

«Σχεδίαση Γραμμικών Ολισθητών με Επιλεκτική Ανάδραση»

Η  
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

Υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνοψης  
του Τμήματος Πληροφορικής  
Εξεταστική Επιτροπή

από τον

Τενέντε Ε. Βασίλειο

ως μέρος των Υποχρεώσεων  
για τη λήψη  
του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ  
ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΙΣ ΤΕΧΝΟΛΟΧΙΕΣ - ΕΦΑΡΜΟΓΕΣ

Μάιος 2007

## **ΕΥΧΑΡΙΣΤΙΕΣ**

---

Η διατριβή αυτή είναι αποτέλεσμα αρκετών κόπων τα τελευταία δύο χρόνια. Στο Πανεπιστήμιο Ιωαννίνων βρέθηκα χωρίς καμία εξειδικευμένη γνώση για Σχεδίαση Ψηφιακών Συστημάτων και ακολούθησα δρόμο του ελέγχου ορθής λειτουργίας ολοκληρωμένων κυκλωμάτων παρακολουθώντας τα μαθήματα της τεχνολογικής κατεύθυνσης. Κάποια στιγμή στη ζωή οι περισσότεροι παίρνουμε το ρόλο του Δον Κιχώτη. Ήταν στιγμές που ένιωσα πως η εργασία αυτή ήταν οι ανεμόμυλοι μου.

Χαίρομαι που στην προσπάθεια αυτή είχα συνοδοιπόρο τον Καθηγητή μου κ. Καβουσιανό Χρυσοβαλάντη και νιώθω την ανάγκη να τον ευχαριστήσω γιατί πίστεψε στις δυνατότητές μου και μου έδωσε την ευκαιρία να ασχοληθώ με αυτό το θέμα. Το πιο σημαντικό είναι ότι απέναντι στο πρόβλημα αυτό ποτέ δεν ένιωσα μόνος. Θέλω επίσης να ευχαριστήσω τον κ. Καλλίγερο γιατί σε κρίσιμο σημείο της εργασίας αυτής σαν εξωτερικός παρατηρητής με την εμπειρία του και τις παρατηρήσεις του έδωσε πολύ χρήσιμες συμβουλές. Τέλος, θέλω να ευχαριστήσω τους γονείς μου γιατί με στήριξαν τόσο χρηματικά όσο και ηθικά στην προσπάθεια αυτή.

## ΠΕΡΙΕΧΟΜΕΝΑ

---

ΕΥΧΑΡΙΣΤΙΕΣ	ii
ΠΕΡΙΕΧΟΜΕΝΑ	iii
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	v
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	vi
ΠΕΡΙΛΗΨΗ	vii
EXTENDED ABSTRACT IN ENGLISH	x
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	1
1.1. Σύγχρονη Σχεδίαση Ψηφιακών Συστημάτων με Ενσωματωμένους Πυρήνες	1
1.2. Ενσωματωμένοι Πυρήνες – Ενσωματωμένος Έλεγχος	2
1.3. Κατηγορίες Μεθόδων Ελέγχου Ενσωματωμένων Πυρήνων	4
1.4. Δομή Διατριβής	5
ΚΕΦΑΛΑΙΟ 2. ΟΛΙΣΘΗΤΕΣ ΓΡΑΜΜΙΚΗΣ ΑΝΑΔΡΑΣΗΣ – LFSRs	7
2.1. Αρχιτεκτονική Ελέγχου Βασισμένη σε Ολισθητές Γραμμικής Ανάδρασης	7
2.2. Υπολογισμός Αρχικής Κατάστασης Ολισθητή Γραμμικής Ανάδρασης	9
2.3. Βελτιστοποιήσεις του Ολισθητή Γραμμικής Ανάδρασης	11
ΚΕΦΑΛΑΙΟ 3. ΠΑΡΑΓΩΓΗ ΜΟΤΙΒΩΝ ΜΕ ΟΛΙΣΘΗΤΗ ΓΡΑΜΜΙΚΗΣ ΑΝΑΔΡΑΣΗΣ	16
3.1. Αρχιτεκτονική Παραγωγής Μοτίβων με Χρήση Επιλεκτικής Ανάδρασης	16
3.2. Επιλογή Μοτίβου	20
3.3. Κωδικοποίηση Σε Παράθυρο L Με Χρήση Μοτίβου	24
3.4. Αρχιτεκτονική Αποκωδικοποίησης	25
3.5. Πειραματικά Αποτελέσματα	27
3.6. Αρχιτεκτονική για Πολλαπλά Μοτίβα	32
3.7. Συμπεράσματα	35
3.8. Μελλοντική Μελέτη	36
ΚΕΦΑΛΑΙΟ 4. ΟΛΙΣΘΗΤΗΣ ΠΑΡΑΒΛΕΨΗΣ ΚΑΤΑΣΤΑΣΕΩΝ	38
4.1. Κύκλωμα Παράβλεψης	38
4.2. Μέθοδος Μείωσης της Ακολουθίας Ελέγχου	42
4.3. Αρχιτεκτονική αποκωδικοποίησης	44
ΚΕΦΑΛΑΙΟ 5. ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΣΥΓΚΡΙΣΕΙΣ ΟΛΙΣΘΗΤΗ ΠΑΡΑΒΛΕΨΗΣ ΚΑΤΑΣΤΑΣΕΩΝ	47

5.1. Βασικά Δεδομένα για Συγκρίσεις	47
5.2. Επίδραση $k$ , $S$ , $L$ στη Βελτίωση της Ακολουθίας Ελέγχου	50
5.3. Κόστος Υλικού του Κυκλώματος Παράβλεψης	52
5.4. Αποτελέσματα Προτεινόμενης Μεθόδου	55
5.5. Συγκρίσεις	57
5.6. Κόστος Υλικού	62
ΚΕΦΑΛΑΙΟ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ	64
ΑΝΑΦΟΡΕΣ	66
ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ ΣΗΜΕΙΩΜΑ	69

## **ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ**

---

Πίνακας 2-1: Κριτήρια επίλυσης γραμμικών συστημάτων	13
Πίνακας 3-1: Παράδειγμα πίνακα	20
Πίνακας 3-2: Αποτελέσματα της μεθόδου παραγωγής μοτίβου.	35
Πίνακας 4-1: Αλγόριθμος απληστίας για επιλογή χρησίων τμημάτων	43
Πίνακας 5-1: Σύνολα Ελέγχου και LFSRs	48
Πίνακας 5-2: Αποτελέσματα της πρωτότυπης μεθόδου ανατροφοδότησης για 32 αλυσίδες ελέγχου	48
Πίνακας 5-3: Αποτελέσματα της πρωτότυπης μεθόδου ανατροφοδότησης για 64 αλυσίδες ελέγχου	49
Πίνακας 5-4: Αποτελέσματα της παρούσας μεθόδου για 32 αλυσίδες ελέγχου	55
Πίνακας 5-5: Αποτελέσματα της παρούσας μεθόδου για 64 αλυσίδες ελέγχου.	56
Πίνακας 5-6: Σύγκριση με [21]	58
Πίνακας 5-7: Σύγκριση με [22]	58
Πίνακας 5-8: Σύγκριση με [32]	59
Πίνακας 5-9: TSL και TDV αποτελέσματα για διάφορες μεθόδους	60
Πίνακας 5-10: Συγκρίσεις με Mintest, [33] και [40]	61

## ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

---

Σχήμα 2-1: Αρχιτεκτονική Ελέγχου με Χρήση Ολισθητών Ανάδρασης (LFSR).....	7
Σχήμα 2-2: Σειριακή εισαγωγή διανυσμάτων στις αλυσίδες ελέγχου .....	8
Σχήμα 2-3: Παράδειγμα Γραμμικών Συστημάτων LFSRs .....	10
Σχήμα 2-4: Βελτίωση κωδικοποίησης με $L > 1$ για 32 αλυσίδες ελέγχου.....	14
Σχήμα 2-5: Βελτίωση κωδικοποίησης με $L > 1$ για τις 64 αλυσίδες ελέγχου .....	15
Σχήμα 3-1: Αρχιτεκτονική Ανάδρασης για Μοντελοποίηση Μοτίβων .....	18
Σχήμα 3-2: Δύσκολα διανύσματα σε μια αρχική κατάσταση σύμφωνα με δύο αλγορίθμους επιλογής .....	23
Σχήμα 3-3: Κόστος υλικού συγκριτικά με το μέγεθος του μοτίβου $p$ .....	26
Σχήμα 3-4: Σύγκριση TDV για μεταβλητή παραμετρική απόδοση $y$ με την κλασική μέθοδο .....	27
Σχήμα 3-5: TDV σταθερό $L$ , $p$ μεταβλητό $y$ .....	29
Σχήμα 3-6: Σταθερό $y$ , $p$ μεταβλητό παράθυρο $L$ .....	30
Σχήμα 3-7: Σταθερό $L$ και $y$ , μεταβλητό $p$ .....	31
Σχήμα 3-8: Βελτίωση στο πλήθος αρχικών καταστάσεων με χρήση πολλαπλών μοτίβων σε μια αρχιτεκτονική.....	34
Σχήμα 4-1: Γραμμικός Ολισθητής Παράβλεψης Καταστάσεων (SSLFSR).....	41
Σχήμα 4-2: Προτεινόμενη αρχιτεκτονική αποκωδικοποιητή.....	44
Σχήμα 5-1: Βελτίωση του μήκους της ακολουθίας ελέγχου για διάφορες τιμές $k$ , $S$ .....	50
Σχήμα 5-2: Βελτίωση του μήκους της ακολουθίας ελέγχου για διάφορες τιμές του $k$ , $L$ (s13207).....	52
Σχήμα 5-3: Κόστος υλικού του κυκλώματος παράβλεψης κατάστασης.....	53
Σχήμα 5-4: Κόστος υλικού της Μονάδας Επιλογής .....	63

## ΠΕΡΙΛΗΨΗ

---

Στην διατριβή αυτή παρουσιάζονται δύο εφαρμογές των Ολισθητών Ανάδρασης με χρήση επιλεκτικής γραμμικής ανάδρασης στην αποκωδικοποίηση διανυσμάτων ελέγχου. Η επιλεκτική ανάδραση είναι ένας τρόπος να παρακάμπτεται το χαρακτηριστικό πολυώνυμο που υπολογίζει την επόμενη κατάσταση του ολισθητή. Στην εργασία αυτή θα γίνει χρήση της επιλεκτικής ανάδρασης από δύο εφαρμογές. Στην πρώτη, ο *γραμμικός ολισθητής παραγωγής μοτίβων* χρησιμοποιεί την επιλεκτική ανάδραση για να παράγει μοτίβα με σκοπό την καλύτερη συμπίεση των διανυσμάτων ελέγχου. Στη δεύτερη, ο *ολισθητής παράβλεψης* χρησιμοποιεί το *κύκλωμα παράβλεψης* αντί του κυκλώματος που εκφράζει το χαρακτηριστικό πολυώνυμο του ολισθητή, για υπολογισμό μελλοντικής κατάστασης του ολισθητή. Έτσι ο ολισθητής μπορεί να παραβλέπει προκαθορισμένο σταθερό αριθμό από καταστάσεις μέσα σε ένα κύκλο, υπολογίζοντας την κατάσταση που τις ακολουθεί. Είναι ιδανικός για εφαρμογές ενσωμάτωσης διανυσμάτων ελέγχου, αφού άχρηστα μέρη των ακολουθιών ελέγχου μπορούν να μειωθούν δραματικά. Στην εργασία παρουσιάζεται συστηματική μελέτη για την ελαχιστοποίηση των ακολουθιών ελέγχου των τεχνικών ενσωμάτωσης διανυσμάτων ελέγχου και προτείνεται μια μικρού κόστους αρχιτεκτονική αποκωδικοποίησης. Ο ολισθητής παράβλεψης εκμεταλλεύεται την υψηλή συμπίεση που παρέχουν οι τεχνικές ενσωμάτωσης διανυσμάτων ελέγχου και μειώνει σημαντικά το μήκος των ακολουθιών ελέγχου (έως και 97%) προσεγγίζοντας έτσι σε χρόνο ακόμη και τις τεχνικές συμπίεσης δεδομένων. Εκτενή πειραματικά αποτελέσματα της τεχνικής αυτής δείχνουν τη σημασία των αποτελεσμάτων της.

Τενέντες Βασίλειος του Ευαγγέλου και της Αναστασίας

MSc Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Μάιος 2007

Τίτλος Διατριβής: «Σχεδίαση Γραμμικών Ολισθητών με Επιλεκτική Ανάδραση»

Επιβλέπων Καθηγητής: Χρυσοβαλάντης Καβουσιανός, Λέκτορας Πανεπιστημίου  
Ιωαννίνων





## EXTENDED ABSTRACT IN ENGLISH

---

This dissertation presents two types of Linear Feedback Shift Register which use selective feedback to encode test vectors. The first type is called Pattern Generator Linear Feedback Shift Register and it consists of a small combinational linear circuit which is used in order to create a predefined pattern and consequently achieve better compression ratios. The second type is called State Skip LFSR and it consists of a primitive polynomial LFSR and a small combinational linear circuit, the State Skip circuit, which substitutes the polynomial and enables the LFSR to perform successive constant jumps in the sequence of its states. The LFSR is reseeded by the ATE and generates pseudorandom vector sequences which embed pre-computed test vectors in order to test IP cores. The major portion of these sequences does not embed test vectors and they are drastically shortened by using the State Skip circuit instead of the polynomial. Test set embedding using window based LFSR reseeding offers very low test data volume at the cost of very long pseudorandom sequences, whereas the State Skip LFSR reduces drastically these sequences, approaching even 97% reduction in some cases. Consequently test set embedding using State Skip LFSRs becomes an attractive test data compression technique. A systematic method for the minimization of the pseudorandom sequence length and low overhead decompressor architecture are presented. Extensive experimentation shows the test data volume – test sequence length advantages of the proposed method.

Tenentes E. Vasilios

MSc Computer Science Department, University of Ioannina, Greece, May 2007

Title of Dissertation: “Selective Feedback Linear Shift Registers”

Thesis Supervisor: Xrisovalantis Kavousianos

## ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

---

1.1. Σύγχρονη Σχεδίαση Ψηφιακών Συστημάτων με Ενσωματωμένους Πυρήνες

1.2. Ενσωματωμένοι Πυρήνες – Ενσωματωμένος Έλεγχος

1.3. Κατηγορίες Μεθόδων Ελέγχου Ενσωματωμένων Πυρήνων

1.4. Δομή Διατριβής

---

### 1.1. Σύγχρονη Σχεδίαση Ψηφιακών Συστημάτων με Ενσωματωμένους Πυρήνες

Η σχεδίαση σύγχρονων ολοκληρωμένων συστημάτων (System On Chips – SOC) έχει ως κύριο στόχο την επίτευξη της οικονομικής βιωσιμότητας και τεχνολογικής ανταγωνιστικότητας των τελικών προϊόντων. Δύο βασικοί παράγοντες που επηρεάζουν τόσο την βιωσιμότητα όσο και την ανταγωνιστικότητα των σχεδιαζόμενων συστημάτων είναι η αξιοπιστία και ο χρόνος που απαιτείται για την είσοδο των συστημάτων στην αγορά (Time To Market). Αδιαμφισβήτητη είναι η αναγκαιότητα των ενσωματωμένων πυρήνων (embedded cores) στην επίτευξη αυτών των στόχων.

Οι πυρήνες είναι προσχεδιασμένα κυκλώματα που ενσωματώνονται αυτούσια σε ένα σύστημα και εκτελούν κάποια συγκεκριμένη λειτουργία. Οι πυρήνες παρόλο που σχεδιάζονται, ελέγχονται και δοκιμάζονται από τους παροχείς τους (vendors), προωθούνται στην αγορά για να χρησιμοποιηθούν από τρίτους σαν τμήματα ψηφιακών συστημάτων. Η ενσωμάτωσή τους σε οποιοδήποτε σύστημα γίνεται εύκολα και γρήγορα μειώνοντας κατά πολύ τον χρόνο σχεδίασης του. Είναι λοιπόν, αξιόπιστα κυκλώματα που έχουν σχεδιαστεί, ελεγχθεί και δοκιμαστεί από τους σχεδιαστές τους. Ωστόσο, για να

είναι αξιόπιστο το τελικό σύστημα δεν αρκεί μόνο να είναι απαλλαγμένη από λάθη η σχεδίασή του, αλλά πρέπει να είναι απαλλαγμένη από σφάλματα και η κατασκευή του.

