω σχημάτων, μπορούν να συγκριθούν με άλλα κυκλώματα που έχουν προταθεί στη σχετική βιβλιογραφία, τα οποία κάνουν χρήση της απλής Carry-Save αναπαράστασης για την αποθήκευση των αποτελεσμάτων μέχρι το τελικό στάδιο άθροισης, στοχεύοντας στην ελάχιστη καθυστέρηση.

# 6

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [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] Zhijun Huang, “**High-Level Optimization Techniques for Low-Power Multiplier Design**“
- [4] Dimitris Bekiaris, George Economakos, Kiamal Z. Pekmestzi, “**A High-Speed Radix-16 Array Multiplier**”.
- [5] O. L. MacSorley, “**High Speed Arithmetic in Binary Computers**”
- [6] G. W. Reitweisner , “**Binary Arithmetic Advance in Computers**” pp 231-308
- [7] C. S. Wallace, “**A Suggestion for a Fast Multiplier**”
- [8] L. Dadda, “**Some Schemes for Parallel Multipliers**”
- [9] Andrew D. Booth, “**A Signed Binary Multiplication Technique**”
- [10] R. P. Brent, H. T. Kung, “**A Regular Layout for Parallel Adders**”
- [11] Synopsys Design Compiler, [www.synopsys.com](http://www.synopsys.com).
- [12] ModelSim Corporation, [www.model.com](http://www.model.com).
- [13] Synopsys Primetime PX, [www.synopsys.com](http://www.synopsys.com).



