ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ ΣΤΗΝ
«ΕΠΙΣΤΗΜΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ»

Μεταπτυχιακή Διπλωματική Εργασία

Τίτλος:

Συνδυασμένες μονάδες πολλαπλασιασμού / αθροίσματος τετραγώνων για αριθμητικά συστήματα υπολοίπων

Αδαμίδης Δημήτριος (ΑΜ: 318)
adamidis@ceid.upatras.gr

Επιβλέπων Καθηγητής :
Χ. Βέργος

Πάτρα, Μάιος 2005

Περιεχόμενα

1. Εισαγωγή

Τα αριθμητικά συστήματα υπολοίπων (Residue Number Systems, RNS), αποτελούν μια αξιόλογη εναλλακτική πρόταση έναντι του δυαδικού συστήματος για εφαρμογές οι οποίες περιορίζονται στις αριθμητικές πράξεις της πρόσθεσης, της αφαίρεσης, του πολλαπλασιασμού και του τετραγωνισμού. Ένα αριθμητικό σύστημα υπολοίπων καθορίζεται από ένα σύνολο F ακεραίων, π.χ. $\{m_1, m_2, ..., m_F\}$, οι οποίοι είναι ανά δύο πρώτοι μεταξύ τους. Ας υποθέσουμε ότι το $|A|_M$ υποδηλώνει το modulo M του A, δηλαδή το μικρότερο μη αρνητικό υπόλοιπο της ακέραιης διαίρεσης του A με το M. Τότε, ένας ακέραιος A έχει μια μοναδική αναπαράσταση στο RNS, η οποία δίνεται από το σύνολο υπολοίπων $\{a_1, a_2, ..., a_F\}$, όπου $a_i = |A|_{m_i}$ αν $A \geq 0$ και $a_i = |M + A|_{m_i}$ αν $A < 0$, όπου $M = m_1 \times m_2 \times ... m_F$ και $1 \leq i \leq F$. Μια πράξη $\Diamond$ του RNS καθορίζεται ως $(z_1, z_2, ..., z_F) = (a_1, a_2, ..., a_F) \Diamond (b_1, b_2, ..., b_F)$, όπου $z_i = |a_i \Diamond b_i|_{m_i}$. Παρατηρούμε ότι ο υπολογισμός του z_i εξαρτάται μόνο από τα α_i, b_i και m_i. Συνεπώς, ο υπολογισμός του κάθε z_i μπορεί να γίνεται παράλληλα σε μια ξεχωριστή αριθμητική μονάδα, η οποία καλείται συχνά κανάλι. Παρατηρούμε ότι κάθε κανάλι χειρίζεται μόνο μικρά υπόλοιπα και όχι μεγάλους αριθμούς, καθώς και ότι όλα τα κανάλια λειτουργούν παράλληλα χωρίς διάδοση κρατουμένου από το ένα στο άλλο. Αυτό σημαίνει ότι μπορεί εύκολα να επιτευχθεί μια σημαντική επιτάχυνση της διαδικασίας έναντι της αντίστοιχης του δυαδικού συστήματος.

Αρκετοί επεξεργαστές ψηφιακών σημάτων (Digital Signal Processors, DSP), οι οποίοι χρησιμοποιούνται σε τηλεπικοινωνιακές και άλλες εφαρμογές, έχουν κατασκευαστεί με χρήση RNS συστημάτων και πολλοί ακόμα αναμένονται στο μέλλον, δεδομένου ότι θα υπάρχουν διαθέσιμα αποδοτικά modulo m_i αριθμητικά υποσυστήματα. Στην ανοιχτή βιβλιογραφία έχουν παρουσιαστεί πολύ αποδοτικές αρχιτεκτονικές για αθροιστές, πολλαπλασιαστές και τετραγωνιστές όπου το m_i είναι $2^n - 1$ ή $2^n + 1$. Είναι επομένως λογικό ότι τα RNS συστήματα που βασίζονται στο σύνολο $\{2^n, 2^n - 1, 2^n + 1\}$ βρίσκονται στο επίκεντρο του ενδιαφέροντος και χρησιμοποιούνται συχνά.

Παρόλα αυτά, το παραπάνω σύνολο ακεραίων δημιουργεί ένα νέο πρόβλημα: το modulo 2^n+1 κανάλι χειρίζεται αριθμούς οι οποίοι είναι κατά ένα bit μεγαλύτεροι σε σχέση με τα άλλα δύο κανάλια. Σε μια πρόχειρη υλοποίηση επομένως η απόδοση του 2^n+1 καναλιού θα ήταν μικρότερη από αυτή των άλλων δύο καναλιών και θα περιόριζε την συνολική ταχύτητα των RNS υπολογισμών. Για να ξεπεραστεί αυτό το πρόβλημα, ο Leibowitz πρότεινε τη χρήση της ελαττωμένης κατά ένα (diminished-1) αναπαράστασης.

Στην ελαττωμένη κατά ένα αναπαράσταση, κάθε αριθμός παρουσιάζεται μειωμένος κατά ένα modulo 2^n+1. Επιπλέον, όλες οι αριθμητικές πράξεις απαγορεύονται για μηδενικούς αριθμούς (οι οποίοι αναγνωρίζονται εύκολα επειδή η ελαττωμένη κατά ένα αναπαράστασή τους έχει μονάδα στην περισσότερο σημαντική θέση). Η αναπαράσταση αυτή έχει το σημαντικό πλεονέκτημα ότι όλοι οι αριθμοί μπορούν να αναπαρασταθούν με τη χρήση n bits. Το μόνο μειονέκτημα είναι ότι χρειάζονται μετατροπείς από και προς την ελαττωμένη κατά ένα αναπαράσταση. Παρόλα αυτά, ο χρόνος που απαιτείται για τις μετατροπές αποτελεί μόνο ένα πολύ μικρό ποσοστό του συνολικού χρόνου των υπολογισμών, δεδομένου ότι ένα RNS σύστημα χρησιμοποιείται όταν πραγματοποιείται μια σειρά αριθμητικών πράξεων προτού απαιτηθεί μια μετατροπή. Στην ανοιχτή βιβλιογραφία έχουν παρουσιαστεί διάφοροι αποδοτικοί αθροιστές, πολλαπλασιαστές και τετραγωνιστές για το ελαττωμένο κατά ένα αριθμητικό σύστημα.

Μια ακόμα λειτουργία που συναντάται συχνά σε αλγορίθμους πολυμέσων και DSP είναι ο υπολογισμός του αθροίσματος τετραγώνων. Αυτή χρησιμοποιείται συχνά σε υπολογισμό Ευκλείδειων διακλαδώσεων, συμπίεση εικόνων, αναγνώριση προτύπων, εκτίμηση κίνησης σε υπολογιστικά συστήματα όρασης, κώδικες συνέλιξης Viterbi, ισοστάθμιση καναλιών, εφαρμογές φίλτρων και στατιστική. Για την εκτέλεση της πράξης του αθροίσματος τετραγώνων μπορεί να χρησιμοποιηθεί ένα ζευγάρι πολλαπλασιαστή και αθροιστή, ή ένα ζευγάρι τετραγωνιστή και αθροιστή. Αυτή η προσέγγιση όμως εισάγει μια σημαντική καθυστέρηση και καθιστά αναγκαία τη χρήση μιας ενδιάμεσης μνήμης αποθήκευσης. Μια ξεχωριστή μονάδα αθροίσματος τετραγώνων μπορεί να δώσει πολύ καλύτερα αποτελέσματα. Αυτή η λύση όμως αυξάνει σημαντικά τις χωρικές απαιτήσεις της υλοποίησης.

Στην παρούσα εργασία προτείνεται μια λύση η οποία τοποθετείται κάπου ανάμεσα στις δύο παραπάνω ακραίες περιπτώσεις, εφαρμόζοντας την ιδέα του

διαμοιρασμού χώρου ανάμεσα σε δύο μονάδες. Αυτές οι νέες μονάδες μπορούν να εκτελέσουν είτε την πράξη του πολλαπλασιασμού είτε την πράξη του αθροίσματος τετραγώνων, ανάλογα με την τιμή ενός σήματος ελέγχου. Εξετάζεται τόσο η περίπτωση $2^n - 1$ όσο και η περίπτωση $2^n + 1$. Οι μονάδες που παρουσιάζονται ονομάζονται στην πρώτη περίπτωση $MMSSU_{-1}$, ενώ στην δεύτερη περίπτωση $MMSSU_{+1}$. Για την περίπτωση $2^n + 1$, υποθέτουμε ότι χρησιμοποιείται για όλους τους αριθμούς η ελαττωμένη κατά ένα αναπαράσταση. Τόσο οι χωρικές όσο και οι χρονικές απαιτήσεις των νέων μονάδων συγκρίνονται με τις απαιτήσεις των αντίστοιχων πολλαπλασιαστών και αποδεικνύεται ότι μπορούν να κατασκευαστούν με σχετικά μικρή επιβάρυνση.

Το υπόλοιπο της εργασίας αυτής είναι οργανωμένο ως εξής. Στο τμήμα 2 παρουσιάζονται δύο προτεινόμενες αρχιτεκτονικές για μονάδες modulo $2^n - 1$. Αντίστοιχα, στο τμήμα 3 παρουσιάζονται δύο προτεινόμενες αρχιτεκτονικές για μονάδες modulo $2^n + 1$. Σε κάθε περίπτωση, παρουσιάζονται οι απαιτήσεις σε χώρο και οι καθυστερήσεις των σχεδιασμών, καθώς και αποτελέσματα ποιοτικών συγκρίσεων. Αποτελέσματα ποσοτικών συγκρίσεων παρουσιάζονται στο τμήμα 4, όπου οι προτεινόμενες αρχιτεκτονικές συγκρίνονται με τους αντίστοιχους πολλαπλασιαστές. Τέλος, στο τμήμα 5 υπάρχει μια σύντομη ανασκόπηση καθώς και διάφορα συμπεράσματα που προκύπτουν από τα προηγούμενα.

2. Προτεινόμενες αρχιτεκτονικές MMSSU$_{-1}$

Θεωρούμε δύο αριθμούς A, B των n bits, όπου $A = a_{n-1}a_{n-2}...a_1a_0$ και $B = b_{n-1}b_{n-2}...b_1b_0$. Έστω ακόμα ότι οι αριθμοί αυτοί ακολουθούν την modulo $2^n - 1$ αναπαράσταση. Προφανώς ισχύει

$$A = \sum_{i=0}^{n-1} a_i 2^i \text{ και } B = \sum_{j=0}^{n-1} b_j 2^j .$$

Για τον modulo $2^n - 1$ πολλαπλασιασμό των αριθμών αυτών θα ισχύει

$$\left|A \times B\right|_{2^n-1} = \left|\sum_{i=0}^{n-1} a_i 2^i \sum_{j=0}^{n-1} b_j 2^j\right|_{2^n-1} = \left|\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i 2^i b_j 2^j\right|_{2^n-1} = \left|\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i b_j 2^{i+j}\right|_{2^n-1} = \left|\sum_{j=0}^{n-1} PP_j\right|_{2^n-1}$$

όπου

$$PP_j = \left|\sum_{i=0}^{n-1} a_i b_j 2^{i+j}\right|_{2^n-1} .$$

Το PP_j αναπαριστά ένα μερικό γινόμενο στον πίνακα του modulo $2^n - 1$ πολλαπλασιασμού των A και B.

Έστω τώρα ένας αριθμός $C = c_{n-1}c_{n-2}...c_1c_0$ ο οποίος βρίσκεται μέσα στο $[0,2^n - 1)$. Στο [12] έχει δειχθεί ότι ισχύει

$$\left|C2^j\right|_{2^n-1} = c_{|n-1-j|_n} c_{|n-2-j|_n} ...c_{|1-j|_n} c_{|0-j|_n} . \quad (1)$$

Αυτό σημαίνει ότι οι όροι PP_j μπορούν να αναπαρασταθούν με n bits, ολισθαίνοντας τα bits των μερικών γινομένων με βάρος $i + j \geq n$ στην θέση $\left|n + i + j\right|_n$. Έστω ότι $n = 8$. Τότε θα έχουμε τον παρακάτω πίνακα.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_7 b_0$	$\alpha_6 b_0$	$\alpha_5 b_0$	$\alpha_4 b_0$	$\alpha_3 b_0$	$\alpha_2 b_0$	$\alpha_1 b_0$	$\alpha_0 b_0$
PP_1 =	$\alpha_6 b_1$	$\alpha_5 b_1$	$\alpha_4 b_1$	$\alpha_3 b_1$	$\alpha_2 b_1$	$\alpha_1 b_1$	$\alpha_0 b_1$	$\alpha_7 b_1$
PP_2 =	$\alpha_5 b_2$	$\alpha_4 b_2$	$\alpha_3 b_2$	$\alpha_2 b_2$	$\alpha_1 b_2$	$\alpha_0 b_2$	$\alpha_7 b_2$	$\alpha_6 b_2$
PP_3 =	$\alpha_4 b_3$	$\alpha_3 b_3$	$\alpha_2 b_3$	$\alpha_1 b_3$	$\alpha_0 b_3$	$\alpha_7 b_3$	$\alpha_6 b_3$	$\alpha_5 b_3$
PP_4 =	$\alpha_3 b_4$	$\alpha_2 b_4$	$\alpha_1 b_4$	$\alpha_0 b_4$	$\alpha_7 b_4$	$\alpha_6 b_4$	$\alpha_5 b_4$	$\alpha_4 b_4$
PP_5 =	$\alpha_2 b_5$	$\alpha_1 b_5$	$\alpha_0 b_5$	$\alpha_7 b_5$	$\alpha_6 b_5$	$\alpha_5 b_5$	$\alpha_4 b_5$	$\alpha_3 b_5$
PP_6 =	$\alpha_1 b_6$	$\alpha_0 b_6$	$\alpha_7 b_6$	$\alpha_6 b_6$	$\alpha_5 b_6$	$\alpha_4 b_6$	$\alpha_3 b_6$	$\alpha_2 b_6$
PP_7 =	$\alpha_0 b_7$	$\alpha_7 b_7$	$\alpha_6 b_7$	$\alpha_5 b_7$	$\alpha_4 b_7$	$\alpha_3 b_7$	$\alpha_2 b_7$	$\alpha_1 b_7$

Πίνακας 1. Μερικά γινόμενα για πολλαπλασιασμό modulo $2^n - 1$.

Για την modulo $2^n - 1$ μείωση των μερικών γινομένων σε δύο μόνο προσθετέους μπορεί να χρησιμοποιηθεί μια αρχιτεκτονική που βασίζεται σε ένα δέντρο πλήρων αθροιστών (δέντρο Wallace). Επιπλέον, στην αρχιτεκτονική αυτή θα πρέπει να έχουμε end-around-carry, δηλαδή το κρατούμενο της πιο σημαντικής βαθμίδας θα πρέπει να επιστρέφει στην λιγότερο σημαντική βαθμίδα κάθε γραμμής. Η χρήση του δέντρου Wallace σε έναν δυαδικό πολλαπλασιαστή περιπλέκει τη δομή του και οι διασυνδέσεις ανάμεσα στους πλήρεις αθροιστές δεν παρουσιάζουν κάποια κανονικότητα. Το γεγονός αυτό αποτελεί ένα μεγάλο πρόβλημα και για το λόγο αυτό δεν υπάρχουν πολλές VLSI υλοποιήσεις δυαδικών πολλαπλασιαστών οι οποίοι χρησιμοποιούν το δέντρο Wallace. Σε έναν δυαδικό πολλαπλασιαστή, η κάθε στήλη του πίνακα των μερικών γινομένων έχει διαφορετικό μέγεθος και αυτό είναι η κυριότερη πηγή των προβλημάτων. Αντίθετα, σε έναν modulo πολλαπλασιαστή το μέγεθος κάθε στήλης είναι το ίδιο. Στην περίπτωση αυτή το δέντρο Wallace παρουσιάζει μεγάλη κανονικότητα και επομένως η υλοποίησή του γίνεται πολύ ευκολότερη. Ας θεωρήσουμε ότι c_n είναι το κρατούμενο της πιο σημαντικής βαθμίδας κατά τη διάρκεια ενός σταδίου i της μείωσης. Προφανώς το c_n έχει βάρος 2^n. Εφόσον

$$\left| c_n 2^n \right|_{2^n - 1} = \left| c_n \right|_{2^n - 1}, \quad (2)$$

το c_n μπορεί πολύ απλά να προστεθεί στην λιγότερο σημαντική βαθμίδα του επόμενου σταδίου της μείωσης. Με τον τρόπο αυτό σχηματίζεται μια αρχιτεκτονική δέντρου modulo άθροισης με end-around-carry ([33]). Μετά από το δέντρο αυτό της modulo άθροισης θα πρέπει να έχουν απομείνει μόνο δύο προσθετέοι. Αυτοί οι δύο προσθετέοι θα πρέπει να προστεθούν modulo $2^n - 1$ προκειμένου να πάρουμε το

τελικό αποτέλεσμα. Για αυτή την τελική άθροιση μπορεί να χρησιμοποιηθεί ένας παράλληλος modulo $2^n - 1$ αθροιστής ([8], [9] και [10]).