Αν και η χρήση πυρήνων αυξάνει κατακόρυφα την αξιοπιστία της σχεδίασης του συστήματος και μειώνει σημαντικά τον χρόνο σχεδίασης του, δημιουργεί μεγάλα προβλήματα στον έλεγχο ορθής λειτουργίας του, ο οποίος τελικά επιβεβαιώνει και την κατασκευαστική ακεραιότητα του. Το μεγαλύτερο πρόβλημα δημιουργείται από την βαθιά ενσωμάτωση των πυρήνων στο σύστημα, η οποία δυσκολεύει αρκετά την πρόσβαση σε αυτούς από εξωτερικούς ελεγκτές, μέσω των ακροδεκτών του ολοκληρωμένου. Επιπρόσθετα, πολλοί πυρήνες αποκρύπτουν τη δομή τους από τους σχεδιαστές των συστημάτων καθώς οι παροχείς τους (που συνήθως είναι διαφορετικοί από τους σχεδιαστές) δεν επιθυμούν να αποκαλύψουν συγκεκριμένα χαρακτηριστικά των πυρήνων (Intellectual Property – IP Cores). Ο μεγάλος όγκος δεδομένων ελέγχου, που πρέπει να εφαρμοσθούν στους πυρήνες, καθώς και οι υψηλές συχνότητες λειτουργίας των σύγχρονων συστημάτων οδηγούν στη δραματική αύξηση του χρόνου και κόστους ελέγχου και συνεπώς στην αύξηση του κόστους του συστήματος. Σύμφωνα με την ITRS (International Technology Roadmap for Semiconductors), μέχρι το 2012 το κόστος ελέγχου ενός τρανζίστορ θα είναι ίσο με αυτό της κατασκευής του.

## 1.2. Ενσωματωμένοι Πυρήνες – Ενσωματωμένος Έλεγχος

Ο ενσωματωμένος έλεγχος (Built-In Self Test - BIST) βοηθάει στην επίλυση των παραπάνω προβλημάτων. Συγκεκριμένα, μαζί με τους πυρήνες ενσωματώνονται στο σύστημα και μικρά κυκλώματα ελέγχου, τα οποία διευκολύνουν την διεξαγωγή του τελικού ελέγχου ορθής λειτουργίας στον κάθε πυρήνα χωριστά και μειώνουν κατά πολύ την εξάρτηση από τους ελεγκτές (ATEs). Διατηρώντας μικρό το κόστος υλικού των κυκλωμάτων ελέγχου, εξασφαλίζεται σημαντικό κέρδος σε χρόνο και κόστος ελέγχου που τελικά ελαχιστοποιεί το συνολικό κόστος του ολοκληρωμένου. Για τους παραπάνω λόγους, ο ενσωματωμένος έλεγχος αποτελεί μια λύση που έχει γίνει ευρέως αποδεκτή από την βιομηχανία και την επιστημονική κοινότητα.

Η βαθιά ενσωμάτωση των πυρήνων σε ένα σύστημα SoC δημιουργεί μεγάλα προβλήματα στην διεκπεραίωση του ελέγχου από τους εξωτερικούς ελεγκτές. Συγκεκριμένα, ο μικρός αριθμός ακροδεκτών δεν επιτρέπει την άμεση πρόσβαση σε οποιοδήποτε σημείο του συστήματος, οπότε δεν είναι εύκολη η πρόσβαση στις εισόδους/εξόδους των βαθιά ενσωματωμένων πυρήνων. Το πρόβλημα επιδεινώνεται σημαντικά όταν οι πυρήνες έχουν άγνωστη δομή (IP cores). Σε αυτή την περίπτωση μαζί με τους πυρήνες παρέχονται και διανύσματα ελέγχου, τα οποία πρέπει να εφαρμοσθούν σε κάθε έναν από αυτούς μετά την κατασκευή του τελικού ολοκληρωμένου συστήματος. Ένα διάνυσμα ελέγχου αποτελείται από δυαδικές τιμές 0, 1 και αδιάφορους όρους X. Οι δυαδικές τιμές πρέπει να εφαρμοσθούν στις εισόδους των πυρήνων και να φορτώσουν τα εσωτερικά στοιχεία μνήμης (flip flops) προκειμένου να ελεγχθούν όλα τα πιθανά κατασκευαστικά σφάλματα του κάθε πυρήνα ενός ολοκληρωμένου κυκλώματος. Οι αδιάφοροι όροι μπορούν να αντικατασταθούν από οποιαδήποτε τιμή 0 ή 1. Συνεπώς, τα σύνολα των διανυσμάτων ελέγχου έχουν μεγάλο μέγεθος για τον κάθε πυρήνα και αν συνυπολογίσουμε και τον μεγάλο αριθμό πυρήνων που ενσωματώνονται σε ένα σύστημα καταλαβαίνουμε ότι απαιτείται τεράστιος χώρος αποθήκευσης στον ελεγκτή. Επιπλέον, οι αυξημένες συχνότητες λειτουργίας των σχεδιαζόμενων συστημάτων, απαιτούν και ανάλογες δυνατότητες στο ρυθμό εφαρμογής των διανυσμάτων ελέγχου από τους ελεγκτές (At Speed Testing). Συνεπώς, απαιτείται πολύ μεγάλος χρόνος εφαρμογής των διανυσμάτων ελέγχου, ακόμη και από ιδιαίτερα προηγμένα συστήματα ελέγχου (Automatic Test Equipment-ATE). Οι περιορισμοί λειτουργίας των σύγχρονων ATE οδηγούν σε περαιτέρω αύξηση του χρόνου και κόστους ελέγχου, κάτι που τελικά αυξάνει δραματικά το κόστος του τελικού συστήματος.

Η βασική λύση στα παραπάνω προβλήματα είναι ο ενσωματωμένος έλεγχος (BIST). Η βασική του ιδέα είναι η ενσωμάτωση μικρών κυκλωμάτων ελέγχου στο σύστημα, τα οποία έχουν μικρό κόστος αλλά παράλληλα εξασφαλίζουν σημαντικό κέρδος σε χρόνο και κόστος ελέγχου και συνεπώς στο κόστος του ολοκληρωμένου συστήματος.

### 1.3. Κατηγορίες Μεθόδων Ελέγχου Ενσωματωμένων Πυρήνων

Τα τελευταία χρόνια η αλματώδης αύξηση της ενσωμάτωσης πυρήνων στη σχεδίαση ολοκληρωμένων συστημάτων (System On Chip) οδήγησε βιομηχανικούς και ακαδημαϊκούς ερευνητές στην προσπάθεια ανεύρεσης λύσεων για τα προβλήματα που δημιουργούνται κατά τον έλεγχο ορθής λειτουργίας των συστημάτων αυτών εξαιτίας της χρήσης τέτοιων πυρήνων. Οι μέθοδοι που αναπτύχθηκαν ανήκουν σε δύο μεγάλες κατηγορίες: α) μέθοδοι ελέγχου πυρήνων γνωστής δομής (soft ή firm Cores) και β) μέθοδοι ελέγχου πυρήνων άγνωστης δομής (Hard Cores). Στην πρώτη κατηγορία ανήκουν οι μέθοδοι που αξιοποιούν τη διαδικασία παραγωγής διανυσμάτων ελέγχου (Automatic Test Pattern Generation – ATPG) καθώς και τη διαδικασία εξομοίωσης σφαλμάτων (fault Simulation). Και οι δύο αυτές διαδικασίες προϋποθέτουν την γνώση της δομής του πυρήνα, με συνέπεια να αποδίδουν πολλούς βαθμούς ελευθερίας στη σχεδίαση του κυκλώματος ελέγχου. Από την άλλη μεριά, στη δεύτερη κατηγορία το μόνο διαθέσιμο δεδομένο είναι το σύνολο διανυσμάτων ελέγχου που παρέχει ο σχεδιαστής του πυρήνα και, το οποίο, θα πρέπει να εφαρμοστεί τελικά στον πυρήνα. Επομένως, η μόνη βελτιστοποίηση είναι η συμπίεση που μπορεί να εφαρμοσθεί στα διανύσματα ελέγχου, που δίδονται από τον κατασκευαστή του πυρήνα. Αρκετές τεχνικές έχουν προταθεί και σε αυτή την κατηγορία μεθοδολογιών. Οι περισσότερες από αυτές χρησιμοποιούν συνδυαστικούς ή ακολουθιακούς γραμμικούς αποκωδικοποιητές [3, 28, 31, 45]. Άλλες συμπιέζουν τα διανύσματα χρησιμοποιώντας διάφορους κώδικες. Οι πιο πρόσφατες τεχνικές μπορούν να κατηγοριοποιηθούν, σύμφωνα με την ιδιότητά τους να εφαρμόζουν τα διανύσματα που παράγουν σε πυρήνες που διαθέτουν μία αλυσίδα ελέγχου [6-8, 10, 15, 16, 23, 36, 41, 43] ή πολλές αλυσίδες ελέγχου [1, 27, 33, 40, 46, 47].

