



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Ηλεκτρονικής & Μηχανικών Υπολογιστών

Διπλωματική Εργασία

**Μελέτη, Σχεδιασμός και Υλοποίηση σε
Αναδιατασσόμενη Λογική, Αρχιτεκτονικής για
τον παράλληλο Υπολογισμό Βέλτιστων Κανόνων
Golomb (GE3)**

Μαλακωνάκης Παύλος

Χανιά 2009

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Καθηγητή κ. Απόστολο Δόλλα για την επίβλεψη, την καθοδήγηση και τις χρήσιμες υποδείξεις του για την ολοκλήρωση αυτής της εργασίας. Επίσης θα ήθελα να ευχαριστήσω τους Καθηγητές κ. Διονύσιο Πνευματικάτο και κ. Ιωάννη Παπαευσταθίου για το ενδιαφέρον που έδειξαν για αυτήν την εργασία και για τις χρήσιμες παρατηρήσεις τους.

Επίσης θα ήθελα να ευχαριστήσω τον κ. Ευριπίδη Σωτηριάδη για τις πολύτιμες συμβουλές του, τη συνεχή καθοδήγησή του και τη βοήθειά του στη λύση διαφόρων προβλημάτων μέχρι την ολοκλήρωση της εργασίας αυτής. Ακόμα τον ευχαριστώ για τις πολύτιμες γνώσεις που μου μετέδωσε για το συγκεκριμένο πρόβλημα.

Επίσης ευχαριστώ τον κ. Μάρκο Κιμιωνή και όλους τους προπτυχιακούς και μεταπτυχιακούς φοιτητές του Εργαστηρίου Μικροεπεξεργαστών και Υλικού για την βοήθεια τους όποτε αυτή χρειάστηκε.

Ακόμα ευχαριστώ το Εργαστήριο Μικροεπεξεργαστών και Υλικού για τα ηλεκτρονικά αναλώσιμα που είχα στη διάθεσή μου κατά τη μελέτη της διπλωματικής μου εργασίας.

Επίσης ευχαριστώ όλους τους φίλους και συμφοιτητές μου για το συνεχές ενδιαφέρον τους και την συμπαράστασή τους σε όλη τη διάρκεια των σπουδών μου.

Τέλος, ευχαριστώ τους γονείς μου για την υπομονή τους και την ηθική και υλική τους συμπαράσταση που μου έδωσε δύναμη να ολοκληρώσω τις σπουδές μου.

Περιεχόμενα

1. Εισαγωγή.....	5
2. Κανόνες Golomb.....	7
2.1 Ορισμοί.....	7
2.1.1 Κανόνας- Δείκτες – Μήκος.....	7
2.1.2 Τρίγωνο Διαφορών.....	8
2.1.3 Κανόνες Golomb.....	9
2.1.4 Γνωστοί Κανόνες Golomb.....	10
2.2 Εφαρμογές κανόνων Golomb.....	12
2.3 Αλγόριθμοι για τον υπολογισμό Βέλτιστων Κανόνων Golomb.....	14
2.4 Shift Algorithm.....	15
2.5 Προηγούμενες σχεδιάσεις Golomb Engine στο Πολυτεχνείο Κρήτης.....	20
2.5.1 Αρχιτεκτονική GE1.....	20
2.5.2 Αρχιτεκτονική GE2.....	22
2.6 Υπολογισμός Κανόνων Golomb Μέσω Διαδικτύου.....	23
3. Αρχιτεκτονική μηχανής GE3.....	25
3.1 Ανάλυση αρχιτεκτονικής της GE3.....	25
3.2 Βελτίωση της GE3 σε σχέση με τη GE2 – πολλαπλά shift.....	32
3.3 Παραλληλισμός Προβλήματος – (Παραγωγή–Κατανάλωση).....	35
3.4 Υλοποίηση δυνατότητας Αποθήκευσης του σημείου που έχουμε φτάσει στο search space.....	37
4. Αποτύπωση της αρχιτεκτονικής της GE3.....	39
4.1 Αποτύπωση της GE3 χωρίς παραλληλισμό.....	39
4.2 Αποτύπωση της GE3 με παραλληλισμό.....	49
4.2.1 Αποτύπωση της αρχιτεκτονικής παραλληλισμού για τον κανόνα Golomb 12.....	49

4.2.2 Έρευνα για τον παραλληλισμό της αρχιτεκτονικής για τον κανόνα Golomb 25.....	50
4.2.3 Αποτύπωση της αρχιτεκτονικής παραλληλισμού για τον κανόνα Golomb 25.....	51
4.3 Πιστοποίηση λειτουργίας GE3.....	56
5. Αποτελέσματα της Σχεδίασης.....	57
5.1 Σύγκριση GE3 με shift - GE3 χωρίς shift.....	57
5.2 Αποτελέσματα από τον παραλληλισμό του προβλήματος.....	60
5.3 Σύγκριση GE3 – λογισμικό.....	60
6. Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	64
6.1 Συμπεράσματα.....	64
6.1.1 Συμπεράσματα από τη σχεδίαση και την αποτύπωση του αλγορίθμου shift.....	64
6.1.2 Συμπεράσματα που αφορούν την απόδοση της GE3.....	65
6.2 Μελλοντικές Επεκτάσεις.....	65
Βιβλιογραφία.....	67

1. Εισαγωγή

Κανόνας Golomb ονομάζεται ένας κανόνας (χάρακας) στον οποίο η απόσταση που μετράται μεταξύ δύο δεικτών είναι μοναδική γι' αυτόν τον κανόνα, δηλαδή η ίδια απόσταση δεν μετράται από δύο διαφορετικά ζεύγη δεικτών. Ως **βέλτιστος κανόνας Golomb** ορίζεται αυτός που έχει το μικρότερο μήκος για ένα δοσμένο αριθμό δεικτών.

Σε αυτή τη διπλωματική εργασία μελετούνται μέθοδοι και αρχιτεκτονικές για αποτύπωση αλγορίθμου μεγάλου υπολογιστικού φόρτου σε αναδιατασσομένη λογική (FPGA). Δηλαδή, μελετάται εάν τέτοιοι αλγόριθμοι μπορούν να σχεδιαστούν και να αποτυπωθούν σε αναδιατασσομένη λογική με ικανοποιητικά αποτελέσματα. Σκοπός της διπλωματικής αυτής είναι ο σχεδιασμός του **αλγορίθμου Shift**, ο οποίος είναι ο αλγόριθμος που επιλέχτηκε για τον υπολογισμό **βέλτιστων κανόνων Golomb**. Το συγκεκριμένο πρόβλημα είναι μεγάλου υπολογιστικού φόρτου.

Δηλαδή η διπλωματική αυτή αφορά :

- Στο σχεδιασμό αρχιτεκτονικής η οποία υλοποιεί τον αλγόριθμο Shift,
- Στη βελτίωση της αρχιτεκτονικής με την υποστήριξη πολλαπλών shift,
- Στην εκμετάλλευση του παραλληλισμού του προβλήματος με την υλοποίηση της αντίστοιχης αρχιτεκτονικής,
- Στη μελέτη της απόδοσης που επιτυγχάνεται από την αποτύπωση των αρχιτεκτονικών αυτών σε αναδιατασσομένη λογική (FPGA).

Στο δεύτερο κεφάλαιο δίνονται οι βασικοί ορισμοί που αφορούν τους κανόνες Golomb. Ακόμα γίνεται αναφορά στις διαφορές εφαρμογές των κανόνων Golomb. Επίσης, περιγράφονται οι αλγόριθμοι που υπάρχουν για τον υπολογισμό βέλτιστων κανόνων Golomb και αναλύεται ο αλγόριθμος που επιλέχτηκε τελικά, δηλαδή ο αλγόριθμος shift καθώς και οι τεχνικές που χρησιμοποιήθηκαν για τη βελτίωσή του. Επίσης αναφέρονται προηγούμενες αρχιτεκτονικές που έχουν αναπτυχτεί στο Πολυτεχνείο Κρήτης για την επίλυση του προβλήματος αυτού. Τέλος γίνεται αναφορά και στην αναζήτηση βέλτιστων κανόνων Golomb η οποία γίνεται μέσω του διαδικτύου.

Στο τρίτο κεφάλαιο, αναλύεται η αρχιτεκτονική της νέας μηχανής αναζήτησης βέλτιστων κανόνων Golomb (GE3), Datapath και Controlpath. Αρχικά δίνεται μια γενική περιγραφή της αρχιτεκτονικής και έπειτα αναλύονται περισσότερο τα βασικά

τμήματα από τα οποία αποτελείται. Έπειτα αναλύεται η βελτίωση, σε σχέση με προηγούμενες αρχιτεκτονικές, η οποία αφορά στην υποστήριξη πολλαπλών shift. Ακόμη αναλύεται ο τρόπος με τον οποίο επιτυγχάνεται ο παραλληλισμός του προβλήματος. Τέλος, γίνεται αναφορά στην, απαραίτητη για το πρόβλημα αυτό, υποστήριξη δυνατότητας αποθήκευσης του σημείου μέχρι το οποίο έχει ολοκληρωθεί η αναζήτηση, έτσι ώστε σε περίπτωση αστοχίας να μπορεί η αναζήτηση να συνεχιστεί από το σημείο αυτό.

Στο τέταρτο κεφάλαιο, αρχικά παρουσιάζονται τα αποτελέσματα από την αποτύπωση της αρχιτεκτονικής σε FPGA, χωρίς παραλληλισμό. Έπειτα αναλύεται ο παραλληλισμός του προβλήματος, ώστε να έχουμε την πιο αποδοτική εκμετάλλευση των πόρων που διαθέτουμε και παρουσιάζονται τα αποτελέσματα της αρχιτεκτονικής με παραλληλισμό (παραγωγή - κατανάλωση). Δηλαδή στο κεφάλαιο αυτό παρουσιάζονται διάφορα αποτελέσματα από την αποτύπωση του προβλήματος σε (FPGA).

Στο πέμπτο κεφάλαιο αναλύονται τα αποτελέσματα που παρατηρούνται στη βελτίωση της απόδοσης από τον υπολογισμό βέλτιστων κανόνων Golomb με αναδιατασσόμενη λογική σε σχέση με τον υπολογισμό τους με λογισμικό.

Στο τελευταίο, έκτο, κεφάλαιο παρατίθενται τα συμπεράσματα που βγαίνουν από τα αποτελέσματα αλλά και από την σχεδίαση. Δηλαδή, το πόσο βελτιώνεται η απόδοση με την απεικόνιση ενός απαιτητικού υπολογιστικά αλγόριθμου σε τεχνολογία FPGA. Επίσης παρατίθενται μελλοντικές επεκτάσεις για την αρχιτεκτονική και γενικά για την εργασία αυτή.

2. Κανόνες Golomb

Στο κεφάλαιο αυτό αναλύονται βασικές έννοιες, όπως ο ορισμός του τι είναι κανόνας Golomb. Ακόμα αναφέρονται διάφορες εφαρμογές των κανόνων Golomb. Επίσης παρουσιάζονται οι αλγόριθμοι με τους οποίους υπολογίζονται βέλτιστοι κανόνες Golomb, καθώς επίσης και τα βήματα έχουν γίνει μέχρι σήμερα για τον υπολογισμό βέλτιστων κανόνων Golomb. Ακόμη αναλύεται ο αλγόριθμος Shift, ο οποίος και επιλέχθηκε προς μελέτη και υλοποίηση. Επίσης οι τεχνικές που μελετήθηκαν και χρησιμοποιήθηκαν για τον αλγόριθμο Shift, ώστε να μειωθεί ο χώρος αναζήτησης και άρα ο υπολογιστικός φόρτος.

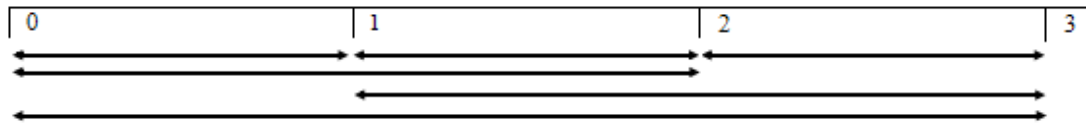
2.1 Ορισμοί

Είναι απαραίτητο να δώσουμε μερικούς βασικούς ορισμούς που χρησιμεύουν για την περιγραφή του κανόνα Golomb. Ακολουθώντας τους όρους αυτούς θα δοθεί ένας τυπικός ορισμός για τους κανόνες Golomb.

2.1.1 Κανόνας - Δείκτες - Μήκος

Ένας συνηθισμένος κανόνας (χάρακας) είναι μια διατεταγμένη σειρά από ακέραιους αριθμούς που ονομάζονται δείκτες (marks) και αντιστοιχούν σε θέσεις σε μια γραμμική κλίμακα. Η διαφορά μεταξύ των τιμών δύο οποιονδήποτε δεικτών ονομάζεται απόσταση μεταξύ αυτών των δεικτών. Η διαφορά μεταξύ του μεγαλύτερου και του μικρότερου αριθμού της παραπάνω σειράς ονομάζεται μήκος του κανόνα και είναι η μεγαλύτερη απόσταση που περιέχει ο κανόνας. Ο πρώτος δείκτης της σειράς είναι κατά σύμβαση στη θέση 0.

Ο κανόνας στο Σχήμα 2-1 έχει 4 δείκτες στις θέσεις 0,1,2,3. Εδώ έχουμε τη δυνατότητα να μετρήσουμε μέχρι 3 αποστάσεις. Οι αποστάσεις μεταξύ των δεικτών είναι σημειωμένες πάνω στο σχήμα. Όπως φαίνεται η απόσταση 1 μπορεί να μετρηθεί με τρεις διαφορετικούς τρόπους, δηλ. 0-1, 1-2, 2-3. Παρόμοια η απόσταση 2 μπορεί να μετρηθεί με δύο τρόπους, ενώ η απόσταση 3 με ένα τρόπο. Το πλεονέκτημα σ' αυτό το κανόνα είναι ότι μπορούν να μετρηθούν όλες οι αποστάσεις από το 0 έως το 3.



Σχήμα 2-1 Αποστάσεις μεταξύ δεικτών σ' έναν κανονικό κανόνα

2.1.2 Τρίγωνο Διαφορών

Πριν οριστεί η έννοια του κανόνα Golomb, είναι χρήσιμο να απεικονισθούν σε μία δομή δεδομένων όλες οι αποστάσεις που μπορούν να μετρηθούν από έναν τέτοιο κανόνα. Ένα τρίγωνο διαφορών είναι ένα πλέγμα αριθμών, που ο καθένας αντιπροσωπεύει την απόσταση μεταξύ ενός προσδιορισμένου ζευγαριού δεικτών. Για ένα χάρακα των m δεικτών το αντίστοιχο τρίγωνο διαφορών θα έχει $m(m-1)/2$ εισαγωγές όπως φαίνεται στο Σχήμα 2-2 για ένα χάρακα με 4 δείκτες.

$$\begin{array}{ccc}
 d_{11} & d_{12} & d_{13} \\
 & d_{21} & d_{22} \\
 & & d_{31}
 \end{array}$$

Σχήμα 2-2 Το τρίγωνο διαφορών 1ης, 2ης και 3ης τάξης

Οι όροι d_{1x} , d_{2x} , d_{3x} είναι οι 1ης, 2ης και 3ης τάξης διαφορές αντίστοιχα. Οι πρώτης τάξης διαφορές είναι οι αποστάσεις που μετρούνται μεταξύ διαδοχικών δεικτών του κανόνα. Οι δεύτερης τάξης διαφορές είναι οι αποστάσεις που μετρούνται μεταξύ δεικτών που είναι τοποθετημένοι ανά δύο θέσεις στον κανόνα κ.ο.κ.. Οι διαφορές αυτές υπολογίζονται από τις παρακάτω εξισώσεις:

$$d_{1_x} = M_{x+1} - M_x$$

$$d_{2_x} = d_{1_x} + d_{1_{x+1}}$$

$$d_{3_x} = d_{2_x} + d_{2_{x+1}} - d_{1_{x+1}}$$

Οι εξισώσεις για μεγαλύτερη τάξη διαφορών είναι απλά μια επέκταση της εξίσωσης για τρίτης τάξης διαφορών. Οι εξίσωση 4ης τάξης είναι :

$$d4_x = d3_x + d3_{x+1} - d2_{x+1}$$

Στο Σχήμα 2-3 φαίνεται ένα παράδειγμα τριγώνου διαφορών για ένα χάρακα με 4 δείκτες (κανόνας Golomb 4). Η πρώτη γραμμή αντιπροσωπεύει τις θέσεις των δεικτών του κανόνα, ενώ από κάτω σχηματίζεται το τρίγωνο διαφορών.

0	1	4	6
1	3	2	
	4	5	
		6	

Σχήμα 2-3 Τρίγωνο Διαφορών 4 δεικτών μήκους 6 (κανόνας Golomb 4)

2.1.3 Κανόνες Golomb

Ο **κανόνας Golomb** ορίζεται ως μια αριθμημένη ταξινομημένη σειρά που αντιστοιχεί στο τρίγωνο διαφορών, στο οποίο δεν περιλαμβάνονται διπλά στοιχεία [1],[2],[3],[4],[5],[6],[7]. Η απόσταση που μετράται μεταξύ δύο δεικτών για ένα κανόνα Golomb είναι μοναδική γι' αυτόν τον κανόνα, δηλαδή η ίδια απόσταση δεν μετράται από δύο διαφορετικά ζεύγη δεικτών. Ας σημειωθεί ότι ο παραπάνω ορισμός δεν περιλαμβάνει την έννοια του μήκους του κανόνα, ερευνητικά όμως το ενδιαφέρον επικεντρώνεται στην εύρεση του κανόνα με το μικρότερο μήκος.

Ως **τέλειος κανόνας Golomb** ορίζεται ένας κανόνας Golomb όπου το τρίγωνο διαφορών περιέχει όλους τους αριθμούς μεταξύ του ενός, και του μήκους του κανόνα. Αφού για ένα κανόνα υπάρχουν είσοδοι στο τρίγωνο διαφορών, το μήκος ενός N δεικτών τέλειου κανόνα Golomb είναι $(N-1)N/2$. Έχει αποδειχθεί ότι τέλειος κανόνας μεγαλύτερος των τεσσάρων δεικτών δεν μπορεί να υπάρξει.

Ως **βέλτιστος κανόνας Golomb** ορίζεται αυτός που έχει το μικρότερο μήκος για ένα δοσμένο αριθμό δεικτών. Αποδεικνύεται ότι μπορεί να υπάρξουν περισσότεροι από έναν βέλτιστοι κανόνες Golomb για ένα δοσμένο αριθμό δεικτών. Οι βέλτιστοι κανόνες Golomb ορίζονται επειδή, όπως προαναφέρθηκε δεν υπάρχει τέλειος κανόνας Golomb με περισσότερους από τέσσερις δείκτες.

Υπάρχουν όμως και άλλοι κανόνες, γνωστοί ως **σχεδόν βέλτιστοι κανόνες Golomb** που είναι απλοί κανόνες Golomb, των οποίων το μήκος είναι κατά μικρό αριθμό μονάδων μεγαλύτερο ή ίσο με το μήκος του αντίστοιχου βέλτιστου κανόνα, ο οποίος δεν έχει αποδειχθεί ακόμα. Οι σχεδόν βέλτιστοι κανόνες Golomb υπολογίζονται με αριθμητικές μεθόδους.

Κάθε τμήμα ενός κανόνα Golomb είναι και αυτό με τη σειρά του ένας κανόνας Golomb με λιγότερους δείκτες. Ένα τέτοιο τμήμα καλείται **υποκανόνας Golomb** (stub).

Επίσης οι κανόνες Golomb εμφανίζονται ανά ζεύγη, ο κανόνας και η **εικόνα καθρέπτη** του. Η εικόνα καθρέπτη είναι ο ίδιος κανόνας απλά αντεστραμμένος. Δηλαδή δημιουργείται αν τοποθετήσουμε τον τελευταίο δείκτη n στο 0 και έπειτα για κάθε άλλον δείκτη x , δημιουργείται ένας νέος δείκτης στη θέση με τιμή $n-x$. Ακόμα ο κανόνας και η εικόνα καθρέπτη του μετρούν τις ίδιες αποστάσεις. Στο Σχήμα 2-4 φαίνεται ένα παράδειγμα με τον κανόνα Golomb με τέσσερις δείκτες και την εικόνα καθρέπτη του.

0	1			4		6

0		2			5	6

Σχήμα 2-4 Κανόνας Golomb 4 δεικτών μήκους 6 και η αντίστοιχη εικόνα καθρέπτη

2.1.4 Γνωστοί Κανόνες Golomb

Στον παρακάτω πίνακα (Πίνακας 2-1), φαίνεται αριστερά ο αριθμός των δεικτών του κανόνα Golomb και δεξιά οι θέσεις που καταλαμβάνουν οι δείκτες. Οι κανόνες αυτοί είναι οι βέλτιστοι κανόνες Golomb που έχουν αποδειχθεί υπολογιστικά μέχρι σήμερα.

order length		marks
1	0	0
2	1	0 1
3	3	0 1 3
4	6	0 1 4 6
5	11	0 1 4 9 11
		0 2 7 8 11
6	17	0 1 4 10 12 17
		0 1 4 10 15 17
		0 1 8 11 13 17
		0 1 8 12 14 17
		0 1 4 10 18 23 25
7	25	0 1 7 11 20 23 25
		0 1 11 16 19 23 25
		0 2 3 10 16 21 25
		0 2 7 13 21 22 25
8	34	0 1 4 9 15 22 32 34
9	44	0 1 5 12 25 27 35 41 44
10	55	0 1 6 10 23 26 34 41 53 55
11	72	0 1 4 13 28 33 47 54 64 70 72
		0 1 9 19 24 31 52 56 58 69 72
12	85	0 2 6 24 29 40 43 55 68 75 76 85
13	106	0 2 5 25 37 43 59 70 85 89 98 99 106
14	127	0 4 6 20 35 52 59 77 78 86 89 99 122 127
15	151	0 4 20 30 57 59 62 76 100 111 123 136 144 145 151
16	177	0 1 4 11 26 32 56 68 76 115 117 134 150 163 168 177
17	199	0 5 7 17 52 56 67 80 81 100 122 138 159 165 168 191 199
18	216	0 2 10 22 53 56 82 83 89 98 130 148 153 167 188 192 205 216
19	246	0 1 6 25 32 72 100 108 120 130 153 169 187 190 204 231 233 242 246
20	283	0 1 8 11 68 77 94 116 121 156 158 179 194 208 212 228 240 253 259
		283
21	333	0 2 24 56 77 82 83 95 129 144 179 186 195 255 265 285 293 296 310
		329 333
22	356	0 1 9 14 43 70 106 122 124 128 159 179 204 223 253 263 270 291 330
		341 353 356
23	372	0 3 7 17 61 66 91 99 114 159 171 199 200 226 235 246 277 316 329
		348 350 366 372
24	425	0 9 33 37 38 97 122 129 140 142 152 191 205 208 252 278 286 326 332
		353 368 384 403 425
25	480	0 12 29 39 72 91 146 157 160 161 166 191 207 214 258 290 316 354
		372 394 396 431 459 467 480

Πίνακας 2-1 Όλοι οι γνωστοί κανόνες Golomb χωρίς τις αντίστοιχες εικόνες καθρέπτη

2.2 Εφαρμογές κανόνων Golomb

Οι κανόνες Golomb χρησιμοποιούνται σε ευρύ φάσμα εφαρμογών πολλών τομέων. Αναφέρονται χαρακτηριστικά οι εξής [8],[9],[10],[11]:

Ράδιο-Τηλεπικοινωνίες: Η πρώτη αναφορά του χάρακα Golomb, όχι όμως με αυτό το όνομα, έγινε σε μία δημοσίευση του W. Babcock πάνω σε τρίτης και πέμπτης τάξης παραμόρφωση ενδοδιαμόρφωσης γειτονικών καναλιών σε μία περιοχή ραδιοσυχνότητας. Ο Babcock ανακάλυψε ότι τοποθετώντας κάθε κανάλι μέσα σ' ένα φάσμα συχνοτήτων σε διαστήματα που αντιστοιχούσαν σε δείκτες ενός χάρακα Golomb, θόρυβοι τρίτης τάξης απαλείφονται, ενώ πέμπτης τάξης μειώνονται πάρα πολύ. Ακόμα και σήμερα η ανάθεση καναλιών στο ηλεκτρομαγνητικό φάσμα γίνεται με αυτή τη μέθοδο.