Ας εξετάσουμε τώρα την περίπτωση του modulo αθροίσματος τετραγώνων. Γνωρίζουμε ότι ισχύει

$$A^2 = A \times A = \sum_{i=0}^{n-1} a_i 2^{2i} + 2\left(\sum_{i=1}^{n-1} \sum_{j=0}^{i-1} a_i a_j 2^{i+j} \right)$$

και

$$B^2 = B \times B = \sum_{i=0}^{n-1} b_i 2^{2i} + 2\left(\sum_{i=1}^{n-1} \sum_{j=0}^{i-1} b_i b_j 2^{i+j} \right).$$

Το άθροισμα τετραγώνων $A^2 + B^2$ παράγει

$$A^2 + B^2 = \sum_{i=0}^{n-1} (a_i + b_i) 2^{2i} + 2\left(\sum_{i=1}^{n-1} \sum_{j=0}^{i-1} (a_i a_j + b_i b_j) 2^{i+j} \right) = \sum_{i=0}^{n-1} (a_i + b_i) 2^{2i} + \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} (a_i a_j + b_i b_j) 2^{i+j+1} \quad (3)$$

Ο όρος $\sum_{i=0}^{n-1} (a_i + b_i) 2^{2i}$ μπορεί να αντικατασταθεί από τα αντίστοιχα αθροίσματα και κρατούμενα για κάθε βαθμίδα. Με άλλα λόγια ισχύει

$$\sum_{i=0}^{n-1} (a_i + b_i) 2^{2i} = \sum_{i=0}^{n-1} (a_i \oplus b_i) 2^{2i} + 2 \sum_{i=0}^{n-1} a_i b_i 2^{2i} = \sum_{i=0}^{n-1} (a_i \oplus b_i) 2^{2i} + \sum_{i=0}^{n-1} a_i b_i 2^{2i+1}.$$

Επομένως το άθροισμα τετραγώνων $A^2 + B^2$ μπορεί να ξαναγραφτεί ως εξής:

$$A^2 + B^2 = \sum_{i=0}^{n-1} (a_i \oplus b_i) 2^{2i} + \sum_{i=0}^{n-1} a_i b_i 2^{2i+1} + \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} (a_i a_j + b_i b_j) 2^{i+j+1}. \quad (4)$$

Τα μερικά γινόμενα του modulo $2^n - 1$ αθροίσματος τετραγώνων μπορούν να παραχθούν με τρόπο ανάλογο της περίπτωσης του πολλαπλασιασμού, εφαρμόζοντας δηλαδή την (1) στους όρους της (4). Παρατηρούμε ότι στην γενική περίπτωση απαιτούνται $n+2$ μερικά γινόμενα για τον σχηματισμό του πίνακα του modulo αθροίσματος τετραγώνων. Όπως και στην περίπτωση του πολλαπλασιασμού, μπορεί να χρησιμοποιηθεί μια βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου με end-around-carry για την μείωση των γραμμών σε δύο προσθετέους, ενώ για την τελική modulo πρόσθεση μπορεί να χρησιμοποιηθεί ένας modulo $2^n - 1$ παράλληλος αθροιστής. Προφανώς η έξοδος του αθροιστή αυτού είναι το επιθυμητό αποτέλεσμα. Για παράδειγμα, όταν $n = 8$ έχουμε $8 + 2 = 10$ γραμμές μερικών γινομένων, όπως φαίνεται και στον παρακάτω πίνακα.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_6\alpha_0$	$\alpha_5\alpha_0$	$\alpha_4\alpha_0$	$\alpha_3\alpha_0$	$\alpha_2\alpha_0$	$\alpha_1\alpha_0$	$\alpha_0 b_0$	$\alpha_7\alpha_0$
PP_1 =	$\alpha_5\alpha_1$	$\alpha_4\alpha_1$	$\alpha_3\alpha_1$	$\alpha_2\alpha_1$	$\alpha_1 b_1$	$b_0 b_1$	$\alpha_7\alpha_1$	$\alpha_6\alpha_1$
PP_2 =	$\alpha_4\alpha_2$	$\alpha_3\alpha_2$	$\alpha_2 b_2$	$b_1 b_2$	$b_0 b_2$	$\alpha_7\alpha_2$	$\alpha_6\alpha_2$	$\alpha_5\alpha_2$
PP_3 =	$\alpha_3 b_3$	$b_2 b_3$	$b_1 b_3$	$b_0 b_3$	$\alpha_7\alpha_3$	$\alpha_6\alpha_3$	$\alpha_5\alpha_3$	$\alpha_4\alpha_3$
PP_4 =	$b_2 b_4$	$b_1 b_4$	$b_0 b_4$	$\alpha_7\alpha_4$	$\alpha_6\alpha_4$	$\alpha_5\alpha_4$	$\alpha_4 b_4$	$b_3 b_4$
PP_5 =	$b_1 b_5$	$b_0 b_5$	$\alpha_7\alpha_5$	$\alpha_6\alpha_5$	$\alpha_5 b_5$	$b_4 b_5$	$b_3 b_5$	$b_2 b_5$
PP_6 =	$b_0 b_6$	$\alpha_7\alpha_6$	$\alpha_6 b_6$	$b_5 b_6$	$b_4 b_6$	$b_3 b_6$	$b_2 b_6$	$b_1 b_6$
PP_7 =	$\alpha_7 b_7$	$b_6 b_7$	$b_5 b_7$	$b_4 b_7$	$b_3 b_7$	$b_2 b_7$	$b_1 b_7$	$b_0 b_7$
PP_8 =	0	$\alpha_3 \oplus b_3$	0	$\alpha_2 \oplus b_2$	0	$\alpha_1 \oplus b_1$	0	$\alpha_0 \oplus b_0$
PP_9 =	0	$\alpha_7 \oplus b_7$	0	$\alpha_6 \oplus b_6$	0	$\alpha_5 \oplus b_5$	0	$\alpha_4 \oplus b_4$

Πίνακας 2. Μερικά γινόμενα για άθροισμα τετραγώνων modulo $2^n - 1$.

Όπως είδαμε παραπάνω, ο πίνακας του αθροίσματος τετραγώνων έχει δύο παραπάνω γραμμές σε σχέση με τον πίνακα του πολλαπλασιασμού. Προκειμένου να γίνει δυνατή η υλοποίηση και των δύο λειτουργιών από το ίδιο κύκλωμα, ο πίνακας του πολλαπλασιασμού ξαναγράφεται ως εξής για μια $MMSSU_{-1}$.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_7 b_0$	$\alpha_6 b_0$	$\alpha_5 b_0$	$\alpha_4 b_0$	$\alpha_3 b_0$	$\alpha_2 b_0$	$\alpha_1 b_0$	$\alpha_0 b_0$
PP_1 =	$\alpha_6 b_1$	$\alpha_5 b_1$	$\alpha_4 b_1$	$\alpha_3 b_1$	$\alpha_2 b_1$	$\alpha_1 b_1$	$\alpha_0 b_1$	$\alpha_7 b_1$
PP_2 =	$\alpha_5 b_2$	$\alpha_4 b_2$	$\alpha_3 b_2$	$\alpha_2 b_2$	$\alpha_1 b_2$	$\alpha_0 b_2$	$\alpha_7 b_2$	$\alpha_6 b_2$
PP_3 =	$\alpha_4 b_3$	$\alpha_3 b_3$	$\alpha_2 b_3$	$\alpha_1 b_3$	$\alpha_0 b_3$	$\alpha_7 b_3$	$\alpha_6 b_3$	$\alpha_5 b_3$
PP_4 =	$\alpha_3 b_4$	$\alpha_2 b_4$	$\alpha_1 b_4$	$\alpha_0 b_4$	$\alpha_7 b_4$	$\alpha_6 b_4$	$\alpha_5 b_4$	$\alpha_4 b_4$
PP_5 =	$\alpha_2 b_5$	$\alpha_1 b_5$	$\alpha_0 b_5$	$\alpha_7 b_5$	$\alpha_6 b_5$	$\alpha_5 b_5$	$\alpha_4 b_5$	$\alpha_3 b_5$
PP_6 =	$\alpha_1 b_6$	$\alpha_0 b_6$	$\alpha_7 b_6$	$\alpha_6 b_6$	$\alpha_5 b_6$	$\alpha_4 b_6$	$\alpha_3 b_6$	$\alpha_2 b_6$
PP_7 =	$\alpha_0 b_7$	$\alpha_7 b_7$	$\alpha_6 b_7$	$\alpha_5 b_7$	$\alpha_4 b_7$	$\alpha_3 b_7$	$\alpha_2 b_7$	$\alpha_1 b_7$
PP_8 =	0	0	0	0	0	0	0	0
PP_9 =	0	0	0	0	0	0	0	0

Πίνακας 3. Μερικά γινόμενα για πολλαπλασιασμό modulo $2^n - 1$, υλοποιημένο από μια $MMSSU_{-1}$.

Από την παραπάνω ανάλυση προκύπτει ότι ο modulo $2^n - 1$ πολλαπλασιαστής και η modulo $2^n - 1$ μονάδα αθροίσματος τετραγώνων έχουν παρόμοια δομή. Αρχικά έχουμε ένα υποκύκλωμα παραγωγής μερικών γινομένων, έπειτα μια μονάδα modulo μείωσης των μερικών γινομένων σε δύο προσθετέους και τέλος έναν modulo $2^n - 1$ παράλληλο αθροιστή. Βασιζόμενοι σε αυτά τα αποτελέσματα, παρακάτω προτείνουμε δύο αρχιτεκτονικές για την ενοποίηση των παραπάνω σχεδιασμών σε ένα μόνο κύκλωμα. Η πρώτη αρχιτεκτονική που παρουσιάζεται καλείται απλή αρχιτεκτονική MMSSU$_{-1}$, λόγω του απλού τρόπου υλοποίησής της. Η δεύτερη αρχιτεκτονική στοχεύει στη μείωση του συνολικού χώρου που απαιτείται για την υλοποίησή της και καλείται αρχιτεκτονική μειωμένου χώρου MMSSU$_{-1}$.

2.1 Παραδείγματα modulo $2^n - 1$ πολλαπλασιασμού και αθροίσματος τετραγώνων

Πριν προχωρήσουμε παρακάτω, είναι χρήσιμο να εξετάσουμε από κοντά πώς ακριβώς πραγματοποιούνται ο πολλαπλασιασμός και το άθροισμα τετραγώνων σε αριθμητική modulo $2^n - 1$. Παρακάτω φαίνεται πώς εκτελείται ο πολλαπλασιασμός από μια MMSSU$_{-1}$. Έστω ότι $n = 8$. Οι αριθμοί που πολλαπλασιάζονται είναι οι 155 (10011011) και 109 (01101101). Το αποτέλεσμα της πράξης 155*109 modulo 255 είναι 65 (01000001).

```
      1 0 0 1 1 0 1 1 (155)
*     0 1 1 0 1 1 0 1 (109)
-------------------------------------
      1 0 0 1 1 0 1 1                  A
      0 0 0 0 0 0 0 0
      0 1 1 0 1 1 1 0
      1 1 0 1 1 1 0 0
      0 0 0 0 0 0 0 0
      0 1 1 1 0 0 1 1
      1 1 1 0 0 1 1 0
```

	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 0 0	
s	1 1 1 1 0 1 0 1	B
c	0 0 0 0 1 0 1 0 0	
s	1 0 1 0 1 1 1 1	
c	0 1 0 1 0 0 0 0 0	
s	1 1 1 0 0 1 1 0	
c	0 0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 0 0	
s	0 1 0 0 1 1 1 0	C
c	1 0 1 1 0 1 0 1 1	
s	0 1 0 0 0 1 1 0	
c	1 0 1 0 0 0 0 0 1	
	0 0 0 0 0 0 0 0	
s	0 1 1 0 0 0 1 1	D
c	0 1 0 0 1 1 1 0 0	
	0 1 0 0 0 0 0 1	
	0 0 0 0 0 0 0 0	
s	1 0 1 1 1 1 1 0	E
c	0 1 0 0 0 0 0 1 0	
	0 0 0 0 0 0 0 0	
s	0 0 1 1 1 1 0 0	F
c	1 0 0 0 0 0 1 0 1	
	0 0 1 0 0 0 0 0 1	G
	0	
	0 1 0 0 0 0 0 1 (65)	H

Κάθε τρεις γραμμές «συμπιέζονται» σε δύο χρησιμοποιώντας πλήρεις αθροιστές. Οι τρεις αυτές γραμμές μπορεί να είναι μερικά γινόμενα, ενδιάμεσα αθροίσματα ή κρατούμενα. Μια γραμμή αθροισμάτων σημειώνεται με s, ενώ μια γραμμή κρατουμένων σημειώνεται με c. Κάθε κρατούμενο της πιο σημαντικής βαθμίδας σε κάθε γραμμή υπογραμμίζεται και μεταφέρεται στην λιγότερο σημαντική

βαθμίδα της γραμμής. Επίσης υπογραμμίζεται το bit που μετακινήθηκε. Το A είναι το βήμα που απαιτείται για την παραγωγή του πίνακα των μερικών γινομένων. Οι δέκα γραμμές που φαίνονται αντιστοιχούν στις γραμμές του πίνακα 3. Τα B, C, D, E και F είναι τα βήματα της συμπίεσης που πραγματοποιείται από το δέντρο Wallace. Χρησιμοποιώντας πλήρεις αθροιστές, οι γραμμές συμπιέζονται από δέκα που ήταν αρχικά σε δύο. Το κρίσιμο μονοπάτι κάθε βήματος αυτής της φάσης παρουσιάζει καθυστέρηση ίση με αυτή ενός πλήρους αθροιστή. Επομένως, η συνολική καθυστέρηση αυτής της φάσης είναι η καθυστέρηση πέντε πλήρων αθροιστών. Τα δύο τελευταία βήματα, G και H, αναπαριστούν έναν γρήγορο modulo 255 αθροιστή. Ειδικότερα, το G αναπαριστά έναν γρήγορο αθροιστή των 8 bits, ενώ το H αναπαριστά την διόρθωση κρατουμένου που απαιτείται για άθροιση modulo 255.

Τώρα θα εξετάσουμε πώς υπολογίζεται το άθροισμα τετραγώνων modulo $2^n - 1$. Θεωρούμε πάλι τους ίδιους αριθμούς, δηλαδή $A = 155$ (10011011) και $B = 109$ (01101101). Ισχύει $A^2 + B^2$ modulo 255 = 206 (11001110). Η διαδικασία για την εύρεση του αποτελέσματος προχωράει όπως στο παρακάτω παράδειγμα.

	1 0 0 1 1 0 1 1 (155)	
	0 1 1 0 1 1 0 1 (109)	
	0 0 1 1 0 1 1 1	A
	0 1 1 0 0 0 1 0	
	0 0 0 0 1 0 0 0	
	1 1 0 1 1 0 0 1	
	0 0 0 1 0 0 0 0	
	0 1 0 0 0 0 1 1	
	1 0 0 1 0 1 1 0	
	0 0 0 0 0 0 0 0	
	0 0 0 1 0 1 0 0	
	0 1 0 1 0 1 0 1	
s	0 1 0 1 1 1 0 1	B
c	<u>0</u> 0 1 0 0 0 1 0 <u>0</u>	
s	1 0 0 0 1 0 1 0	
c	<u>0</u> 1 0 1 0 0 0 1 <u>0</u>	
s	1 0 0 0 0 0 1 0	

c	0 0 0 1 0 1 0 0 0 0 1 0 1 0 1 0 1	
s	1 0 0 1 0 0 1 1	C
c	0 1 0 0 1 1 0 0 0	
s	0 0 0 0 1 0 0 0	
c	1 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 1	
s	0 0 0 0 0 0 1 1	D
c	1 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 1	
s	0 1 1 1 0 1 1 1	E
c	0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 1	
s	0 0 1 0 0 0 0 0	F
c	0 1 0 1 0 1 1 1 0	
	0 1 1 0 0 1 1 1 0 0	G
	1 1 0 0 1 1 1 0 (206)	H

Η μόνη διαφορά της πράξης του αθροίσματος τετραγώνων με την πράξη του πολλαπλασιασμού είναι ο τρόπος με τον οποίο σχηματίζονται τα αρχικά μερικά γινόμενα. Από εκεί και πέρα ακολουθείται ακριβώς η ίδια λογική προκειμένου να φτάσουμε στο τελικό αποτέλεσμα. Όπως και στο παράδειγμα του πολλαπλασιασμού, το A είναι το βήμα που απαιτείται για την παραγωγή του πίνακα των μερικών γινομένων. Οι δέκα γραμμές του βήματος A αντιστοιχούν στα μερικά γινόμενα του πίνακα 2. Τα B, C, D, E και F είναι τα βήματα της συμπίεσης που πραγματοποιείται από το δέντρο Wallace. Χρησιμοποιώντας πλήρεις αθροιστές, οι γραμμές συμπιέζονται από δέκα που ήταν αρχικά σε δύο. Το κρίσιμο μονοπάτι κάθε βήματος αυτής της φάσης παρουσιάζει καθυστέρηση ίση με αυτή ενός πλήρους αθροιστή. Επομένως, η συνολική καθυστέρηση αυτής της φάσης είναι η καθυστέρηση πέντε πλήρων αθροιστών. Τα δύο τελευταία βήματα, G και H, αναπαριστούν έναν γρήγορο modulo 255 αθροιστή. Ειδικότερα, το G αναπαριστά έναν γρήγορο αθροιστή των 8

bits, ενώ το H αναπαριστά την διόρθωση κρατουμένου που απαιτείται για άθροιση modulo 255.