Οι τεχνικές ενσωμάτωσης διανυσμάτων ελέγχου κωδικοποιούν τα προϋπολογισμένα διανύσματα ελέγχου των πυρήνων σε μεγάλες ψευδοτυχαίες ακολουθίες, οι οποίες παράγονται πάνω στο κύκλωμα που πρόκειται να ελεγχθεί. Στο [42] μια τεχνική ενσωμάτωσης ελέγχου ανά κύκλο παρουσιάζεται, ενώ στα [18, 19] οι συγγραφείς χρησιμοποιούν απαριθμητές για να παράγουν τα διανύσματα. Στο [32]

παρουσιάζεται μια παραμετροποιήσιμη αρχιτεκτονική υλικού με αλληλοσυνδέσεις και δικτυακή μορφή (RIN – Rinconfigurable Interconnection Network), η οποία επιτυγχάνει πολύ μεγάλη συμπίεση των δεδομένων που πρέπει να αποθηκευτούν στον ελεγκτή, αλλά μεγάλο κόστος για την υλοποίηση του αποκωδικοποιητή. Παρόλο που η ενσωμάτωση διανυσμάτων χρειάζεται πολύ λίγα δεδομένα ελέγχου σε σχέση με τις τεχνικές χωρίς ενσωμάτωση, οι ακολουθίες ελέγχου που απαιτούνται για την αποκωδικοποίηση των διανυσμάτων είναι συνήθως απαγορευτικά μεγάλες. Η πολυφασική τεχνική ενσωμάτωσης που προτάθηκε στο [21] έχει μικρό κόστος σε υλικό και παράγει μικρότερες ακολουθίες ελέγχου από τις παραπάνω μεθόδους. Μια ακόμη μεγαλύτερη βελτίωση της ακολουθίας ελέγχου επιτυγχάνεται στο [22], με ελάχιστο παραπάνω κόστος σε δεδομένα ελέγχου. Ωστόσο, και οι δυο μέθοδοι [21] και [22] χρειάζονται πολύ μεγάλο χρόνο ελέγχου.

#### 1.4. Δομή Διατριβής

Στη διατριβή αυτή παρουσιάζεται, αρχικά, η κλασική αντιμετώπιση ενσωμάτωσης διανυσμάτων ελέγχου με χρήση Γραμμικών Ολισθητών Ανάδρασης (Linear Feedback Shift Registers - LFSR) και γνωστές βελτιώσεις της τεχνικής αυτής (κεφάλαιο 2). Στη συνέχεια (κεφάλαιο 3) θα παρουσιαστεί και θα προταθεί μια νέα αρχιτεκτονική Γραμμικού Ολισθητή Ανάδρασης (LFSR) με δυνατότητα επιλεκτικής ανάδρασης. Η επιλεκτική ανάδραση θα χρησιμοποιηθεί με διαφορετικό τρόπο για να επιτύχει δύο στόχους. Ο πρώτος της στόχος θα είναι η μείωση των δεδομένων ελέγχου δηλαδή η απόδοση της κωδικοποίησης των διανυσμάτων ελέγχου (κεφάλαιο 3). Για την επίτευξη αυτού του στόχου θα παρουσιαστεί μια αρχιτεκτονική επιλεκτικής ανάδρασης με τη χρήση ενός γραμμικού κυκλώματος, το οποίο έχει σαν στόχο την επαναληπτική παραγωγή προκαθορισμένων μοτίβων μέσα στην ψευδοτυχαία ακολουθία. Ο δεύτερος στόχος είναι η μείωση του μήκους της ψευδοτυχαίας ακολουθίας ελέγχου. Για την επίτευξη αυτού του στόχου θα παρουσιαστεί (κεφάλαιο 4) μια αρχιτεκτονική επιλεκτικής ανάδρασης με τη χρήση ενός γραμμικού κυκλώματος, το οποίο έχει τη δυνατότητα να υπολογίζει σε ένα κύκλο την επόμενη  $k$ -οστή κατάσταση του γραμμικού ολισθητή

(υπολογισμός που θα απαιτούσε  $k$  κύκλους). Ο νέος αυτός τύπος γραμμικού ολισθητής ανάδρασης με επιλεκτική ανάδραση θα ονομαστεί *Γραμμικός Ολισθητής Παράβλεψης* (State Skip Linear FeedBack Shift Register - SSLFSR) και το κύκλωμα ανάδρασης *Κύκλωμα Παράβλεψης* (State Skip Circuit).

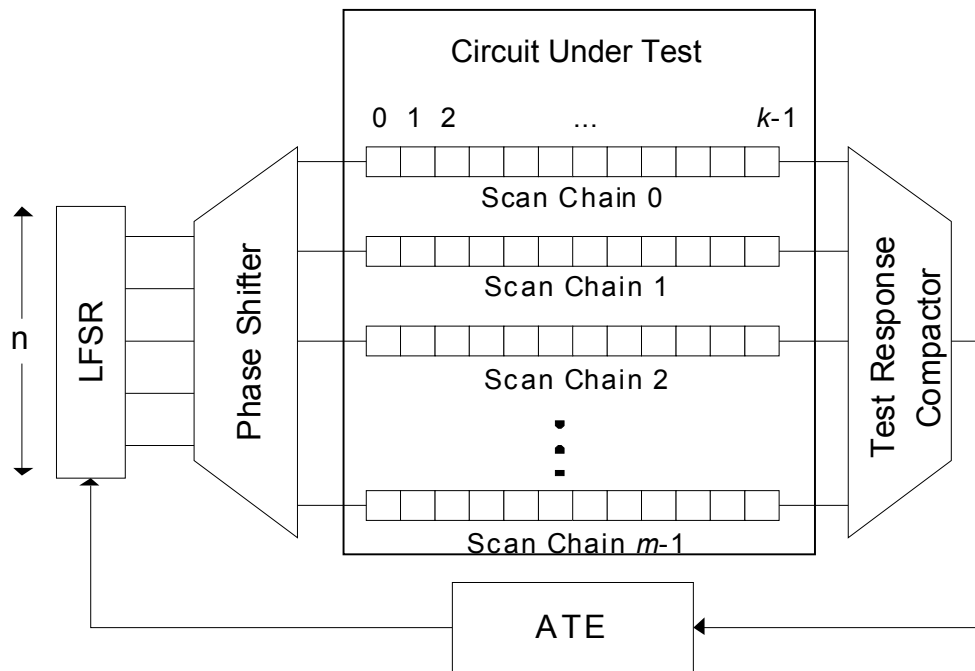
Εκτός από την γραμμική ανάδραση, που χαρακτηρίζει τους ολισθητές γραμμικής ανάδρασης, ο SSLFSR είναι εφοδιασμένος με ένα μικρό γραμμικό κύκλωμα που ονομάζεται *Κύκλωμα Παράβλεψης Καταστάσεων* (State Skip Circuit). Η ανάδραση του ολισθητή παραμετροποιείται, ώστε να μπορεί να λειτουργήσει τόσο με την κλασική λειτουργία του χαρακτηριστικού πολωνύμου (Normal Mode) όσο και με την ανάδραση που προκαλεί παράβλεψη καταστάσεων (State Skip Mode). Στη κατάσταση παράβλεψης, ο ολισθητής μέσω ανάδρασης, αυτοπροκαλεί μεταπηδήσεις σε επόμενες καταστάσεις που απέχουν ένα σταθερό μήκος μέσα στην ακολουθία καταστάσεων υπολογίζοντας την επόμενη από αυτές. Οι ολισθητές αυτού του τύπου ελαττώνουν δραματικά το μήκος των ακολουθιών των κλασικών μεθόδων ανατροφοδότησης με χρήση γραμμικών ολισθητών, επειδή μπορούν να λειτουργήσουν στην κατάσταση παράβλεψης για να προσπεράσουν γρήγορα τα άχρηστα τμήματα των ακολουθιών ελέγχου. Παρέχοντας τελικά υψηλή συμπίεση διανυσμάτων ελέγχου και ταυτόχρονα μείωση των ακολουθιών ελέγχου (έως και 97%), οι γραμμικοί ολισθητές παράβλεψης γεφυρώνουν το χάσμα των τεχνικών συμπίεσης διανυσμάτων ελέγχου και αυτών της ενσωμάτωσής τους. Οι προτεινόμενοι ολισθητές συνοδεύονται και από μία τεχνική διαμέρισης και διαχωρισμού των ακολουθιών ελέγχου σε χρήσιμα και άχρηστα τμήματα, η οποία οδηγεί σε σημαντική μείωση της ακολουθίας ελέγχου και σε απλή σχετικά αρχιτεκτονική με μικρό μέγεθος υλικού. Στο 5<sup>ο</sup> κεφάλαιο παρουσιάζονται αποτελέσματα και συγκρίσεις της μεθόδου μείωσης της ακολουθίας ελέγχου με τις καλύτερες τεχνικές στη βιβλιογραφία.