Κρυσταλλογραφία με ακτίνες X: Η ανάλυση με ακτίνες X κρυσταλλικών δομών είναι μερικές φορές ασαφής γιατί δύο διαφορετικές κρυσταλλικές δομές δημιουργούν όμοια πρότυπα παράθλασης ακτινών X. Οι κανόνες Golomb χρησιμοποιούνται για τη μοντελοποίηση του πλέγματος της δομής έτσι ώστε να επιλυθεί αυτή η ασάφεια, με βάση την εμπειρική παρατήρηση ότι δύο κανόνες Golomb διαφορετικού μήκους δεν έχουν το ίδιο σύνολο διαφορών.

Θεωρία Κωδίκων: Οι κανόνες Golomb έχουν χρησιμοποιηθεί στη θεωρία κωδίκων για την εύρεση και τη διόρθωση λαθών. Σύνολα κανόνων χρησιμοποιούνται για να δημιουργούν αυτοορθογωνικούς κώδικες που παίζουν σημαντικό ρόλο στις τηλεπικοινωνίες.

Αυτοί οι κώδικες χρησιμοποιούν αυτό που ονομάζουμε σύνολο τριγώνου διαφορών (Σ.Τ.Δ.). Ένα Σ.Τ.Δ. αποτελείται από μία ομάδα κανόνων, όπου όλοι έχουν τον ίδιο αριθμό δεικτών. Ένα Σ.Τ.Δ. έχει την πρόσθετη ιδιότητα, ότι το τρίγωνο διαφορών για κάθε μέλος του συνόλου περιέχει τιμές μοναδικές για το σύνολο αυτό. Για παράδειγμα ένα Σ.Τ.Δ. φαίνεται στο Σχήμα 2-5 για το σύνολο κανόνων $\{\{0,6,11,13\},\{0,8,17,18\},\{0,3,5,19\}\}$.

Οι κανόνες που αποτελούν Σ.Τ.Δ. χρησιμοποιούνται για ισοτιμία διανυσμάτων κωδικοποίησης, όπου υπάρχει αντιστοιχία των bits με τη θέση των δεικτών στους κανόνες. Δύο διανύσματα δεν μπορούν να έχουν το σύνολο των bits

τους σε ίδιες θέσεις, γιατί εάν συνέβαινε αυτό θα σήμαινε ότι τα διανύσματα περιέχουν μία κοινή διαφορά.

6	11	13	8	17	18	3	15	19
5	7		9	10		12	16	
2				1		4		

Σχήμα 2-2 Τρίγωνο διαφορών για το σύνολο $\{0,6,11,13\}$, $\{0,8,17,18\}$, $\{0,3,15,19\}$

Διανύσματα Αισθητήρων: Τα διάνυσμα αισθητήρων (sensor arrays) παίζουν σημαντικό ρόλο σε μία ευρεία περιοχή εφαρμογών, και περισσότερο στις τηλεπικοινωνίες και στη ραδιοαστρονομία.

Στην ραδιοαστρονομία οι κανόνες Golomb χρησιμοποιούνται στον εντοπισμό μακρινών εκπομπών ραδιοσυχνοτήτων, ουσιαστικά για να προσδιοριστεί η γωνία μεταξύ του εισερχόμενου μετώπου κυμάτων και της γραμμής βάσης των κεραιών. Οι κεραίες είναι όλες ρυθμισμένες να δέχονται την ίδια συχνότητα και ο ακριβής χρόνος λήψης του εισερχόμενου σήματος να μπορεί να μετρηθεί και να συγκριθεί μεταξύ όλων των ζευγαριών των κεραιών. Η κατεύθυνση του εισερχόμενου σήματος μπορεί να προσδιορισθεί αναλύοντας τη διαφορά φάσης μεταξύ ζευγαριών κεραιών. Ο μέγιστος αριθμός διαφορών φάσης μπορεί να επιτευχθεί τοποθετώντας τις κεραίες σε θέσεις σύμφωνα με τους δείκτες των κανόνων Golomb. Αν χρησιμοποιούμε βέλτιστους ή σχεδόν-βέλτιστους κανόνες Golomb, το συνολικό μήκος του διανύσματος κεραιών είναι ελαχιστοποιημένο.

Άλλη μία εφαρμογή του κανόνα Golomb, στη ραδιοαστρονομία, είναι στην επιλογή συχνοτήτων πιθανής λήψης (scanning). Ο σκοπός αυτής της εφαρμογής δεν είναι τόσο να εντοπιστεί η μακρινή πηγή, όσο ο προσδιορισμός της ίδιας της κεραίας. Χρησιμοποιώντας ένα απλό ζευγάρι κεραιών και ανιχνεύοντας ένα σύνολο συχνοτήτων από ένα απομακρυσμένο σημείο πηγής, η θέση των στοιχείων της κεραίας μπορεί να προσδιορισθεί με ακρίβεια εκατοστών σε διηπειρωτικές αποστάσεις. Η ανάθεση των ανιχνευτικών συχνοτήτων σε τιμές που αντιστοιχούν στους δείκτες του κανόνα Golomb, εγγυάται τη μέγιστη ακρίβεια για ελάχιστο αριθμό ανιχνευτικών συχνοτήτων.

Pulse Phase Modulation(PPM) Τηλεπικοινωνίες: Σε Pulse Phase Modulation τηλεπικοινωνίες μία σειριακή είσοδος είναι διαιρημένη σε μία σειρά από

ισομήκη πλαίσια. Κάθε πλαίσιο είναι μία ομάδα από slots. Μέσα σε κάθε πλαίσιο μπορεί να τοποθετηθεί ένας παλμός και να καταλαμβάνει τη θέση ενός slot. Η θέση του παλμού μέσα στο πλαίσιο δημιουργεί την πληροφορία.

Όπως μπορούμε να συμπεράνουμε, κατάλληλος συγχρονισμός είναι απαραίτητος για σίγουρη, χωρίς λάθη επικοινωνία. Συγχρονισμός των πλαισίων επιτυγχάνεται συγκρίνοντας τη ροή εισόδου με μία αποθηκευμένη σειρά και δημιουργώντας ένα συντελεστή συσχέτισης βασισμένο πάνω στο άθροισμα των συγκρινόμενων εισόδων.

Μία επιθυμητή Pulse Phase Modulation σειρά μεγιστοποιεί αυτό τον συντελεστή συσχέτισης αν τα πλαίσια είναι συγχρονισμένα και ελαχιστοποιεί τη συσχέτιση για κάθε μετατοπισμένη σειρά. Οι Gargliardi, Robbins και Taylor αναφέρουν ότι η μέγιστη συσχέτιση μπορεί να επιτευχθεί αν δεν υπάρχει χώρος μεταξύ κάθε ζευγαριού των εισερχόμενων παλμών και να επαναλαμβάνεται για κάθε ζευγάρι. Αυτή είναι η πρώτη απαίτηση για ένα κανόνα Golomb. Οι Gargliardi, Robbins και Taylor προσπάθησαν να δείξουν πως η χρήση βέλτιστων και σχεδόν βέλτιστων κανόνων Golomb μεγιστοποιεί το συντελεστή συσχέτισης, αφού ελαχιστοποιείται το μήκος της σειράς.

Fiber Optic Microcellular system Composite Triple Beat (CTB) reduction: Το CTB που εμφανίζεται κατά την διαμόρφωση της διόδου laser μειώνει την απόδοση του micro-cellular συστήματος οπτικών ινών, καθώς δε μπορεί να φιλτραριστεί. Το CTB αυξάνεται στα μεσαία κανάλια όπου οι αποστάσεις μεταξύ των φορέων είναι ίδιες. Μείωση του CTB μπορεί να επιτευχθεί εάν τοποθετηθούν οι φορείς σύμφωνα με κάποιο κανόνα Golomb, κάνοντας έτσι τις αποστάσεις των γειτονικών φορέων να διαφέρουν.

2.3 Αλγόριθμοι για τον υπολογισμό Βέλτιστων Κανόνων Golomb

Από την επιστημονική κοινότητα έχουν προταθεί οι παρακάτω αλγόριθμοι για τον υπολογισμό των βέλτιστων κανόνων Golomb:

Scientific American Algorithm: Παρουσιάστηκε από την Scientific American. Αποτελείται από δύο μέρη, μία συνάρτηση που δημιουργεί τον κανόνα και μία συνάρτηση που αποδεικνύει ότι ο κανόνας αυτός είναι κανόνας Golomb. Είναι ο

πρώτος αλγόριθμος που προτάθηκε για τον υπολογισμό κανόνων Golomb, ενώ όλοι οι υπόλοιποι αποτελούν εξέλιξη αυτού.

Token Passing Algorithm: Δημιουργήθηκε από τον καθηγητή Απόστολο Δόλλα στο πανεπιστήμιο Duke. Αποτελείται από δύο μέρη. Το πρώτο δημιουργεί μία σειρά κανόνων, ενώ το δεύτερο μέρος ελέγχει αν καθένας από αυτούς είναι κανόνας Golomb. Σε αντίθεση με τους υπολοίπους αλγόριθμους, δε προσπαθεί να δημιουργήσει νέο κανόνα προσθέτοντας νέους δείκτες στον ήδη υπάρχοντα κανόνα. Δηλαδή, χειρίζεται ταυτόχρονα όλους τους δείκτες που έχουν ορισθεί, ελέγχοντας όλες τις πιθανές θέσεις των δεικτών, μέχρι να βρεθεί μία που να τηρεί τα κριτήρια για κάποιο κανόνα Golomb. Αυτός ο αλγόριθμος είναι αδημοσίευτος.

Tree Algorithm: Ο Tree Algorithm προσπαθεί να αποφύγει το πρόβλημα ορισμού πάνω ορίου στο μήκος του κανόνα. Ο χώρος αναζήτησης παρουσιάζεται ως μία γενική δομή δένδρου, με κόμβους να αντιπροσωπεύουν τους δείκτες και οι ακμές τις αποστάσεις. Αφού βρεθεί ένας έγκυρος δείκτης, φορτώνεται σ' έναν κόμβο σε κατάλληλο βάθος. Το πλεονέκτημα αυτού του αλγορίθμου είναι η γρήγορη επεξεργασία του και ότι δεν χρειάζεται να οριστεί από πριν το ελάχιστο μήκος του κανόνα και ο μέγιστος χώρος αναζήτησης. Το μειονέκτημά του είναι ότι το μέγεθος της δομής δεδομένων για το δένδρο αυξάνεται όσο αυξάνεται ο χώρος αναζήτησης, ο οποίος αυξάνεται γεωμετρικά με κάθε επιπλέον δείκτη.

Feiri - Levet Enhanced GARSP Engine (FLEGE): Ο αλγόριθμος αυτός δημιουργήθηκε από τους Didier Levet και Michael Feiri. Στηρίζεται στη χρήση μεγάλης βάσης δεδομένων για τη μείωση του χώρου αναζήτησης. Ουσιαστικά μειώνονται αισθητά οι κόμβοι που πρέπει να ελεγχτούν, έως και δέκα φορές.

Shift Algorithm: Αναπτύχθηκε από τον καθ. Α. Δόλλα και τους φοιτητές του στο πανεπιστήμιο Duke ενώ παρόμοιος αλγόριθμος αναπτύχθηκε ανεξάρτητα από τον ερευνητή Oljan Sibert χωρίς όμως να δημοσιευτεί [12],[13],[14],[15]. Είναι ο αλγόριθμος που έχει αποδειχτεί ότι είναι ο πιο αποδοτικός ως προς τον χρόνο αναζήτησης για βέλτιστους κανόνες Golomb και αποδεικνύεται ως ο πιο κατάλληλος για παράλληλες εφαρμογές.

Για τους παραπάνω λόγους ο **Shift Algorithm** επιλέχθηκε για την υλοποίηση της μηχανής GE3. Ακολουθεί αναλυτικότερη παρουσίαση του αλγορίθμου.

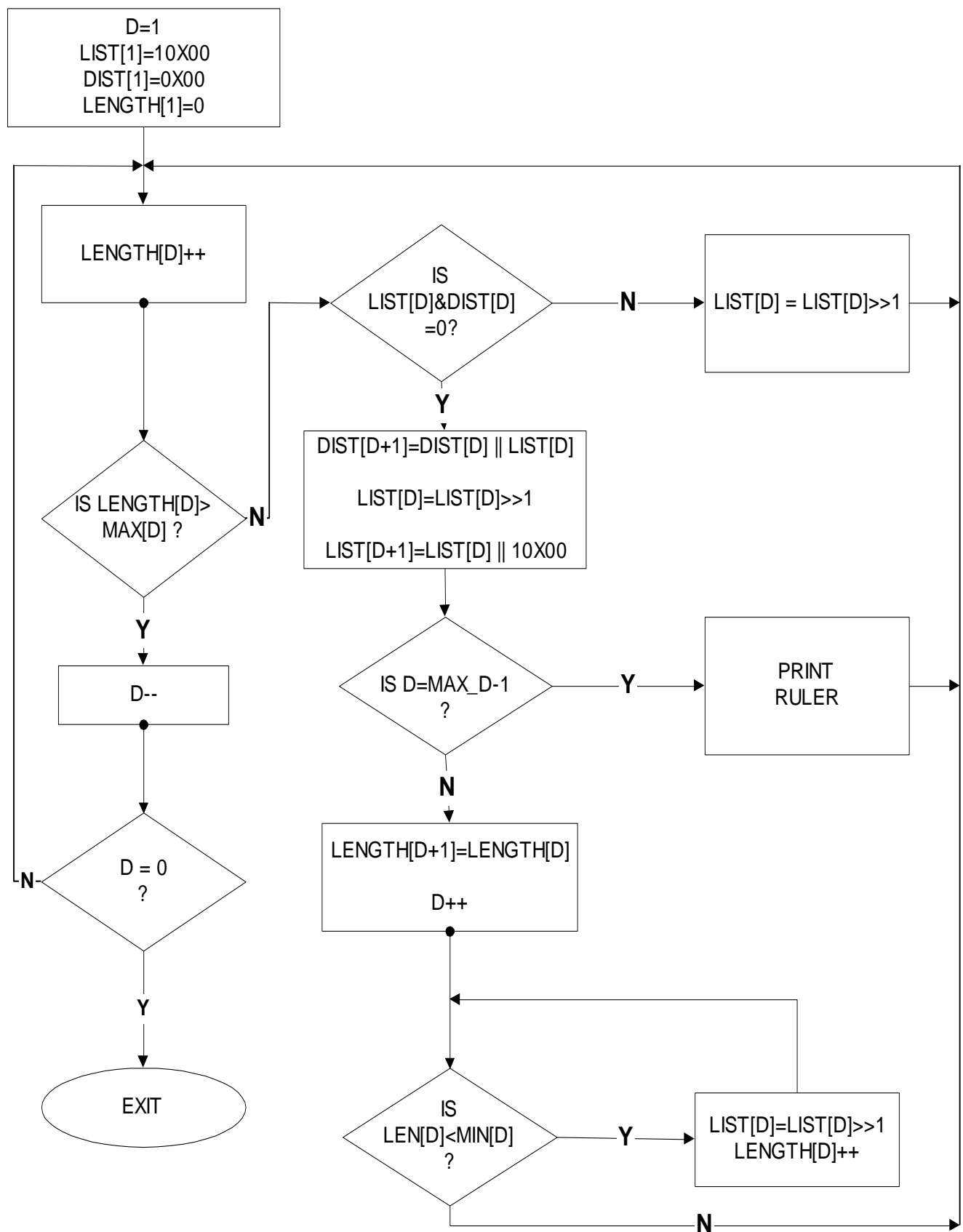
2.4 Shift Algorithm

Ο **Shift Algorithm** (Σχήμα 2-7) δημιουργήθηκε από τον David McCracken, σαν βελτίωση άλλων, αδημοσίευτων αλγορίθμων του A. Δόλλα και ήταν ένα μέρος ενός προπτυχιακού ερευνητικού προγράμματος στο πανεπιστήμιο του Duke [15].

Από προηγούμενες μελέτες έχει αποδειχτεί ότι ο πλέον αποδοτικός αλγόριθμος ως προς τον χρόνο αναζήτησης για βέλτιστους κανόνες Golomb είναι ο **Shift Algorithm**. Η κεντρική ιδέα του αλγορίθμου είναι ίδια με αυτή του Scientific American Algorithm, αλλά με βελτιώσεις στο πεδίο ελέγχου του κανόνα. Η καλύτερη απόδοση του Shift Algorithm οφείλεται στο ότι αν αποθηκευτεί ο πίνακας των αποστάσεων μεταξύ των υπολογισμών και προστεθεί ένας νέος δείκτης, τότε μόνο οι νέες αποστάσεις που δημιουργούνται απ' αυτόν τον δείκτη πρέπει να προστεθούν και να ελεγχθεί ξανά ο πίνακας. Έτσι αντί να πρέπει να υπολογίζονται $N(N-1)/2$ αποστάσεις, τώρα υπολογίζονται μόνο οι N νέες αποστάσεις που δημιουργούνται από το νέο δείκτη. Ακόμα ο έλεγχος για σύγκρουση ή μη των αποστάσεων που μετράει ο κανόνας μειώνεται, καθώς χρειάζεται λιγότερους υπολογισμούς (γίνεται με μια πράξη AND).

Πρόκειται για έναν αλγόριθμο depth-first search, ο οποίος εφαρμόζεται σε διανύσματα δυαδικών αριθμών με μήκος ίσο με το μέγιστο μήκος του σχεδόν βέλτιστου κανόνα Golomb που υπολογίζεται. Οι δομές που χρησιμοποιούνται για δεδομένα από τον αλγόριθμο είναι το διάνυσμα LIST το οποίο περιέχει την αποτύπωση των δεικτών του κανόνα σε bit (σε όποιο σημείο υπάρχει δείκτης η τιμή του αντίστοιχου bit είναι 1 διαφορετικά είναι 0) και το διάνυσμα DIST το οποίο αποτυπώνει το σύνολο των αποστάσεων που μετρούνται από τους δείκτες όπως αυτοί είναι τοποθετημένοι στο διάνυσμα LIST. Όπως φαίνεται στο Σχήμα 2-7 για να προστεθεί ένας καινούργιος δείκτης στο διάνυσμα LIST, το διάνυσμα αυτό πρέπει να ολισθήσει (πράξη SHIFT) έως ότου οι αποστάσεις σ' αυτό το διάνυσμα να μην επικαλύπτονται με αποστάσεις που έχουν ήδη μετρηθεί στο διάνυσμα DIST. Αυτός ο έλεγχος ολοκληρώνεται εύκολα, υλοποιώντας απλά την πράξη AND ανά ζεύγος Bit (bitwise) στα δύο διανύσματα, LIST και DIST. Εάν οι τιμές των bits της λογικής πράξης AND δεν είναι όλες 0 το διάνυσμα LIST πρέπει να ολισθήσει και να

επαναληφτεί ο έλεγχος. Στην αντίθετη περίπτωση (δηλαδή είναι όλες 0) ο νέος δείκτης μπορεί να προστεθεί πλέον στην αρχή του διανύσματος LIST δημιουργώντας έναν εν δυνάμει υποκανόνα του κανόνα Golomb που υπολογίζεται. Το νέο διάνυσμα DIST υπολογίζεται με μια πράξη OR ανά ζεύγος Bit (bitwise) μεταξύ των LIST και DIST. Ταυτόχρονα με την προσθήκη του νέου δείκτη αποθηκεύεται η προηγούμενη κατάσταση των διανυσμάτων, σε μια στοίβα (πράξη PUSH), ώστε αν αποδειχτεί ότι ο υποκανόνας που δημιουργήθηκε δεν είναι υποκανόνας του κανόνα Golomb που υπολογίζεται, να συνεχίσει ο αλγόριθμος την αναζήτηση από το σημείο εκείνο (πράξη POP). Αν το διάνυσμα LIST ολισθήσει παραπάνω από συγκεκριμένα όρια που προκύπτουν για κάθε κανόνα Golomb τότε διαπιστώνεται ότι ο συγκεκριμένος υποκανόνας δεν είναι κομμάτι του κανόνα Golomb που υπολογίζεται και ο αλγόριθμος συνεχίζει από το σημείο που προστέθηκε ο τελευταίος δείκτης (πράξη POP).