2.2 Απλή αρχιτεκτονική MMSSU-1

Μία απλή λύση για την ενοποίηση των δύο πράξεων σε ένα μόνο κύκλωμα είναι η χρησιμοποίηση ενός πολυπλέκτη για κάθε bit που διαφέρει στους πίνακες των $A \times B$ και $A^2 + B^2$. Μια τέτοια αρχιτεκτονική απαιτεί στην χειρότερη περίπτωση:

- n πύλες XOR για τον σχηματισμό των bits της μορφής $a_i \oplus b_i$.
- $2\binom{n}{2}$ πύλες AND για την παραγωγή των bits της μορφής $\alpha_i\alpha_j$ ή b_ib_j.
- n^2 πύλες AND για την παραγωγή των bits της μορφής α_ib_j.
- n^2 πολυπλέκτες για την επιλογή των σωστών μερικών γινομένων ανάλογα με την επιθυμητή πράξη. Οι πολυπλέκτες αυτοί χρησιμοποιούνται για τον σχηματισμό των n πρώτων γραμμών.
- n πύλες AND για την παραγωγή των όρων $s(a_i \oplus b_i)$.
- Μία βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου με end-around-carry για την μείωση των $n+2$ μερικών γινομένων σε δύο τελικούς προσθετέους. Έστω ότι $k = n+2$. Τότε ο αριθμός των σταδίων του δέντρου αυτού είναι μια συνάρτηση του αριθμού των μερικών γινομένων, D(k). Το D(k) φαίνεται στον πίνακα 4 για όλες τις πρακτικές τιμές που μπορεί να πάρει το n. Ο αριθμός των πλήρων αθροιστών που απαιτούνται για το δέντρο αυτό είναι $n \times n = n^2$.
- Έναν modulo $2^n - 1$ παράλληλο αθροιστή.

Προφανώς το κρίσιμο μονοπάτι αυτής της αρχιτεκτονικής αρχίζει στην είσοδο μιας πύλης XOR και περνάει μέσα από μια πύλη AND, το δέντρο των αθροιστών και τελικά μέσα από τον παράλληλο αθροιστή.

k	D(k)
4	2
$5 \le k \le 6$	3
$7 \le k \le 9$	4
$10 \le k \le 13$	5
$14 \le k \le 19$	6
$20 \le k \le 28$	7
$29 \le k \le 42$	8
$43 \le k \le 63$	9
$64 \le k \le 94$	10

Πίνακας 4. Βήματα που απαιτούνται για ένα δέντρο με k εισόδους.

Ένα συχνά χρησιμοποιούμενο μοντέλο για τη σύγκριση διαφόρων αρχιτεκτονικών είναι το μοντέλο unit-gate ([32]). Σύμφωνα με το μοντέλο αυτό, όλες οι μονοτονικές πύλες δύο εισόδων ισοδυναμούν με μία πύλη τόσο από άποψη χώρου όσο και καθυστέρησης. Επιπλέον, μια πύλη XOR ή XNOR δύο εισόδων, καθώς και ένας δύο-σε-ένα πολυπλέκτης, ισοδυναμεί με δύο πύλες τόσο από άποψη χώρου όσο και καθυστέρησης. Χρησιμοποιώντας το παραπάνω μοντέλο, μπορούμε να υπολογίσουμε τον χώρο $A_{simple,-1}$ και την καθυστέρηση $T_{simple,-1}$ της απλής αρχιτεκτονικής $MMSSU_{-1}$. Υποθέτουμε ότι για τον τελικό παράλληλο αθροιστή χρησιμοποιείται η υλοποίηση που προτείνεται στο [8]. Τότε έχουμε

$$A_{simple,-1} = 11n^2 + 3n\log n + 6n$$

$$T_{simple,-1} = 4D(n+2) + 2\log n + 6$$

2.3 Αρχιτεκτονική μειωμένου χώρου $MMSSU_{-1}$

Η απλή αρχιτεκτονική $MMSSU_{-1}$ χρησιμοποιεί πολλούς πολυπλέκτες προκειμένου να επιλέξει τα σωστά μερικά γινόμενα ανάλογα με την επιθυμητή πράξη. Η αρχιτεκτονική μειωμένου χώρου στοχεύει στη μείωση αυτών ακριβώς των πολυπλεκτών, εκμεταλλευόμενη τις ομοιότητες που υπάρχουν στα μερικά γινόμενα των δύο πράξεων. Έστω ότι το s είναι το σήμα επιλογής της επιθυμητής πράξης. Όταν

$s=0$ επιλέγεται η πράξη του modulo 2^n-1 πολλαπλασιασμού, ενώ όταν $s=1$ επιλέγεται η πράξη του modulo 2^n-1 αθροίσματος τετραγώνων.

Βασιζόμενοι στη μεταβλητή s, ορίζουμε τις παρακάτω μεταβλητές για $0 \le i \le n-1$:

$$c_i = a_i s + b_i \bar{s}$$

$$d_i = a_i \bar{s} + b_i s$$

$$e_i = s(a_i \oplus b_i)$$

Στις παραπάνω σχέσεις το σύμβολο '+' υποδηλώνει την λογική πράξη OR. Στη συνέχεια ξαναγράφουμε τους πίνακες πολλαπλασιασμού και αθροίσματος τετραγώνων χρησιμοποιώντας τις παραπάνω μεταβλητές. Ο μετασχηματισμός αυτός των πινάκων γίνεται με πολύ απλό τρόπο. Κάθε bit της μορφής $\alpha_i b_j$ αντικαθίσταται με $\alpha_i c_j$ όταν $i > j$, ενώ όταν $i < j$ αντικαθίσταται με $d_i b_j$. Τα bits της μορφής $\alpha_i b_i$ παραμένουν ως έχουν. Στην περίπτωση του πίνακα αθροίσματος τετραγώνων, τα bits της μορφής $\alpha_i \alpha_j$ αντικαθίστανται με $\alpha_i c_j$, ενώ τα bits της μορφής $b_i b_j$ αντικαθίστανται με $d_i b_j$. Ακόμα, τα bits της μορφής $a_i \oplus b_i$ αντικαθίστανται με e_i. Και πάλι τα bits της μορφής $\alpha_i b_i$ παραμένουν ως έχουν. Για $n=8$, οι πίνακες πολλαπλασιασμού και αθροίσματος τετραγώνων μετασχηματίζονται ως εξής.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_7 c_0$	$\alpha_6 c_0$	$\alpha_5 c_0$	$\alpha_4 c_0$	$\alpha_3 c_0$	$\alpha_2 c_0$	$\alpha_1 c_0$	$\alpha_0 b_0$
PP_1 =	$\alpha_6 c_1$	$\alpha_5 c_1$	$\alpha_4 c_1$	$\alpha_3 c_1$	$\alpha_2 c_1$	$\alpha_1 b_1$	$d_0 b_1$	$\alpha_7 c_1$
PP_2 =	$\alpha_5 c_2$	$\alpha_4 c_2$	$\alpha_3 c_2$	$\alpha_2 b_2$	$d_1 b_2$	$d_0 b_2$	$\alpha_7 c_2$	$\alpha_6 c_2$
PP_3 =	$\alpha_4 c_3$	$\alpha_3 b_3$	$d_2 b_3$	$d_1 b_3$	$d_0 b_3$	$\alpha_7 c_3$	$\alpha_6 c_3$	$\alpha_5 c_3$
PP_4 =	$d_3 b_4$	$d_2 b_4$	$d_1 b_4$	$d_0 b_4$	$\alpha_7 c_4$	$\alpha_6 c_4$	$\alpha_5 c_4$	$\alpha_4 b_4$
PP_5 =	$d_2 b_5$	$d_1 b_5$	$d_0 b_5$	$\alpha_7 c_5$	$\alpha_6 c_5$	$\alpha_5 b_5$	$d_4 b_5$	$d_3 b_5$
PP_6 =	$d_1 b_6$	$d_0 b_6$	$\alpha_7 c_6$	$\alpha_6 b_6$	$d_5 b_6$	$d_4 b_6$	$d_3 b_6$	$d_2 b_6$
PP_7 =	$d_0 b_7$	$\alpha_7 b_7$	$d_6 b_7$	$d_5 b_7$	$d_4 b_7$	$d_3 b_7$	$d_2 b_7$	$d_1 b_7$
PP_8 =	e_0	0	e_3	0	e_2	0	e_1	0
PP_9 =	e_4	0	e_7	0	e_6	0	e_5	0

Πίνακας 5. Νέα μορφή των μερικών γινομένων του πολλαπλασιασμού.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	α_6c_0	α_5c_0	α_4c_0	α_3c_0	α_2c_0	α_1c_0	α_0b_0	α_7c_0
PP_1 =	α_5c_1	α_4c_1	α_3c_1	α_2c_1	α_1b_1	d_0b_1	α_7c_1	α_6c_1
PP_2 =	α_4c_2	α_3c_2	α_2b_2	d_1b_2	d_0b_2	α_7c_2	α_6c_2	α_5c_2
PP_3 =	α_3b_3	d_2b_3	d_1b_3	d_0b_3	α_7c_3	α_6c_3	α_5c_3	α_4c_3
PP_4 =	d_2b_4	d_1b_4	d_0b_4	α_7c_4	α_6c_4	α_5c_4	α_4b_4	d_3b_4
PP_5 =	d_1b_5	d_0b_5	α_7c_5	α_6c_5	α_5b_5	d_4b_5	d_3b_5	d_2b_5
PP_6 =	d_0b_6	α_7c_6	α_6b_6	d_5b_6	d_4b_6	d_3b_6	d_2b_6	d_1b_6
PP_7 =	α_7b_7	d_6b_7	d_5b_7	d_4b_7	d_3b_7	d_2b_7	d_1b_7	d_0b_7
PP_8 =	0	e_3	0	e_2	0	e_1	0	e_0
PP_9 =	0	e_7	0	e_6	0	e_5	0	e_4

Πίνακας 6. Νέα μορφή των μερικών γινομένων του αθροίσματος τετραγώνων.

Εύκολα διαπιστώνει κανείς ότι ο πίνακας του πολλαπλασιασμού μπορεί να γίνει ίδιος με αυτόν του αθροίσματος τετραγώνων, αρκεί η πρώτη του στήλη να μεταφερθεί τελευταία. Χρησιμοποιούμε λοιπόν σαν βάση για την $MMSSU_{-1}$ τον πίνακα του αθροίσματος τετραγώνων. Η ίδια αρχιτεκτονική δέντρου για τη μείωση των μερικών γινομένων μπορεί να χρησιμοποιηθεί και πάλι, όπως και ο τελικός παράλληλος αθροιστής. Στην περίπτωση του πολλαπλασιασμού όμως, όταν δηλαδή $s = 0$, το πιο σημαντικό bit του αποτελέσματος βρίσκεται στην λιγότερο σημαντική θέση. Για τον λόγο αυτό, μετά από τον παράλληλο αθροιστή, χρησιμοποιούνται n πολυπλέκτες προκειμένου να πραγματοποιήσουν μια δεξιά κυκλική ολίσθηση του αποτελέσματος όταν $s = 0$.

Η παραπάνω αρχιτεκτονική απαιτεί λοιπόν:

- n πύλες XOR για τον σχηματισμό των bits της μορφής $a_i \oplus b_i$.
- n πύλες AND για τον σχηματισμό των όρων e_i.
- 2n δύο-σε-ένα πολυπλέκτες για τον σχηματισμό των όρων c_i και d_i.
- $2\binom{n}{2}$ πύλες AND για την παραγωγή των bits της μορφής $\alpha_i c_j$ ή $d_i b_j$.
- n πύλες AND για την παραγωγή των όρων $\alpha_i b_i$.
- Μία βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου για τη μείωση των μερικών γινομένων. Το βάθος του δέντρου αυτού είναι $D(n+2)$, ενώ ο αριθμός των πλήρων αθροιστών που απαιτούνται είναι n^2.

- Έναν modulo $2^n - 1$ παράλληλο αθροιστή.
- n δύο-σε-ένα πολυπλέκτες προκειμένου να πραγματοποιήσουν μια δεξιά κυκλική ολίσθηση του αποτελέσματος όταν $s = 0$.

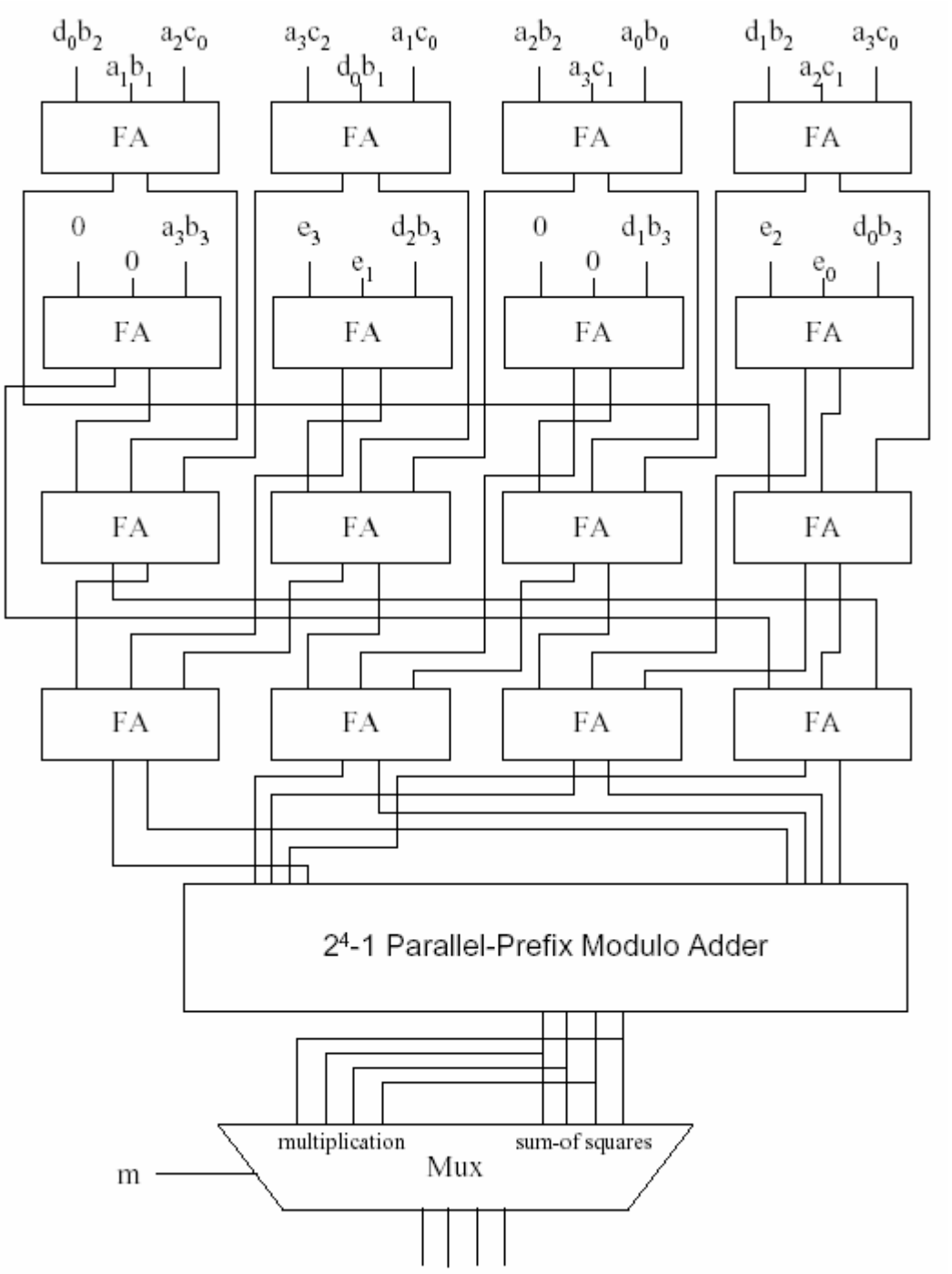
Το κρίσιμο μονοπάτι της παραπάνω αρχιτεκτονικής αρχίζει στην είσοδο μιας πύλης XOR και περνάει μέσα από μια πύλη AND για την παραγωγή ενός όρου e_i, μέσα από το δέντρο των πλήρων αθροιστών, τον παράλληλο αθροιστή και τελικά από έναν πολυπλέκτη ο οποίος χρησιμοποιείται για την δεξιά κυκλική ολίσθηση.

Χρησιμοποιώντας και πάλι το unit-gate μοντέλο, μπορούμε να υπολογίσουμε τον χώρο $A_{reduced,-1}$ και την καθυστέρηση $T_{reduced,-1}$ της αρχιτεκτονικής μειωμένου χώρου $MMSSU_{-1}$. Υποθέτουμε ότι για τον τελικό παράλληλο αθροιστή χρησιμοποιείται η υλοποίηση που προτείνεται στο [8]. Τότε έχουμε

$$A_{reduced,-1} = 8n^2 + 3n\log n + 13n$$

$$T_{reduced,-1} = 4D(n+2) + 2\log n + 8$$

Το σχήμα 1 παρακάτω απεικονίζει την αρχιτεκτονική μειωμένου χώρου $MMSSU_{-1}$ για $n = 4$. Οι παραγωγή των όρων c_i, d_i και e_i δεν φαίνεται στο σχήμα για λόγους απλότητας. Έπειτα από την παραγωγή των μερικών γινομένων, χρησιμοποιείται μια βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου για τη μείωσή τους σε δύο τελικούς προσθετέους. Τον παράλληλο modulo αθροιστή ακολουθούν οι πολυπλέκτες που είναι απαραίτητοι για την δεξιά κυκλική ολίσθηση του αποτελέσματος για την περίπτωση που ισχύει $s = 0$.



Σχήμα 1. 4-bit αρχιτεκτονική μειωμένου χώρου MMSSU$_{-1}$.

2.4 Ποιοτικές συγκρίσεις

Θεωρούμε δύο βασικές αρχιτεκτονικές οι οποίες μπορούν να πραγματοποιήσουν modulo πολλαπλασιασμό και άθροισμα τετραγώνων, τις οποίες

καλούμε σύστημα Α και σύστημα Β. Το σύστημα Α διαθέτει μόνο έναν πολλαπλασιαστή και έναν modulo αθροιστή. Επομένως, ο πολλαπλασιασμός στο σύστημα αυτό πραγματοποιείται σε έναν κύκλο, ενώ το άθροισμα τετραγώνων απαιτεί τρεις συνεχόμενους κύκλους: δύο κύκλους πολλαπλασιασμού και έναν κύκλο πρόσθεσης. Το σύστημα Β διαθέτει έναν πολλαπλασιαστή, έναν τετραγωνιστή και έναν modulo αθροιστή. Ο πολλαπλασιασμός στο σύστημα αυτό πραγματοποιείται σε έναν κύκλο, ενώ το άθροισμα τετραγώνων απαιτεί τρεις συνεχόμενους κύκλους: δύο κύκλους τετραγωνισμού και έναν κύκλο πρόσθεσης. Προφανώς το σύστημα Β υπολογίζει το άθροισμα τετραγώνων γρηγορότερα από το σύστημα Α, διότι ένας τετραγωνιστής είναι πολύ γρηγορότερος από έναν πολλαπλασιαστή που χρησιμοποιείται για τον υπολογισμό του τετραγώνου ενός αριθμού.