## ΚΕΦΑΛΑΙΟ 2. ΟΛΙΣΘΗΤΕΣ ΓΡΑΜΜΙΚΗΣ ΑΝΑΔΡΑΣΗΣ – LFSRS

- 
- 2.1. Αρχιτεκτονική Ελέγχου Βασισμένη σε Ολισθητές Γραμμικής Ανάδρασης
  - 2.2. Υπολογισμός Αρχικής Κατάστασης Ολισθητή Γραμμικής Ανάδρασης
  - 2.3. Βελτιστοποιήσεις του Ολισθητή Γραμμικής Ανάδρασης
- 

### 2.1. Αρχιτεκτονική Ελέγχου Βασισμένη σε Ολισθητές Γραμμικής Ανάδρασης



Σχήμα 2-1: Αρχιτεκτονική Ελέγχου με Χρήση Ολισθητών Ανάδρασης (LFSR)

Η θεμελιώδης και πιο κοινή αρχιτεκτονική ελέγχου με χρήση LFSR φαίνεται στο σχήμα 2.1. Κάθε διάνυσμα των  $N$  ψηφίων μεταφέρεται από τον ελεγκτή (ATE) στο LFSR, όπου επεκτείνεται σε  $m \cdot k$  ψηφία (το διάνυσμα φορτώνεται σε  $m$  αλυσίδες ελέγχου, μήκους  $k$  η κάθε μία, του υπό έλεγχο κυκλώματος). Επειδή οι ακολουθίες ψηφίων που παράγονται από τα κελιά του LFSR παρουσιάζουν γραμμικές εξαρτήσεις, ένα γραμμικό κύκλωμα που ονομάζεται Ολισθητής Φάσης (Phase Shifter) παρεμβάλλεται μεταξύ του LFSR και των αλυσίδων ελέγχου, ώστε να μειωθούν αυτές οι εξαρτήσεις. Ταυτόχρονα με τη φόρτωση ενός διανύσματος στις αλυσίδες ελέγχου, η απόκριση της εφαρμογής του προηγούμενου διανύσματος ελέγχου ολισθαίνει σταδιακά προς τον ελεγκτή των αποκρίσεων ελέγχου (Test Response Compactor).

	capture	shift	shift	1 0 1 1	0 0 0 1 1 1
cycle	1	2	3	shift 4	capture 5
sc 1	x x x	1 x x	0 1 x	1 0 1	0 1 0
sc 2	x x x	0 x x	0 0 x	1 0 0	1 0 0
sc 3	x x x	1 x x	1 1 x	0 1 1	1 0 0
				1 0 1 0	1 1 1 0 0 0
		shift 6	shift 7	shift 8	capture 9
cycle		6	7	8	9
sc 1		1 0 1	0 1 0	1 0 1	1 0 0
sc 2		1 1 0	1 1 1	0 1 1	1 1 1
sc 3		0 1 0	0 0 1	1 0 0	0 1 0

Σχήμα 2-2: Σειριακή εισαγωγή διανυσμάτων στις αλυσίδες ελέγχου

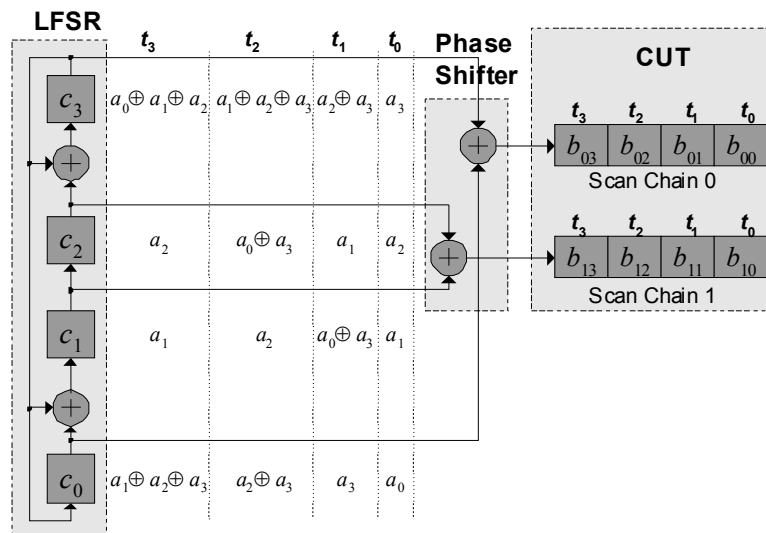
Στο παραπάνω σχήμα φαίνονται τα περιεχόμενα των αλυσίδων ελέγχου καθώς ένας ολισθητής φορτώνεται με μια αρχική κατάσταση και προχωράει στην σειριακή φόρτωση των αλυσίδων ελέγχου με τα ψηφία που παράγει. Στο παράδειγμα αυτό, τα διανύσματα που παράγονται έχουν μέγεθος 9 ψηφία και φορτώνονται σε τρεις αλυσίδες ελέγχου εισάγοντας σειριακά διανύσματα ελέγχου μέσα στο υπό έλεγχο κύκλωμα. Ταυτόχρονα, όμως με την σειριακή εισαγωγή των διανυσμάτων γίνεται και σειριακή εξαγωγή αποκρίσεων διανυσμάτων που εφαρμόστηκαν. Από αριστερά προς τα δεξιά στο σχήμα βλέπουμε ότι αρχικά στον κύκλο 0 τα περιεχόμενα των αλυσίδων είναι αδιάφορα και θεωρούμε ότι η αρχική κατάσταση φορτώνεται στα κελιά του ολισθητή. Στην πορεία

για τρεις κύκλους λαμβάνονται ψηφία από τον ολισθητή και φορτώνουν τις τρεις αλυσίδες ελέγχου. Έτσι μετά από τρεις κύκλους, στον 4<sup>ο</sup> κύκλο, παράγεται το διάνυσμα. Στην συνέχεια το διάνυσμα αυτό εφαρμόζεται στο υπό έλεγχο κύκλωμα. Αυτό συμβαίνει τον 5<sup>ο</sup> κύκλο που ονομάζεται κύκλος σύλληψης (capture cycle). Κατά τον κύκλο αυτό δίνεται η δυνατότητα στις αποκρίσεις του υπό έλεγχο κυκλώματος στο διάνυσμα εισόδου  $\delta_1$  να φορτωθούν στην αλυσίδα ελέγχου και να εξαχθούν προς τον αναλυτή των αποκρίσεων ταυτόχρονα με τις ολισθήσεις που θα ακολουθήσουν για την εισαγωγή του επόμενου διανύσματος μέσα στο υπο έλεγχο κύκλωμα. Ο κύκλος σύλληψης, λοιπόν είναι ο κύκλος κατά τον οποίο δεν λαμβάνονται οι τιμές από τις εξόδους του ολισθητή ανάδρασης, διότι στον κύκλο αυτό στις αλυσίδες ελέγχου γράφονται οι αποκρίσεις του υπο έλεγχο κυκλώματος. Οι υπόλοιποι κύκλοι, πλην των κύκλων σύλληψης λέγονται κύκλοι ολίσθησης (shift cycles). Όμοια, στους επόμενους κύκλους εισάγεται σειριακά το  $\delta_2$  διάνυσμα και στον 9<sup>ο</sup> κύκλο λαμβάνεται η απόκριση του υπό έλεγχο κυκλώματος στο  $\delta_2$ .