Σχήμα 2-7 Αλγόριθμος Shift για υπολογισμό κανόνων Golomb

Για να βελτιωθεί η απόδοση του αλγορίθμου έχουν εισαχθεί κάποιες μέθοδοι μείωσης του χώρου αναζήτησης (search space reduction) βασισμένες στις μαθηματικές ιδιότητες των κανόνων Golomb [12],[13]:

1. First Mark Preclusion: Η τεχνική αυτή στηρίζεται στο γεγονός ότι εάν ένας κανόνας έχει τοποθετημένο το δεύτερο δείκτη στην δεύτερη θέση (μετρώντας μία απόσταση μιας μονάδας), τότε δεν μπορεί να υπάρχει δείκτης στην πρώτη θέση στον αντίστοιχο του κανόνα καθρέπτη. Για αυτό μπορούμε να ξεκινήσουμε βάζοντας το δεύτερο δείκτη στην τρίτη θέση. Έτσι, αν δε βρεθεί ένας κανόνας, θα βρεθεί ο κανόνας καθρέπτη αυτού του κανόνα.

2. Midpoint Reduction: Η μέθοδος αυτή στηρίζεται στο γεγονός ότι μία εικόνα καθρέπτη ενός κανόνα μετράει τις ίδιες αποστάσεις με τον κανονικό κανόνα. Άρα μπορεί κανείς να περιορίσει τη θέση του μεσαίου δείκτη του κανόνα από τη μία μεριά του κανόνα, αφού είτε ο κανόνας είτε ο κανόνας καθρέπτη αυτού πρέπει να έχει το μεσαίο δείκτη πριν το γεωμετρικό κέντρο του κανόνα. Επισημαίνεται ότι η μέθοδος αυτή δεν μπορεί να χρησιμοποιηθεί ταυτόχρονα με τη μέθοδο First Mark Preclusion.

3. Maximum Position Reduction: Αυτή τη μέθοδος στηρίζεται στο γεγονός, ότι για κάθε δείκτη στον κανόνα, οι υπόλοιποι δείκτες (μεγαλύτεροι) δεν είναι δυνατόν να τοποθετηθούν σε χώρο μικρότερο από τον αντίστοιχο χώρο που απαιτείται για ένα βέλτιστο κανόνα Golomb με τον ίδιο αριθμό δεικτών. Η μέγιστη θέση για κάθε δείκτη δεν μπορεί να είναι μεγαλύτερη από το μέγιστο μήκος του κανόνα που αναζητείται, μείον το μήκος του κανόνα Golomb για τον αριθμό των δεικτών, που δεν έχουν τοποθετηθεί ακόμα, μείον ένα. Ακόμα η τεχνική βελτίωσης αυτή μπορεί να συνδυαστεί με τη 2^η (Midpoint Reduction), δηλαδή η μέγιστη θέση για κάθε δείκτη δεν μπορεί να είναι μεγαλύτερη από το κεντρικό σημείο, μείον το μήκος του κανόνα Golomb του μεσαίου δείκτη μείον τον αριθμό του δείκτη που τοποθετείται μειωμένο κατά ένα.

4. Minimum Ruler Preclusion: Η τελευταία μέθοδος που περιγράφεται στηρίζεται στο ότι κάθε κανόνας με ένα αριθμό δεικτών, μετράει συγκεκριμένες αποστάσεις. Κρατώντας μία δομή με όλες τις πληροφορίες που προκύπτουν από την

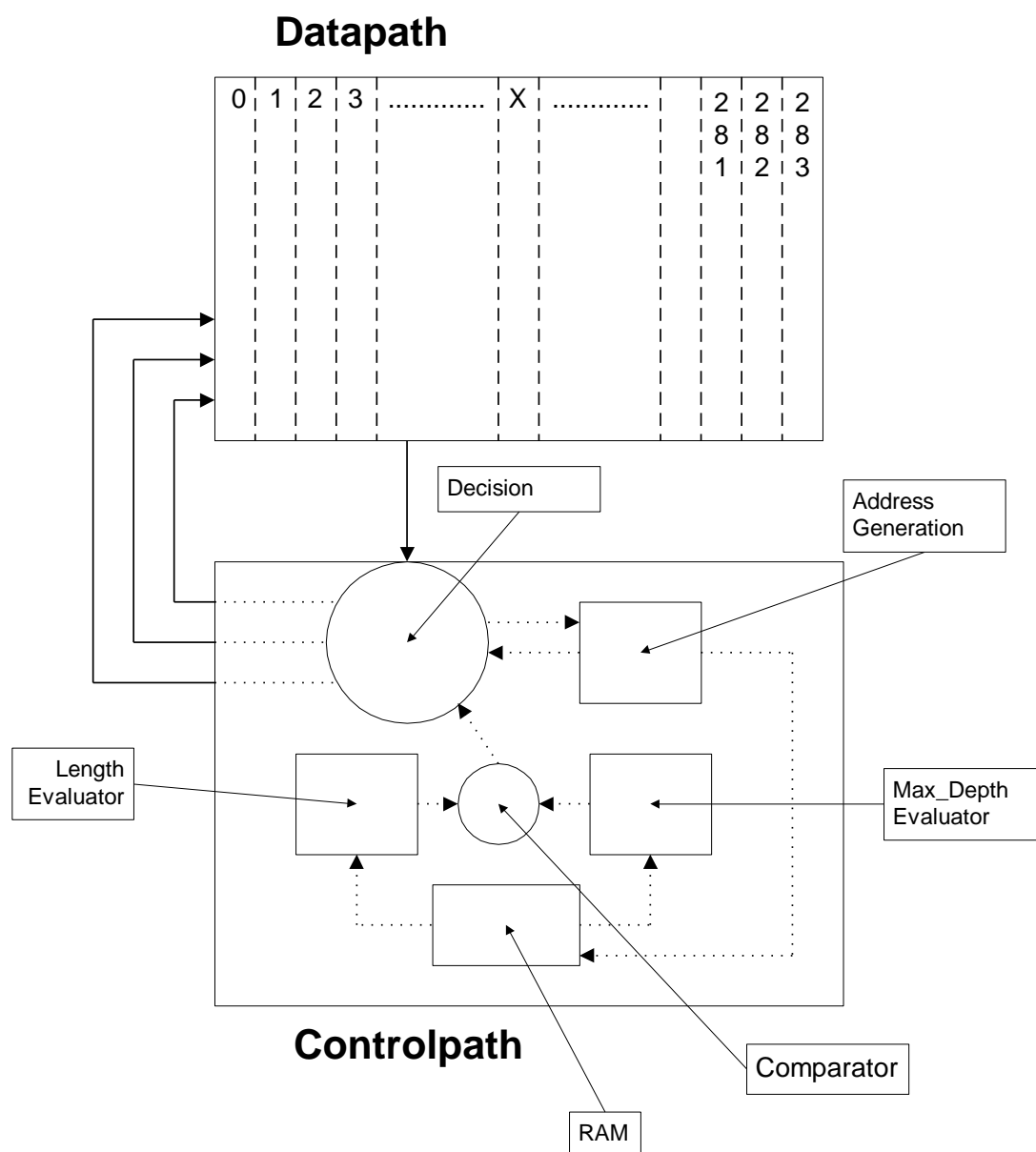
παραπάνω ιδιότητα μειώνεται το εύρος τιμών που μπορεί να τοποθετηθεί ένας δείκτης άρα και ο χώρος αναζήτησης.

Από τις παραπάνω τεχνικές μείωσης του χώρου αναζήτησης, στη σχεδίαση της GE3 χρησιμοποιούνται οι 2,3 και 4.

2.5 Προηγούμενες σχεδιάσεις Golomb Engine στο Πολυτεχνείο Κρήτης

2.5.1 Αρχιτεκτονική GE1

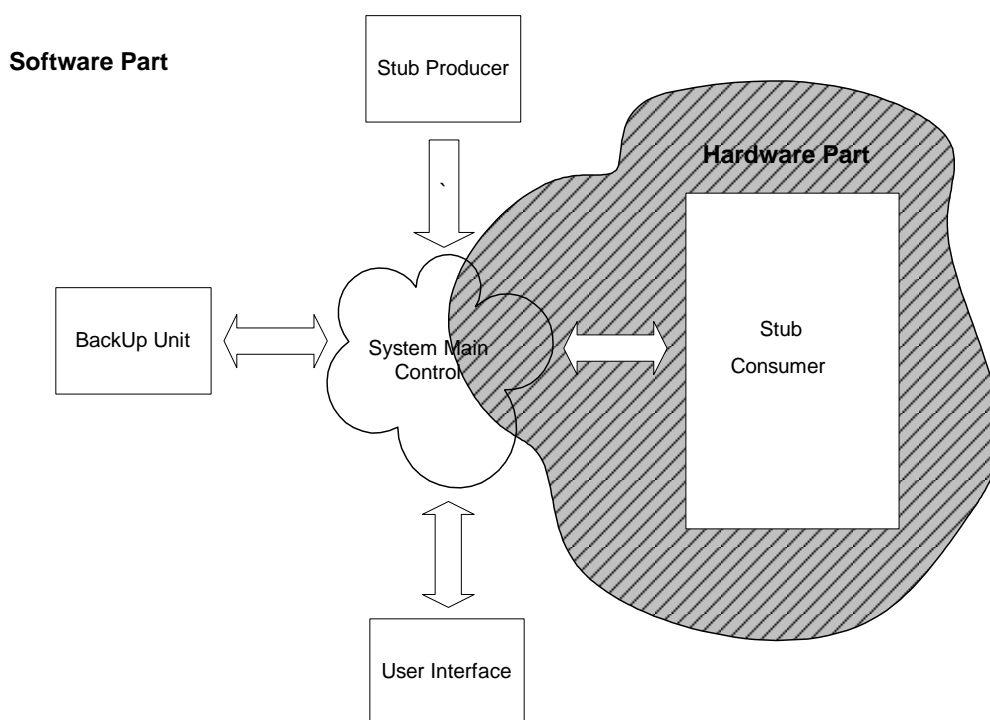
Μια προσπάθεια για την κατασκευή μιας μηχανής υπολογισμού βέλτιστων κανόνων Golomb ήταν η σχεδίαση της GE1. Η σχεδίαση έγινε στο Πολυτεχνείο Κρήτης από τους Α. Εμμανουηλίδη και Ε. Σωτηριάδη υπό την επίβλεψη του Καθηγητή Α. Δόλλα [13],[16]. Συνολική λειτουργία της σχεδίασης δεν υπήρξε αλλά αποδείχτηκε με συντηρητικούς υπολογισμούς ότι ήταν 27 φορές πιο γρήγορη από έναν αντίστοιχο, σε τεχνολογικούς όρους, υπολογιστή γενικής χρήσης.



Σχήμα 2-8 Σχηματικό διάγραμμα γενικής αρχιτεκτονικής GE1

2.5.2 Αρχιτεκτονική GE2

Μια ακόμα προσπάθεια για την κατασκευή μιας μηχανής υπολογισμού βέλτιστων κανόνων Golomb ήταν η σχεδίαση της GE2. Η σχεδίαση έγινε στο Πολυτεχνείο Κρήτης από τον Ε. Σωτηριάδη υπό την επίβλεψη του Καθηγητή Α. Δόλλα [12],[30]. Η μηχανή GE2 αποτελεί συσχέδιαση υλικού λογισμικού [18],[19],[20] για τον υπολογισμό βέλτιστων κανόνων Golomb, στόχος της οποίας είναι ο παραλληλισμός και ταχύτερη λύση του προβλήματος. Σύμφωνα με αυτή την προσέγγιση ο χώρος αναζήτησης τέμνεται με την παραγωγή stubs (software), δηλαδή με την τοποθέτηση των πρώτων n δεικτών ενός κανόνα, προϋπολογίζοντας έτσι υποκανόνες Golomb του ζητούμενου κανόνα. Υπάρχουν πολλές ίδιες μηχανές GE που εξετάζουν (καταναλώνουν) τα stubs (hardware) μέχρι να εξαντληθεί ο χώρος αναζήτησης. Η GE2 ήταν περίπου 20 φορές πιο γρήγορη από έναν αντίστοιχο, σε τεχνολογικούς όρους, υπολογιστή γενικής χρήσης.



Σχήμα 2-9 Γενική Αρχιτεκτονική Σχεδίασης GE2 με Διαχωρισμό των υποσυστημάτων σε κομμάτια Υλικού και λογισμικού

2.6 Υπολογισμός Κανόνων Golomb Μέσω Διαδικτύου

Μια μέθοδος για την επίλυση προβλημάτων με πολύ μεγάλες απαιτήσεις σε υπολογιστική ισχύ είναι η χρήση πολλών υπολογιστών μέσω διαδικτύου (Massive Internet Search). Η διάτμηση δηλαδή του προβλήματος σε πολλά μικρότερα, και η επιμέρους επίλυση του από πολλούς υπολογιστές σε διάφορα μέρη του κόσμου. Η διασύνδεση όλων των υπολογιστών με κάποιο κέντρο αναφοράς γίνεται μέσω του διαδικτύου. Έτσι προβλήματα με μεγάλες υπολογιστικές απαιτήσεις που μπορούν να τμηθούν εύκολα και να λυθούν ανεξάρτητα τα επιμέρους κομμάτια τους προσφέρονται για επίλυση με τη μέθοδο αυτή. Βασική προϋπόθεση, για να υπάρχει πραγματικό υπολογιστικό όφελος, είναι να μην στοιχίζει πολύ υπολογιστικά η διάτμηση του προβλήματος, ο καταμερισμός δηλαδή της εργασίας στους διάφορους υπολογιστές, καθώς και η ανασύνθεση των λύσεων σε μία ενιαία λύση (μικρό overhead).

Το πρόβλημα υπολογισμού των κανόνων Golomb συγκεντρώνει τα παραπάνω χαρακτηριστικά και μια τέτοια προσπάθεια ξεκίνησε από τους Mark Garry, David Vanderschel και αρκετούς εθελοντές το 1998. Η προσπάθεια αυτή οδήγησε στον υπολογισμό των κανόνων Golomb 20 έως και 22. Ο σχετικός δικτυακός τόπος (site) είναι <http://members.aol.com/golomb20/>. Για τον κανόνα Golomb 23 και πάνω η προσπάθεια μεταφέρθηκε στον οργανισμό *distributed.net* ο οποίος έχει ως σκοπό τη χρήση υπολογιστικής ισχύος μέσω διαδικτύου για την επίλυση προβλημάτων με μεγάλο υπολογιστικό φόρτο. Η ομάδα των εθελοντών αποτελείται από δεκάδες χιλιάδες μέλη τα οποία συμμετέχουν στην επίλυση όποιου προβλήματος επιθυμεί ο καθένας.

Για τον υπολογισμό κανόνων Golomb η προσπάθεια αυτή βασιζόταν στον αλγόριθμο Shift, ενώ τώρα βασίζεται στον αλγόριθμο FLEGE. Έχει αναπτυχθεί μία εφαρμογή client - server που επιλύει κανόνες Golomb. Ο server παράγει υποκανόνες Golomb (stubs) και με αυτούς τροφοδοτεί τους υπολογιστές - clients. Ο υπολογιστής - client επιλύει το πρόβλημα διαπιστώνοντας αν ο συγκεκριμένος υποκανόνας είναι μέρος του αναζητούμενου κανόνα Golomb ή όχι. Ο υπολογιστής - client αποστέλλει στη συνέχεια το αποτέλεσμα στον server ο οποίος είναι υπεύθυνος για την διασφάλιση του αποτελέσματος. Να διαπιστώσει δηλαδή ότι ο χώρος αναζήτησης εξαντλείται χωρίς καμία απολύτως εξαίρεση. Με αυτόν τον τρόπο ελέγχεται ο χώρος αναζήτησης (search space) κομμάτι-κομμάτι ανάλογα με τον υποκανόνα που παίρνει ο κάθε υπολογιστής με κεντρική ευθύνη του Server. Για την διευκόλυνση των

χρηστών παγκοσμίως υπάρχουν αρκετοί διαφορετικοί servers σε διάφορα σημεία του κόσμου όπου κατά την ίδια λογική ο κάθε server εξυπηρετεί ένα κομμάτι του χώρου αναζήτησης και οι οποίοι υπόκεινται με την σειρά τους σε κεντρικό έλεγχο. Μ' αυτόν τον τρόπο μπορούν να επιλύουν το πρόβλημα πάρα πολλοί υπολογιστές ταυτόχρονα αρκεί να έχει προϋπολογισθεί για τον καθένα διαφορετικός αρχικός υποκανόνας.

Το βασικό πλεονέκτημα ενός υπολογιστή γενικής χρήσης στο συγκεκριμένο πρόβλημα είναι η τεράστια ποσότητα πληροφορίας την οποία μπορεί να διαχειριστεί για την μείωση του χώρου αναζήτησης, σε αντίθεση με μια υπολογιστική μηχανή ειδικής χρήσης βασισμένη σε FPGAs. Αυτό το πλεονέκτημα σε συνδυασμό με τον μεγάλο αριθμό προσφερόμενων υπολογιστών για τέτοιου τύπου εφαρμογές οδήγησαν στον υπολογισμό των κανόνων Golomb 20 έως και 25. Ο οργανισμός *distributed.net* έχει αποδείξει τους κανόνες από 23 έως 25. Ο κανόνας Golomb 25 αποδείχτηκε μετά από 8 χρόνια υπολογισμών, στις 22-9-2008.

3. Αρχιτεκτονική μηχανής GE3

Στόχος της μηχανής GE3 είναι η αποτύπωση του αλγορίθμου shift σε υλικό. Στο κεφάλαιο αυτό αναλύεται η αρχιτεκτονική της μηχανής GE3, με ποιο τρόπο επιτυγχάνεται ο παραλληλισμός του προβλήματος, καθώς και μια βελτίωση σε σχέση με τη μηχανή GE2, η οποία αφορά στην υποστήριξη πολλαπλών shift. Τέλος αναλύεται η υλοποίηση σε λογισμικό της δυνατότητας αποθήκευσης κατά τακτά χρονικά διαστήματα του σημείου στο οποίο βρίσκεται η αναζήτηση, ώστε να μη χάνεται πολύτιμος χρόνος αναζήτησης σε περιπτώσεις αστοχιών.

Οι διαφορές της GE3 σε σχέση με τη GE2 είναι οι ακόλουθες:

- Υλοποίηση υποστήριξης πολλαπλών shift για τη βελτίωση της απόδοσης
- Στην αρχιτεκτονική παραλληλισμού και η παράγωγή των stubs και η κατανάλωση τους, γίνεται σε υλικό

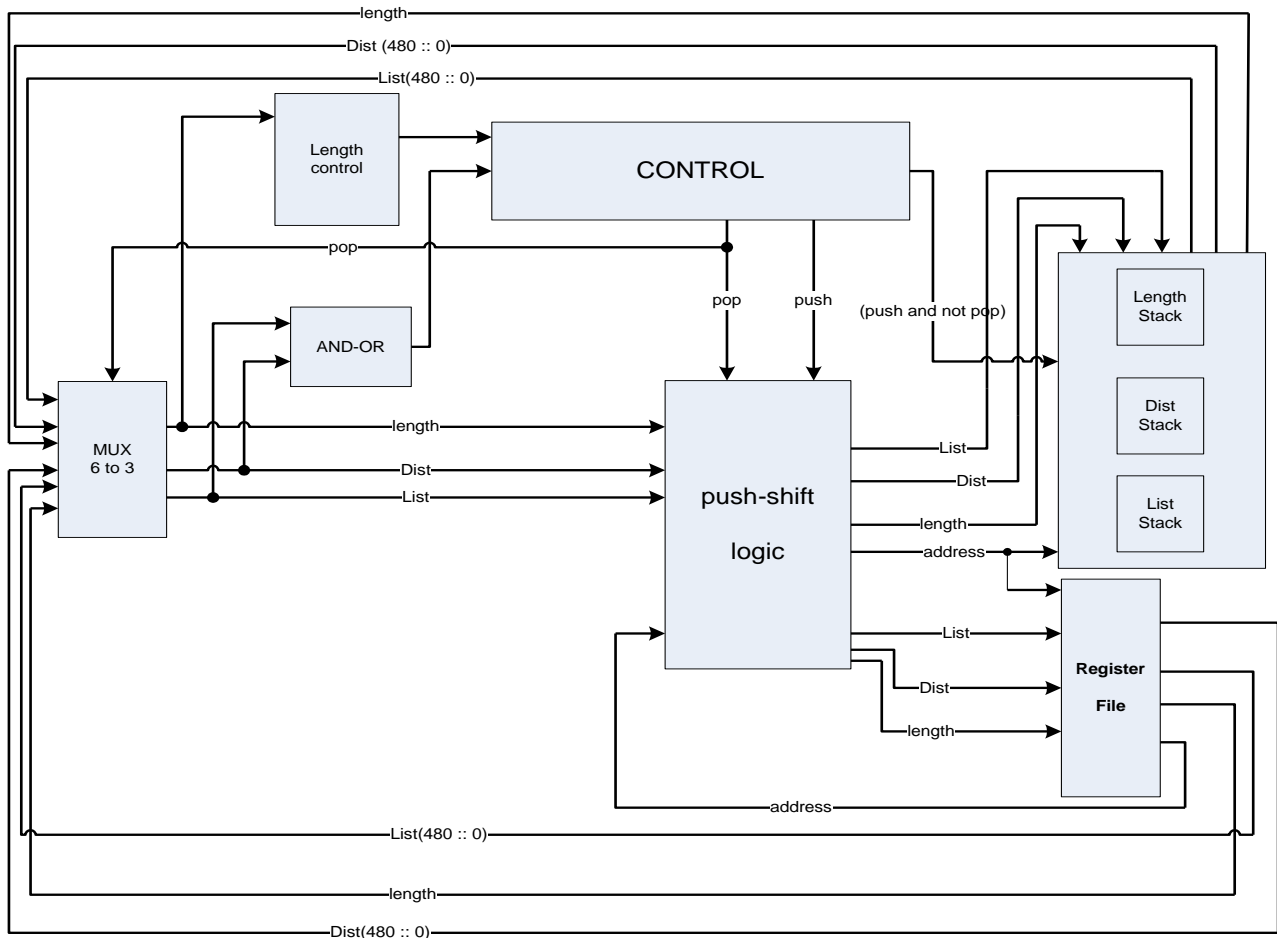
3.1 Ανάλυση αρχιτεκτονικής της GE3

Η αρχιτεκτονική της GE3 στηρίζεται σε δύο διανύσματα. Στο διάνυσμα List, το οποίο περιέχει αποτύπωση των δεικτών του κανόνα σε bit (σε όποιο σημείο υπάρχει δείκτης η τιμή του αντίστοιχου bit είναι 1 διαφορετικά είναι 0) και το διάνυσμα Dist το οποίο αποτυπώνει το σύνολο των αποστάσεων που μετρούνται από τους δείκτες όπως αυτοί είναι τοποθετημένοι στο διάνυσμα List. Τα διανύσματα αυτά έχουν μέγεθος ίσο με το μέγεθος του σχεδόν βέλτιστου κανόνα Golomb που θέλουμε να υπολογίσουμε. Η αρχιτεκτονική της GE3 αποτελείται από 5 κυρία τμήματα όπως φαίνεται στο Σχήμα 3-1:

- 1) Το σύστημα AND-OR που ελέγχει αν υπάρχει σύγκρουση μεταξύ των List και Dist.
- 2) Η μονάδα έλεγχου για το μέγεθος του διανύσματος List (Length Control), στην οποία υλοποιούνται οι βελτιστοποιήσεις του αλγορίθμου shift, οι οποίες δείχνουν μέχρι πιο σημείο μπορεί να φτάσει ο κάθε δείκτης για τον κανόνα που θέλουμε να αποδείξουμε.
- 3) Η γενική μονάδα ελέγχου, η οποία ελέγχοντας τα αποτελέσματα από το σύστημα AND-OR και τη μονάδα Length Control αποφασίζει ποια θα είναι η επόμενη εντολή που θα εκτελεστεί.
- 4) Η στοίβα (Stack) στην οποία αποθηκεύονται οι προηγούμενες καταστάσεις των διανυσμάτων List και Dist ώστε αν αποδειχτεί ότι ο υποκανόνας που δημιουργήθηκε

δεν είναι υποκανόνας του κανόνα Golomb που υπολογίζεται, να συνεχίσει ο αλγόριθμος την αναζήτηση από το σημείο εκείνο (εντολή POP).