Παρακάτω θα χρησιμοποιήσουμε και πάλι το μοντέλο unit-gate προκειμένου να πραγματοποιήσουμε συγκρίσεις ανάμεσα στις προτεινόμενες αρχιτεκτονικές MMSSU_{-1} και τα συστήματα Α και Β. Θεωρούμε ότι τα συστήματα Α και Β χρησιμοποιούν τον πολλαπλασιαστή που προτείνεται στο [33] και τον αθροιστή που προτείνεται στο [8]. Μπορούμε τότε να υπολογίσουμε τον χώρο και την καθυστέρηση των αρχιτεκτονικών αυτών όταν πραγματοποιούν την πράξη του αθροίσματος τετραγώνων.

$$A_{multiplier,-1} = 8n^2 + 3n\log n - 10n$$

$$T_{multiplier,-1} = 4D(n) + 2\log n + 4$$

$$A_{adder,-1} = 3n\log n + 4n$$

$$T_{adder,-1} = 2\log n + 3$$

Σύμφωνα με το [14], ένας modulo $2^n - 1$ τετραγωνιστής έχει τα μισά μερικά γινόμενα από ότι ο αντίστοιχος modulo $2^n - 1$ πολλαπλασιαστής. Μπορούμε επομένως να υπολογίσουμε τον χώρο και την καθυστέρηση του τετραγωνιστή ως εξής.

$$A_{squarer,-1} = 7n\left\lceil \frac{n}{2} \right\rceil + 3n\log n + \frac{n^2 - 21n}{2}$$

$$T_{squarer,-1} = 4D\left(\left\lceil \frac{n}{2} \right\rceil\right) + 2\log n + 4$$

Χρησιμοποιώντας τις παραπάνω προσεγγίσεις, υπολογίζουμε τον χώρο και την καθυστέρηση των βασικών συστημάτων Α και Β ως εξής:

$$A_{A,-1} = 8n^2 + 6n\log n - 6n$$

$$T_{A,-1} = 8D(n) + 6\log n + 11$$

$$A_{B,-1} = 8n^2 + 7n\left\lceil \frac{n}{2} \right\rceil + 9n\log n + \frac{n^2 - 33n}{2}$$

$$T_{B,-1} = 8D\left(\left\lceil \frac{n}{2} \right\rceil\right) + 6\log n + 11$$

Τώρα είμαστε έτοιμοι να συγκρίνουμε τα βασικά συστήματα A και B με συστήματα τα οποία χρησιμοποιούν κάποια από τις προτεινόμενες αρχιτεκτονικές MMSSU_{-1}. Ο παρακάτω πίνακας παρουσιάζει για διάφορες τιμές του n τις προσεγγίσεις για τον χώρο και την καθυστέρηση των συγκρινόμενων συστημάτων.

n	$A_{A,-1}$	$A_{B,-1}$	$A_{simple,-1}$	$A_{reduced,-1}$	$T_{A,-1}$	$T_{B,-1}$	$T_{simple,-1}$	$T_{reduced,-1}$
4	152	198	224	204	39	23	22	24
8	608	852	824	688	61	45	32	34
12	1368	1917	1785	1437	72	56	37	39
16	2336	3384	3104	2448	83	67	38	40
20	3598	5247	4779	3719	92	76	42	44
24	5124	7506	6810	5250	94	78	43	45
28	6911	10157	9195	7039	95	87	47	49
32	8960	13200	11936	9088	105	89	48	50

Πίνακας 7. Ποιοτικές συγκρίσεις για συστήματα modulo $2^n - 1$.

Όπως ήταν αναμενόμενο, οι τιμές του παραπάνω πίνακα δείχνουν ότι ο χώρος που απαιτούν οι προτεινόμενες αρχιτεκτονικές MMSSU_{-1} βρίσκεται μεταξύ του χώρου του συστήματος A και του χώρου του συστήματος B. Η προτεινόμενη αρχιτεκτονική μειωμένου χώρου απαιτεί μόνο 6% περισσότερο χώρο σε σχέση με το βασικό σύστημα A όταν $n \geq 16$. Από την άλλη πλευρά, ο χώρος του βασικού συστήματος B είναι αρκετά μεγαλύτερος από τον χώρο οποιασδήποτε από τις προτεινόμενες αρχιτεκτονικές. Για $n \geq 12$, ο χώρος του βασικού συστήματος B είναι 40% μεγαλύτερος από τον χώρο που απαιτεί η προτεινόμενη αρχιτεκτονική μειωμένου χώρου. Λόγω των υψηλών δυνατοτήτων ολοκλήρωσης που προσφέρουν οι σύγχρονες κατασκευαστικές τεχνολογίες, έχει δημιουργηθεί εσφαλμένα η εντύπωση ότι το υλικό

είναι φτηνό. Δεδομένου όμως ότι μια 20% αύξηση του χώρου του ολοκληρωμένου οδηγεί σε 50% αύξηση του συνολικού κόστους ([34]), η μείωση του χρησιμοποιούμενου υλικού παραμένει ένας πολύ σημαντικός στόχος.

Και οι δύο προτεινόμενες αρχιτεκτονικές έχουν πολύ μικρότερες καθυστερήσεις για το άθροισμα τετραγώνων σε σχέση με οποιοδήποτε από τα βασικά συστήματα για $n \geq 8$. Στην περίπτωση που χρησιμοποιείται η απλή αρχιτεκτονική και $n = 32$, επιτυγχάνεται 45% μείωση της συνολικής καθυστέρησης. Προκειμένου να εξαλειφθεί αυτή η διαφορά, θα μπορούσαμε βέβαια να προσθέσουμε μια ξεχωριστή μονάδα αθροίσματος τετραγώνων σε οποιοδήποτε από τα βασικά συστήματα Α και Β. Μια τέτοια κίνηση θα οδηγούσε όμως σε περαιτέρω αύξηση του απαιτούμενου χώρου. Παρόλα αυτά, οι προτεινόμενες αρχιτεκτονικές $MMSSU_{-1}$ αναπόφευκτα έχουν μεγαλύτερες καθυστερήσεις από τα βασικά συστήματα όταν πραγματοποιείται πολλαπλασιασμός. Για παράδειγμα, παρατηρούμε μια μέση αύξηση 11% στον χρόνο που απαιτεί ο πολλαπλασιασμός όταν $n \geq 16$.

Φυσικά όλα τα αποτελέσματα που παρουσιάστηκαν παραπάνω βασίζονται σε ένα απλό μοντέλο το οποίο δεν λαμβάνει υπόψη διάφορες σημαντικές παραμέτρους, όπως π.χ. τη δρομολόγηση και την ικανότητα οδήγησης. Για τον λόγο αυτό, στο τμήμα 4 παρουσιάζονται αποτελέσματα που βασίζονται σε πλήρεις στατικές υλοποιήσεις CMOS.

3. Προτεινόμενες αρχιτεκτονικές MMSSU$_{+1}$

Ο πολλαπλασιασμός και το άθροισμα τετραγώνων modulo 2^n+1 παρουσιάζει πολλές ομοιότητες με τις αντίστοιχες πράξεις modulo 2^n-1. Για όλες τις πράξεις modulo 2^n+1 θεωρούμε ότι χρησιμοποιείται η ελαττωμένη κατά ένα αναπαράσταση για όλους τους αριθμούς. Με άλλα λόγια, τόσο οι είσοδοι A και B, όσο και το τελικό αποτέλεσμα ακολουθούν την ελαττωμένη κατά ένα αναπαράσταση. Στο [21] έχει δειχθεί ότι τα μερικά γινόμενα του modulo 2^n+1 πολλαπλασιασμού δύο αριθμών μπορούν να αναπαρασταθούν με n bits. Τα bits με βάρος 2^{n+j}, όπου $0 \le j \le n-1$, συμπληρώνονται και τοποθετούνται στη θέση $|n+j|_n$. Για κάθε τέτοια συμπλήρωση και ολίσθηση όμως, πρέπει να προστεθεί ένας παράγοντας διόρθωσης ο οποίος ισούται με 2^{n+j}.

Έστω ότι το X_{-1} δηλώνει την ελαττωμένη κατά ένα αναπαράσταση του X, δηλαδή ισχύει $X_{-1} = X-1$ και $X \neq 0$. Έστω ακόμα ότι το P αναπαριστά το αποτέλεσμα του modulo 2^n+1 πολλαπλασιασμού των A και B, ενώ το SS αναπαριστά το αποτέλεσμα του modulo 2^n+1 αθροίσματος τετραγώνων, όπου $A_{-1} = a_{n-1}a_{n-2}...a_1a_0$ και $B_{-1} = b_{n-1}b_{n-2}...b_1b_0$. Τότε για την ελαττωμένη κατά ένα αναπαράσταση των αποτελεσμάτων των παραπάνω πράξεων θα ισχύει:

$$|P_{-1}|_{2^n+1} = |P-1|_{2^n+1} = |(A_{-1}+1)(B_{-1}+1)-1|_{2^n+1} = |A_{-1}B_{-1}+A_{-1}+B_{-1}|_{2^n+1} \quad (5)$$

$$|SS_{-1}|_{2^n+1} = |SS-1|_{2^n+1} = |(A_{-1}+1)^2+(B_{-1}+1)^2-1|_{2^n+1} = |A_{-1}^2+B_{-1}^2+2A_{-1}+2B_{-1}+1|_{2^n+1}. \quad (6)$$

Οι παραπάνω σχέσεις δείχνουν ότι χρειάζονται δύο επιπλέον μερικά γινόμενα σε σχέση με την περίπτωση modulo 2^n-1. Για τον πολλαπλασιασμό απαιτείται μία επιπλέον γραμμή για το A_{-1} και μία ακόμα για το B_{-1}. Για το άθροισμα τετραγώνων απαιτείται μία επιπλέον γραμμή για το $2A_{-1}$ και μία ακόμα για το $2B_{-1}$. Τέλος, απαιτείται μία ακόμα γραμμή για τον παράγοντα συνολικής διόρθωσης που πρέπει να ληφθεί υπόψη. Σημειώνεται ότι στην περίπτωση του αθροίσματος τετραγώνων ο παραπάνω παράγοντας θα πρέπει να ενσωματώνει τον όρο +1 της (6). Εφόσον στην περίπτωση modulo 2^n-1 είχαμε $n+2$ μερικά γινόμενα, για την περίπτωση modulo 2^n+1 θα πρέπει να έχουμε $n+2+2+1=n+5$ μερικά γινόμενα.

Παρόλα αυτά, στην συγκεκριμένη περίπτωση οι παράγοντες συνολικής διόρθωσης μπορούν να ενσωματωθούν μέσα στα υπόλοιπα μερικά γινόμενα. Επομένως, στην περίπτωση αυτή δεν χρειάζεται να χρησιμοποιηθεί μία επιπλέον γραμμή για τον παράγοντα συνολικής διόρθωσης. Οι προτεινόμενες αρχιτεκτονικές $MMSSU_{+1}$ χρησιμοποιούν λοιπόν $n+4$ μερικά γινόμενα.

Αρχικά ας θεωρήσουμε την περίπτωση του πολλαπλασιασμού. Τα μερικά γινόμενα που απαιτούνται όταν $n=8$ φαίνονται στον πίνακα 8. Σημειώνεται ότι το σύμβολο "~" στους επόμενους πίνακες υποδηλώνει το συμπλήρωμα. Για την αναπαράσταση καθενός από τα μερικά γινόμενα χρησιμοποιούνται n bits, με τις συμπληρώσεις και ολισθήσεις που αναφέρθηκαν προηγουμένως. Παρακάτω θα υπολογίσουμε τον παράγοντα συνολικής διόρθωσης που απαιτείται και θα δείξουμε ότι μπορεί να αναπαρασταθεί με δύο επιπλέον σειρές, οδηγώντας έτσι σε $n+4$ μερικά γινόμενα συνολικά. Έστω ότι CP είναι ο παράγοντας συνολικής διόρθωσης για την περίπτωση του πολλαπλασιασμού.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$PP_0 =$	$\alpha_7 b_0$	$\alpha_6 b_0$	$\alpha_5 b_0$	$\alpha_4 b_0$	$\alpha_3 b_0$	$\alpha_2 b_0$	$\alpha_1 b_0$	$\alpha_0 b_0$
$PP_1 =$	$\alpha_6 b_1$	$\alpha_5 b_1$	$\alpha_4 b_1$	$\alpha_3 b_1$	$\alpha_2 b_1$	$\alpha_1 b_1$	$\alpha_0 b_1$	$\sim\alpha_7 b_1$
$PP_2 =$	$\alpha_5 b_2$	$\alpha_4 b_2$	$\alpha_3 b_2$	$\alpha_2 b_2$	$\alpha_1 b_2$	$\alpha_0 b_2$	$\sim\alpha_7 b_2$	$\sim\alpha_6 b_2$
$PP_3 =$	$\alpha_4 b_3$	$\alpha_3 b_3$	$\alpha_2 b_3$	$\alpha_1 b_3$	$\alpha_0 b_3$	$\sim\alpha_7 b_3$	$\sim\alpha_6 b_3$	$\sim\alpha_5 b_3$
$PP_4 =$	$\alpha_3 b_4$	$\alpha_2 b_4$	$\alpha_1 b_4$	$\alpha_0 b_4$	$\sim\alpha_7 b_4$	$\sim\alpha_6 b_4$	$\sim\alpha_5 b_4$	$\sim\alpha_4 b_4$
$PP_5 =$	$\alpha_2 b_5$	$\alpha_1 b_5$	$\alpha_0 b_5$	$\sim\alpha_7 b_5$	$\sim\alpha_6 b_5$	$\sim\alpha_5 b_5$	$\sim\alpha_4 b_5$	$\sim\alpha_3 b_5$
$PP_6 =$	$\alpha_1 b_6$	$\alpha_0 b_6$	$\sim\alpha_7 b_6$	$\sim\alpha_6 b_6$	$\sim\alpha_5 b_6$	$\sim\alpha_4 b_6$	$\sim\alpha_3 b_6$	$\sim\alpha_2 b_6$
$PP_7 =$	$\alpha_0 b_7$	$\sim\alpha_7 b_7$	$\sim\alpha_6 b_7$	$\sim\alpha_5 b_7$	$\sim\alpha_4 b_7$	$\sim\alpha_3 b_7$	$\sim\alpha_2 b_7$	$\sim\alpha_1 b_7$
$PP_8 =$	α_7	α_6	α_5	α_4	α_3	α_2	α_1	α_0
$PP_9 =$	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

Πίνακας 8. Μερικά γινόμενα για πολλαπλασιασμό modulo $2^n + 1$.

Χρησιμοποιώντας τις διαδικασίες που παρουσιάζονται στα [21] και [22], το CP μπορεί να υπολογιστεί ως $CP = CP_1 + CP_2$, όπου CP_1 είναι η διόρθωση που απαιτείται για την παραγωγή των μερικών γινομένων και CP_2 είναι η διόρθωση που απαιτείται κατά τη μείωση των μερικών γινομένων. Στον παραπάνω πίνακα, κάθε όρος ο οποίος εμφανίζεται συμπληρωμένος προέρχεται από ολίσθηση και επομένως

απαιτεί κάποια διόρθωση. Έστω ότι ένας τέτοιος όρος βρίσκεται στη θέση με βάρος j, όπου $0 \le j \le n-1$. Τότε ο όρος αυτός απαιτεί μια διόρθωση ίση με 2^{n+j}. Παρατηρούμε ότι κάθε μερικό γινόμενο PP_j, με $1 \le j \le n-1$, απαιτεί μια διόρθωση ίση με $2^n(2^j-1)$. Τα PP_0, PP_n και PP_{n+1} δεν απαιτούν κάποια διόρθωση. Επομένως το CP_1 υπολογίζεται εύκολα ως

$$CP_1 = \sum_{j=1}^{n-1} 2^n\left(2^j - 1\right) = 2^n\left(2^n - n - 1\right).$$

Συνολικά έχουμε $n+4$ μερικά γινόμενα τα οποία μειώνονται σε δύο τελικούς προσθετέους με τη χρήση μιας modulo 2^n+1 βασιζόμενης σε πλήρεις αθροιστές αρχιτεκτονικής με end-around-carry. Έστω ότι το c_n δηλώνει το κρατούμενο της πιο σημαντικής βαθμίδας σε κάποιο βήμα i της μείωσης. Το c_n έχει βάρος 2^n. Εφόσον

$$\left|c_n 2^n\right|_{2^n+1} = \left|-c_n\right|_{2^n+1} = \left|2^n + \overline{c_n}\right|_{2^n+1},$$

το c_n συμπληρώνεται και προστίθεται στην λιγότερο σημαντική θέση του επόμενου βήματος. Για κάτι τέτοιο πρέπει να ληφθεί υπόψη μια διόρθωση ίση με 2^n. Κατά τη μείωση των $n+4$ μερικών γινομένων παράγονται $n+2$ κρατούμενα με βάρος 2^n. Προκύπτει λοιπόν ότι

$$CP_2 = 2^n(n+2).$$

Επομένως ο παράγοντας συνολικής διόρθωσης CP που απαιτείται για την περίπτωση του πολλαπλασιασμού υπολογίζεται ως εξής:

$$\left|CP\right|_{2^n+1} = \left|CP_1 + CP_2\right|_{2^n+1} = 0.$$

Αφού το CP θεωρείται σαν ένα επιπλέον μερικό γινόμενο στις προτεινόμενες αρχιτεκτονικές, κατά τη μείωση πρέπει να χρησιμοποιηθεί η ελαττωμένη κατά ένα αναπαράστασή του. Συνεπώς, η συνολική διόρθωση που απαιτείται για τον πολλαπλασιασμό είναι $CP_{-1} = 2^n$. Αφού αυτή η τιμή δεν μπορεί να αναπαρασταθεί με n bits, προστίθενται δύο ακόμα μερικά γινόμενα. Το πρώτο είναι μια γραμμή από άσσους, ενώ το δεύτερο είναι η γραμμή 00…01. Π.χ. για $n=8$ προσθέτουμε τις δύο γραμμές του επόμενου πίνακα.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_{10} =	0	0	0	0	0	0	0	1
PP_{11} =	1	1	1	1	1	1	1	1

Πίνακας 9. Τα δύο τελευταία μερικά γινόμενα για πολλαπλασιασμό modulo $2^n + 1$.