## 2.2. Υπολογισμός Αρχικής Κατάστασης Ολισθητή Γραμμικής Ανάδρασης

Η αρχική κατάσταση (seed- η δυαδικά ψηφία), που εισάγεται στον ολισθητή, είναι συμπιεσμένη έκδοση ενός ή περισσοτέρων διανυσμάτων ελέγχου. Πολύ μεγάλη συμπίεση ( $n \ll m \cdot k$ ) επιτυγχάνεται όταν τα διανύσματα περιέχουν μεγάλο αριθμό από αδιάφορους όρους – ‘x’ ψηφία (διανύσματα ελέγχου που έχουν ‘x’ ψηφία ονομάζονται κύβοι ελέγχου – test cubes). Κάθε αρχική κατάσταση κωδικοποιεί ένα κύβο ελέγχου και υπολογίζεται λύνοντας ένα σύστημα γραμμικών εξισώσεων σύμφωνα με τα καθορισμένα ψηφία του κύβου ελέγχου [24] ( στα ‘x’ ψηφία τοποθετούνται ψευδοτυχαίοι αριθμοί κατά την αποκωδικοποίηση). Πιο συγκεκριμένα: η αρχική κατάσταση του LFSR θεωρείται σαν ένα σύνολο από δυαδικές μεταβλητές  $a_0, a_1, \dots, a_{n-1}$  όπου  $n$  είναι το μήκος του ολισθητή. Σε κάθε κύκλο ρολογιού  $m$  γραμμικές εκφράσεις των μεταβλητών αυτών σχηματίζονται στις  $m$  εξόδους του ολισθητή φάσης, που οδηγούν στις αλυσίδες ελέγχου. Έτσι, κάθε ψηφίο των κύβων ελέγχου αντιστοιχίζεται με ακριβώς μία τέτοια έκφραση. Κάθε έκφραση, η οποία αντιστοιχίζεται με καθορισμένο ψηφίο (0 ή 1) στον κύβο

ελέγχου, πρέπει να τεθεί ίση με αυτό το ψηφίο. Με τον τρόπο αυτό σχηματίζεται ένα σύστημα από γραμμικές εξισώσεις και η λύση αυτού του συστήματος είναι η αρχική κατάσταση του LFSR.



Σχήμα 2-3: Παράδειγμα Γραμμικών Συστημάτων LFSRs

*Παράδειγμα:* Στο σχήμα 2.3 ένας γραμμικός ολισθητής ανάδρασης με 4 κελιά ( $c_0, c_1, c_2, c_3$ ) και έναν ολισθητή φάσης που αποτελείται από δύο XOR πύλες χρησιμοποιούνται για την φόρτωση δύο αλυσίδων ελέγχου. Το μήκος των δύο αλυσίδων 0 και 1 είναι ίσο με 4, και τα κελιά κάθε μιας από αυτές συμβολίζονται ως  $b_{00} \dots b_{03}$  και  $b_{10} \dots b_{13}$  αντίστοιχα. Στον πρώτο κύκλο ρολογιού ( $t_0$ ) η αρχική κατάσταση του ολισθητή είναι  $(c_0, c_1, c_2, c_3) = (a_0, a_1, a_2, a_3)$ . Έτσι οι πρώτες δύο εκφράσεις που παράγονται στις εξόδους του ολισθητή φάσης είναι  $a_0 \oplus a_3$  και  $a_1 \oplus a_2$ , οι οποίες φορτώνονται στα κελιά  $b_{00}, b_{10}$  αντίστοιχα (το σύμβολο  $\oplus$  δηλώνει αποκλειστικό-Η λογική πράξη). Κατά το δεύτερο κύκλο του ρολογιού ( $t_1$ ) η καταστάσεις στα κελιά του LFSR γίνονται  $(c_0, c_1, c_2, c_3) = (a_3, a_0 \oplus a_3, a_1, a_2 \oplus a_3)$ , σύμφωνα με το χαρακτηριστικό πολυώνυμο και επομένως  $b_{01} = a_2, b_{11} = a_0 \oplus a_1 \oplus a_3$ . Με τον ίδιο τρόπο και τα τους  $t_2$  ( $t_3$ ) έχουμε  $b_{02} = a_1, b_{12} = a_0 \oplus a_2 \oplus a_3$  ( $b_{03} = a_0 \oplus a_3, b_{13} = a_1 \oplus a_2$ ).

Ας υποθέσουμε τώρα τον κύβο ελέγχου  $T = T_{00}T_{01}T_{02}T_{03}T_{10}T_{11}T_{12}T_{13} = \text{xx1xxx00}$ , όπου  $T_{ij}$  συμβολίζει το κελί στις αλυσίδες ελέγχου  $b_{ij}$ . Για να καθοριστεί μια αρχική κατάσταση που θα παράγει το  $T$ , πρέπει να λυθεί το ακόλουθο γραμμικό σύστημα:

$$\begin{bmatrix} b_{02} = 1 \\ b_{12} = 0 \\ b_{13} = 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} a_1 = 1 \\ a_0 \oplus a_2 \oplus a_3 = 1 \\ a_1 \oplus a_2 = 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} a_1 = 1 \\ a_0 = a_3 \\ a_2 = 1 \end{bmatrix}$$

Αντικαθιστώντας το  $a_3$  είτε με 0 είτε με 1, η αρχική κατάσταση 0110 ή η 1111 γίνεται η λύση του συστήματος. Και οι δύο παραπάνω καταστάσεις παράγουν το ζητούμενο κύβο ελέγχου.

### 2.3. Βελτιστοποιήσεις του Ολισθητή Γραμμικής Ανάδρασης

Έχουν παρουσιαστεί κατά καιρούς πολλές βελτιώσεις της κωδικοποίησης με ολισθητές ανάδρασης, οι οποίες στοχεύουν στην ενίσχυση των δυνατοτήτων του ολισθητή να ενσωματώνει διανύσματα.

#### 2.3.1. Ελαχιστοποίηση Μήκους Ολισθητή

Μια πολύ σημαντική παράμετρος που επηρεάζει την απόδοση της μεθόδου συμπίεσης είναι το μήκος του LFSR ή με άλλα λόγια το πλήθος των δυαδικών μεταβλητών που διατίθενται για την επίλυση του συστήματος. Εάν για παράδειγμα, το πλήθος των μεταβλητών είναι μικρότερο από το πλήθος των εξισώσεων τότε το σύστημα μπορεί πιθανώς να μην λύνεται. Το σύστημα με το μέγιστο πλήθος εξισώσεων ανήκει στον κύβο ελέγχου με το μεγαλύτερο πλήθος ορισμένων ψηφίων (specified bits – αυτά που δεν είναι ‘x’)  $s_{ax}$ , το οποίο με τη σειρά του καθορίζει το ελάχιστο απαιτούμενο μήκος LFSR. Έχει αποδειχθεί στο [24] ότι εάν το μήκος του LFSR  $n$  είναι ίσο με  $s_{ax}+20$ , τότε η πιθανότητα να είναι αδύνατο το σύστημα των εξισώσεων που προκύπτει για τη κωδικοποίηση ενός κύβου ελέγχου είναι λιγότερη από  $10^{-6}$ .