5) Το σύστημα το οποίο περιέχει την απαραίτητη λογική για την υλοποίηση των εντολών Push και Shift (push– shift logic).



Σχήμα 3-1 Γενική Αρχιτεκτονική Σχεδίασης GE3

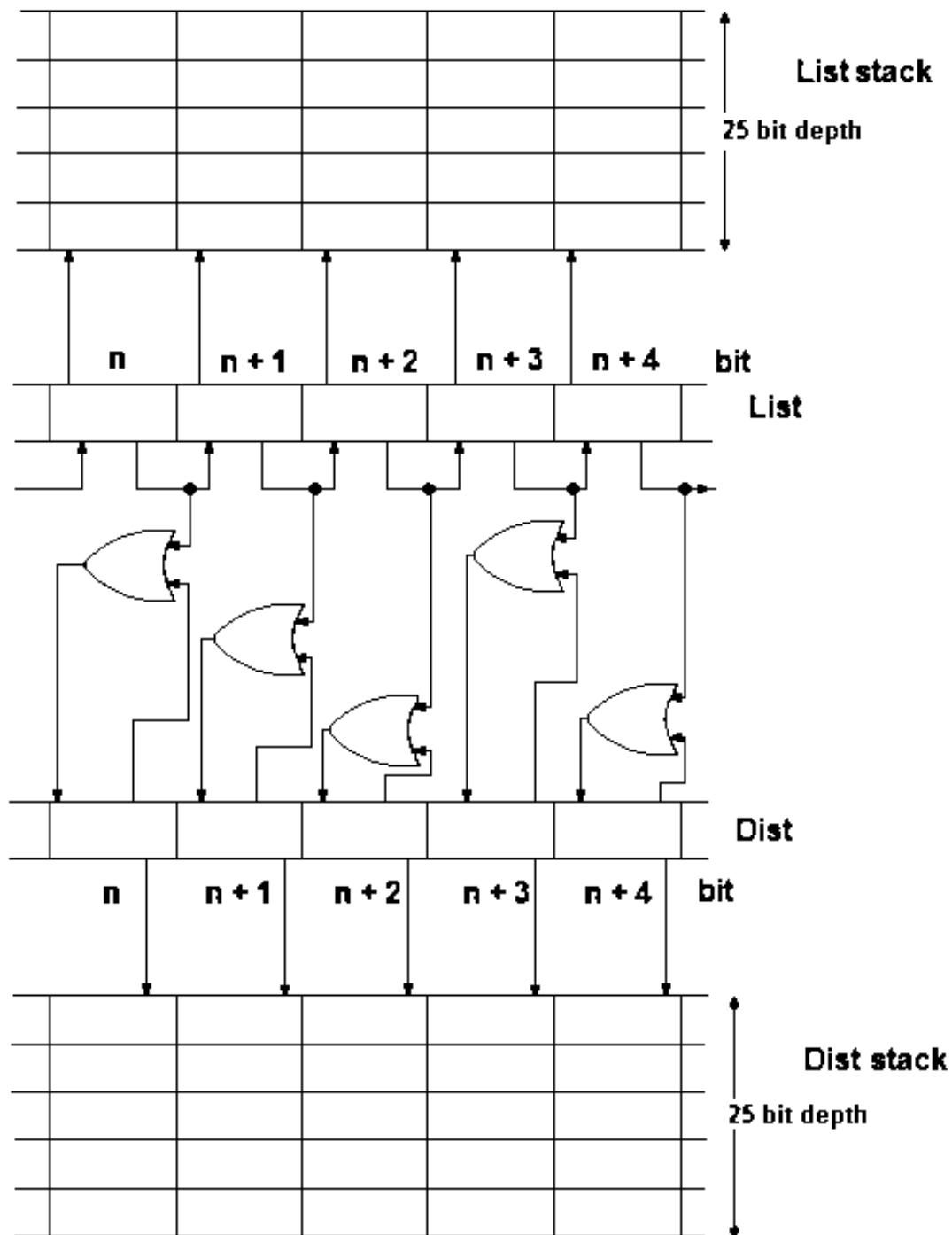
Ακολουθεί αναλυτικότερη περιγραφή των επιμέρους τμημάτων ανάλογα την εντολή που υλοποιούν.

1. Datapath

Η GE3 υποστηρίζει 3 εντολές, τις Push, Shift και Pop.

Η εντολή **Push** αντιστοιχεί στην εισαγωγή ενός ακόμα mark στο διάνυσμα List. Το νέο διάνυσμα List προκύπτει από τη δεξιά ολίσθησή του κατά μια θέση και την εισαγωγή ενός '1' στη θέση του MSB. Για την υλοποίηση αυτής της εντολής χρησιμοποιείται ένας right shifter για τη δεξιά ολίσθηση, ενώ για την εισαγωγή του '1' στη θέση του MSB το διάνυσμα List, αφού έχει ολισθήσει, περνάει από μια πύλη

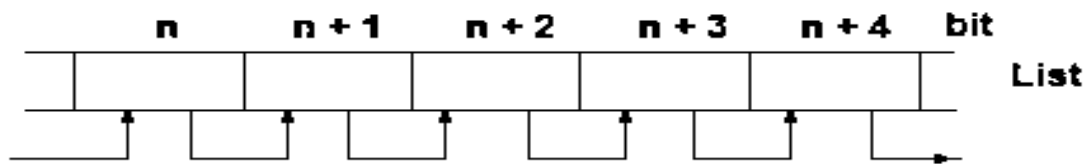
OR η οποία σαν δεύτερη είσοδο έχει ένα διάνυσμα ίδιου μεγέθους το οποίο έχει έναν '1' στη θέση του MSB του. Ακόμα το μέγεθος του List αυξάνεται κατά 1. Το νέο διάνυσμα Dist προκύπτει από μια λογική πράξη OR μεταξύ του νέου List και του Dist. Ακόμα κατά την εκτέλεση της εντολής Push η προηγούμενη κατάσταση των διανυσμάτων List (μετά τη δεξιά ολίσθηση) και Dist αποθηκεύεται σε μια στοίβα, ώστε αν αποδειχτεί ότι ο υποκανόνας που δημιουργήθηκε δεν είναι υποκανόνας του κανόνα Golomb που υπολογίζεται, να συνεχίσει ο αλγόριθμος την αναζήτηση από το σημείο εκείνο (εντολή POP). Η στοίβα αυτή είναι RAM based. Υλοποιείται με μία μνήμη Dual Port Ram για κάθε διάνυσμα, με μήκος όσο με το μήκος των διανυσμάτων και βάθος όσο ο αριθμός των marks του κανόνα που ψάχνουμε (το βάθος, λόγω της τεχνολογίας, πρέπει να είναι δύναμη του 2). Ακόμα αποθηκεύεται το αυξημένο κατά 1 μέγεθος του διανύσματος List για να ξέρουμε το μέγεθος του όταν θα πρέπει να συνεχίσουμε από το σημείο αυτό. Οι μνήμες είναι Dual Port Ram, γιατί χρειαζόμαστε δύο θύρες, τη μια για εγγραφή κατά την εντολή Push και την άλλη για ανάγνωση κατά την εντολή POP, αφού χρησιμοποιούμε διαφορετικές διευθύνσεις στις δύο περιπτώσεις. Η διεύθυνση εγγραφής είναι ίση με τη διεύθυνση ανάγνωσης μειωμένη κατά 1. Τέλος κατά την εκτέλεση της εντολής Push οι διευθύνσεις εγγραφής και ανάγνωσης αυξάνονται κατά 1. Τα παραπάνω παρουσιάζονται στο Σχήμα 3-2.



Σχήμα 3-2 Υλοποίηση εντολής Push

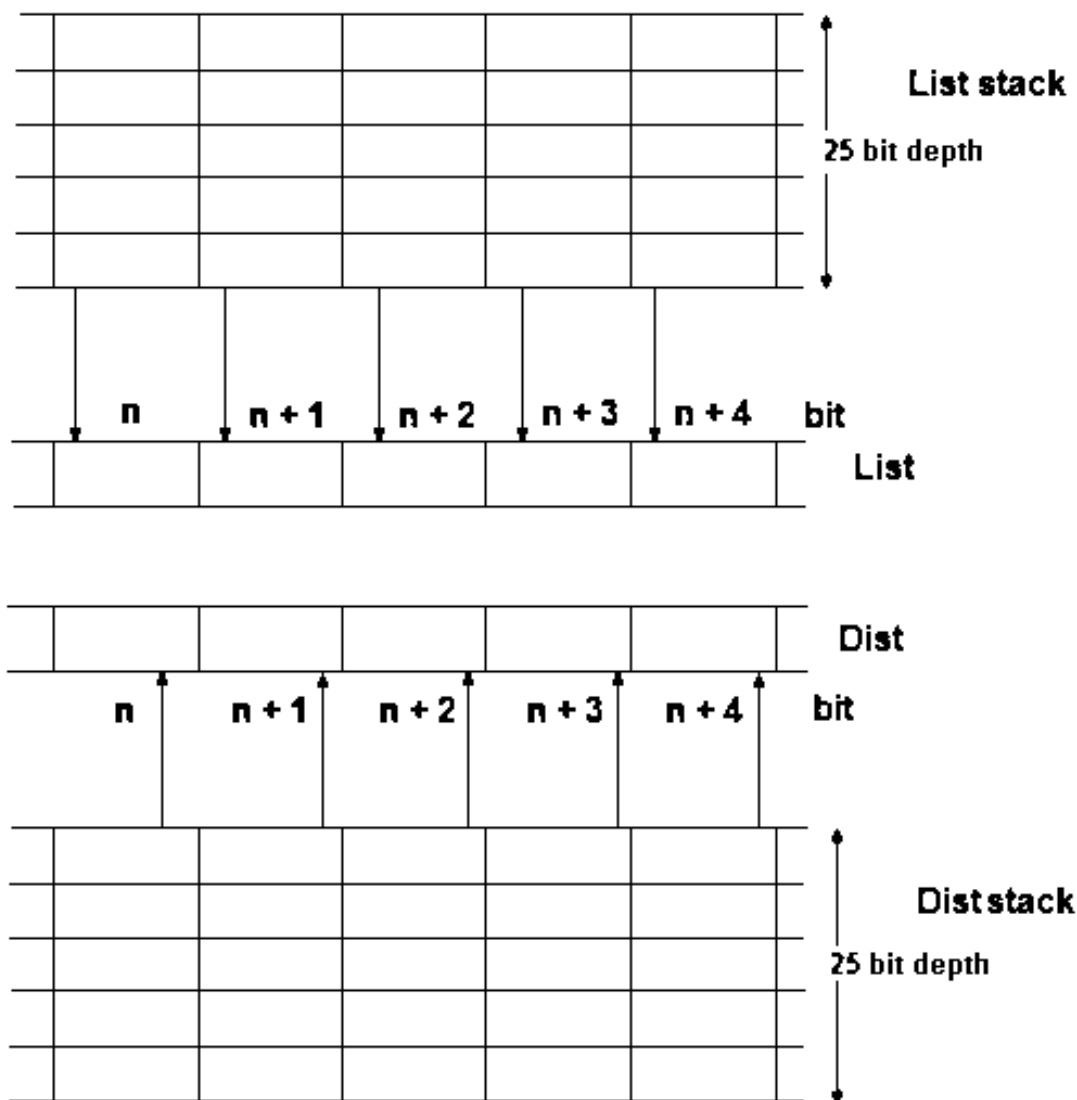
Η εντολή **Shift** αντιστοιχεί στη δεξιά ολίσθηση του διανύσματος List σε περίπτωση που δε μπορεί να εισαχτεί δείκτης στο σημείο εκείνο, λόγω του ότι οι αποστάσεις σ' αυτό το διάνυσμα επικαλύπτονται με αποστάσεις που έχουν ήδη μετρηθεί στο διάνυσμα Dist. Για την υλοποίηση αυτής της εντολής χρησιμοποιείται ένας right shifter για τη δεξιά ολίσθηση του διανύσματος List, ενώ το διάνυσμα Dist

παραμένει ίδιο. Ακόμα το μέγεθος του List αυξάνεται κατά 1. Τα παραπάνω παρουσιάζονται στο Σχήμα 3-3.



Σχήμα 3-3 Υλοποίηση εντολής Shift

Η εντολή **Pop** αντιστοιχεί στην ανάγνωση από τη μνήμη της προηγούμενης αποθηκευμένης κατάστασης των διανυσμάτων List και Dist, και του μεγέθους του List, στην περίπτωση που ο τελευταίος δείκτης που έχει τοποθετηθεί στο List έχει περάσει το άνω όριο στο οποίο μπορεί να φτάσει (σύμφωνα με τους κανόνες βελτίωσης του shift algorithm που περιγράφονται παραπάνω). Δηλαδή επιλέγονται μέσω ενός πολυπλέκτη τα νέα List, Dist και το μέγεθος του List οι αντίστοιχες έξοδοι των μνημών, Dual Port Ram, που υλοποιούν τη στοίβα. Τέλος κατά την εκτέλεση της εντολής Pop οι διευθύνσεις εγγραφής και ανάγνωσης μειώνονται κατά 1. Τα παραπάνω παρουσιάζονται στο Σχήμα 3-4.



Σχήμα 3-4 Υλοποίηση εντολής Pop

Η αύξηση και μείωση της διεύθυνσης ανάλογα με την εντολή γίνεται με έναν up – down counter, ενώ η αύξηση του μεγέθους του διανύσματος List γίνεται με έναν up counter.

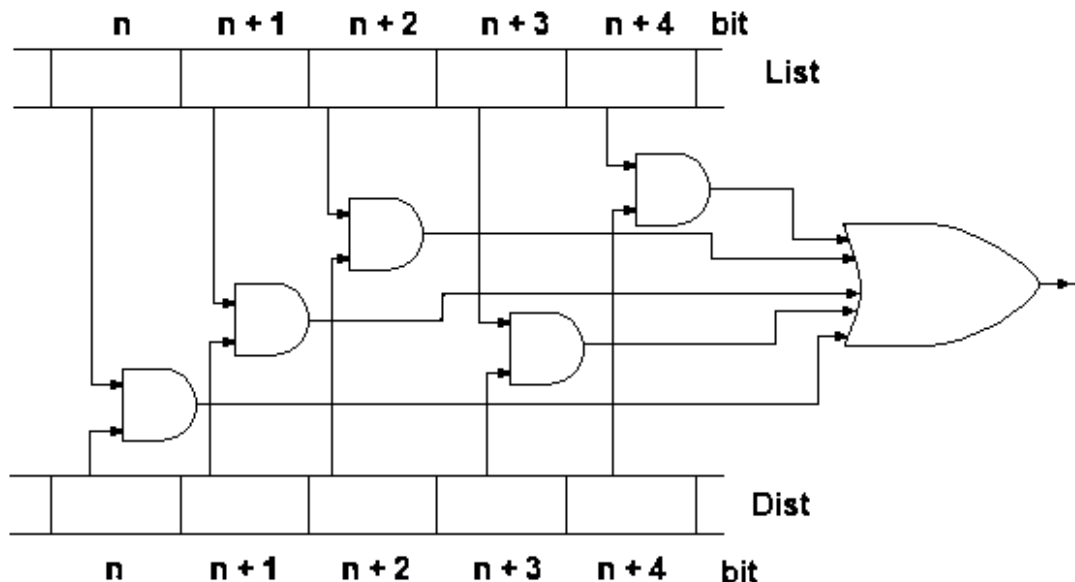
Τέλος, στην αρχιτεκτονική έχει προστεθεί ένας καταχωρητής στον οποίο αποθηκεύεται το αποτέλεσμα, εάν βρεθεί, δηλαδή όταν τοποθετηθεί και ο τελευταίος δείκτης του κανόνα Golomb που ψάχνουμε.

2. Controlpath

Η επιλογή της εντολής που εκτελείται κάθε κύκλο ορίζεται από τη μονάδα ελέγχου.

Η GE3 έχει CPI 1, δηλαδή εκτελείται μία από τις τρεις εντολές που περιγράφηκαν παραπάνω σε κάθε κύκλο ρολογιού. Η μονάδα ελέγχου της GE3 είναι μία FSM που αποτελείται από τρεις καταστάσεις που αντιστοιχούν στις εντολές Push, Shift και Pop, τις Pushstate, Shiftstate και Popstate αντίστοιχα.

Η μονάδα ελέγχου παίρνει σαν είσοδο το αποτέλεσμα της λογικής πράξης AND μεταξύ των διανυσμάτων List και Dist (σύστημα AND-OR), καθώς επίσης το μέγεθος του List και τον αριθμό των δεικτών που έχουν εισαχτεί σε αυτό, ο οποίος αντιστοιχεί στη διεύθυνση εγγραφής στη μνήμη (stack). Στην περίπτωση που ο τελευταίος δείκτης που έχει τοποθετηθεί στο διάνυσμα List έχει περάσει το άνω όριο στο οποίο επιτρέπεται να φτάσει (Length Control), η επόμενη εντολή που εκτελείται είναι Pop. Στην κατάσταση Popstate ένα σήμα pop γίνεται '1' και οδηγεί έναν πολυπλέκτη ο οποίος επιλέγει τις εξόδους της στοίβας ως νέα διανύσματα List και Dist. Εάν ο τελευταίος δείκτης που έχει τοποθετηθεί στο διάνυσμα List δεν έχει περάσει το άνω όριο στο οποίο επιτρέπεται να φτάσει, ελέγχεται το αποτέλεσμα του συστήματος AND-OR (Σχήμα 3-5). Εάν είναι 0, δηλαδή δεν υπάρχει σύγκρουση μεταξύ των διανυσμάτων List και Dist, η επομένη κατάσταση είναι Pushstate, ενώ στην αντίθετη περίπτωση η επόμενη κατάσταση είναι Shiftstate. Εάν η επόμενη κατάσταση είναι Pushstate ένα σήμα push γίνεται '1' ενώ αν είναι Shiftstate γίνεται '0'. Το σήμα αυτό οδηγεί τρεις πολυπλέκτες, μέσα στη μονάδα push– shift logic, έτσι τα νέα διανύσματα List και Dist παίρνουν τιμές ανάλογα με την εντολή που εκτελείται (Push ή Shift). Επίσης το σήμα push αντιστοιχεί και στο σήμα write enable (WE) των μνημών που υλοποιούν τη στοίβα δίνοντας την εντολή για εγγραφή της προηγούμενης κατάστασης των List και Dist στην κατάσταση Pushstate.



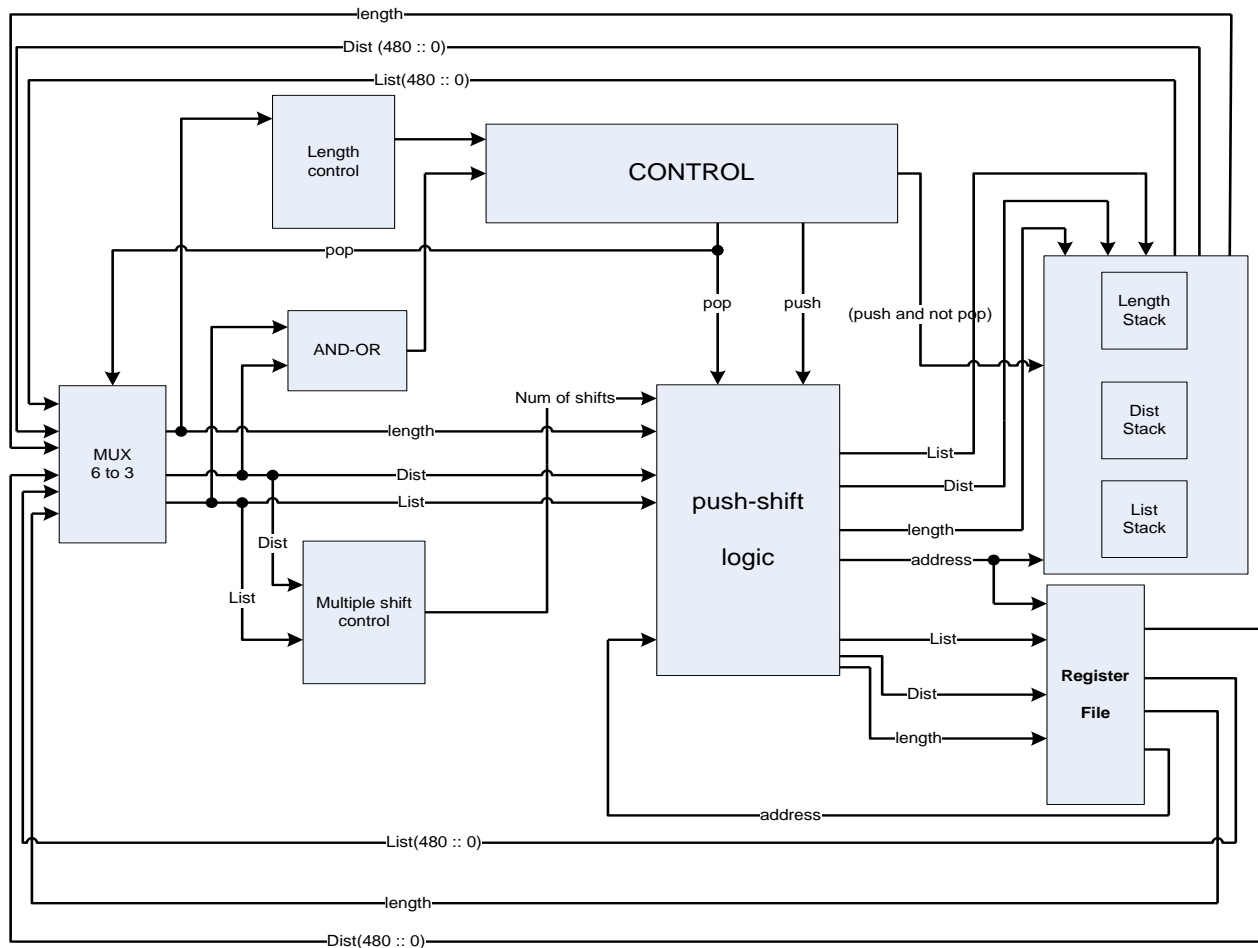
Σχήμα 3-5 Υλοποίηση συστήματος AND-OR

3.2 Βελτίωση της GE3 σε σχέση με τη GE2 – πολλαπλά shift

Η βελτίωση της GE3 σε σχέση με τη GE2 που αναλύεται σε αυτό το κεφάλαιο είναι η υποστήριξη **πολλαπλών shift**.