Επομένως, ο πλήρης πίνακας μερικών γινομένων για $n = 8$ έχει την παρακάτω μορφή. Σημειώνεται ότι έχει γίνει μια αναδιάταξη των γραμμών του πίνακα προκειμένου να μοιάζει περισσότερο με τον αντίστοιχο πίνακα για την περίπτωση modulo $2^n - 1$.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_7 b_0$	$\alpha_6 b_0$	$\alpha_5 b_0$	$\alpha_4 b_0$	$\alpha_3 b_0$	$\alpha_2 b_0$	$\alpha_1 b_0$	$\alpha_0 b_0$
PP_1 =	$\alpha_6 b_1$	$\alpha_5 b_1$	$\alpha_4 b_1$	$\alpha_3 b_1$	$\alpha_2 b_1$	$\alpha_1 b_1$	$\alpha_0 b_1$	$\sim\alpha_7 b_1$
PP_2 =	$\alpha_5 b_2$	$\alpha_4 b_2$	$\alpha_3 b_2$	$\alpha_2 b_2$	$\alpha_1 b_2$	$\alpha_0 b_2$	$\sim\alpha_7 b_2$	$\sim\alpha_6 b_2$
PP_3 =	$\alpha_4 b_3$	$\alpha_3 b_3$	$\alpha_2 b_3$	$\alpha_1 b_3$	$\alpha_0 b_3$	$\sim\alpha_7 b_3$	$\sim\alpha_6 b_3$	$\sim\alpha_5 b_3$
PP_4 =	$\alpha_3 b_4$	$\alpha_2 b_4$	$\alpha_1 b_4$	$\alpha_0 b_4$	$\sim\alpha_7 b_4$	$\sim\alpha_6 b_4$	$\sim\alpha_5 b_4$	$\sim\alpha_4 b_4$
PP_5 =	$\alpha_2 b_5$	$\alpha_1 b_5$	$\alpha_0 b_5$	$\sim\alpha_7 b_5$	$\sim\alpha_6 b_5$	$\sim\alpha_5 b_5$	$\sim\alpha_4 b_5$	$\sim\alpha_3 b_5$
PP_6 =	$\alpha_1 b_6$	$\alpha_0 b_6$	$\sim\alpha_7 b_6$	$\sim\alpha_6 b_6$	$\sim\alpha_5 b_6$	$\sim\alpha_4 b_6$	$\sim\alpha_3 b_6$	$\sim\alpha_2 b_6$
PP_7 =	$\alpha_0 b_7$	$\sim\alpha_7 b_7$	$\sim\alpha_6 b_7$	$\sim\alpha_5 b_7$	$\sim\alpha_4 b_7$	$\sim\alpha_3 b_7$	$\sim\alpha_2 b_7$	$\sim\alpha_1 b_7$
PP_8 =	0	0	0	0	0	0	0	1
PP_9 =	1	1	1	1	1	1	1	1
PP_{10} =	α_7	α_6	α_5	α_4	α_3	α_2	α_1	α_0
PP_{11} =	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

Πίνακας 10. Όλα τα μερικά γινόμενα για πολλαπλασιασμό modulo $2^n + 1$.

Προχωρώντας στην περίπτωση του αθροίσματος τετραγώνων, έχουμε:

$$\left|SS_{-1}\right|_{2^n+1} = \left|A_{-1}^2 + B_{-1}^2 + 2A_{-1} + 2B_{-1} + 1\right|_{2^n+1} = \left|\sum_{i=0}^{n-1}(a_i \oplus b_i)2^{2i} + \sum_{i=0}^{n-1} a_i b_i 2^{2i+1} + NPP + 1\right|_{2^n+1} \quad (7)$$

όπου

$$NPP = \left(\sum_{i=1}^{n-1}\sum_{j=0}^{i-1}(a_i a_j + b_i b_j)2^{i+j+1}\right) + 2A_{-1} + 2B_{-1}.$$

Τα μερικά γινόμενα που απαιτούνται για το NPP όταν $n = 8$ φαίνονται στον πίνακα 11. Όπως και προηγουμένως, κάθε μερικό γινόμενο αναπαρίσταται με n bits μέσω συμπληρώσεων και ολισθήσεων.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_6\alpha_0$	$\alpha_5\alpha_0$	$\alpha_4\alpha_0$	$\alpha_3\alpha_0$	$\alpha_2\alpha_0$	$\alpha_1\alpha_0$	0	$\sim\alpha_7\alpha_0$
PP_1 =	$\alpha_5\alpha_1$	$\alpha_4\alpha_1$	$\alpha_3\alpha_1$	$\alpha_2\alpha_1$	0	b_0b_1	$\sim\alpha_7\alpha_1$	$\sim\alpha_6\alpha_1$
PP_2 =	$\alpha_4\alpha_2$	$\alpha_3\alpha_2$	0	b_1b_2	b_0b_2	$\sim\alpha_7\alpha_2$	$\sim\alpha_6\alpha_2$	$\sim\alpha_5\alpha_2$
PP_3 =	0	b_2b_3	b_1b_3	b_0b_3	$\sim\alpha_7\alpha_3$	$\sim\alpha_6\alpha_3$	$\sim\alpha_5\alpha_3$	$\sim\alpha_4\alpha_3$
PP_4 =	b_2b_4	b_1b_4	b_0b_4	$\sim\alpha_7\alpha_4$	$\sim\alpha_6\alpha_4$	$\sim\alpha_5\alpha_4$	0	$\sim b_3b_4$
PP_5 =	b_1b_5	b_0b_5	$\sim\alpha_7\alpha_5$	$\sim\alpha_6\alpha_5$	0	$\sim b_4b_5$	$\sim b_3b_5$	$\sim b_2b_5$
PP_6 =	b_0b_6	$\sim\alpha_7\alpha_6$	0	$\sim b_5b_6$	$\sim b_4b_6$	$\sim b_3b_6$	$\sim b_2b_6$	$\sim b_1b_6$
PP_7 =	0	$\sim b_6b_7$	$\sim b_5b_7$	$\sim b_4b_7$	$\sim b_3b_7$	$\sim b_2b_7$	$\sim b_1b_7$	$\sim b_0b_7$
PP_8 =	α_6	α_5	α_4	α_3	α_2	α_1	α_0	$\sim\alpha_7$
PP_9 =	b_6	b_5	b_4	b_3	b_2	b_1	b_0	$\sim b_7$

Πίνακας 11. Μερικά γινόμενα για άθροισμα τετραγώνων modulo $2^n + 1$.

Οι όροι $\sum_{i=0}^{n-1}(a_i \oplus b_i)2^{2i} + \sum_{i=0}^{n-1} a_i b_i 2^{2i+1}$ μπορούν να αναπαρασταθούν με μερικά γινόμενα των n bits με δύο τρόπους. Για $n = 8$, αυτοί οι τρόποι φαίνονται στον πίνακα 12. Παρακάτω χρησιμοποιείται ο δεύτερος τρόπος, καθώς οι όροι $\alpha_i b_i$ και $\sim\alpha_i b_i$ μπορούν να αντικαταστήσουν τα μηδενικά των μερικών γινομένων PP_j, οδηγώντας έτσι σε έναν πολύ κανονικό πίνακα μερικών γινομένων με $n + 4$ γραμμές. Επίσης, όπως θα δούμε παρακάτω, ο δεύτερος τρόπος μας δίνει τη δυνατότητα να καταλήξουμε σε έναν παράγοντα συνολικής διόρθωσης ο οποίος δεν εξαρτάται από το n.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
				Πρώτος τρόπος				
PP_{10}	$\alpha_3 b_3$	$\alpha_3 \oplus b_3$	$\alpha_2 b_2$	$\alpha_2 \oplus b_2$	$\alpha_1 b_1$	$\alpha_1 \oplus b_1$	$\alpha_0 b_0$	$\alpha_0 \oplus b_0$
PP_{11}	$\sim\alpha_7 b_7$	$\sim(\alpha_7 \oplus b_7)$	$\sim\alpha_6 b_6$	$\sim(\alpha_6 \oplus b_6)$	$\sim\alpha_5 b_5$	$\sim(\alpha_5 \oplus b_5)$	$\sim\alpha_4 b_4$	$\sim(\alpha_4 \oplus b_4)$
				Δεύτερος τρόπος				
PP_{10}	$\alpha_3 b_3$	0	$\alpha_2 b_2$	0	$\alpha_1 b_1$	0	$\alpha_0 b_0$	0
PP_{11}	0	$\alpha_3 \oplus b_3$	0	$\alpha_2 \oplus b_2$	0	$\alpha_1 \oplus b_1$	0	$\alpha_0 \oplus b_0$
PP_{12}	$\sim\alpha_7 b_7$	0	$\sim\alpha_6 b_6$	0	$\sim\alpha_5 b_5$	0	$\sim\alpha_4 b_4$	0
PP_{13}	1	$\sim(\alpha_7 \oplus b_7)$	1	$\sim(\alpha_6 \oplus b_6)$	1	$\sim(\alpha_5 \oplus b_5)$	1	$\sim(\alpha_4 \oplus b_4)$

Πίνακας 12. Εναλλακτικοί τρόποι παρουσίασης των υπόλοιπων όρων.

Σύμφωνα με τα παραπάνω, ο πλήρης πίνακας μερικών γινομένων για $n = 8$ φαίνεται παρακάτω. Σημειώνεται ότι στον πίνακα αυτό έχει ενσωματωθεί και ο παράγοντας συνολικής διόρθωσης. Επίσης έχει γίνει και πάλι αναδιάταξη των γραμμών προκειμένου να μοιάζει περισσότερο με τον αντίστοιχο πίνακα για την περίπτωση modulo $2^n - 1$.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$PP_0 =$	$\alpha_6 \alpha_0$	$\alpha_5 \alpha_0$	$\alpha_4 \alpha_0$	$\alpha_3 \alpha_0$	$\alpha_2 \alpha_0$	$\alpha_1 \alpha_0$	$\alpha_0 b_0$	$\sim\alpha_7 \alpha_0$
$PP_1 =$	$\alpha_5 \alpha_1$	$\alpha_4 \alpha_1$	$\alpha_3 \alpha_1$	$\alpha_2 \alpha_1$	$\alpha_1 b_1$	$b_0 b_1$	$\sim\alpha_7 \alpha_1$	$\sim\alpha_6 \alpha_1$
$PP_2 =$	$\alpha_4 \alpha_2$	$\alpha_3 \alpha_2$	$\alpha_2 b_2$	$b_1 b_2$	$b_0 b_2$	$\sim\alpha_7 \alpha_2$	$\sim\alpha_6 \alpha_2$	$\sim\alpha_5 \alpha_2$
$PP_3 =$	$\alpha_3 b_3$	$b_2 b_3$	$b_1 b_3$	$b_0 b_3$	$\sim\alpha_7 \alpha_3$	$\sim\alpha_6 \alpha_3$	$\sim\alpha_5 \alpha_3$	$\sim\alpha_4 \alpha_3$
$PP_4 =$	$b_2 b_4$	$b_1 b_4$	$b_0 b_4$	$\sim\alpha_7 \alpha_4$	$\sim\alpha_6 \alpha_4$	$\sim\alpha_5 \alpha_4$	$\sim\alpha_4 b_4$	$\sim b_3 b_4$
$PP_5 =$	$b_1 b_5$	$b_0 b_5$	$\sim\alpha_7 \alpha_5$	$\sim\alpha_6 \alpha_5$	$\sim\alpha_5 b_5$	$\sim b_4 b_5$	$\sim b_3 b_5$	$\sim b_2 b_5$
$PP_6 =$	$b_0 b_6$	$\sim\alpha_7 \alpha_6$	$\sim\alpha_6 b_6$	$\sim b_5 b_6$	$\sim b_4 b_6$	$\sim b_3 b_6$	$\sim b_2 b_6$	$\sim b_1 b_6$
$PP_7 =$	$\sim\alpha_7 b_7$	$\sim b_6 b_7$	$\sim b_5 b_7$	$\sim b_4 b_7$	$\sim b_3 b_7$	$\sim b_2 b_7$	$\sim b_1 b_7$	$\sim b_0 b_7$
$PP_8 =$	0	$\alpha_3 \oplus b_3$	0	$\alpha_2 \oplus b_2$	0	$\alpha_1 \oplus b_1$	1	$\alpha_0 \oplus b_0$
$PP_9 =$	1	$\sim(\alpha_7 \oplus b_7)$	1	$\sim(\alpha_6 \oplus b_6)$	1	$\sim(\alpha_5 \oplus b_5)$	1	$\sim(\alpha_4 \oplus b_4)$
PP_{10}	α_6	α_5	α_4	α_3	α_2	α_1	α_0	$\sim\alpha_7$
PP_{11}	b_6	b_5	b_4	b_3	b_2	b_1	b_0	$\sim b_7$

Πίνακας 13. Όλα τα μερικά γινόμενα για άθροισμα τετραγώνων modulo $2^n + 1$.

Έστω ότι το CSS αναπαριστά τον παράγοντα συνολικής διόρθωσης που απαιτείται για το άθροισμα τετραγώνων. Παρακάτω υπολογίζεται το CSS και αποδεικνύεται ότι είναι ίσο με 3 ανεξάρτητα από το n. Ο παράγοντας συνολικής διόρθωσης ισούται με $CSS = CSS_1 + CSS_2 + CSS_3$, όπου:

- Το CSS_1 είναι ο παράγοντας διόρθωσης που απαιτείται για την παραγωγή των μερικών γινομένων. Εύκολα παρατηρεί κανείς ότι οι πρώτες n – 1 γραμμές του πίνακα 13 απαιτούν διόρθωση ίση με το CP_1 της περίπτωσης του πολλαπλασιασμού, δηλαδή $2^n(2^n - n - 1)$. Η επόμενη γραμμή, δηλαδή το $PP_{n-1,}$ απαιτεί διόρθωση ίση με $2^n(2^n - 1)$. Το PP_{n+1} επίσης απαιτεί διόρθωση ίση με $2^n(2^n - 1)$. Για το PP_n δεν απαιτείται κάποια διόρθωση, ενώ για κάθε ένα από τα PP_{n+2} και PP_{n+3} απαιτείται διόρθωση ίση με 2^n. Επομένως έχουμε
$$CSS_1 = 2^n\left(2^n - n - 1\right) + 2^n\left(2^n - 1\right) + 2^n\left(2^n - 1\right) + 2^n + 2^n = 2^n\left(3 \times 2^n - n - 1\right)$$
- Το CSS_2 είναι ο παράγοντας διόρθωσης που απαιτείται για τη μείωση των μερικών γινομένων. Εφόσον κατά την μείωση n + 4 μερικών γινομένων σε δύο τελικούς προσθετέους παράγονται $n + 2$ κρατούμενα βάρους 2^n, ισχύει:
$$CSS_2 = 2^n\left(n + 2\right)$$
- Το CSS_3 ισούται με 1 και είναι ο άσσος που πρέπει να ενσωματωθεί μέσα στον παράγοντα συνολικής διόρθωσης.

Σύμφωνα με τα παραπάνω έχουμε:

$$\left|CSS\right|_{2^n+1} = \left|CSS_1 + CSS_2 + CSS_3\right|_{2^2+1} = \left|2^n\left(3 \times 2^n - n - 1\right) + 2^n\left(n + 2\right) + 1\right|_{2^n+1} = 3.$$

Εφόσον χρησιμοποιούμε την ελαττωμένη κατά ένα αναπαράσταση του CSS στον πίνακα μερικών γινομένων, έχουμε $CSS_{-1} = 2$, το οποίο εξηγεί την ύπαρξη του 1 στο δεύτερο από δεξιά bit του PP_8 στον πίνακα 13.

Από τα παραπάνω γίνεται φανερό ότι ο modulo $2^n + 1$ πολλαπλασιαστής και η modulo $2^n + 1$ μονάδα αθροίσματος τετραγώνων έχουν παρόμοια δομή. Και τα δύο κυκλώματα αποτελούνται από ένα υποκύκλωμα παραγωγής μερικών γινομένων, μια μονάδα modulo μείωσης των μερικών γινομένων και έναν τελικό παράλληλο αθροιστή. Παρακάτω προτείνονται δύο αρχιτεκτονικές για την ενοποίηση των δύο σχεδιασμών σε ένα μόνο κύκλωμα. Η πρώτη αρχιτεκτονική που παρουσιάζεται καλείται απλή αρχιτεκτονική $MMSSU_{+1}$, λόγω του απλού τρόπου υλοποίησής της. Η

δεύτερη αρχιτεκτονική στοχεύει στη μείωση του συνολικού χώρου που απαιτείται για την υλοποίησή της και καλείται αρχιτεκτονική μειωμένου χώρου $MMSSU_{+1}$.