Ωστόσο στην πράξη μπορούμε, συνήθως, να βρούμε χαρακτηριστικά πολυώνυμα με μέγεθος μικρότερο από  $s_{max}+20$ , τα οποία κωδικοποιούν όλους τους κύβους ελέγχου [34].

### 2.3.2. Επέκταση της Ψευδοτυχαίας Ακολουθίας σε Παράθυρο

Ακόμα και μετά την ελαχιστοποίηση του μήκους του LFSR, υπάρχουν πολλά περιθώρια βελτίωσης της συμπίεσης. Αυτό είναι συνέπεια της σπανιότητας των καθορισμένων ψηφίων μέσα σε ένα κύβο ελέγχου. Συνήθως, υπάρχουν πολλοί κύβοι ελέγχου με μικρότερο πλήθος ορισμένων ψηφίων από  $s_{max}$ . Σαν αποτέλεσμα, πολλές μεταβλητές μένουν «άλυτες» μετά τη λύση των αντίστοιχων συστημάτων και δεν εξαντλούνται οι δυνατότητες συμπίεσης του LFSR.

Πολλές μέθοδοι έχουν προταθεί για την λύση αυτού του προβλήματος, όπως η χρήση πολλαπλών πολυωνύμων ανάδρασης και συνένωση των κύβων ελέγχου [13], χρήση πολλαπλών ολισθητών με διαφορετικά μήκη [49], μερική [26] ή δυναμική [38] ανατροφοδότηση, για να αναφέρουμε λίγες από αυτές. Μια τακτική βελτίωσης της συμπίεσης που έχει πολύ καλά αποτελέσματα, χρησιμοποιεί την ίδιο αρχική κατάσταση για την κωδικοποίηση περισσότερων από έναν κύβων ελέγχου σε μια ακολουθία από  $L$  ψευδοτυχαίων διανυσμάτων. Με άλλα λόγια η ακολουθία κάθε αρχικής κατάστασης επεκτείνεται σε ένα παράθυρο από  $L$  διαδοχικά διανύσματα, αντί του ενός. Το πλήθος των κύβων ελέγχου που κωδικοποιούνται μέσα στο παράθυρο είναι πολύ μικρότερο από  $L$ , που σημαίνει ότι άχρηστα διανύσματα εφαρμόζονται επίσης στο υπό έλεγχο κύκλωμα.

Αυτή η μέθοδος είναι πολύ αποδοτική, αφού για κάθε κύβο ελέγχου  $L > 1$  συστήματα γραμμικών εξισώσεων κατασκευάζονται και ανάμεσα σε αυτά που οδηγούν σε λύση μπορεί να επιλεγθεί αυτό που επιλύει τις λιγότερες μεταβλητές και συνεπώς παρέχει την καλύτερη συμπίεση. Με άλλα λόγια κάθε κύβος ελέγχου κωδικοποιείται με τέτοιο τρόπο, ώστε να μεγιστοποιεί την απόδοση της κωδικοποίησης.

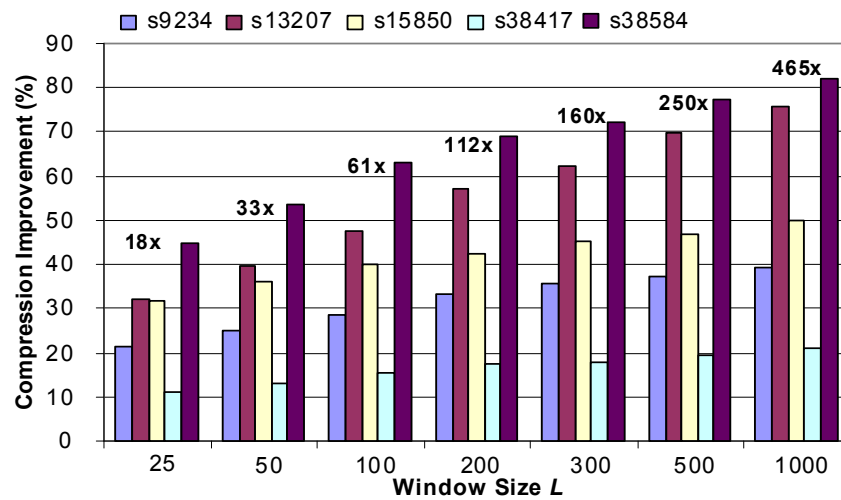
### 2.3.3. Κριτήρια Επίλυσης Γραμμικών Συστημάτων για Μέγιστη Συμπύεση

Προς αυτή την κατεύθυνση στο [22] τα παρακάτω τέσσερα κριτήρια επιλογής συστήματος επιλέχθηκαν:

Κριτ.	Περιγραφή
1	Επέλεξε από τα συστήματα που επιλύονται αυτά που αντιστοιχούν στους κύβους ελέγχου που περιέχουν το μεγαλύτερο πλήθος από ορισμένα ψηφία
2	Εάν υπάρχουν περισσότερα του ενός συστήματα που επιλέχθηκαν από το πρώτο κριτήριο, επέλεξε αυτό που η επίλυσή του αντιστοιχίζεται στις λιγότερες αντικαταστάσεις μεταβλητών μέσα στο L μήκους παράθυρο
3	Εάν υπάρχουν περισσότερα του ενός συστήματα που επιλέχθηκαν από το δεύτερο κριτήριο επέλεξε αυτό που λύνεται τις λιγότερες φορές μέσα στο παράθυρο
4	Εάν υπάρχουν περισσότερα του ενός συστήματα που επιλέχθηκαν από το τρίτο κριτήριο, επέλεξε αυτό που βρίσκεται πιο κοντά στο πρώτο διάνυσμα μέσα στο παράθυρο L

Πίνακας 2-1: Κριτήρια επίλυσης γραμμικών συστημάτων

Τα πρώτα τρία κριτήρια μεγιστοποιούν την απόδοση της συμπύεσης, ενώ το τελευταίο μειώνει το μήκος της ακολουθίας ελέγχου. Πρέπει να σημειωθεί ότι για το πρώτο κύβο ελέγχου που κωδικοποιείται, μόνο το πρώτο κριτήριο εφαρμόζεται και επιλέγεται πάντα το σύστημα που αντιστοιχεί στο πρώτο διάνυσμα του παραθύρου. Η επαναληπτική εφαρμογή των παραπάνω κριτηρίων οδηγεί στην σταδιακή δημιουργία μιας αρχικής κατάστασης, που μπορεί να παράγει περισσότερους του ενός κύβους ελέγχου (ένα σύστημα που ανήκει σε διαφορετικό κύβο ελέγχου επιλέγεται σε κάθε επανάληψη και έτσι κάποιες μεταβλητές που παραμένουν άλυτες μετά την κωδικοποίηση του προηγούμενου κύβου ελέγχου θέτονται σε 0 ή 1). Ο υπολογισμός μιας αρχικής κατάστασης ολοκληρώνεται όταν κανένα σύστημα, για οποιοδήποτε από τους κύβους ελέγχους που δεν έχουν κωδικοποιηθεί, δεν μπορεί να επιλυθεί σε καμία θέση μέσα στο παράθυρο των L διανυσμάτων.