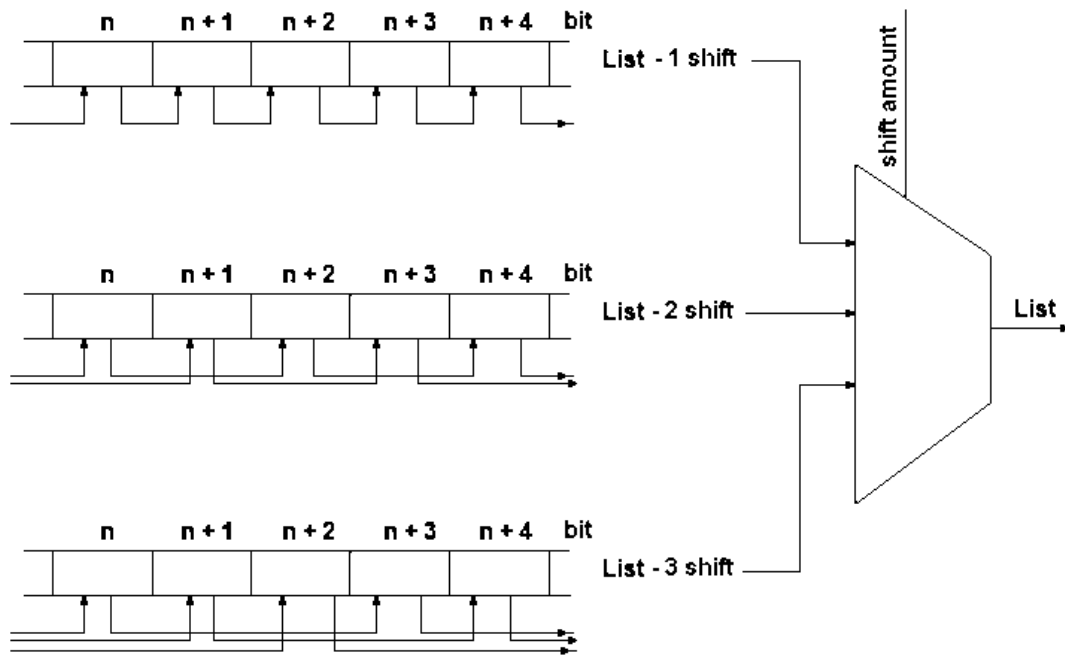
Πριν την εισαγωγή στη GE3 του απαραίτητου υλικού για να υποστηρίζει πολλαπλά shift έγιναν προσομοιώσεις σε λογισμικό. Οι μετρήσεις έδειξαν, για τον κανόνα Golomb 25 ότι περίπου το 98% των εντολών είναι shift. Ακόμα έγιναν μετρήσεις στο λογισμικό και για διαφορές περιπτώσεις πολλαπλών shift οι οποίες έδειξαν ποσό αποδοτική είναι η υλοποίηση για τη GE3. Για shift 1 – 16 θέσεων οι μετρήσεις έδειξαν ότι πάνω από το 60% των shift ήταν 16 θέσεων η μεγαλύτερα.

Έτσι υλοποιήθηκε, για τη GE3, ένα σύστημα ελέγχου (multiple shift control) που αποφασίζει, σε περίπτωση εκτέλεσης εντολής shift, πόσο δεξιά θα ολισθήσει το διάνυσμα List, από 1 - 16 θέσεις (Σχήμα 3-6). Το σύστημα αυτό ουσιαστικά αποτελείται από 16 συστήματα AND-OR, 1 για κάθε περίπτωση shift.



Σχήμα 3-6 Γενική Αρχιτεκτονική Σχεδίασης GE3 η οποία υποστηρίζει πολλαπλά shift

Το σύστημα ελέγχου multiple shift control ψάχνει μετά από πόσα δεξιά shift, εξετάζοντας από το μικρότερο προς το μεγαλύτερο, δεν επικαλύπτονται αποστάσεις που μετράει το διάνυσμα List με αυτές που έχουν ήδη μετρηθεί στο διάνυσμα Dist και βγάζει ως έξοδο τον αριθμό των δεξιών shift που πρέπει να γίνουν. Η έξοδος του συστήματος αυτού ελέγχει ένα right shifter ο οποίος κάνει shift το διάνυσμα List ανάλογα με την τιμή αυτή (Σχήμα 3-7). Ακόμα ο αριθμός των shift που γίνονται πρέπει να προστεθεί και στο μέγεθος του List. Έτσι ο up counter αντικαθίσταται με ένα αθροιστή, ο οποίος σαν εισόδους παίρνει το προηγούμενο μέγεθος του List και τον αριθμό των shift, και βγάζει στην έξοδο το νέο μέγεθος του List.



Σχήμα 3-7 Υλοποίηση shifter πολλαπλών shift

Με αυτή τη βελτίωση αντί να χρειάζονται 16 κύκλοι ρολογιού για να γίνουν 16 shift, τώρα χρειάζεται μόνο 1 κύκλος. Από έρευνα που έγινε σε λογισμικό, για τον κανόνα Golomb 25, για shift από 1-16 θέσεων παρατηρήθηκε ότι πάνω από το 60% είναι shift 16 θέσεων και πάνω, ενώ για εξυπηρέτηση shift 1-24 θέσεων παρατηρήθηκε ότι το 25% είναι shift 24 θέσεων και πάνω. Επίσης υπολογίζεται ότι το 98% των εντολών είναι shift, για τον κανόνα Golomb 25. Έτσι για shift 1-16 θέσεων έχουμε μια αύξηση περίπου 12 φορές στην ταχύτητα της μηχανής, ενώ για shift 1-24 θέσεων έχουμε μια αύξηση περίπου 14 φορές στην ταχύτητα της μηχανής (συμπεριλαμβανομένης της μικρής μείωσης του ρολογιού 3%). Επειδή η απόδοση που επιτυγχάνεται σε σχέση με τους πόρους που απαιτούνται (σε FPGA) για την εξυπηρέτηση 1-24 shift είναι μικρή, η περίπτωση 1-16 θέσεων shift κρίνεται πιο αποδοτική.

Ακόμα με αυτή τη βελτίωση παρατηρείται πολύ μικρή μείωση του ρολογιού (περίπου 3%), λόγω του επιπλέον επιπέδου ελέγχου. Επίσης παρατηρείται σχεδόν τριπλασιασμός του απαιτούμενου χώρου πάνω στις FPGA, το οποίο θεωρείται μικρό ποσοστό σε σχέση με τη βελτίωση που επιτυγχάνεται στην ταχύτητα.

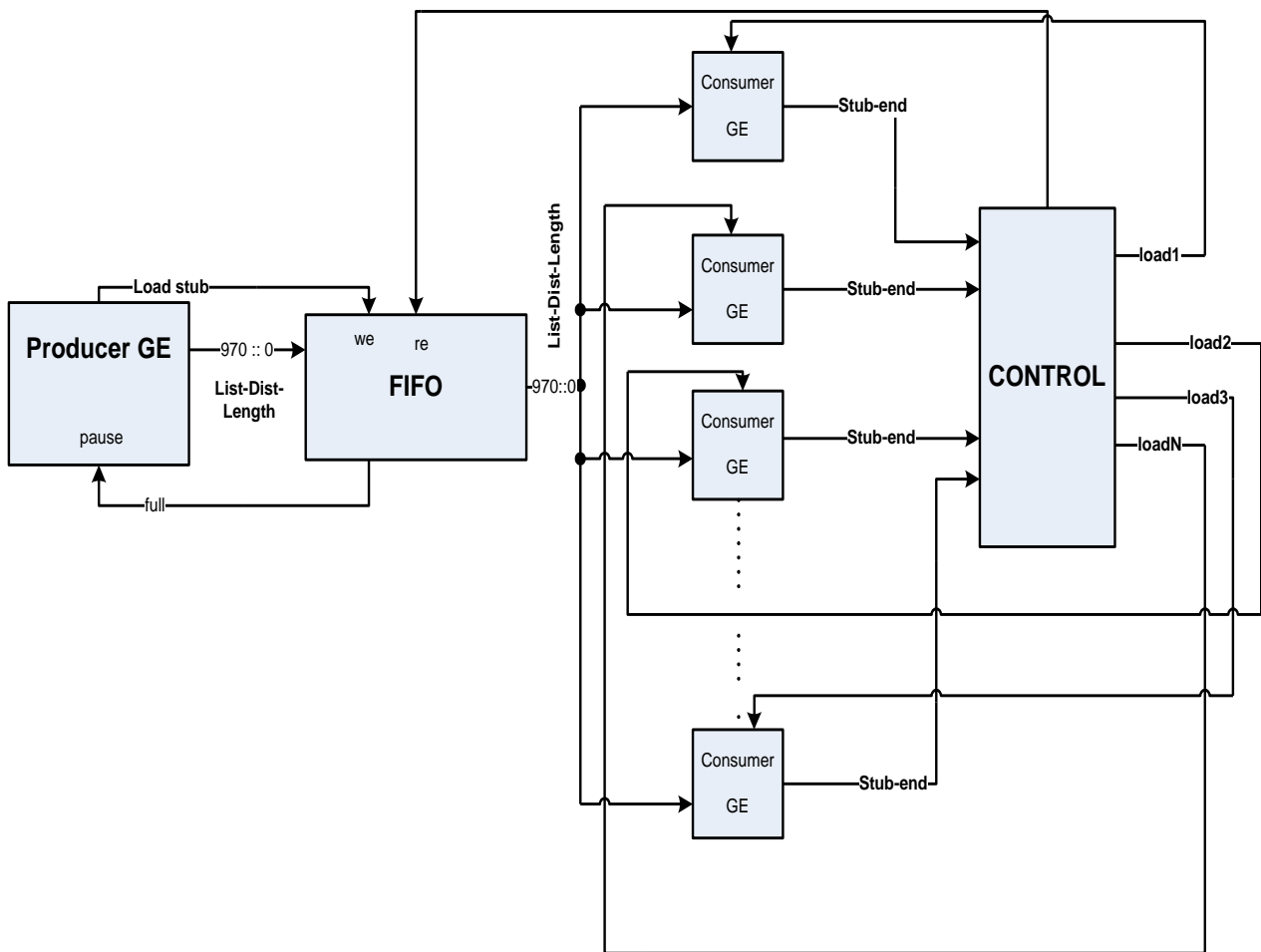
Τα παραπάνω επαληθεύτηκαν σε FPGA Spartan3, XC3S1000, στην οποία για τον κανόνα Golomb 13 χρειάζονται 46 δευτερόλεπτα για την ολοκλήρωση της αναζήτησης χωρίς πολλαπλά shift, ενώ μόλις 11 δευτερόλεπτα με πολλαπλά shift, 1-16 θέσεων. Με την υποστήριξη των πολλαπλών shift η μηχανή γίνεται περίπου 3

φορές πιο γρήγορη. Θα είχαμε ακόμα καλύτερα αποτελέσματα αλλά για τον κανόνα Golomb 13 το ποσοστό των shift 16 θέσεων είναι αμελητέο (0,01%) σε σχέση με τον κανόνα Golomb 25. Το μεγαλύτερο ποσοστό των shift, περίπου 90%, παρατηρείται σε shift 1 – 7 θέσεων για τον κανόνα Golomb 13, ενώ πάνω από 60% είναι 1 – 3 θέσεων.

3.3 Παραλληλισμός Προβλήματος – (Παραγωγή – Κατανάλωση)

Ανάλογα με τη μηχανή GE2, στόχος της GE3 είναι ο παραλληλισμός με τον οποίο επιτυγχάνεται ταχύτερη λύση του προβλήματος εύρεσης Βέλτιστων Κανόνων Golomb. Σύμφωνα με αυτή την προσέγγιση ο χώρος αναζήτησης τέμνεται με την παραγωγή stubs, δηλαδή με την τοποθέτηση των πρώτων n δεικτών ενός κανόνα, προϋπολογίζοντας έτσι υποκανόνες Golomb του ζητούμενου κανόνα. Υπάρχουν πολλές ίδιες μηχανές GE που εξετάζουν (καταναλώνουν) τα stubs μέχρι να εξαντληθεί ο χώρος αναζήτησης. Δηλαδή από το σημείο στο χώρο αναζήτησης που σταματά η μια μηχανή που καταναλώνει κάποιο stub, αρχίζει η επόμενη μηχανή. Η ουσιαστική διαφορά της GE3 από τη GE2 είναι ότι και η παραγωγή και η κατανάλωση των stubs γίνεται σε υλικό (HW).

Η αρχιτεκτονική η οποία υλοποιεί τον παραλληλισμό του προβλήματος φαίνεται στο Σχήμα 3-8. Ακολουθεί αναλυτικότερη περιγραφή της αρχιτεκτονικής αυτής.



Σχήμα 3-8 Γενική Αρχιτεκτονική Σχεδίασης GE3 με παραλληλισμό (παραγωγή - κατανάλωση)

1. Datapath

Παραγωγή - Κατανάλωση

Για την παραγωγή των stubs η δεικτών χρησιμοποιείται μια μηχανή GE, στην οποία τα διανύσματα List και Dist έχουν μέγεθος όσο το άνω όριο μέχρι το οποίο μπορεί να τοποθετηθεί ο n-οστός δείκτης στον κανόνα Golomb που ψάχνουμε. Με αυτό τον τρόπο μειώνεται ο χώρος που καταλαμβάνεται από τη μηχανή παραγωγής.

Τα αποτελέσματα από τη μηχανή παραγωγής, δηλαδή τα stubs, αποθηκεύονται σε μια ουρά (FIFO) η οποία τροφοδοτεί τις μηχανές που τα καταναλώνουν.

Τέλος στις μηχανές που καταναλώνουν τα stubs τα διανύσματα List και Dist έχουν μέγεθος όσο το άνω όριο του κανόνα που ψάχνουμε. Οι μηχανές αυτές δουλεύουν παράλληλα και καταναλώνουν τα stubs που διαβάζουν από τη FIFO. Με

αυτό τον τρόπο ο χώρος αναζήτησης διαιρείται διά όσες μηχανές κατανάλωσης διαθέτουμε.

2. Controlpath

Για το συγχρονισμό της μηχανής παραγωγής stubs με την ουρά έχει υλοποιηθεί μια μηχανή πεπερασμένων καταστάσεων (FSM). Η μηχανή παραγωγής παράγει stubs όσο η ουρά δεν είναι γεμάτη. Αν η ουρά είναι γεμάτη, η παραγωγή πηγαίνει σε κατάσταση παύσης (Pause state) έως ότου αδειάσει κάποια θέση, όταν δηλαδή κάποια μηχανή κατανάλωσης τελειώσει κάποιο stub και διαβάσει νέο από την ουρά. Η παραγωγή τερματίζει όταν έχουν παραχθεί όλα τα stubs για τον κανόνα που ψάχνουμε.

Για το συγχρονισμό των μηχανών κατανάλωσης stubs έχει υλοποιηθεί μια ακόμα μηχανή πεπερασμένων καταστάσεων (FSM). Η FSM αυτή εξασφαλίζει ότι δε θα διαβάσουν δύο μηχανές ταυτόχρονα από την ουρά το ίδιο stub. Έτσι αποφεύγεται το φαινόμενο δύο ή περισσότερες μηχανές να επεξεργάζονται τον ίδιο χώρο αναζήτησης. Αν η ουρά είναι άδεια οι μηχανές κατανάλωσης περιμένουν μέχρι να παραχθεί κάποιο stub (Pause). Αν η ουρά είναι άδεια και η μηχανή παραγωγής έχει τερματίσει, οι μηχανές κατανάλωσης, μόλις τελειώσουν το stub που επεξεργάζονται, τερματίζουν.

3.4 Υλοποίηση δυνατότητας Αποθήκευσης του σημείου που έχουμε φτάσει στο search space

Επειδή το πρόβλημα αυτό χρειάζεται πολύ χρόνο για να ολοκληρωθεί είναι απαραίτητη η αποθήκευση του σημείου μέχρι το οποίο έχει εξεταστεί ο χώρος αναζήτησης, ώστε σε περίπτωση αστοχίας να μην εξεταστεί ξανά ο χώρος αυτός.

Για το λόγο αυτό υλοποιήθηκε κατάλληλο λογισμικό και υλικό για να υποστηρίζεται η δυνατότητα αποθήκευσης του σημείου μέχρι το οποίο έχει ολοκληρωθεί η αναζήτηση ανά τακτά χρονικά διαστήματα.

Στην αρχιτεκτονική προστέθηκε ένας καταχωρητής ο οποίος κρατάει το τελευταίο stub που διαβάστηκε από τη FIFO. Επίσης σε κάθε μηχανή κατανάλωσης έχει προστεθεί ένας καταχωρητής, ο οποίος κρατάει το stub που υπολογίζεται για κάθε μηχανή καθώς και τον κύκλο ρολογιού στον οποίο ξεκίνησε η αναζήτηση για το

συγκεκριμένο stub, ο οποίος δίνεται από ένα 32 bit counter. Έτσι, αποθηκεύοντας το τελευταίο stub που διαβάστηκε από τη FIFO ξέρουμε έως που έχει ελεγχτεί ο χώρος αναζήτησης, πλην τα stubs που υπολογίζονται εκείνη τη στιγμή στις μηχανές κατανάλωσης. Για αυτό το λόγο αποθηκεύουμε και το stub που επεξεργάζεται η κάθε μηχανή κατανάλωσης. Τέλος τον κύκλο ρολογιού στον οποίο ξεκίνησε κάθε stub τον αποθηκεύουμε για να μπορούμε να αναγνωρίζουμε, σε περίπτωση που κάποια μηχανή υπολογίζει για πολύ χρόνο κάποιο stub, ότι η μηχανή έχει αστοχήσει και άρα το συγκεκριμένο stub πρέπει να ξαναυπολογιστεί.

Τα παραπάνω στοιχεία στέλνονται ανά τακτά χρονικά διαστήματα μέσω της σειριακής θήρας σε υπολογιστή γενικής χρήσης όπου και τα αποθηκεύονται. Έτσι σε περίπτωση αστοχίας να μπορούμε να τα ανακαλέσουμε και να συνεχίσουμε την αναζήτηση από το σημείο πριν την αστοχία.

Ακόμα στον υπολογιστή γενικής χρήσης μέσω της ίδιας διεπαφής στέλνονται τα αποτελέσματα όταν βρεθούν από τις μηχανές που καταναλώνουν τα stubs. Δηλαδή όταν βρεθεί ο κανόνας Golomb που ψάχνουμε αποθηκεύεται σαν αποτέλεσμα της αναζήτησης στον υπολογιστή γενικής χρήσης.

4. Αποτύπωση της αρχιτεκτονικής της GE3

Στο κεφάλαιο αυτό παρουσιάζονται διάφορα αποτελέσματα από την αποτύπωση της αρχιτεκτονικής της GE3 σε FPGA. Για την αποτύπωση της GE3 σε FPGA χρησιμοποιήθηκαν τα εργαλεία της Xilinx, ISE 9.1 και EDK9.1, στα οποία έγινε και προσομοίωση του συστήματος.

4.1 Αποτύπωση της GE3 χωρίς παραλληλισμό

Αρχικά η GE3 αποτυπώθηκε και ελέγχθηκε η σωστή λειτουργία της σε πλατφόρμα με FPGA Spartan3, XC3S1000 για τον κανόνα Golomb 4 χωρίς καμία τεχνική βελτίωσης. Όλη η διάρκεια της αναζήτησης (Πίνακας 4-1) ελέγχθηκε και στον ISE simulator και επιβεβαιώθηκε με λογισμικό, το οποίο αποτελεί υλοποίηση του αλγορίθμου shift. Οι πόροι που καταλαμβάνει η μηχανή που επιλύει τον κανόνα Golomb 4 παρουσιάζονται στον Πίνακα 4-2. Η μηχανή αυτή λειτουργεί στα 90 Mhz.