3.1 Παραδείγματα modulo $2^n + 1$ πολλαπλασιασμού και αθροίσματος τετραγώνων

Πριν προχωρήσουμε παρακάτω, είναι χρήσιμο να εξετάσουμε από κοντά πώς ακριβώς πραγματοποιούνται ο πολλαπλασιασμός και το άθροισμα τετραγώνων σε αριθμητική modulo 2^n+1. Αρχικά θα εξετάσουμε πώς πραγματοποιείται ο πολλαπλασιασμός από μια $MMSSU_{+1}$. Για το παρακάτω παράδειγμα θα χρησιμοποιηθούν τα ίδια αριθμητικά δεδομένα όπως και στην περίπτωση modulo 2^n-1. Έστω λοιπόν ότι $A_{-1}=155$ (10011011) και $B_{-1}=109$ (01101101). Το αποτέλεσμα της πράξης θα πρέπει να είναι (155 + 1)*(109 + 1) – 1 modulo 257, δηλαδή 197 (11000101).

	1 0 0 1 1 0 1 1 (155)	
*	0 1 1 0 1 1 0 1 (109)	
	1 0 0 1 1 0 1 1	A
	0 0 0 0 0 0 0 1	
	0 1 1 0 1 1 0 1	
	1 1 0 1 1 0 1 1	
	0 0 0 0 1 1 1 1	
	0 1 1 0 1 1 0 0	
	1 1 0 1 1 0 0 1	
	0 1 1 1 1 1 1 1	
	0 0 0 0 0 0 0 1	
	1 1 1 1 1 1 1 1	
	1 0 0 1 1 0 1 1	
	0 1 1 0 1 1 0 1	
s	1 1 1 1 0 1 1 1	B
c	0 0 0 0 1 0 0 1 1	
s	1 0 1 1 1 0 0 0	

c	0 1 0 0 1 1 1 1 1	
s	1 0 1 0 0 1 1 1	
c	0 1 0 1 1 0 0 1 1	
s	0 0 0 0 1 0 0 1	
c	1 1 1 1 1 1 1 1 0	
s	0 1 0 1 1 1 0 0	C
c	1 0 1 1 0 0 1 1 0	
s	1 0 0 0 1 0 1 1	
c	1 0 1 1 0 1 1 1 0	
	0 0 0 0 1 0 0 1	
	1 1 1 1 1 1 1 0	
s	1 0 1 1 0 0 0 1	D
c	0 1 0 0 1 1 1 0 1	
s	1 0 0 1 1 0 0 1	
c	0 1 1 0 1 1 1 0 1	
s	1 0 1 1 0 1 0 1	E
c	1 0 0 1 1 0 0 1 0	
	1 1 0 1 1 1 0 1	
s	0 1 0 1 1 0 1 0	F
c	1 0 1 1 0 1 0 1 0	
	0 1 1 0 0 0 1 0 0	G
	1	
	1 1 0 0 0 1 0 1 (197)	H

Κάθε κρατούμενο της πιο σημαντικής βαθμίδας σε κάθε γραμμή υπογραμμίζεται και μεταφέρεται αντεστραμμένο στην λιγότερο σημαντική βαθμίδα της γραμμής. Υπογραμμίζεται επίσης το bit που αντιστράφηκε και μετακινήθηκε. Το A είναι το βήμα που απαιτείται για την παραγωγή του πίνακα των μερικών γινομένων. Οι δώδεκα γραμμές που φαίνονται αντιστοιχούν σε αυτές του πίνακα 10. Τα B, C, D, E και F είναι τα βήματα της συμπίεσης που πραγματοποιείται από το δέντρο Wallace. Χρησιμοποιώντας πλήρεις αθροιστές, οι γραμμές συμπιέζονται από δώδεκα που ήταν αρχικά σε δύο. Το κρίσιμο μονοπάτι κάθε βήματος αυτής της φάσης παρουσιάζει καθυστέρηση ίση με αυτή ενός πλήρους αθροιστή και ενός αντιστροφέα.

Επομένως, η συνολική καθυστέρηση αυτής της φάσης είναι η καθυστέρηση πέντε πλήρων αθροιστών και πέντε αντιστροφέων. Τα δύο τελευταία βήματα, G και H, αναπαριστούν έναν γρήγορο modulo 257 αθροιστή. Ειδικότερα, το G αναπαριστά έναν γρήγορο αθροιστή των 8 bits, ενώ το H αναπαριστά την διόρθωση κρατουμένου που απαιτείται για άθροιση modulo 257.

Όπως και στην περίπτωση του πολλαπλασιασμού modulo $2^n - 1$, η δομή του δέντρου Wallace είναι εντελώς κανονική. Παρόλα αυτά, είναι αναγκαία η χρήση μερικών αντιστροφέων, κάτι που δεν συνέβαινε στην προηγούμενη περίπτωση.

Παρακάτω θα δούμε ένα παράδειγμα για το πώς πραγματοποιείται το άθροισμα τετραγώνων modulo $2^n + 1$. Έστω ότι $A_{-1} = 155$ (10011011) και $B_{-1} = 109$ (01101101). Το αποτέλεσμα της πράξης του αθροίσματος τετραγώνων θα πρέπει να είναι $(155 + 1)^2 + (109 + 1)^2 - 1$ modulo 257, δηλαδή 198(11000110).

	1 0 0 1 1 0 1 1 (155)	
	0 1 1 0 1 1 0 1 (109)	
	0 0 1 1 0 1 1 0	A
	0 1 1 0 0 0 0 1	
	0 0 0 0 1 1 1 1	
	1 1 0 1 0 1 1 0	
	0 0 0 0 1 1 1 1	
	0 1 1 1 1 1 0 0	
	1 1 1 0 1 0 0 1	
	1 1 1 1 1 1 1 1	
	0 0 0 1 0 1 1 0	
	1 0 1 0 1 0 1 0	
	0 0 1 1 0 1 1 0	
	1 1 0 1 1 0 1 1	
s	0 1 0 1 1 0 0 0	B
c	0 0 1 0 0 1 1 1 1	
s	1 0 1 0 0 1 0 1	
c	0 1 0 1 1 1 1 0 1	
s	0 0 0 0 0 0 0 0	
c	1 1 1 1 1 1 1 1 0	

s	0 1 0 0 0 1 1 1	
c	1 0 1 1 1 0 1 0 0	
s	1 0 1 1 0 0 1 0	C
c	0 1 0 0 1 1 0 1 1	
s	0 1 0 0 0 0 1 1	
c	1 0 1 1 1 1 0 0 0	
	0 1 0 0 0 1 1 1	
	0 1 1 1 0 1 0 0	
s	0 1 1 0 1 0 1 0	D
c	1 0 0 1 0 0 1 1 0	
s	0 1 0 0 1 0 1 1	
c	0 1 1 1 0 1 0 0 1	
s	0 0 0 0 0 1 1 1	E
c	0 1 1 0 1 0 1 0 1	
	1 1 1 0 1 0 0 1	
s	0 0 1 1 1 0 1 1	F
c	1 1 0 0 0 1 0 1 0	
	0 1 1 0 0 0 1 0 1	G
	1	
	1 1 0 0 0 1 1 0 (198)	H

Η σημειογραφία που χρησιμοποιείται είναι η ίδια με του προηγούμενου παραδείγματος. Κάθε κρατούμενο της πιο σημαντικής βαθμίδας σε κάθε γραμμή υπογραμμίζεται και μεταφέρεται αντεστραμμένο στην λιγότερο σημαντική βαθμίδα της γραμμής. Υπογραμμίζεται επίσης το bit που αντιστράφηκε και μετακινήθηκε. Το A είναι το βήμα που απαιτείται για την παραγωγή του πίνακα των μερικών γινομένων. Οι δώδεκα γραμμές που φαίνονται αντιστοιχούν σε αυτές του πίνακα 13. Τα B, C, D, E και F είναι τα βήματα της συμπίεσης που πραγματοποιείται από το δέντρο Wallace. Χρησιμοποιώντας πλήρεις αθροιστές, οι γραμμές συμπιέζονται από δώδεκα που ήταν αρχικά σε δύο. Το κρίσιμο μονοπάτι κάθε βήματος αυτής της φάσης παρουσιάζει καθυστέρηση ίση με αυτή ενός πλήρους αθροιστή και ενός αντιστροφέα. Επομένως, η συνολική καθυστέρηση αυτής της φάσης είναι η καθυστέρηση πέντε πλήρων αθροιστών και πέντε αντιστροφέων. Τα δύο τελευταία βήματα, G και H,

αναπαριστούν έναν γρήγορο modulo 257 αθροιστή. Ειδικότερα, το G αναπαριστά έναν γρήγορο αθροιστή των 8 bits, ενώ το H αναπαριστά την διόρθωση κρατουμένου που απαιτείται για άθροιση modulo 257.

3.2 Απλή αρχιτεκτονική MMSSU$_{+1}$

Η απλή αρχιτεκτονική MMSSU$_{+1}$ ενοποιεί τις πράξεις του πολλαπλασιασμού και του αθροίσματος τετραγώνων χρησιμοποιώντας έναν πολυπλέκτη για κάθε bit το οποίο διαφέρει στους πίνακες των δύο λειτουργιών. Η επιθυμητή πράξη επιλέγεται μέσω ενός σήματος s. Όταν $s=0$ επιλέγεται ο πολλαπλασιασμός, ενώ όταν $s=1$ επιλέγεται το άθροισμα τετραγώνων. Παρατηρούμε ότι για τα μερικά γινόμενα PP_n και PP_{n+1} χρειάζονται μόνο n πολυπλέκτες. Στον πίνακα 14 φαίνονται τα δύο αυτά μερικά γινόμενα και για τις δύο λειτουργίες όταν $n=8$.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
				Πολλαπλασιασμός				
PP_8	0	0	0	0	0	0	s	0
PP_9	1	1	1	1	1	1	1	1
				Άθροισμα τετραγώνων				
PP_8	0	$\alpha_3 \oplus b_3$	0	$\alpha_2 \oplus b_2$	0	$\alpha_1 \oplus b_1$	s	$\alpha_0 \oplus b_0$
PP_9	1	$\sim(\alpha_7 \oplus b_7)$	1	$\sim(\alpha_6 \oplus b_6)$	1	$\sim(\alpha_5 \oplus b_5)$	1	$\sim(\alpha_4 \oplus b_4)$

Πίνακας 14. Τα μερικά γινόμενα PP_8 και PP_9.

Η απλή αρχιτεκτονική MMSSU$_{+1}$ απαιτεί επομένως στην χειρότερη περίπτωση:

- n πύλες XOR και XNOR για την παραγωγή των bits της μορφής $a_i \oplus b_i$ και $\sim(a_i \oplus b_i)$.
- $2\binom{n}{2}$ πύλες AND και NAND για τον σχηματισμό των bits της μορφής $\alpha_i\alpha_j$, b_ib_j και $\sim \alpha_i\alpha_j$, $\sim b_ib_j$.
- n^2 πύλες AND και NAND για την παραγωγή των bits της μορφής α_ib_j και $\sim\alpha_ib_j$.

- $n^2 + 3n$ πολυπλέκτες οι οποίοι επιλέγουν τα σωστά bits των μερικών γινομένων ανάλογα με την επιλεγμένη λειτουργία.
- Μια βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου για τη μείωση των μερικών γινομένων. Η αρχιτεκτονική αυτή χρησιμοποιεί $n(n+2)$ πλήρεις αθροιστές.
- Έναν modulo $2^n + 1$ παράλληλο αθροιστή. Θεωρούμε ότι χρησιμοποιείται η αρχιτεκτονική που προτείνεται στο [18].

Το κρίσιμο μονοπάτι της απλής αρχιτεκτονικής $MMSSU_{+1}$ αρχίζει στην είσοδο μιας πύλης XOR ή XNOR και περνάει μέσα από έναν πολυπλέκτη, το δέντρο των αθροιστών και τον παράλληλο αθροιστή.

Χρησιμοποιώντας τις προσεγγίσεις του μοντέλου unit-gate, μπορούμε να υπολογίσουμε τον χώρο $A_{simple,+1}$ και την καθυστέρηση $T_{simple,+1}$ της απλής αρχιτεκτονικής $MMSSU_{+1}$ ως εξής:

$$A_{simple,+1} = 11n^2 + \frac{9}{2} n \log n + \frac{43}{2} n + 6$$

$$T_{simple,+1} = 4D(n+4) + 2\log n + 7$$

3.3 Αρχιτεκτονική μειωμένου χώρου $MMSSU_{+1}$

Η αρχιτεκτονική μειωμένου χώρου που θα περιγραφεί παρακάτω στοχεύει στη μείωση των πολυπλεκτών που απαιτούνται. Βασιζόμενοι στη μεταβλητή s, ορίζουμε τις παρακάτω μεταβλητές:

$$c_i = a_i s + b_i \bar{s},\ 0 \le i \le n-1$$

$$d_i = a_i \bar{s} + b_i s,\ 0 \le i \le n-1$$

$$e_i = s(a_i \oplus b_i),\ 0 \le i < \frac{n}{2}$$

$$e_i = \overline{s(a_i \oplus b_i)},\ \frac{n}{2} \le i \le n-1$$

Στις παραπάνω σχέσεις το σύμβολο ‘+’ υποδηλώνει την λογική πράξη OR. Χρησιμοποιώντας τις μεταβλητές αυτές και ακολουθώντας τους κανόνες αντικατάστασης που περιγράφηκαν για την περίπτωση modulo $2^n - 1$, καταλήγουμε

σε παρόμοιους πίνακες για τον πολλαπλασιασμό και το άθροισμα τετραγώνων. Οι μόνες διαφορές μεταξύ των δύο πινάκων είναι οι εξής:

- Η πρώτη στήλη του πίνακα του πολλαπλασιασμού μεταφέρεται στην λιγότερο σημαντική θέση του πίνακα του αθροίσματος τετραγώνων και
- κάποια bits της στήλης αυτής είναι συμπληρωμένα.

Για $n=8$, οι πίνακες του πολλαπλασιασμού και του αθροίσματος τετραγώνων παίρνουν τις παρακάτω μορφές.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	$\alpha_7 c_0$	$\alpha_6 c_0$	$\alpha_5 c_0$	$\alpha_4 c_0$	$\alpha_3 c_0$	$\alpha_2 c_0$	$\alpha_1 c_0$	$\alpha_0 b_0$
PP_1 =	$\alpha_6 c_1$	$\alpha_5 c_1$	$\alpha_4 c_1$	$\alpha_3 c_1$	$\alpha_2 c_1$	$\alpha_1 b_1$	$d_0 b_1$	$\sim\alpha_7 c_1$
PP_2 =	$\alpha_5 c_2$	$\alpha_4 c_2$	$\alpha_3 c_2$	$\alpha_2 b_2$	$d_1 b_2$	$d_0 b_2$	$\sim\alpha_7 c_2$	$\sim\alpha_6 c_2$
PP_3 =	$\alpha_4 c_3$	$\alpha_3 b_3$	$d_2 b_3$	$d_1 b_3$	$d_0 b_3$	$\sim\alpha_7 c_3$	$\sim\alpha_6 c_3$	$\sim\alpha_5 c_3$
PP_4 =	$d_3 b_4$	$d_2 b_4$	$d_1 b_4$	$d_0 b_4$	$\sim\alpha_7 c_4$	$\sim\alpha_6 c_4$	$\sim\alpha_5 c_4$	$\sim\alpha_4 b_4$
PP_5 =	$d_2 b_5$	$d_1 b_5$	$d_0 b_5$	$\sim\alpha_7 c_5$	$\sim\alpha_6 c_5$	$\sim\alpha_5 b_5$	$\sim d_4 b_5$	$\sim d_3 b_5$
PP_6 =	$d_1 b_6$	$d_0 b_6$	$\sim\alpha_7 c_6$	$\sim\alpha_6 b_6$	$\sim d_5 b_6$	$\sim d_4 b_6$	$\sim d_3 b_6$	$\sim d_2 b_6$
PP_7 =	$b_0 b_7$	$\sim\alpha_7 b_7$	$\sim d_6 b_7$	$\sim d_5 b_7$	$\sim d_4 b_7$	$\sim d_3 b_7$	$\sim d_2 b_7$	$\sim d_1 b_7$
PP_8 =	e_0	0	e_3	0	e_2	0	e_1	1
PP_9 =	e_4	1	e_7	1	e_6	1	e_5	1
PP_{10} =	α_7	α_6	α_5	α_4	α_3	α_2	α_1	α_0
PP_{11} =	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

Πίνακας 15. Νέα μορφή των μερικών γινομένων του πολλαπλασιασμού.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
PP_0 =	α_6c_0	α_5c_0	α_4c_0	α_3c_0	α_2c_0	α_1c_0	α_0b_0	$\sim\alpha_7c_0$
PP_1 =	α_5c_1	α_4c_1	α_3c_1	α_2c_1	α_1b_1	d_0b_1	$\sim\alpha_7c_1$	$\sim\alpha_6c_1$
PP_2 =	α_4c_2	α_3c_2	α_2b_2	d_1b_2	d_0b_2	$\sim\alpha_7c_2$	$\sim\alpha_6c_2$	$\sim\alpha_5c_2$
PP_3 =	α_3b_3	d_2b_3	d_1b_3	d_0b_3	$\sim\alpha_7c_3$	$\sim\alpha_6c_3$	$\sim\alpha_5c_3$	$\sim\alpha_4c_3$
PP_4 =	d_2b_4	d_1b_4	d_0b_4	$\sim\alpha_7c_4$	$\sim\alpha_6c_4$	$\sim\alpha_5c_4$	$\sim\alpha_4b_4$	$\sim d_3b_4$
PP_5 =	d_1b_5	d_0b_5	$\sim\alpha_7c_5$	$\sim\alpha_6c_5$	$\sim\alpha_5b_5$	$\sim d_4b_5$	$\sim d_3b_5$	$\sim d_2b_5$
PP_6 =	d_0b_6	$\sim\alpha_7c_6$	$\sim\alpha_6b_6$	$\sim d_5b_6$	$\sim d_4b_6$	$\sim d_3b_6$	$\sim d_2b_6$	$\sim d_1b_6$
PP_7 =	$\sim\alpha_7b_7$	$\sim d_6b_7$	$\sim d_5b_7$	$\sim d_4b_7$	$\sim d_3b_7$	$\sim d_2b_7$	$\sim d_1b_7$	$\sim b_0b_7$
PP_8 =	0	e_3	0	e_2	0	e_1	1	e_0
PP_9 =	1	e_7	1	e_6	1	e_5	1	e_4
PP_{10} =	α_6	α_5	α_4	α_3	α_2	α_1	α_0	$\sim\alpha_7$
PP_{11} =	b_6	b_5	b_4	b_3	b_2	b_1	b_0	$\sim b_7$

Πίνακας 16. Νέα μορφή των μερικών γινομένων του αθροίσματος τετραγώνων.