1	Push	1100000	15	Pop	0000110	29	Pop	0001010	43	Push	1000100
2	Shift	0110000	16	Push	1000011	30	Push	1000101	44	Push	1100010
3	Push	1011000	17	Pop	0000011	31	Pop	0000101	45	Shift	0110001
4	Shift	0101100	18	Pop	0100000	32	Pop	0010000	46	Pop	0100010
5	Shift	0010110	19	Push	1010000	33	Push	1001000	47	Push	1010001
6	Shift	0001011	20	Push	1101000	34	Push	1100100	48	Pop	0010001
7	Pop	0011000	21	Shift	0110100	35	Shift	0110010	49	Pop	0000100
8	Push	1001100	22	Shift	0011010	36	Shift	0011001	50	Push	1000010
9	Shift	0100110	23	Shift	0001101	37	Pop	0100100	51	Push	1100001
10	Push	1010011	24	Pop	0101000	38	Push	1010010	52	Pop	0100001
11	Pop	0010011	25	Shift	0010100	39	Shift	0101001	53	Pop	0000010
12	Pop	0001100	26	Push	1001010	40	Pop	0010010	54	Push	1000001
13	Push	1000110	27	Push	1100101	41	Shift	0001001	55	Pop	0000001
14	Shift	0100011	28	Pop	0100101	42	Pop	0001000	56	END	

Πίνακας 4-1 Ακολουθία πράξεων για τον υπολογισμό του κανόνα Golomb 4, μήκους 6, χωρίς καμία τεχνική βελτιστοποίησης

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	49	15,360	1%
Number used as Flip Flops	29		
Number used as Latches	20		
Number of 4 input LUTs	121	15,360	1%
Logic Distribution			
Number of occupied Slices	84	7,680	1%
Number of Slices containing only related logic	84	84	100%
Number of Slices containing unrelated logic	0	84	0%
Total Number of 4 input LUTs	164	15,360	1%
Number used as logic	121		
Number used as a route-thru	1		
Number used for Dual Port RAMs	42		
Number of bonded IOBs	9	173	5%

Πίνακας 4-2 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 4, μήκους 6, χωρίς καμία τεχνική βελτιστοποίησης (Spartan3, XC3S1000)

Τα πειράματα που ακολούθησαν, στην ίδια πλατφόρμα, αφορούν μεγαλύτερους κανόνες Golomb. Η σχεδίαση επιβεβαιώθηκε για κανόνες Golomb από τον κανόνα Golomb 10 δεικτών μέχρι τον κανόνα Golomb 15 δεικτών, με χρήση των τεχνικών βελτιστοποίησης. Για μεγαλύτερους κανόνες ο χρόνος που χρειαζόταν για να τελειώσει η αναζήτηση ήταν απαγορευτικός για μια μηχανή GE3 και την πλατφόρμα με FPGA Spartan3, XC3S1000.

Οι κανόνες Golomb 10 και 11 τερματίζουν σε λιγότερο από μισό δευτερόλεπτο. Ο κανόνας Golomb 12 τερματίζει σε 2 δευτερόλεπτα. Για αυτό το λόγο δεν έγιναν δοκιμές σε αυτούς τους κανόνες με την αρχιτεκτονική που υποστηρίζει πολλαπλά shift. Ενδεικτικά στον Πίνακα 4-3 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή που επιλύει τον κανόνα Golomb 12. Η μηχανή αυτή λειτουργεί στα 55 Mhz.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	367	15,360	2%
Number used as Flip Flops	189		
Number used as Latches	178		
Number of 4 input LUTs	793	15,360	5%
Logic Distribution			
Number of occupied Slices	644	7,680	8%
Number of Slices containing only related logic	644	644	100%
Number of Slices containing unrelated logic	0	644	0%
Total Number of 4 input LUTs	1,152	15,360	7%
Number used as logic	793		
Number used as a route-thru	1		
Number used for Dual Port RAMs	358		
Number of bonded IOBs	9	173	5%

Πίνακας 4-3 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 12, μήκους 85 (Spartan3, XC3S1000)

Ο κανόνας Golomb 13 χρειάζεται 46 δευτερόλεπτα περίπου για να τερματίσει χωρίς την εξυπηρέτηση πολλαπλών shift, ενώ 11 περίπου δευτερόλεπτα με πολλαπλά shift (1-16 θέσεων). Στον Πίνακα 4-4 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή που επιλύει τον κανόνα Golomb 13 χωρίς πολλαπλά shift, η οποία λειτουργεί στα 57 Mhz. Στον Πίνακα 4-5 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή που επιλύει τον ίδιο κανόνα Golomb με πολλαπλά shift, η οποία λειτουργεί στα 49 Mhz. Όπως φαίνεται και από τους πίνακες οι πόροι που καταλαμβάνει η μηχανή που υποστηρίζει πολλαπλά shift είναι τρεις φορές περισσότεροι από τους πόρους που καταλαμβάνει η μηχανή χωρίς πολλαπλά shift. Τέλος ο κανόνας Golomb 13 προσομοιώθηκε και με το σύστημα παράγωγης – κατανάλωσης με μια μηχανή παράγωγης, η οποία τοποθετεί τους πρώτους 8 δείκτες και μια μηχανή κατανάλωσης η οποία τοποθετεί τους υπόλοιπους 5. Η μηχανή αυτή χρειάζεται 60 δευτερόλεπτα περίπου για να τερματίσει χωρίς την εξυπηρέτηση πολλαπλών shift, ενώ 18 περίπου δευτερόλεπτα με πολλαπλά shift. Η αύξηση αυτή στο χρόνο για να ολοκληρωθεί η αναζήτηση είναι λόγω της παρεμβολής της FIFO, η οποία προσθέτει 2 κύκλους για κάθε stub που καταναλώνεται. Εάν η παραγωγή τοποθετεί λιγότερους δείκτες τα stubs που δημιουργούνται είναι λιγότερα και έτσι η αύξηση στο χρόνο είναι

μικρότερη. Για παράδειγμα εάν από την παραγωγή τοποθετούνται μόνο 3 δείκτες και οι υπόλοιποι 10 από την κατανάλωση, χρειάζεται 12 δευτερόλεπτα για να τερματίσει με πολλαπλά shift, ενώ 53 δευτερόλεπτα χωρίς πολλαπλά shift. Οι πόροι που καταλαμβάνει η μηχανή με το σύστημα παραγωγής - κατανάλωσης, η οποία υποστηρίζει πολλαπλά shift, παρουσιάζονται στον Πίνακα 4-6. Η μηχανή αυτή λειτουργεί στα 50 Mhz. Ο διπλασιασμός των πόρων που καταλαμβάνει αυτή η μηχανή σε σχέση με αυτή χωρίς το σύστημα παραγωγής – κατανάλωσης οφείλεται στο ότι αποτελείται από δύο μηχανές GE και στον επιπλέον έλεγχο που απαιτείται για το συγχρονισμό αυτών. Ακόμα οι πόροι σε Block RAM καταλαμβάνονται από τη FIFO που παρεμβάλλεται μεταξύ της παραγωγής και της κατανάλωσης.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	450	15,360	2%
Number used as Flip Flops	230		
Number used as Latches	220		
Number of 4 input LUTs	829	15,360	5%
Logic Distribution			
Number of occupied Slices	726	7,680	9%
Number of Slices containing only related logic	726	726	100%
Number of Slices containing unrelated logic	0	726	0%
Total Number of 4 input LUTs	1,271	15,360	8%
Number used as logic	829		
Number used for Dual Port RAMs	442		
Number of bonded IOBs	9	173	5%

Πίνακας 4-4 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 13, μήκους 106, χωρίς πολλαπλά shift (Spartan3, XC3S1000)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	484	15,360	3%
Number used as Flip Flops	264		
Number used as Latches	220		
Number of 4 input LUTs	3,598	15,360	23%
Logic Distribution			
Number of occupied Slices	2,135	7,680	27%
Number of Slices containing only related logic	2,135	2,135	100%
Number of Slices containing unrelated logic	0	2,135	0%
Total Number of 4 input LUTs	4,043	15,360	26%
Number used as logic	3,598		
Number used as a route-thru	3		
Number used for Dual Port RAMs	442		
Number of bonded IOBs	9	173	5%

Πίνακας 4-5 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 13, μήκους 106, με πολλαπλά shift (Spartan3, XC3S1000)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,288	15,360	8%
Number used as Flip Flops	872		
Number used as Latches	416		
Number of 4 input LUTs	7,220	15,360	47%
Logic Distribution			
Number of occupied Slices	4,556	7,680	59%
Number of Slices containing only related logic	4,556	4,556	100%
Number of Slices containing unrelated logic	0	4,556	0%
Total Number of 4 input LUTs	8,066	15,360	52%
Number used as logic	7,220		
Number used as a route-thru	10		
Number used for Dual Port RAMs	836		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		
Number of Block RAMs	7	24	29%

Πίνακας 4-6 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 13, μήκους 106, με παραγωγή – κατανάλωση και πολλαπλά shift (Spartan3, XC3S1000)

Οι κανόνες Golomb 14 και 15 προσομοιώθηκαν με το σύστημα παραγωγής - κατανάλωσης. Η μηχανή παραγωγής τοποθετεί τους πρώτους 8 δείκτες ενώ η μηχανή κατανάλωσης λύνει τους υπόλοιπους 6 και 7 για τους κανόνες Golomb 14 και 15 αντίστοιχα. Ο κανόνας Golomb 14 χρειάζεται 22 λεπτά περίπου για να τερματίσει χωρίς την εξυπηρέτηση πολλαπλών shift, ενώ λιγότερο από 7 λεπτά με πολλαπλά shift. Ο κανόνας Golomb 15 χρειάζεται περίπου 2 ώρες και 14 λεπτά χωρίς πολλαπλά shift, ενώ 35 λεπτά με πολλαπλά shift. Στους Πινάκες 4-7 και 4-8 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή για τον κανόνα Golomb 14 χωρίς (λειτουργεί στα 59 Mhz) και με πολλαπλά shift (λειτουργεί στα 47 Mhz) αντίστοιχα. Ενώ στους Πινάκες 4-9 και 4-10 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή για τον κανόνα Golomb 15 χωρίς (λειτουργεί στα 56 Mhz) και με πολλαπλά shift (λειτουργεί στα 46 Mhz) αντίστοιχα.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,295	15,360	8%
Number used as Flip Flops	803		
Number used as Latches	492		
Number of 4 input LUTs	2,599	15,360	16%
Logic Distribution			
Number of occupied Slices	2,229	7,680	29%
Number of Slices containing only related logic	2,229	2,229	100%
Number of Slices containing unrelated logic	0	2,229	0%
Total Number of 4 input LUTs	3,589	15,360	23%
Number used as logic	2,599		
Number used as a route-thru	2		
Number used for Dual Port RAMs	988		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		
Number of Block RAMs	8	24	33%

Πίνακας 4-7 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 14, μήκους 127, με παραγωγή – κατανάλωση χωρίς πολλαπλά shift (Spartan3, XC3S1000)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,392	15,360	9%
Number used as Flip Flops	900		
Number used as Latches	492		
Number of 4 input LUTs	8,491	15,360	55%
Logic Distribution			
Number of occupied Slices	5,253	7,680	68%
Number of Slices containing only related logic	5,253	5,253	100%
Number of Slices containing unrelated logic	0	5,253	0%
Total Number of 4 input LUTs	9,489	15,360	61%
Number used as logic	8,491		
Number used as a route-thru	10		
Number used for Dual Port RAMs	988		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		
Number of Block RAMs	8	24	33%

Πίνακας 4-8 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 14, μήκους 127, με παραγωγή – κατανάλωση και πολλαπλά shift (Spartan3, XC3S1000)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,487	15,360	9%
Number used as Flip Flops	915		
Number used as Latches	572		
Number of 4 input LUTs	2,886	15,360	18%
Logic Distribution			
Number of occupied Slices	2,508	7,680	32%
Number of Slices containing only related logic	2,508	2,508	100%
Number of Slices containing unrelated logic	0	2,508	0%
Total Number of 4 input LUTs	4,035	15,360	26%
Number used as logic	2,886		
Number used as a route-thru	1		
Number used for Dual Port RAMs	1,148		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		
Number of Block RAMs	10	24	41%

Πίνακας 4-9 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 15, μήκους 151, με παραγωγή – κατανάλωση χωρίς πολλαπλά shift (Spartan3, XC3S1000)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,550	15,360	10%
Number used as Flip Flops	978		
Number used as Latches	572		
Number of 4 input LUTs	9,926	15,360	64%
Logic Distribution			
Number of occupied Slices	6,216	7,680	80%
Number of Slices containing only related logic	6,216	6,216	100%
Number of Slices containing unrelated logic	0	6,216	0%
Total Number of 4 input LUTs	11,084	15,360	72%
Number used as logic	9,926		
Number used as a route-thru	10		
Number used for Dual Port RAMs	1,148		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		
Number of Block RAMs	10	24	41%

Πίνακας 4-10 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 15, μήκους 151, με παραγωγή – κατανάλωση και πολλαπλά shift (Spartan3, XC3S1000)

Στον παρακάτω Πίνακα (4-11) παρουσιάζονται ποσοστιαία οι πόροι σε Slices που καταλαμβάνουν οι παραπάνω μηχανές GE3 στην FPGA Spartan3, XC3S1000.

	Ποσοστό των Slices που καταλαμβάνονται
Golomb 4	1%
Golomb 12	8%
Golomb 13	9%
Golomb 13 με πολλαπλά shift 1-16	27%
Golomb 13 με παραγωγή – κατανάλωση και πολλαπλά shift 1-16	59%
Golomb 14 με παραγωγή – κατανάλωση	29%
Golomb 14 με παραγωγή – κατανάλωση και πολλαπλά shift 1-16	68%
Golomb 15 με παραγωγή – κατανάλωση	32%
Golomb 15 με παραγωγή – κατανάλωση και πολλαπλά shift 1-16	80%

Πίνακας 4-11 Ποσοστά πόρων που καταλαμβάνουν στην Spartan3 οι μηχανές GE, ανάλογα τον κανόνα Golomb και την αρχιτεκτονική

Τέλος προσομοιώθηκε ο κανόνας Golomb 25 στο ISE simulator για αντιπροσωπευτικό αριθμό κύκλων ρολογιού και επιβεβαιώθηκε η σωστή λειτουργία της αρχιτεκτονικής. Επίσης αποτυπώθηκε στην πλατφόρμα με FPGA Spartan3, XC3S1000, όπου και διαπιστώθηκε ότι δε χωράει αν εξυπηρετηθούν μέχρι 16 θέσεων πολλαπλά shift, καθώς καταλαμβάνεται πάνω από 60% της πλατφόρμας χωρίς την εξυπηρέτηση πολλαπλών shift. Για αυτό το λόγο χρησιμοποιήθηκε μεγαλύτερη πλατφόρμα, η Virtex4, XC4VLX60. Στον Πίνακα 4-12 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή για τον κανόνα Golomb 25 χωρίς την εξυπηρέτηση πολλαπλών shift, η οποία λειτουργεί στα 92Mhz, ενώ στον Πίνακα 4-13 παρουσιάζονται οι πόροι που καταλαμβάνει η μηχανή με την εξυπηρέτηση πολλαπλών shift (1-16 θέσεων), η οποία λειτουργεί στα 91Mhz.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,955	53,248	3%
Number used as Flip Flops	985		
Number used as Latches	970		
Number of 4 input LUTs	5,622	53,248	10%
Logic Distribution			
Number of occupied Slices	5,276	26,624	19%
Number of Slices containing only related logic	5,276	5,276	100%
Number of Slices containing unrelated logic	0	5,276	0%
Total Number of 4 input LUTs	9,507	53,248	17%
Number used as logic	5,622		
Number used as a route-thru	1		
Number used for Dual Port RAMs	3,884		
Number of bonded IOBs	9	640	1%

Πίνακας 4-12 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 25, μήκους 480, χωρίς πολλαπλά shift (Virtex4, XC4V60)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	2,123	53,248	3%
Number used as Flip Flops	1,153		
Number used as Latches	970		
Number of 4 input LUTs	21,018	53,248	39%
Logic Distribution			
Number of occupied Slices	13,612	26,624	51%
Number of Slices containing only related logic	13,612	13,612	100%
Number of Slices containing unrelated logic	0	13,612	0%
Total Number of 4 input LUTs	25,255	53,248	47%
Number used as logic	21,018		
Number used as a route-thru	353		
Number used for Dual Port RAMs	3,884		
Number of bonded IOBs	9	640	1%

Πίνακας 4-13 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 25, μήκους 480, με πολλαπλά shift (Virtex4, XC4V60)

Στον παρακάτω Πίνακα (4-14) παρουσιάζονται ποσοστιαία οι πόροι που καταλαμβάνει η GE3 για τον κανόνα Golomb 25 στην FPGA Virtex4, XC4VLX60.

	Ποσοστό των Slices που καταλαμβάνονται
Golomb 25	19%
Golomb 25 με πολλαπλά shift 1-16	51%

Πίνακας 4-14 Πόροι που καταλαμβάνει η GE3 για τον κανόνα Golomb 25 ανάλογα την αρχιτεκτονική (με και χωρίς πολλαπλά shift)

4.2 Αποτύπωση της GE3 με παραλληλισμό

Αφού επαληθεύτηκε η σωστή λειτουργία της μηχανής GE3 έγιναν οι κατάλληλες προσθήκες για να επιτευχθεί ο παραλληλισμός του προβλήματος. Δηλαδή να έχουμε μια μηχανή η οποία παράγει stubs και πολλές μηχανές που τα καταναλώνουν παράλληλα.

4.2.1 Αποτύπωση της αρχιτεκτονικής παραλληλισμού για τον κανόνα Golomb 12

Η αρχιτεκτονική παραγωγής – κατανάλωσης προσομοιώθηκε στο ISE simulator και η λειτουργία της επαληθεύτηκε για τον κανόνα Golomb 12, καθώς επίσης έτρεξε και στην πλατφόρμα με FPGA Spartan3, XC3S1000. Η μηχανή αποτελείται από μία μηχανή παραγωγής η οποία παράγει stubs 9 δεικτών, ενώ τρεις μηχανές κατανάλωσης επιλύουν τους υπόλοιπους 3 δείκτες. Στον Πίνακα 4-15 παρουσιάζονται οι πόροι που καταλαμβάνονται από αυτή τη μηχανή, η οποία λειτουργεί στα 60Mhz. Η συγκεκριμένη μηχανή καταλαμβάνει 4 φορές περισσότερους πόρους από τη μηχανή χωρίς παραλληλισμό (Πίνακας 4-2), αφού ουσιαστικά αποτελείται από 4 μηχανές GE.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,638	15,360	10%
Number used as Flip Flops	940		
Number used as Latches	698		
Number of 4 input LUTs	3,673	15,360	23%
Logic Distribution			
Number of occupied Slices	2,897	7,680	37%
Number of Slices containing only related logic	2,897	2,897	100%
Number of Slices containing unrelated logic	0	2,897	0%
Total Number of 4 input LUTs	5,077	15,360	33%
Number used as logic	3,673		
Number used for Dual Port RAMs	1,404		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		
Number of Block RAMs	5	24	20%

Πίνακας 4-15 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 12, μήκους 85, με 1 μηχανή παραγωγής και 3 μηχανές κατανάλωσης (Spartan3, XC3S1000)

4.2.2 Έρευνα για τον παραλληλισμό της αρχιτεκτονικής για τον κανόνα Golomb 25

Για τον κανόνα Golomb 25 αρχικά ερευνήθηκε το πρόβλημα το οποίο αφορά στο πόσοι δείκτες θα τοποθετούνται από τη μηχανή παραγωγής, ώστε ο φόρτος να μοιράζεται σωστά μεταξύ παραγωγής και κατανάλωσης των stubs. Δηλαδή, στόχος είναι καμία μηχανή να μην είναι ανενεργή για μεγάλο χρονικό διάστημα. Αν κάποια μηχανή πρέπει να είναι ανενεργή, αυτή προτιμάται να είναι η μηχανή παραγωγής και όχι οι μηχανές κατανάλωσης των stubs. Δηλαδή η μηχανή παραγωγής πρέπει να παράγει stubs με υψηλότερο ρυθμό από ότι τα καταναλώνουν οι μηχανές κατανάλωσης.

Για το λόγο αυτό δημιουργήθηκε λογισμικό με το οποίο προσομοιώθηκε η διαδικασία παραγωγής – κατανάλωσης. Με το λογισμικό αυτό μπορούμε να δούμε, ανάλογα τον αριθμό των δεικτών που θα έχουν τα stubs που θα βγαίνουν από την παραγωγή, ανά πόσους κύκλους παράγεται ένα stub. Για παράδειγμα για stubs 9 δεικτών η μηχανή παράγει stubs με ρυθμό 1 ανά 2,5 κύκλους ρολογιού περίπου, ενώ

για stubs 15 δεικτών η μηχανή παράγει stubs με ρυθμό 1 ανά 3,5 κύκλους ρολογιού περίπου και για stubs 14 δεικτών η μηχανή παράγει stubs με ρυθμό περίπου 1 ανά 3 κύκλους ρολογιού.

Επίσης μετρήθηκε ο ρυθμός με τον οποίο καταναλώνονται τα stubs από τις μηχανές κατανάλωσης. Υπολογίστηκε ότι αν στις μηχανές κατανάλωσης εισέρχονται stubs 15 δεικτών χρειάζονται κατά μέσο όρο περίπου 5 χιλιάδες κύκλοι για να τελειώσει η αναζήτηση σε καθένα από αυτά, ενώ για stubs 14 δεικτών χρειάζονται κατά μέσο όρο περίπου 70 χιλιάδες κύκλοι. Όσο μειώνεται ο αριθμός των δεικτών που τοποθετούνται από τη μηχανή παραγωγής τόσο περισσότεροι κύκλοι χρειάζονται για την επίλυση κάθε stub.

Ακόμα δεν πάρθηκαν μετρήσεις για stubs παραγωγής μικρότερα των 9 δεικτών, αφού αυτό θα έβαζε περισσότερους από 16 δείκτες να επιλυθούν από την κατανάλωση. Έτσι θα διπλασιαζόταν ο χώρος που καταλαμβάνουν οι μνήμες που υλοποιούν τη στοίβα, Dual Port Ram, σε κάθε μηχανή κατανάλωσης, αφού η τεχνολογία υποστηρίζει μνήμες των οποίων το βάθος είναι δύναμη του 2. Άρα θα χρειαζόμασταν μνήμες βάθους 32 σε κάθε μηχανή κατανάλωσης για πάνω από 16 δείκτες. Για τον ίδιο λόγο δεν εξεταστήκαν stubs παραγωγής μεγαλύτερα των 16 δεικτών, καθώς έτσι θα διπλασιαζόταν ο χώρος που καταλαμβάνουν οι μνήμες της μηχανής παραγωγής.

Έχοντας τα παραπάνω υπόψη και το ότι η κάθε μηχανή κατανάλωσης αρχικοποιείται σε 2 κύκλους ρολογιού, από τη FIFO, μπορούμε να υπολογίσουμε σε ποιο σημείο πρέπει να «κόψουμε» τον κανόνα, δηλαδή πόσοι δείκτες θα επιλύονται από την παραγωγή και πόσοι από τις μηχανές κατανάλωσης. Έτσι ανάλογα το πόσες μηχανές κατανάλωσης έχουμε στη διάθεσή μας πρέπει να αποφασίσουμε σε ποιο σημείο πρέπει να μοιράσουμε την αναζήτηση μεταξύ παραγωγής και κατανάλωσης, το οποίο εξαρτάται από τους διαθέσιμους πόρους της κάθε πλατφόρμας. Όσο περισσότερες μηχανές κατανάλωσης διαθέτουμε τόσο λιγότερους δείκτες αναθέτουμε στη μηχανή παραγωγής, δίνοντας έτσι περισσότερο χώρο αναζήτησης στις μηχανές κατανάλωσης των stubs.

4.2.3 Αποτύπωση της αρχιτεκτονικής παραλληλισμού για τον κανόνα Golomb 25

Η αποτύπωση της αρχιτεκτονικής παραλληλισμού για τον κανόνα Golomb 25 αρχικά έγινε στην πλατφόρμα με FPGA Spartan3, XC3S1000, και επιβεβαιώθηκε η

σωστή λειτουργία της. Σε αυτή την πλατφόρμα χωράει μόνο 1 μηχανή παραγωγής και 1 μηχανή κατανάλωσης με την υποστήριξη λίγων μόνο περιπτώσεων πολλαπλών shift (από 1 - 3 θέσεις). Σε αυτή την περίπτωση, δηλαδή όταν έχουμε μια μηχανή παραγωγής και μια μηχανή κατανάλωσης, δεν έχει σημασία σε πιο σημείο θα μοιράσουμε τον κανόνα μεταξύ παραγωγής και κατανάλωσης, όμως πιο αποδοτική λύση είναι αν η παραγωγή τοποθετεί λίγους δείκτες, ώστε να γλυτώνουμε το χρόνο αρχικοποίησης της μηχανής κατανάλωσης. Όσο περισσότεροι δείκτες τοποθετούνται από τη μηχανή παραγωγής τόσο περισσότερα stubs παράγονται, και άρα τόσο περισσότερες αρχικοποιήσεις της μηχανής κατανάλωσης. Στον Πίνακα 4-16 παρουσιάζονται οι πόροι που καταλαμβάνονται από αυτή τη μηχανή, η οποία λειτουργεί στα 40Mhz.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	4,640	15,360	30%
Number used as Flip Flops	2,809		
Number used as Latches	1,831		
Number of 4 input LUTs	10,666	15,360	69%
Logic Distribution			
Number of occupied Slices	7,678	7,680	99%
Number of Slices containing only related logic	6,761	7,678	88%
Number of Slices containing unrelated logic	917	7,678	11%
Total Number of 4 input LUTs	14,817	15,360	96%
Number used as logic	10,666		
Number used as a route-thru	487		
Number used for Dual Port RAMs	3,664		
Number of bonded IOBs	9	173	5%
IOB Flip Flops	7		

Πίνακας 4-16 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 25, μήκους 480, με 1 μηχανή παραγωγής και 1 μηχανή κατανάλωσης (Spartan3, XC3S1000)

Ακόμα η αρχιτεκτονική αποτυπώθηκε στην πλατφόρμα XUP, με FPGA XC2VP30. Στην πλατφόρμα αυτή τοποθετούνται 2 μηχανές κατανάλωσης χωρίς την υποστήριξη πολλαπλών shift ή 1 μηχανή κατανάλωσης με την υποστήριξη πολλαπλών shift (1 – 8 θέσεων). Η δεύτερη περίπτωση με τη μία μηχανή

κατανάλωσης είναι πιο αποδοτική, αφού είναι περίπου 7 φορές πιο γρήγορη από μια μηχανή που δεν υποστηρίζει πολλαπλά shift. Στην πρώτη περίπτωση έχουμε δύο μηχανές κατανάλωσης που δεν υποστηρίζουν πολλαπλά shift, δηλαδή είναι περίπου 2 φορές πιο γρήγορη από μία μηχανή που δεν υποστηρίζει πολλαπλά shift. Επίσης δεν έχει σημασία σε πιο σημείο θα μοιράσουμε τον κανόνα μεταξύ παραγωγής και κατανάλωσης για τη δεύτερη περίπτωση, ενώ στην πρώτη τα stubs παραγωγής πρέπει να είναι 15 ή 16 δεικτών, για να έχουμε αποδοτικό διαμοιρασμό της αναζήτησης μεταξύ παραγωγής και κατανάλωσης. Στον Πίνακα 4-17 παρουσιάζονται οι πόροι που καταλαμβάνονται από τη μηχανή που αποτελείται από 1 μηχανή παράγωγης και 1 μηχανή κατανάλωσης. Η μηχανή αυτή λειτουργεί στα 63Mhz.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	4,639	27,392	16%
Number used as Flip Flops	2,808		
Number used as Latches	1,831		
Number of 4 input LUTs	19,054	27,392	69%
Logic Distribution			
Number of occupied Slices	13,694	13,696	99%
Number of Slices containing only related logic	12,764	13,694	93%
Number of Slices containing unrelated logic	930	13,694	6%
Total Number of 4 input LUTs	22,734	27,392	82%
Number used as logic	19,054		
Number used as a route-thru	16		
Number used for Dual Port RAMs	3,664		
Number of bonded IOBs	9	556	1%
IOB Flip Flops	7		
Number of Block RAMs	28	136	20%

Πίνακας 4-17 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 25, μήκους 480, με 1 μηχανή παραγωγής και 1 μηχανή κατανάλωσης (XUP, FPGA XC2VP30)

Επίσης αποτυπώθηκε στην πλατφόρμα με FPGA Virtex4 (ML410, XC4V60), όπου μπορούν να τοποθετηθούν 2 μηχανές κατανάλωσης οι οποίες υποστηρίζουν από 1 – 11 θέσεων πολλαπλά shift. Αυτό σημαίνει ότι είναι περίπου 20 φορές πιο γρήγορη από μια μηχανή που δεν υποστηρίζει πολλαπλά shift. Επιπλέον στην πλατφόρμα

αυτή, αποδοτικό σπάσιμο του κανόνα είναι η παραγωγή stubs 15 ή 16 δεικτών. Στον Πίνακα 4-18 παρουσιάζονται οι πόροι που καταλαμβάνονται από αυτή τη μηχανή, η οποία λειτουργεί στα 87Mhz.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	6,688	53,248	12%
Number used as Flip Flops	4,739		
Number used as Latches	1,949		
Number of 4 input LUTs	39,806	53,248	74%
Logic Distribution			
Number of occupied Slices	25,783	26,624	96%
Number of Slices containing only related logic	25,783	25,783	100%
Number of Slices containing unrelated logic	0	25,783	0%
Total Number of 4 input LUTs	48,550	53,248	91%
Number used as logic	39,806		
Number used as a route-thru	976		
Number used for Dual Port RAMs	7,768		
Number of bonded IOBs	9	448	2%
Number of FIFO16/RAMB16s	28	160	17%

Πίνακας 4-18 Πόροι που καταλαμβάνει η μηχανή για τον υπολογισμό του κανόνα Golomb 25, μήκους 480, με 1 μηχανή παραγωγής και 2 μηχανές κατανάλωσης (Virtex4, XC4V60)

Τέλος αποτυπώθηκε στην πλατφόρμα με FPGA Virtex5, XC5V330XL, στην οποία, επειδή διαθέτει τους περισσότερους πόρους, μπορούν να τοποθετηθούν 18 μηχανές κατανάλωσης. Οι μηχανές αυτές υποστηρίζουν από 1 – 16 θέσεων πολλαπλά shift. Επίσης το ρολόι σε αυτή την πλατφόρμα είναι πολύ γρηγορότερο από τις μικρότερες (φτάνει τα 120 Mhz). Σε αυτή τη μηχανή, η μηχανή παραγωγής πρέπει να παράγει stubs με λιγότερους δείκτες. Καλή και αποδοτική λύση σε αυτή την περίπτωση κρίνεται η παραγωγή stubs 14 δεικτών με την οποία μειώνεται πολύ η πιθανότητα να τερματίζουν πολλές μηχανές ταυτόχρονα.

Η αρχιτεκτονική με τις 18 μηχανές κατανάλωσης δεν ήταν δυνατό να προσομοιωθεί στα εργαλεία της Xilinx λόγω των αυξημένων απαιτήσεων σε μνήμη που χρειάζονται για να γεμίσει η συγκεκριμένη FPGA. Για αυτό το λόγο δεν ήταν δυνατή η παρουσίαση των πόρων που καταλαμβάνονται από αυτή τη σχεδίαση.

Στον παρακάτω Πίνακα (4-19) παρουσιάζονται συγκεντρωτικά οι παραπάνω μηχανές GE3 και οι ιδιότητες της αρχιτεκτονικής που υποστηρίζουν, για τον κανόνα Golomb 25, ανάλογα την FPGA στην οποία αποτυπώνονται, καθώς και η συχνότητα του ρολογιού που μπορούμε να έχουμε. Στην τελευταία στήλη παρουσιάζεται η αύξηση στην απόδοση που έχουμε σε σχέση με μια μηχανή GE3 η οποία δεν υποστηρίζει πολλαπλά shift, με ρολόι στα 50Mhz.

Board	Πόροι της FPGA (Slices)	Πολλαπλά shift	Μηχανές κατανάλωσης	Συχνότητα ρολογιού	SpeedUp - 1 μηχανή χωρίς shift
Spartan3 XC3S1000	7680	1-3 θέσεων	1	40MHz	2
Virtex2p XC2VP30 (XUP)	13696	1-8 θέσεων	1	63MHz	7
Virtex4 XC4V60	26624	1-11 θέσεων	2	87MHz	35
Virtex5 XC5V330XL	51840 (207360)	1-16 θέσεων	18	120MHz	520

Πίνακας 4-19 Συγκεντρωτικά αποτελέσματα από την αποτύπωση της GE3 για τον κανόνα Golomb 25 σε διάφορες FPGA

Το γενικό συμπέρασμα από την αποτύπωση της αρχιτεκτονικής για τον κανόνα Golomb 25 είναι ότι στην περίπτωση που διαθέτουμε λίγους πόρους, όπως για παράδειγμα με τις πλατφόρμες με FPGA Spartan3 και XUP, δε είναι αποδοτική η αποτύπωση της αρχιτεκτονικής με παραλληλισμό. Η αποτύπωση της αρχιτεκτονικής χωρίς παραλληλισμό κρίνεται πιο αποδοτική, καθώς επιτρέπει την υποστήριξη πολλαπλών shift περισσότερων θέσεων. Αντίθετα για πλατφόρμες με μεγαλύτερες FPGA, όπως η Virtex4 και η Virtex5 330, η αποτύπωση με παραλληλισμό κρίνεται πιο αποδοτική, επειδή δεν έχουμε συνεχή αύξηση στην απόδοση όσο προσθέτουμε περισσότερων θέσεων πολλαπλά shift.

4.3 Πιστοποίηση λειτουργίας GE3

Αρχικά η σωστή και αξιόπιστη λειτουργία της αρχιτεκτονικής επιβεβαιώθηκε με προσομοίωση (simulation) και σύγκριση με τα αποτελέσματα από το λογισμικό που υλοποιεί τον αλγόριθμο Shift για τον κανόνα Golomb 4. Έπειτα η συγκεκριμένη μηχανή κατέβηκε σε FPGA Spartan3 όπου και διαπιστώθηκε ότι επιστρέφει σωστά αποτελέσματα. Η αρχιτεκτονική επεκτάθηκε ώστε να υπολογίζει μεγαλύτερους κανόνες Golomb. Η μηχανή αυτή μετά από simulation κατέβηκε επίσης σε FPGA Spartan3 όπου και επιβεβαιώθηκε η σωστή της λειτουργία για τους κανόνες Golomb από 10 έως και 15. Δεν έγινε επαλήθευση για μεγαλύτερους κανόνες αφού ο χρόνος αναζήτησης ήταν αποτρεπτικός. Παράλληλα επαληθεύτηκε η σωστή λειτουργία των πολλαπλών shift. Μετά την υλοποίηση της αρχιτεκτονικής παραγωγής - κατανάλωσης επιβεβαιώθηκε η σωστή της λειτουργία για τους παραπάνω κανόνες Golomb, ενώ έγινε σύγκριση με τα προηγούμενα αποτελέσματα. Τέλος για τον κανόνα Golomb 25 έγινε simulation για μεγάλο αριθμό κύκλων ρολογιού, ενώ παράλληλα τα αποτελέσματα συγκρίνονταν με αυτά του λογισμικού που υλοποιεί τον αλγόριθμο Shift. Τελικά η αρχιτεκτονική παράγωγης - κατανάλωσης κατέβηκε στο board XUP με και χωρίς πολλαπλά shift. Τα αποτελέσματα από την αποτύπωση της αρχιτεκτονικής στέλνονται μέσω της RS232 σε υπολογιστή γενικής χρήσης. Μετά από ανάλυση των αποτελεσμάτων αποδείχτηκε η σωστή λειτουργία της αρχιτεκτονικής.

5. Αποτελέσματα της Σχεδίασης

Στο κεφάλαιο αυτό παρουσιάζονται και αναλύονται τα πειραματικά αποτελέσματα από την αποτύπωση της GE3 σε FPGA. Ακόμα γίνεται σύγκριση των αποτελεσμάτων αυτών με αυτά από το λογισμικό που σχεδιάστηκε για την προσομοίωση του αλγορίθμου shift. Επίσης γίνεται σύγκριση με τα αποτελέσματα από το λογισμικό που αναπτύχθηκε από τον συνάδελφο Μουντάκη Κυριάκο στα πλαίσια της διπλωματικής του εργασίας.

5.1 Σύγκριση GE3 με shift - GE3 χωρίς shift

Στον Πίνακα 5-1 παρουσιάζονται τα αποτελέσματα από την αποτύπωση της GE3 σε FPGA (Spartan3, XC3S1000), με και χωρίς την υποστήριξη πολλαπλών shift. Η συχνότητα του ρολογιού με την οποία έγιναν τα πειράματα είναι 50 Mhz. Η αρχιτεκτονική, η οποία χρησιμοποιήθηκε για τα πειράματα, αποτελείται από μια μηχανή παράγωγης stubs και μια μηχανή κατανάλωσης, ενώ υποστηρίζει από 1-16 πολλαπλά shift. Όπως έχει αναφερθεί και παραπάνω, μια μηχανή GE χωρίς το σύστημα παραγωγής - κατανάλωσης είναι πιο αποδοτική, αφού το σύστημα αυτό προσθέτει 2 κύκλους για την αρχικοποίηση της μηχανής κατανάλωσης για κάθε stub.

	GE3 χωρίς shift	GE3 με shift 1-16	SpeedUp
Golomb 12	2 sec	0,7 sec	3
Golomb 13	57 sec	18 sec	3,2
Golomb 14	22 m 50sec	7m 3sec	3,3
Golomb 15	2 h 14 m	36m	3,7

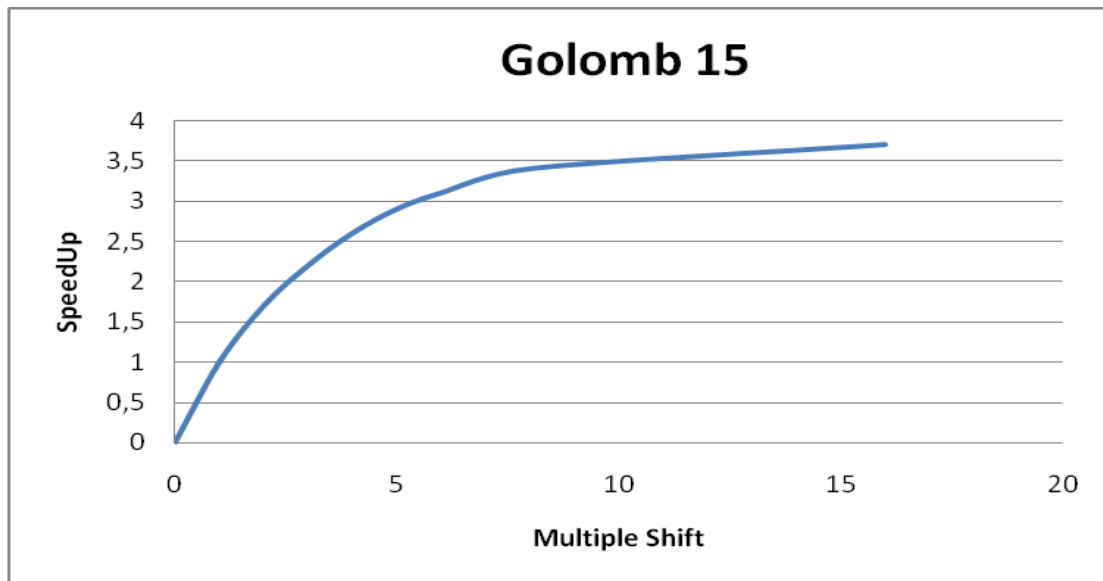
Πίνακας 5-1 Σύγκριση GE3 με shift - GE3 χωρίς shift

Από τον παραπάνω πίνακα παρατηρούμε ότι για τους κανόνες Golomb 12, 13 και 14 έχουμε μια αύξηση στην ταχύτητα περίπου της τάξης του 3. Στο κανόνα Golomb 15 η αύξηση στην ταχύτητα είναι της τάξης του 4. Αυτό συμβαίνει επειδή όσο μεγαλύτερος είναι ο κανόνας τόσο καλύτερη απόδοση παρατηρείται από την υποστήριξη των πολλαπλών shift.

Στον Πίνακα 5-2 παρουσιάζονται τα αποτελέσματα από την αποτύπωση της GE3 σε FPGA για τον κανόνα Golomb 15 για διαφορές περιπτώσεις πολλαπλών shift. Όπως φαίνεται από τον πίνακα ο ρυθμός μείωσης του χρόνου αναζήτησης είναι μεγαλύτερος για shift λίγων περιπτώσεων. Η μηχανή που υποστηρίζει πολλαπλά shift 1-2 θέσεων χρειάζεται σχεδόν το μισό χρόνο σε σχέση με τη μηχανή η οποία δεν υποστηρίζει πολλαπλά shift. Η απόδοση από την εισαγωγή των πολλαπλών shift για τον κανόνα Golomb 15 είναι ικανοποιητική για shift έως 6 θέσεων, αφού πολλαπλά shift περισσότερων θέσεων μας δίνουν μικρή αύξηση στην απόδοση σε σχέση με τους πόρους που απαιτούνται για την αποτύπωση τους. Στον πίνακα φαίνεται ότι η αύξηση στην απόδοση για 1-16 shift είναι ελάχιστη σε σχέση με την αύξηση που παρατηρείται για 1- 8 shift. Η ικανοποιητική αύξηση στην απόδοση που παρατηρείται για shift 1-6 θέσεων είναι επειδή για τον κανόνα Golomb 15 το 90% των shift είναι 1-6 θέσεων. Η αύξηση στην απόδοση σε σχέση με τα πολλαπλά shift παρουσιάζεται επίσης στο Διάγραμμα 5-1.

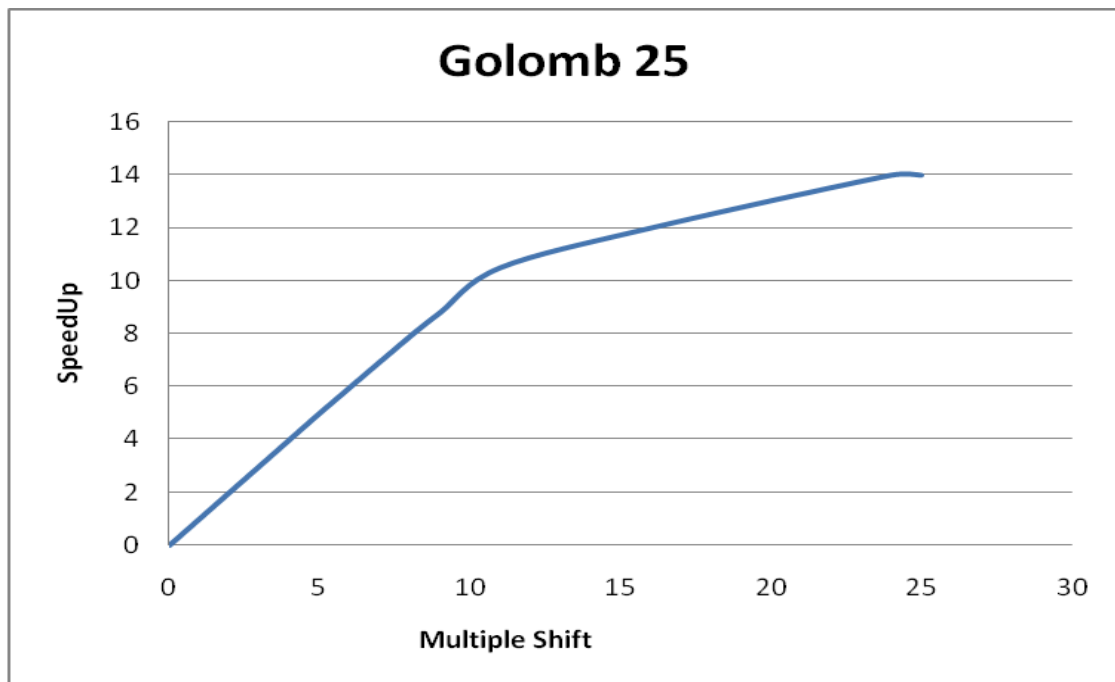
shifts	run time	SpeedUp
1 (χωρίς πολλαπλά shift)	2h 14m	1
1-2	1h 18m	1,7
1-3	60m	2,2
1-4	52m	2,6
1-5	46m	2,9
1-6	43m	3,1
1-8	39m	3,4
1-16	36m	3,7

Πίνακας 5-2 Απόδοση διαφόρων περιπτώσεων πολλαπλών shift για τον κανόνα Golomb 15 σε σχέση με μια μηχανή που δεν υποστηρίζει πολλαπλά shift



Διάγραμμα 5-1 SpeedUp ανάλογα τον αριθμό των πολλαπλών shift για τον κανόνα Golomb 15 σε σχέση με μια μηχανή που δεν υποστηρίζει πολλαπλά shift

Για τον κανόνα Golomb 25 υπολογίστηκε, μετά από ανάλυση σε λογισμικό, ότι με υποστήριξη 1-16 πολλαπλών shift, η αύξηση στην απόδοση είναι της τάξης του 12, αφού πάνω από το 60% των shift είναι 16 θέσεων. Τα αποτελέσματα στην αύξηση απόδοσης από την υποστήριξη πολλαπλών shift φαίνονται στο Διάγραμμα 5-2. Στο διάγραμμα φαίνεται ότι ο ρυθμός αύξησης της απόδοσης ελαττώνεται για πάνω από 16 πολλαπλά shift.