Η αρχιτεκτονική μειωμένου χώρου $MMSSU_{+1}$ ξεκινάει λοιπόν από τον πίνακα του αθροίσματος τετραγώνων και χρησιμοποιεί πολυπλέκτες στα ακόλουθα σημεία:

- Στην λιγότερο σημαντική θέση, προκειμένου να γίνει η επιλογή ανάμεσα στην κανονική και την συμπληρωμένη μορφή των bits.
- Πριν από τον τελικό παράλληλο αθροιστή, προκειμένου να πραγματοποιηθεί δεξιά κυκλική ολίσθηση των εισόδων του όταν $s = 0$.
- Στις εισόδους κρατουμένου των πλήρων αθροιστών της περισσότερο και της λιγότερο σημαντικής θέσης, προκειμένου να γίνει η επιλογή ανάμεσα στην κανονική και την συμπληρωμένη μορφή των bits. Απαιτούνται επομένως δύο πολυπλέκτες για κάθε γραμμή πλήρων αθροιστών του δέντρου της μείωσης. Εφόσον έχουμε $n+4$ μερικά γινόμενα, το δέντρο της μείωσης θα έχει $n+2$ γραμμές πλήρων αθροιστών, οπότε απαιτούνται $2(n+2) = 2n+4$ πολυπλέκτες.

Η αρχιτεκτονική μειωμένου χώρου απαιτεί λοιπόν:

- n πύλες XOR και XNOR για τον σχηματισμό των bits της μορφής $a_i \oplus b_i$ και $\sim(a_i \oplus b_i)$.
- n πύλες AND για τον σχηματισμό των όρων e_i.

- 2n πολυπλέκτες για τον σχηματισμό των όρων c_i και d_i.
- $2\binom{n}{2}$ πύλες AND και NAND για τον σχηματισμό των bits της μορφής $\alpha_i c_j$, $d_i b_j$ και ~ $\alpha_i c_j$, ~ $d_i b_j$.
- n πύλες AND και NAND για την παραγωγή των όρων $\alpha_i b_i$ και ~ $\alpha_i b_i$.
- $n+2$ πολυπλέκτες για την επιλογή των σωστών bits της τελευταίας στήλης.
- Μια βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου για τη μείωση των $n+4$ μερικών γινομένων σε δύο. Η αρχιτεκτονική αυτή απαιτεί $n(n+2)$ πλήρεις αθροιστές.
- $2n+4$ πολυπλέκτες για την επιλογή των σωστών κρατουμένων στις εισόδους των πλήρων αθροιστών.
- 2n πολυπλέκτες για την πραγματοποίηση δεξιάς κυκλικής ολίσθησης των εισόδων του παράλληλου αθροιστή όταν $s=0$.
- Έναν modulo 2^n+1 παράλληλο αθροιστή. Θεωρούμε ότι ο αθροιστής αυτός ακολουθεί την αρχιτεκτονική που παρουσιάζεται στο [18].

Το κρίσιμο μονοπάτι της αρχιτεκτονικής αυτής ξεκινάει στην είσοδο ενός πολυπλέκτη για την παραγωγή ενός όρου c_j ή d_i, περνάει μέσα από μια πύλη AND για την παραγωγή ενός όρου $\alpha_i c_j$ ή $d_i b_j$, μέσα από έναν πολυπλέκτη ο οποίος αποφασίζει αν ο όρος αυτός πρέπει να παρουσιασθεί αυτούσιος ή συμπληρωμένος στο δέντρο της μείωσης, μέσα από το δέντρο των πλήρων αθροιστών, τους πολυπλέκτες για την δεξιά κυκλική ολίσθηση και τον τελικό παράλληλο αθροιστή. Για $n \geq 4$, ισχύει $D(n+4) \leq n$, οπότε το κρίσιμο μονοπάτι μέσα από το δέντρο των πλήρων αθροιστών περιλαμβάνει δύο πολυπλέκτες.

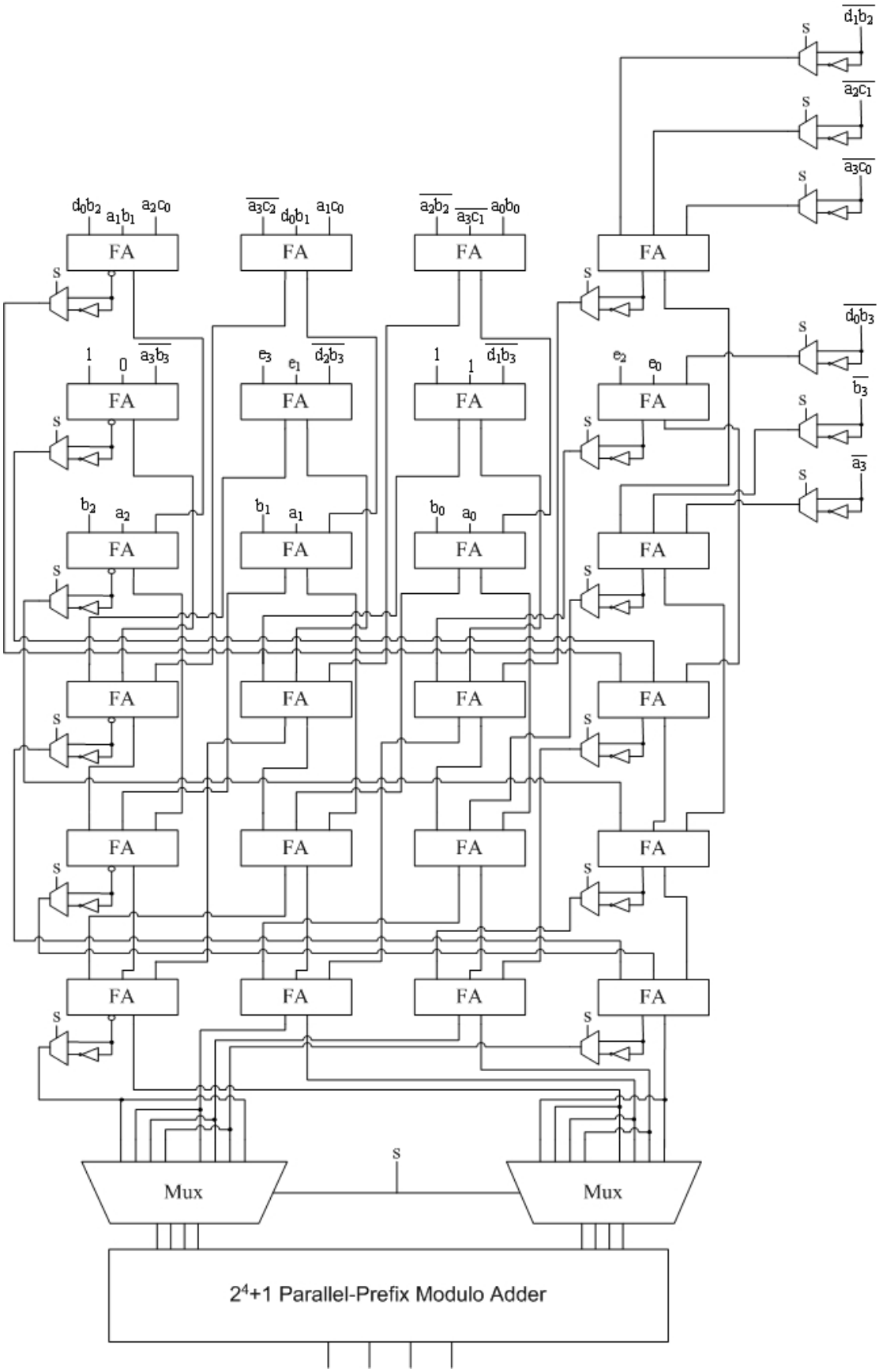
Χρησιμοποιώντας τις προσεγγίσεις του μοντέλου unit-gate, μπορούμε να υπολογίσουμε τον χώρο $A_{reduced,+1}$ και την καθυστέρηση $T_{reduced,+1}$ της αρχιτεκτονικής μειωμένου χώρου $MMSSU_{+1}$ ως εξής:

$$A_{reduced,+1} = 8n^2 + \frac{9}{2} n \log n + \frac{63}{2} n + 18$$

$$T_{reduced,+1} = 4D(n+4) + 2\log n + 12$$

Το σχήμα 2 παρακάτω απεικονίζει την αρχιτεκτονική μειωμένου χώρου $MMSSU_{+1}$ για $n=4$. Οι παραγωγή των όρων c_i, d_i και e_i δεν φαίνεται στο σχήμα για λόγους απλότητας. Έπειτα από την παραγωγή των μερικών γινομένων, χρησιμοποιείται μια βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου για τη μείωσή τους σε δύο τελικούς προσθετέους. Φαίνονται οι πολυπλέκτες και οι

αντιστροφείς οι οποίοι είναι απαραίτητοι για την ορθή λειτουργία του δέντρου μείωσης. Ο πρώτος πλήρης αθροιστής κάθε σειράς παράγει το συμπλήρωμα του κρατουμένου της πρόσθεσης. Παρατηρούμε ότι πριν από τον τελικό modulo αθροιστή υπάρχουν πολυπλέκτες οι οποίοι είναι απαραίτητοι για την δεξιά κυκλική ολίσθηση των προσθετέων στην περίπτωση που ισχύει $s = 0$.



Σχήμα 2. 4-bit αρχιτεκτονική μειωμένου χώρου MMSSU$_{+1}$.

3.4 Ποιοτικές συγκρίσεις

Χρησιμοποιώντας και πάλι τις προσεγγίσεις του μοντέλου unit-gate, συγκρίνουμε τις προτεινόμενες αρχιτεκτονικές $MMSSU_{+1}$ με δύο βασικές αρχιτεκτονικές A και B. Η αρχιτεκτονική A προσφέρει μόνο έναν πολλαπλασιαστή και έναν modulo αθροιστή, ενώ η αρχιτεκτονική B προσφέρει έναν πολλαπλασιαστή, έναν τετραγωνιστή και έναν modulo αθροιστή. Θεωρούμε ότι ο πολλαπλασιαστής υιοθετεί την αρχιτεκτονική του [21], ο παράλληλος αθροιστής την αρχιτεκτονική του [18] και ο τετραγωνιστής την αρχιτεκτονική του [22]. Ο χώρος και η καθυστέρηση των μονάδων αυτών τότε είναι:

$$A_{multiplier,+1} = 8n^2 + \frac{9}{2}n\log n + \frac{7}{2}n + 6$$

$$T_{multiplier,+1} = 4D(n+2) + 2\log n + 6$$

$$A_{adder,+1} = \frac{9}{2}n\log n + \frac{n}{2} + 6$$

$$T_{adder,+1} = 2\log n + 3$$

$$A_{squarer,+1} = \frac{n^2}{2} + 7n\left\lfloor \frac{n+1}{2} \right\rfloor + \frac{9}{2}n\log n + \frac{n}{2} + 6$$

$$T_{squarer,+1} = 4D\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) + 2\log n + 4$$

Χρησιμοποιώντας τις παραπάνω προσεγγίσεις, εκτιμούμε τις απαιτήσεις σε χώρο καθώς και την καθυστέρηση των βασικών συστημάτων A και B όταν εκτελούν τη λειτουργία του αθροίσματος τετραγώνων:

$$A_{A,+1} = 8n^2 + 9n\log n + 4n + 12$$

$$T_{A,+1} = 8D(n+2) + 6\log n + 15$$

$$A_{B,+1} = \frac{17}{2}n^2 + 7n\left\lfloor \frac{n+1}{2} \right\rfloor + \frac{27}{2}n\log n + \frac{9}{2}n + 18$$

$$T_{B,+1} = 8D\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) + 6\log n + 11$$

Ο παρακάτω πίνακας παρουσιάζει συγκριτικά τις χωρικές απαιτήσεις και την καθυστέρηση των βασικών συστημάτων και των προτεινόμενων αρχιτεκτονικών για διάφορες τιμές του n.

n	$A_{A,+1}$	$A_{B,+1}$	$A_{simple,+1}$	$A_{reduced,+1}$	$T_{A,+1}$	$T_{B,+1}$	$T_{simple,+1}$	$T_{reduced,+1}$
4	228	336	304	308	51	23	27	32
8	772	1146	990	890	73	45	33	38
12	1599	2380	2041	1741	84	56	38	43
16	2700	4026	3454	2858	87	67	43	48
20	4069	6074	5224	4236	96	76	43	48
24	5706	8523	7353	5877	98	78	44	49
28	7607	11369	9837	7777	107	87	48	53
32	9772	14610	12678	9938	109	89	49	54

Πίνακας 17. Ποιοτικές συγκρίσεις για συστήματα modulo $2^n + 1$.

Οι τιμές του παραπάνω πίνακα δείχνουν ότι ο χώρος που απαιτούν οι προτεινόμενες αρχιτεκτονικές $MMSSU_{-1}$ βρίσκεται μεταξύ του χώρου του συστήματος A και του χώρου του συστήματος B. Η προτεινόμενη αρχιτεκτονική μειωμένου χώρου απαιτεί λιγότερο από 5.9% επιπλέον χώρο σε σχέση με την βασική αρχιτεκτονική A όταν $n \geq 16$. Από την άλλη πλευρά, οι χωρικές απαιτήσεις της βασικής αρχιτεκτονικής B είναι σημαντικά μεγαλύτερες. Θεωρώντας τον μέσο όρο των εξεταζόμενων τιμών, η αρχιτεκτονική B απαιτεί περισσότερο από 15% και 37% επιπλέον χώρο σε σχέση με την απλή αρχιτεκτονική και την αρχιτεκτονική μειωμένου χώρου αντίστοιχα.

Στρέφουμε τώρα την προσοχή μας στην καθυστέρηση των παραπάνω αρχιτεκτονικών. Παρατηρούμε ότι και οι δύο προτεινόμενες αρχιτεκτονικές παρουσιάζουν σημαντικά μικρότερη καθυστέρηση για τη λειτουργία του αθροίσματος τετραγώνων και αρκετά μεγάλες τιμές του n. Όπως είναι φυσικό, η απλή αρχιτεκτονική είναι πιο γρήγορη. Θεωρώντας και πάλι τον μέσο όρο των εξεταζόμενων περιπτώσεων, η απλή αρχιτεκτονική παρουσιάζει 53.4% και 31,8% μικρότερη καθυστέρηση σε σχέση με τις βασικές αρχιτεκτονικές A και B αντίστοιχα. Οι επιπλέον πολυπλέκτες στο κρίσιμο μονοπάτι των προτεινόμενων αρχιτεκτονικών μειωμένου χώρου τις καθιστούν λίγο πιο αργές σε σύγκριση με την απλή αρχιτεκτονική. Η αρχιτεκτονική μειωμένου χώρου παρουσιάζει κατά μέσο όρο 45.1% και 18.9% μικρότερη καθυστέρηση σε σχέση με τις βασικές αρχιτεκτονικές A και B αντίστοιχα.

4. Ποσοτικές συγκρίσεις

Προκειμένου να παρθούν μετρήσεις χώρου και καθυστέρησης βασισμένες σε CMOS υλοποιήσεις, αναπτύχθηκαν προγράμματα σε C τα οποία είναι ικανά να παράγουν Verilog περιγραφές των προτεινόμενων αρχιτεκτονικών. Οι προτεινόμενες αρχιτεκτονικές συγκρίνονται με τους αντίστοιχους modulo πολλαπλασιαστές. Οι πολλαπλασιαστές αυτοί, όπως και οι MMSSUs, χρησιμοποιούν μια βασιζόμενη σε πλήρεις αθροιστές αρχιτεκτονική δέντρου για τη μείωση των μερικών γινομένων. Επιπλέον, χρησιμοποιούν τον ίδιο παράλληλο αθροιστή για την τελική modulo πρόσθεση. Παρακάτω εξετάζονται μεγέθη των 4, 8, 16 και 32 bits.

Αρχικά πραγματοποιήθηκαν προσομοιώσεις της λειτουργίας των συγκρινόμενων μονάδων προκειμένου να επαληθευτεί η ορθότητά τους. Ακολούθησε σύνθεση των παραπάνω μονάδων με χρήση μιας στατικής CMOS τεχνολογίας των 0.25 μm. Για τη σύνθεση των μονάδων χρησιμοποιήθηκε το εργαλείο Synopsys Design Compiler. Οι σχεδιασμοί βελτιστοποιήθηκαν τόσο για χώρο όσο και για καθυστέρηση. Μετά από την αρχική σύνθεση των σχεδιασμών, πραγματοποιήθηκε μια δεύτερη σύνθεση προκειμένου να παραχθούν καλύτερα αποτελέσματα. Για όλους τους σχεδιασμούς χρησιμοποιήθηκε το 8 ως μέγιστη τιμή για το fan-out. Οι συνθήκες λειτουργίας θεωρήθηκαν τυπικές, όπως και οι καθυστερήσεις των επιμέρους στοιχείων. Σημειώνεται ακόμα ότι δεν χρησιμοποιήθηκε boundary optimization για τη σύνθεση των μονάδων.