Διάγραμμα 5-2 SpeedUp ανάλογα τον αριθμό των πολλαπλών shift για τον κανόνα Golomb 25 σε σχέση με μια μηχανή που δεν υποστηρίζει πολλαπλά shift

5.2 Αποτελέσματα από τον παραλληλισμό του προβλήματος

Στον παρακάτω Πίνακα 5-3 παρουσιάζονται τα αποτελέσματα από τον παραλληλισμό του προβλήματος. Η αρχιτεκτονική παράγωγης - κατανάλωσης αποτυπώθηκε σε FPGA (Spartan3, XC3S1000) για τον κανόνα Golomb 15 με 1 έως 4 μηχανές κατανάλωσης. Οι μηχανές GE που χρησιμοποιήθηκαν για τις παρακάτω μετρήσεις υποστηρίζουν πολλαπλά shift 1-2 θέσεων. Τα αποτελέσματα από τα πειράματα αυτά είναι τα αναμενόμενα, δηλαδή ότι η αύξηση της απόδοσης είναι ανάλογη του αριθμού των μηχανών κατανάλωσης που διαθέτουμε. Όσες περισσότερες μηχανές κατανάλωσης έχουμε στη διάθεση μας, δια τόσο διαιρείται ο χρόνος αναζήτησης.

Μηχανές κατανάλωσης	run time	SpeedUp
1	1h 18m	1
2	39m	2
3	26m	3
4	19m	4

Πίνακας 5-3 Απόδοση από τον παραλληλισμό της αρχιτεκτονικής για τον κανόνα Golomb 15

5.3 Σύγκριση GE3 – λογισμικό

Στον Πίνακα 5-4 παρουσιάζονται τα αποτελέσματα από την αποτύπωση της GE3 σε FPGA, σε αντιδιαστολή με τα αποτελέσματα από το λογισμικό που προσομοιώνει τον αλγόριθμο shift, καθώς και με το λογισμικό που υλοποιήθηκε από τον συνάδελφο Μουντάκη Κυριάκο στα πλαίσια της διπλωματικής του εργασίας για την εύρεση βέλτιστων κανόνων Golomb. Το λογισμικό που έχει υλοποιηθεί από το συνάδελφο αποτελεί μια παραλλαγή του αλγορίθμου token passing χρησιμοποιώντας κάποια στοιχεία από τον shift algorithm, ενώ η συγκεκριμένη έκδοση του λογισμικού με την οποία έγιναν τα πειράματα τρέχει σε ένα μόνο νήμα μιας διεργασίας. Τα πειράματα για τα δύο λογισμικά έγιναν σε υπολογιστή με λειτουργικό σύστημα Linux Ubuntu, με επεξεργαστή Core2 Duo 2.13 Ghz και με 2 Gb RAM. Η αρχιτεκτονική

που αποτυπώθηκε στην FPGA (Spartan3, XC3S1000) αποτελείται από 1 μηχανή παραγωγής και 1 μηχανή κατανάλωσης.

Στον Πίνακα 5-5 παρουσιάζονται τα αποτελέσματα στην αύξηση της απόδοσης (SpeedUps) από τη σύγκριση της GE3 με τα παραπάνω λογισμικά.

	GE3 χωρίς shift	GE3 με shift 1-16	Λογισμικό Μαλακωνάκη Παύλου	Λογισμικό Μουντάκη Κυριάκου
Golomb 12	2 sec	0,7 sec	46 m	1 m 23 sec
Golomb 13	57 sec	18 sec	13h	33 m 22 sec
Golomb 14	22 m 50sec	7m 3sec	N/A	7h 13m 28sec
Golomb 15	2 h 14 m	36m	N/A	N/A

Πίνακας 5-4 Σύγκριση GE3 – λογισμικό

	GE3 χωρίς shift	GE3 με shift 1-16	Λογισμικό Μαλακωνάκη Παύλου	Λογισμικό Μουντάκη Κυριάκου
Golomb 12	42	119	1/30	1
Golomb 13	35	112	1/23	1
Golomb 14	19	62	-	1

Πίνακας 5-5 SpeedUps GE3 – λογισμικό

Από τους παραπάνω πίνακες παρατηρούμε ότι το λογισμικό που υλοποιήθηκε ως προσομοίωση του αλγορίθμου shift είναι πολύ αργό σε σχέση με τη GE3 καθώς επίσης και με το λογισμικό που υλοποιήθηκε από το συνάδελφο Μουντάκη Κυριάκο. Για αυτό το λόγο δε θα υπολογίσουμε την αύξηση στην απόδοση σε σχέση με το λογισμικό μας, αλλά με το λογισμικό του συναδέλφου.

Σε σχέση λοιπόν με το λογισμικό του συναδέλφου η GE3, για τους κανόνες Golomb 12 – 14, μας δίνει μια αύξηση στην απόδοση της τάξης του 20 – 40, ανάλογα τον κανόνα, χωρίς την υποστήριξη πολλαπλών shift. Αν συμπεριλάβουμε την

υποστήριξη πολλαπλών shift, η αύξηση στην απόδοση διαφέρει από κανόνα σε κανόνα, αφού όσο αυξάνεται το μέγεθος του κανόνα, τόσο καλύτερη απόδοση επιτυγχάνεται με την υποστήριξη των πολλαπλών shift. Με πολλαπλά shift η GE3 μας δίνει αύξηση στην απόδοση από 60 – 120, ανάλογα τον κανόνα. Ακόμα σε πλατφόρμες με FPGA πιο σύγχρονης τεχνολογίας το ρολόι, για το συγκεκριμένο πρόβλημα, είναι διπλάσιο από αυτό που μας δίνει η Spartan3 (50Mhz), το οποίο μας δίνει διπλασιασμό στην απόδοση.

Από την αποτύπωση της GE3 για τον κανόνα Golomb 25 στην πλατφόρμα XUP, με FPGA XC2VP30 και με το ρολόι στα 50Mhz υπολογίστηκε ότι επιλύονται 720 stubs, 14 δεικτών, ανά δευτερόλεπτο χωρίς πολλαπλά shift, ενώ με 1-7 θέσεων πολλαπλά shift επιλύονται 5000 stubs ανά δευτερόλεπτο. Η αρχιτεκτονική που αποτυπώθηκε αποτελείται από μια μηχανή παραγωγής και μια μηχανή κατανάλωσης χωρίς πολλαπλά shift. Δυστυχώς δεν ήταν δυνατός ο υπολογισμός κάποιας ανάλογης τιμής από το δικό μου λογισμικό ή από του συναδέλφου.

Η σύγκριση με λογισμικό για τον κανόνα Golomb 25 τελικά αποφασίστηκε να γίνει με τα δεδομένα από τον οργανισμό distributed.net. Αρχικά ζητήθηκε από τον οργανισμό να μου στείλει κάποιο αποτέλεσμα σε stubs/sec ώστε να μπορεί να συγκριθεί με τα παραπάνω αποτελέσματα. Δυστυχώς δεν ήταν δυνατό να μας δώσουν μια τέτοιου είδους μέτρηση, αλλά με παρέπεμψαν στην ιστοσελίδα με τα στατιστικά τους (<http://n0cgi.distributed.net/speed/query.php>). Στην ιστοσελίδα αυτή παρουσιάζεται η ταχύτητα με την οποία επιλύει το πρόβλημα κάθε υπολογιστής, ανάλογα τον επεξεργαστή, σε Nodes/sec, δηλαδή πόσους κόμβους επισκέπτεται ανά δευτερόλεπτο. Την καλύτερη επίδοση έχει ο Core 2 Duo στα 3,08Ghz με 46 Mnodes/sec. Έτσι υπολογίστηκε το αντίστοιχο αποτέλεσμα από την αποτύπωση της GE3 σε FPGA. Το ανάλογο αποτέλεσμα για τη GE3 είναι το άθροισμα των push, shift και pop που εκτελούνται ανά δευτερόλεπτο. Έτσι υπολογίσαμε το ανάλογο αποτέλεσμα από την αποτύπωση της GE3 στην πλατφόρμα XUP με FPGA XC2VP30 και με ρολόι στα 50Mhz. Στη συγκεκριμένη FPGA αποτυπώνονται μια μηχανή παραγωγής και μια μηχανή κατανάλωσης με πολλαπλά shift 1-8 θέσεων. Άρα η μηχανή αυτή επισκέπτεται 7 nodes κάθε κύκλο ρολογιού, δηλαδή επισκέπτεται $7 * 50 * 10^6 = 350 * 10^6$ nodes ανά δευτερόλεπτο. Ακόμα το αντίστοιχο αποτέλεσμα υπολογίστηκε για την FPGA Virtex5, XC5V330LX, όπου μπορούν να τοποθετηθούν 1 μηχανή παραγωγής και 18 μηχανές κατανάλωσης οι οποίες υποστηρίζουν πολλαπλά shift 1-16 θέσεων, με ρολόι 120MHz. Άρα η συγκεκριμένη GE επισκέπτεται $18 * 12$

* $120 * 10^6 = 26 * 10^9$ nodes ανά δευτερόλεπτο. Τα αποτελέσματα αυτά μας δίνουν αύξηση στην απόδοση σε σχέση με τον υπολογιστή γενικής χρήσης της τάξης του 7,5 για την πλατφόρμα XUP και 565 για την FPGA Virtex5, XC5V330LX. Τα παραπάνω αποτελέσματα παρουσιάζονται στον Πίνακα 5-6.

Processor - FPGA	Nodes/sec	Αρχιτεκτονική FPGA	SpeedUP
Core 2 Duo 3,08Ghz	$46 * 10^6$	-	1
XUP, XC2VP30 (50Mhz)	$350 * 10^6$	1 μηχανή κατανάλωσης με 1-8 shift	7,6
Virtex5, XC5V330LX (120 Mhz)	$26 * 10^9$	18 μηχανές κατανάλωσης με 1-16 shift	565,5

Πίνακας 5-6 Σύγκριση GE3 με το λογισμικό της οργάνωσης distributet.net

Τέλος πρέπει να αναφερθεί ότι η αύξηση στην απόδοση είναι ακόμα μεγαλύτερη για κανόνες Golomb με περισσότερους δείκτες, αφού όσο μεγαλύτερος είναι ο κανόνας τόσο καλύτερη απόδοση έχουμε από την υλοποίηση των πολλαπλών shift.

6. Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σε αυτό το κεφάλαιο παρουσιάζονται τα συμπεράσματα που προκύπτουν από την αποτύπωση του προβλήματος υπολογισμού Βέλτιστων κανόνων Golomb σε αναδιατασσόμενη λογική. Επίσης, προτείνονται μελλοντικές επεκτάσεις για την εργασία αυτή.

6.1 Συμπεράσματα

6.1.1 Συμπεράσματα από τη σχεδίαση και την αποτύπωση του αλγορίθμου shift

Μετά την προσεκτική μελέτη του προβλήματος του υπολογισμού βέλτιστων κανόνων Golomb προέκυψε η αρχιτεκτονική της μηχανής GE3. Η αρχιτεκτονική της GE3 εκμεταλλεύεται πλήρως τον παραλληλισμό του προβλήματος και έτσι εκμεταλλεύεται πλήρως τους περιορισμένους πόρους σε υλικό. Μετά από ανάλυση του παραλληλισμού υπολογίζεται πόσοι δείκτες πρέπει να τοποθετούνται από τη μηχανή η οποία παράγει stubs και πόσοι από τις μηχανές που τα καταναλώνουν, έτσι ώστε να έχουμε όσο το δυνατόν πιο αποδοτική διαίρεση του χρόνου αναζήτησης. Ακόμα ανάλογα τους πόρους σε υλικό υπολογίζεται αν είναι πιο αποδοτικό να έχουμε περισσότερες μηχανές κατανάλωσης ή η υποστήριξη περισσότερων περιπτώσεων πολλαπλών shift. Ουσιαστικά, δηλαδή, ο κύριος στόχος είναι η πιο αποδοτική εκμετάλλευση των πόρων σε υλικό, ώστε να έχουμε την καλύτερη δυνατή μείωση στο χρόνο αναζήτησης.

Επίσης από την ανάλυση πριν την υλοποίηση των πολλαπλών shift παρατηρείται το φαινόμενο ότι επειδή όσο περισσότερους δείκτες έχει ένας κανόνας Golomb τόσο αυξάνεται το μέγεθός του και άρα τόσο καλύτερη απόδοση έχουμε από την υποστήριξη πολλαπλών shift.

Ακόμα η αρχιτεκτονική της GE3 είναι εύκολα επεκτάσιμη ώστε να μπορεί να υπολογίζει κανόνες Golomb με περισσότερους δείκτες. Επίσης είναι εύκολη η προσαρμογή της ανάλογα με τους πόρους σε υλικό της εκάστοτε πλατφόρμας.

6.1.2 Συμπεράσματα που αφορούν την απόδοση της GE3

Η εξάπλωση του διαδικτύου και η ανάπτυξη αντίστοιχου λογισμικού δίνει μεγάλες δυνατότητες στην διασύνδεση των υπολογιστών και στην εκμετάλλευση χρόνου επεξεργαστή από εκατοντάδες χιλιάδες υπολογιστές ταυτόχρονα. Η λύση αυτή φαίνεται ιδιαίτερα ελκυστική και αποδοτική για προβλήματα μεγάλου υπολογιστικού φόρτου όπως αυτό του υπολογισμού βέλτιστων κανόνων Golomb. Από την άλλη μεριά όμως υπάρχει η αντίστοιχη τάση και στην κοινότητα αυτών που κατασκευάζουν μηχανές ειδικής χρήσης με FPGAs. Η τεχνολογική πρόοδος των ίδιων των chip, αλλά και των πλατφόρμων που τις υποστηρίζουν, είναι μεγαλύτερη από την αντίστοιχη τεχνολογική πρόοδο των υπολογιστών γενικής χρήσης. Τα αποτελέσματα από τις διάφορες πειραματικές μετρήσεις που παρουσιάστηκαν στο προηγούμενο κεφάλαιο επιβεβαιώνουν το παραπάνω συμπέρασμα και δείχνουν την χρησιμότητα των FPGAs σε προβλήματα μεγάλου υπολογιστικού φόρτου. Ειδικά στην περίπτωση του προβλήματος της εύρεσης βέλτιστων κανόνων Golomb η αύξηση στην απόδοση που παρατηρείται από την αποτύπωση του αλγορίθμου shift σε υλικό, είναι επειδή όλοι οι έλεγχοι για το αν έχουμε σύγκρουση μεταξύ των διανυσμάτων List και Dist γίνονται ταυτόχρονα, κάτι αδύνατο για τους υπολογιστές γενικής χρήσης και λογισμικό. Ακόμα η υλοποίηση των πολλαπλών shift δεν είναι αποδοτική σε λογισμικό. Από τα πειραματικά αποτελέσματα φαίνεται ότι ένα μεγάλο υπολογιστικά επαναπρογραμματιζόμενο συστήματα (FPGA) είναι ισοδύναμο με πάνω από 550 υπολογιστές γενικής χρήσης. Έτσι λίγα μεγάλα υπολογιστικά επαναπρογραμματιζόμενα συστήματα είναι ισοδύναμα με ολόκληρη την υπολογιστική ισχύ που μπορεί να συγκεντρωθεί στο διαδίκτυο.

6.2 Μελλοντικές Επεκτάσεις

Δημιουργία server για τον υπολογισμό του κανόνα Golomb 26

Επόμενος στόχος αυτής της εργασίας είναι η επέκταση της αρχιτεκτονικής ώστε να υπολογίζει τον κανόνα Golomb 26 και η αποτύπωση της σε πλατφόρμα με

μεγάλη FPGA. Η πλατφόρμα αυτή θα συνδέεται με ένα υπολογιστή γενικής χρήσης στον οποίο θα στέλνονται και θα αποθηκεύονται τα αποτελέσματα καθώς επίσης και ανά τακτά χρονικά διαστήματα το σημείο μέχρι το οποίο έχει ολοκληρωθεί η αναζήτηση. Ακόμα εξετάζεται το ενδεχόμενο να υλοποιηθεί κατάλληλο interface ώστε η πρόοδος της αναζήτησης να παρουσιάζεται σε ιστοσελίδα στο internet.

Βιβλιογραφία

- [1] A. K. Dewdney. “Computer Recreations”. Scientific American, pages 16-26, December 1985.
- [2] A. K. Dewdney. “Computer Recreations”. Scientific American, pages 14-21, March 1986.
- [3] A.K. Dewdney. “The Armchair Universe”, W.H.Freeman and Company, 1988.
- [4] W. T. Rankin. “Optimal Golomb Rulers: An Exhaustive Parallel Search Implementation”. Master thesis, Duke University, 1993.
- [5] J. P. Robinson. “Optimal Golomb Rulers”. IEEE Transactions on Computers, C-28:943-944, December 1979.
- [6] M. Gardner. “Mathematical Games”. Scientific American, 226:108-112, March 1972.
- [7] E. M. Reingold, J. Nievergelt, and N. Deo. “Combinational Algorithms: Theory and Practice”. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.
- [8] E. Blum, F. Biraud, J. Rides. “On Optimal Synthetic Linear Arrays with Applications to Radioastronomy”. IEEE Transactions on Antennas and Propagation, AP 22:108-109, 1974.
- [9] Gary S. Bloom and Solomon W. Golomb. “Applications of Numbered Unidirected Graphs”. Proceedings of the IEEE, 65:562-570, April 1977.
- [10] J. Robinson, A. Bernstein. “A Class of Binary Recurent Codes with Limited Error”. IEEE Transacions on Informtion Theorions on Informtion Theory, IT-13:106-113, Jan 1967.
- [11] Yong, L. Rahman, A.A. Yusof, B.P.H. Suaidi, M.K.. “Fiber optic microcellular system composite triple beat reduction using the Golomb ruler”. Proceedings of the 12th IEEE International Conference, 2:590-594, 16-19 Nov. 2004.

- [12] E. Sotiriades. "Hardware - Software Codesign of Integrated Parallel Systems for Computation of Optimal Golomb Rulers". Master thesis, TUC, 2001.
- [13] A. Emmanouelides, E. Sotiriades. "Study, Design and Verification of FPGA-based Architecture for Computation of Optimal Golomb Rulers". Diploma thesis, TUC, 1997.
- [14] L. H. House. "Programming FPGA Based Custom Computing Engines in VHDL to Solve Computationally Intensive Problems". Master thesis, Duke University, 1993.
- [15] A. Dollas, W. Rankin, and D. McCracken. "New Algorithms for Golomb Ruler Evaluation and the Proof of the 19-Mark Ruler". IEEE Transactions on Information Theory, January 1998.
- [16] A. Dollas, E. Sotiriades, A. Emmanouelides. "Architecture and Design of GE1, a FCCM for Golomb Ruler Derivation, Proceedings". International Symposium on FPGA's for Custom Computing Machines, FCCM 98, Napa, California, pp.48-56, April 1998.
- [17] J. Hennessy and D. Patterson. "Computer Architecture a Quantive Approach". Morgan Kaufmann Publishers, Inc. 1990.
- [18] Gupta, R.K. and G.D. Micheli. "Constrained Software Generation for Hardware-Software Systems". Third International Workshop on Hardware/Software Codesign, 1994, Grenoble, France, IEEE Computer Society Press, 56-63.
- [19] Edwards, M., et al. "A Development System for Hardware - Software Co-Synthesis using FPGAs". Second IFIP International Workshop on Hardware/Software Codesign (Codes/CASHE'93), 1993, Innsbruck-Igls, Austria, May 24-27: Sponsored by IFIP WG 10.5 in cooperation with IFIP WG 10.2.
- [20] A. Luk, W., T. Wu, and I. Page. "Hardware-Software Codesign of Multidimensional Programs". IEEE Workshop on FPGA for Custom Computing Machines (FCCM '94), 1994, Napa, CA, USA, April 11-13, preliminary proceedings (final proceedings will become available in mid-Summer 1994): IEEE Computer Society, p. 10.

- [21] B. Box and J. Nieznanski. "Common Processor Element Packaging for CHAMP". In Proceedings, IEEE Workshop on FPGAs for Custom Computing Machines, pages 39-44, IEEE Computer Society Press, 1995.
- [22] J. Carrera, E. Martinez, S. Fernandez, and M. Chaus. "Architecture of an FPGA-based Coprocessor: the PAR-1". In Proceedings, IEEE Workshop on FPGAs for Custom Computing Machines, pages 20-29, IEEE Computer Society Press, 1995.
- [23] P. Stogiannos, A. Dollas, V. Digalakis. "A Configurable Logic Based Architecture for Real-Time Continuous Speech Recognition using Hidden Markov Models". Journal of VLSI Signal Processing, Kluwer Academic Publishers, vol. 24/(2/3), pp.223-240, March 2000.
- [24] S. Guccione, D. Levi. "Design Advantages of Run-time Reconfiguration, in Reconfigurable Technology: FPGAs for Computing and Applications". Part of SPIE East 99, Boston, MA, pp. 87-92, September 1999.
- [25] J. Badcock. "Synthesis of Microelectronic Systems with Reusability of Nonsynthesizable Subsystems". 1995.
- [26] A. Dollas, B. Ward, and J. D. S. Babcock. "FPGA Based Low Cost Generic Reusable Module for the Rapid Prototyping of Subsystems". In R. W. Hartenstein and M. Z. Servit, editors, Proceedings, The Forth International Workshop on Field Programmable Logic and Applications, FPL 94, pages 259-270, Springer-Verlag, 1994.
- [27] J. McHenry, P. Dowd, T. Carrozzi, et al. "An FPGA-Based Coprocessor for ATM Firewalls". Proceedings, FCCM '97, Napa, California, pp. 30-39, April 1997.
- [28] M. Jones, L. Scharf, J. Scott, M. Yaconis, K. Yao, P. Athanas, B. Schott. "Implementing an API for Distributed Adaptive Computing Systems". FCCM 99, Napa, California, pp. 22-231, April 1999.
- [29] W. Andrews. "Prototyping System Emulates up to Six Million Gates". Computer Design 31(3):104-105, Nov. 1992.
- [30] E. Sotiriades, A. Dolals, P. Athanas. "Hardware - Software Codesign and Parallel Implementation of a Golomb Ruler Derivation Engine". Proceedings,

Field Programmable Custom Computing Machines, pages 227-235. IEEE Computer Society, 2000.