Ο πίνακας 18 παρουσιάζει αποτελέσματα για την περίπτωση modulo $2^n - 1$. Παρατηρούμε ότι η απόδοση των προτεινόμενων αρχιτεκτονικών σε σχέση με τον απαιτούμενο χώρο υλοποίησης αυξάνεται σύμφωνα με το n. Για $n = 32$, η απλή αρχιτεκτονική $MMSSU_{-1}$ και η αρχιτεκτονική μειωμένου χώρου $MMSSU_{-1}$ απαιτούν 9.3% και 2.5% περισσότερο χώρο αντίστοιχα σε σχέση με τον απλό modulo πολλαπλασιαστή. Η επιπλέον καθυστέρηση των προτεινόμενων μονάδων $MMSSU_{-1}$ είναι κατά μέσο όρο 0.6 ns και 0.83 ns αντίστοιχα. Πρέπει να σημειωθεί ότι αυτή είναι η μέγιστη καθυστέρηση των μονάδων, ανεξάρτητα από την επιθυμητή λειτουργία. Αντιθέτως, ένα σύστημα το οποίο χρησιμοποιεί μόνο έναν πολλαπλασιαστή και έναν modulo αθροιστή, απαιτεί δύο κύκλους πολλαπλασιασμού και έναν κύκλο πρόσθεσης. Λαμβάνοντας υπόψη ότι στην ίδια τεχνολογία

υλοποίησης ένας modulo $2^{16}-1$ αθροιστής απαιτεί χώρο 10586 μm^2 και παρουσιάζει καθυστέρηση 1.77 ns ([17]), συμπεραίνουμε ότι η λύση του πολλαπλασιαστή και του αθροιστή απαιτεί χώρο 115230 μm^2 και παρουσιάζει καθυστέρηση 9.05 ns για τη λειτουργία του αθροίσματος τετραγώνων και για $n=16$. Αυτό είναι περισσότερο από το διπλάσιο της καθυστέρησης που παρουσιάζουν και οι δύο προτεινόμενες αρχιτεκτονικές MMSSU$_{-1}$. Επομένως, οι προτεινόμενες μονάδες παρουσιάζουν πολύ καλή απόδοση σε μία εφαρμογή που χρησιμοποιεί συχνά τη λειτουργία του αθροίσματος τετραγώνων.

	Πολλαπλασιαστής		Απλή MMSSU$_{-1}$		MMSSU$_{-1}$ μειωμένου χώρου	
n	Χώρος (μm^2)	Καθυστέρηση (ns)	Χώρος (μm^2)	Καθυστέρηση (ns)	Χώρος (μm^2)	Καθυστέρηση (ns)
4	6660	1.88	12377	2.43	11401	2.69
8	27377	2.68	40521	3.28	38967	3.54
16	104644	3.64	130213	4.20	119165	4.42
32	415023	4.52	453808	5.19	425274	5.40

Πίνακας 18. Αποτελέσματα υλοποιήσεων για μονάδες modulo 2^n-1.

Στον πίνακα 19 παρουσιάζονται αποτελέσματα για την περίπτωση modulo 2^n+1. Παρατηρούμε ότι η αρχιτεκτονική μειωμένου χώρου παρουσιάζει καλύτερα αποτελέσματα σε σχέση με την απλή αρχιτεκτονική για $n \geq 16$. Για μικρότερες τιμές του n, οι πολυπλέκτες που απαιτούνται για τον χειρισμό του κρατουμένου καθώς και για τις ολισθήσεις πριν από την τελική modulo πρόσθεση, αντισταθμίζουν τη μείωση των πολυπλεκτών για την παραγωγή των μερικών γινομένων. Και πάλι η απόδοση των προτεινόμενων μονάδων αυξάνει με το n, καθώς οι χωρικές τους απαιτήσεις προσεγγίζουν αυτές του αντίστοιχου πολλαπλασιαστή. Για $n=16$ και $n=32$, η αρχιτεκτονική μειωμένου χώρου οδηγεί σε υλοποιήσεις που απαιτούν μόλις 2.2% και -0.1% περισσότερο χώρο αντίστοιχα σε σχέση με τον πολλαπλασιαστή. Η απλή αρχιτεκτονική και η αρχιτεκτονική μειωμένου χώρου είναι κατά μέσο όρο 0.66 ns και 1.07 ns πιο αργές σε σύγκριση με τον αντίστοιχο πολλαπλασιαστή. Λαμβάνοντας υπόψη ότι στην ίδια τεχνολογία υλοποίησης ένας modulo $2^{16}+1$ αθροιστής απαιτεί χώρο 12489 μm^2 και παρουσιάζει καθυστέρηση 1.68 ns ([17]), συμπεραίνουμε ότι η

λύση του πολλαπλασιαστή και του αθροιστή απαιτεί χώρο 146422 μm^2 και παρουσιάζει καθυστέρηση 9.14 ns για τη λειτουργία του αθροίσματος τετραγώνων και για $n = 16$. Επομένως και πάλι οι προτεινόμενες μονάδες $MMSSU_{+1}$ παρουσιάζουν πολύ καλή απόδοση σε μία εφαρμογή που χρησιμοποιεί συχνά τη λειτουργία του αθροίσματος τετραγώνων.

	Πολλαπλασιαστής		Απλή $MMSSU_{+1}$		$MMSSU_{+1}$ μειωμένου χώρου	
n	Χώρος (μm^2)	Καθυστέρηση (ns)	Χώρος (μm^2)	Καθυστέρηση (ns)	Χώρος (μm^2)	Καθυστέρηση (ns)
4	10703	2.27	17524	3.33	17865	3.28
8	38653	2.94	45553	3.46	46048	4.00
16	133933	3.73	141419	4.21	136886	4.83
32	442089	4.54	467319	5.10	441652	5.66

Πίνακας 19. Αποτελέσματα υλοποιήσεων για μονάδες modulo $2^n + 1$.

5. Συμπεράσματα

Η χρήση ενός RNS είναι ιδιαίτερα ελκυστική σε αρκετές εφαρμογές DSP και πολυμέσων οι οποίες χρησιμοποιούν πάρα πολύ συχνά τις λειτουργίες του πολλαπλασιασμού και του αθροίσματος τετραγώνων. Ο χώρος υλοποίησης που απαιτείται στη περίπτωση που χρησιμοποιούνται ξεχωριστές μονάδες για τις παραπάνω λειτουργίες είναι πολύ μεγάλος. Από την άλλη πλευρά, λύσεις που βασίζονται στη χρήση ενός ζευγαριού πολλαπλασιαστή και αθροιστή ή ενός ζευγαριού τετραγωνιστή και αθροιστή για την εκτέλεση της λειτουργίας του αθροίσματος τετραγώνων απαιτούν αρκετούς κύκλους μηχανής.

Για τους παραπάνω λόγους, στην παρούσα εργασία παρουσιάστηκαν δύο αρχιτεκτονικές για μονάδες που μπορούν να εκτελέσουν είτε την λειτουργία $\left|X \times Y\right|_R$ είτε την λειτουργία $\left|X^2 + Y^2\right|_R$, ανάλογα με την τιμή ενός σήματος ελέγχου. Θεωρήθηκε ότι το R μπορεί να έχει είτε την τιμή $2^n - 1$ είτε την τιμή $2^n + 1$, καθώς και ότι στην τελευταία περίπτωση χρησιμοποιείται η ελαττωμένη κατά ένα αναπαράσταση.

Γενικά, η επί τοις εκατό διαφορά ανάμεσα στις MMSSUs και τον αντίστοιχο πολλαπλασιαστή, τόσο σε χώρο υλοποίησης όσο και σε καθυστέρηση, μειώνεται καθώς το n αυξάνεται. Για εισόδους με μέγεθος 4 bits ή 8 bits οι προτεινόμενες μονάδες δεν προσφέρουν μεγάλη οικονομία. Αντίθετα, όταν το μέγεθος των εισόδων είναι 16 bits και περισσότερο, οι MMSSUs παρουσιάζουν πολύ καλή απόδοση και προσφέρουν σημαντική οικονομία σε χώρο υλοποίησης. Παρόλο που στις ποσοτικές συγκρίσεις του τμήματος 4 οι προτεινόμενες μονάδες συγκρίνονται με τον αντίστοιχο πολλαπλασιαστή, η σύγκριση θα μπορούσε κάλλιστα να γίνει με την αντίστοιχη μονάδα αθροίσματος τετραγώνων. Στην περίπτωση αυτή, οι επί τοις εκατό διαφορές θα ήταν ακόμη μικρότερες.

Η απλή υλοποίηση μιας MMSSU είναι σε γενικές γραμμές ταχύτερη από την αντίστοιχη υλοποίηση μειωμένου χώρου. Παρόλα αυτά, αν μας ενδιαφέρει κυρίως η οικονομία σε χώρο υλοποίησης, οι υλοποιήσεις μειωμένου χώρου αποδίδουν πολύ καλύτερα όταν συζητάμε για μονάδες modulo $2^n - 1$. Για μονάδες modulo $2^n + 1$, οι υλοποιήσεις μειωμένου χώρου προσφέρουν καλύτερα αποτελέσματα μόνο όταν το μέγεθος των εισόδων είναι 16 bits και περισσότερο. Αυτό οφείλεται κυρίως στους

πολυπλέκτες που απαιτούνται μετά από κάθε σειρά πλήρων αθροιστών στο δέντρο της μείωσης, κάτι που δεν υπάρχει στις απλές υλοποιήσεις.

Όταν το πρωταρχικό ενδιαφέρον σε έναν σχεδιασμό είναι ο χώρος υλοποίησης, οι προτεινόμενες μονάδες προσφέρουν μεγάλη οικονομία, ειδικά για μεγέθη των 16 bits και παραπάνω. Αυτά τα μεγέθη χρησιμοποιούνται ευρέως σήμερα. Στο κοντινό μέλλον, ακόμα μεγαλύτερα μεγέθη θα γίνουν κοινά και επομένως η απόδοση των προτεινόμενων μονάδων θα συνεχίσει να αυξάνεται.

Αναφορές

1. N. Szabo and R. Tanaka, “Residue Arithmetic and its Applications to Computer Technology”, McGraw-Hill, 1967.
2. M. Soderstrand, M. A. W. Jenkins, G. Jullien and F. Taylor, “Residue Number System Arithmetic : Modern Applications in Digital Signal Processing”, IEEE Press, New York, 1986.
3. T. Kwan and T. Martin, “Adaptive detection and enhancement of multiple sinusoids using a cascade IIR filter”, IEEE Trans. Circuits and Systems, **36**, 1989, pp. 937–945.
4. J. Ramirez, and U. Meyer-Baese, “High Performance, Reduced Complexity Programmable RNS-FPL Merged FIR Filters”, Electronics Letters, **38**, 2002, pp. 199–200.
5. J. Ramirez, A. Garcia, S. Lopez-Buedo and A. Lloris, “RNS-enabled digital signal processor design”, Electronics Letters, **38**, 2002, pp. 266–268.
6. T. Keller, T.H. Liew and L. Hanzo, “Adaptive redundant residue number system coded multicarrier modulation”, IEEE J. on Selected Areas in Communication, **18**, 2000, pp. 2292–2301.
7. J. Ramirez, A. Garcia, U. Meyers-Baese and A. Lloris, “Fast RNS FPL-based communications receiver design and implementation”, in Proc. of the 12th International Conference on Field Programmable Logic, Lecture Notes in Computer Science, Vol. 2438, Springer-Verlag, 2002, pp. 472–481.
8. L. Kalamboukas, D. Nikolos, C. Efstathiou, H. T. Vergos and J. Kalamatianos, “High-speed parallel-prefix modulo 2^n-1 adders”, IEEE Trans. on Computers, **49**, 2000, pp. 673–680.
9. G. Dimitrakopoulos, H. T. Vergos, D. Nikolos and C. Efstathiou, “A systematic methodology for designing area-time efficient parallel-prefix modulo 2^n-1 adders”, in Proc. of the IEEE International Symposium on Circuits and Systems, 2003, pp. 225–228.
10. G. Dimitrakopoulos, H. T. Vergos, D. Nikolos and C. Efstathiou, “A family of Parallel-Prefix Modulo 2^n-1 Adders”, in Proc. of the IEEE International

Conference on Application-Specific Systems, Architectures and Processors, 2003, pp. 326–336.

11. C. Efstathiou, H. T. Vergos, and D. Nikolos, “Fast parallel-prefix modulo 2^n+1 adders”, in Proc. of the XVII Conference on Design of Circuits and Integrated Systems, 2002, pp. 65–70.
12. C. Efstathiou, H. T. Vergos and D. Nikolos, “Modified Booth Modulo 2^n-1 Multipliers”, IEEE Trans. on Computers, **53**, 2004, pp. 370–374.
13. A. Wrzyszcz and D. Milford, “A new modulo 2^a+1 multiplier”, in Proc. of the International Conference on Computer Design (ICCD’93), 1995, pp. 614–617.
14. S. Piestrak, “Design of squarers modulo A with low-level pipelining”, IEEE Trans. on Computers, **49**, 2002, pp. 31–41.
15. L. M. Leibowitz, “A simplified binary arithmetic for the fermat number transform”, IEEE Trans. Acoust., Speech, Signal Processing, **24**, 1976, pp. 356–359.
16. R. Zimmermann, “Binary adder architectures for cell-based VLSI and their synthesis”, Ph.D. dissertation, Swiss Federal Institute of Technology, 1997.
17. C. Efstathiou, H. T. Vergos and D. Nikolos, “Modulo $2^n\pm1$ adder design using select-prefix blocks”, IEEE Transactions on Computers, **52**, 2003, pp. 1399–1406.
18. H. T. Vergos, C. Efstathiou and D. Nikolos, “Diminished-one modulo 2^n+1 adder design”, IEEE Trans. Comput., **51**, 2002, pp. 1389–1399.
19. Z. Wang, G. A. Jullien, and W. C. Miller, “An efficient tree architecture for modulo 2^n+1 multiplication”, Journal of VLSI Signal Processing, **14**, 1996, pp. 241–248.
20. Y. Ma, “A simplified architecture for modulo (2^n+1) multiplication”, IEEE Trans. Comput., **47**, 1998, pp. 333–337.
21. C. Efstathiou, H. T. Vergos, G. Dimitrakopoulos and D. Nikolos, “Efficient modulo 2^n+1 tree multipliers for diminished-1 operands”, in Proc. of the 10th IEEE Int. Conference on Electronics, Circuits and Systems, 2003, pp. 200–203.

22. H. T. Vergos, and C. Efstathiou, “Diminished-1 modulo 2^n+1 squarer Design”, Proc. of the 2004 Euromicro Symposium on Digital Systems Design, to appear.
23. R. K. Kolagotla, W. R. Griesbach and H. R. Srinivas, “VLSI implementation of 350 MHz 0.35 micron 8 bit merged squarer”, Electronics Letters, **34**, 1998, pp. 47–48.
24. J. T. Yoo, K. F. Smith and G. Gopalakrishnan, “A fast parallel squarer based on divide-and-conquer”, IEEE J. Solid-State Circuits, **32**, 1997, pp. 909–912.
25. J. Pihl and E. J. Aas, “A multiplier and squarer generator for high performance DSP applications”, in Proc. of the 39th Midwest Symposium on Circuits and Systems, vol. 1, 1996, pp. 109–112.
26. A. Eshraghi, T. S. Fiez, K. D. Winters and T. R. Fischer, “Design of a new squaring function for the Viterbi algorithm”, IEEE J. Solid-State Circuits, **29**, 1994, pp. 1102–1107.
27. G. Kempa and P. Jung, “FPGA based logic synthesis of squarers using VHDL”, in Proc. of the 2nd International Workshop on Field Programmable Logic and Applications, 1993, pp. 112–123.
28. R. H. Strandberg, L. G. Bustamante, V. G. Oklobdzija, M. Sonderstrand and J. C. Le Duc, “Efficient realizations of squaring circuit and reciprocal used in adaptive sample rate notch filters”, J. of VLSI Signal Processing, **14**, 1996, pp. 303–309.
29. R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W.Fichtner, “A 177 Mb/s VLSI implementation of the International Data Encryption Algorithm”, IEEE J. Solid-State Circuits, **29**, 1994, pp. 303–307.
30. C. S. Wallace, “A suggestion for a fast multiplier”, IEEE Trans. Electron. Comput., **EC-13**, 1964, pp. 14–17.
31. L. Dadda, “Some schemes for parallel multipliers”, Alta Frequenza, **34**, 1965, pp. 349–356.
32. A. Tyagi, “A reduced-area scheme for carry-select adders”, IEEE Trans. Comput., **42**, 1993, pp. 1163–1170.
33. Z. Wang, G. A. Jullien and W. C. Miller, “An algorithm for multiplication modulo 2^N-1”, in Proc. of the 39th IEEE Midwest Symposium on Circuits and Systems”, 1996, pp. 1301–1304.

34. M. G. Walker, “Modeling the wiring of deep submicron ICs”, IEEE Spectrum, vol. 37, no. 3, Mar. 2000, pp. 65–71.
35. H. Bhatnagar, “Advanced ASIC Chip Synthesis: using Synopsys Design Compiler, Physical Compiler, and Primetime”, Kluwer Academic Publishers, 2nd ed., 2001.