Σχήμα 2-4: Βελτίωση κωδικοποίησης με  $L > 1$  για 32 αλυσίδες ελέγχου

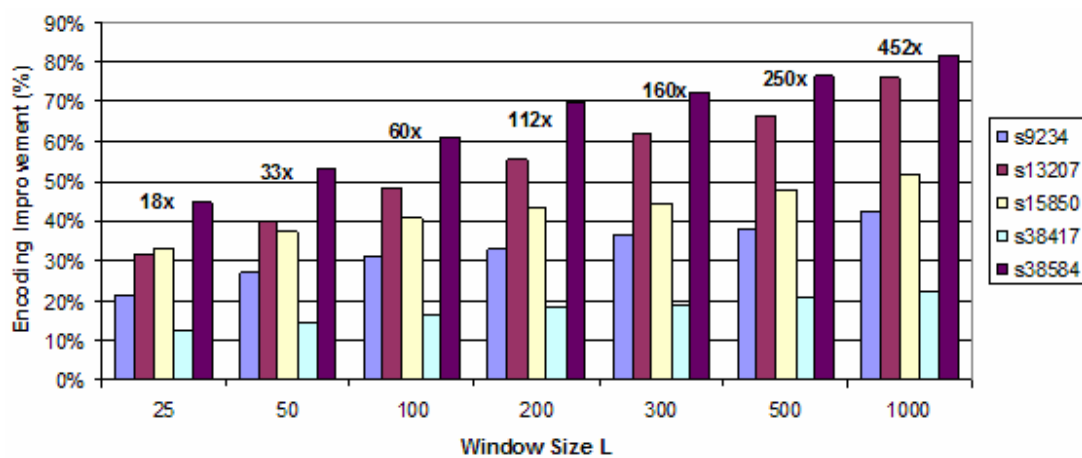
Για να παρουσιαστεί η ανωτερότητα της μεθόδου που επεκτείνει κάθε αρχική κατάσταση σε παράθυρο από  $L > 1$  διανυσμάτων, έναντι της κλασικής κωδικοποίησης στην οποία κάθε αρχική κατάσταση παράγει 1 διάνυσμα ( $L = 1$ ), παρουσιάζεται το εξής πείραμα: Ασυμπίεστα σύνολα διανυσμάτων ελέγχου που παράγονται από το εργαλείο ψηφιακής σχεδίασης Atalanta (λεπτομέρειες για τα διανύσματα μπορούν να βρεθούν στο 5<sup>ο</sup> κεφάλαιο) για τα μεγαλύτερα ISCAS 89 κυκλώματα σύγκρισης επιδόσεων κωδικοποιήθηκαν με την κλασική μέθοδο ( $L = 1$ ) καθώς επίσης και με χρήση διάφορων μεγεθών παραθύρου  $L=25, 50, 100, 200, 300, 500$  και  $1000$  θεωρώντας 32 αλυσίδες ελέγχου. Και στις δύο περιπτώσεις κάθε αρχική κατάσταση μπορεί να κωδικοποιεί όσους κύβους ελέγχου είναι δυνατόν. Η βελτίωση της συμπίεσης για κάθε κωδικοποίηση με  $L \geq 25$ , σε σύγκριση με την κωδικοποίηση με  $L = 1$  υπολογίστηκε με βάση τον ακόλουθο τύπο:

$$\text{Compr. Impr.}(L) \% = \frac{\text{TestBits}(\text{WindowSize} = 1) - \text{TestBits}(\text{WindowSize} = L)}{\text{TestBits}(\text{WindowSize} = 1)} \cdot 100$$

, όπου TestBits είναι το πλήθος των ψηφίων που χρειάζονται για την αποθήκευση των αρχικών καταστάσεων. Τα αποτελέσματα παρουσιάζονται στο σχήμα 2.4 και καθαρά δείχνουν ότι καθώς το μέγεθος του παραθύρου  $L$  αυξάνει η κωδικοποίηση διαρκώς βελτιώνεται (για κάποια κυκλώματα η βελτίωση είναι πολύ μεγάλη). Για κάθε κύκλωμα



ο λόγος του μήκους της ακολουθίας ελέγχου για  $L \geq 25$  προς το μήκος της ακολουθίας για  $L = 1$  έχει επίσης υπολογιστεί. Ο μέσος όρος των λόγων αυτών, για όλα τα κυκλώματα για κάθε μήκος παραθύρου  $L$ , φαίνεται πάνω από τις αντίστοιχες στήλες στο σχήμα. Είναι προφανές ότι καθώς το  $L$  αυξάνει ο χρόνος που απαιτείται για τον έλεγχο αυξάνει δραματικά και γίνεται απαγορευτικά μεγάλος, ειδικά για πολύ μεγάλα παράθυρα (για  $L = 1000$  η ακολουθία ελέγχου είναι 465 φορές μεγαλύτερη από αυτή για  $L = 1$ ).



Σχήμα 2-5: Βελτίωση κωδικοποίησης με  $L > 1$  για τις 64 αλυσίδες ελέγχου

## ΚΕΦΑΛΑΙΟ 3. ΠΑΡΑΓΩΓΗ ΜΟΤΙΒΩΝ ΜΕ ΟΛΙΣΘΗΤΗ ΓΡΑΜΜΙΚΗΣ ΑΝΑΔΡΑΣΗΣ

- 
- 3.1. Αρχιτεκρονική Παραγωγής Μοτίβων με Χρήση Επιλεκτικής Ανάδρασης
  - 3.2. Επιλογή Μοτίβου
  - 3.3. Κωδικοποίηση Σε Παράθυρο L Με Χρήση Μοτίβου
  - 3.4. Αρχιτεκτονική Αποκωδικοποίησης
  - 3.5. Πειραματικά Αποτελέσματα
  - 3.6. Αρχιτεκτονική για Πολλαπλά Μοτίβα
  - 3.7. Συμπεράσματα
  - 3.8. Μελλοντική Μελέτη
- 

Στο κεφάλαιο αυτό παρουσιάζεται η μέθοδος μοντελοποίησης μοτίβων η οποία αποτελεί μια εφαρμογή της επιλεκτικής ανάδρασης για βελτίωση της συμπίεσης των διανυσμάτων ελέγχου. Θα παρουσιαστεί η αρχιτεκτονική για τη μοντελοποίηση μοτίβων, όπως παράγεται όταν στο κλασσικό σχήμα του ολισθητή ανάδρασης προσαρτηθεί ένα μικρό κύκλωμα επιλεκτικής ανάδρασης που δίνει στον ολισθητή τη δυνατότητα να αναπαράγει προκαθορισμένα μοτίβα σε κάθε παράθυρο χωρίς να προκαλείται μείωση της δυνατότητας κωδικοποίησης του ολισθητή. Θα παρουσιαστούν εναλλακτικοί αλγόριθμοι επιλογής μοτίβων και διάφορες τακτικές εναλλαγής μοτίβων κατά την αποκωδικοποίηση για μεγαλύτερη συμπίεση.

### 3.1. Αρχιτεκρονική Παραγωγής Μοτίβων με Χρήση Επιλεκτικής Ανάδρασης

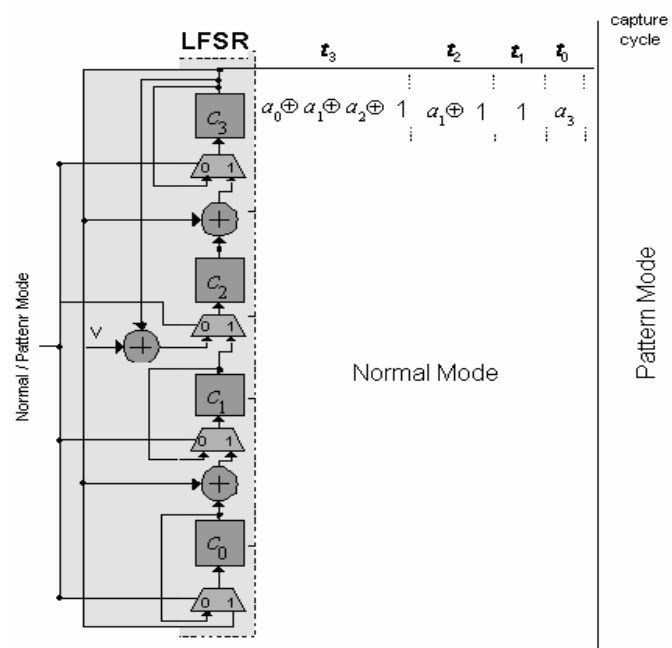
Για να γίνει αντιληπτή η λειτουργία της επιλεκτικής ανάδρασης με δυνατότητα αναπαραγωγής μοτίβων πρέπει να παρουσιαστούν αναλυτικά τα περιεχόμενα των κελιών του LFSR σύμφωνα με το σχήμα 2.2. Για να είναι πιο απλοί οι υπολογισμοί, χωρίς να

βλάψουμε τη γενικότητα θα επικεντρωθούμε σε απλό LFSR χωρίς χρήση ολισθητή φάσης και θα θεωρήσουμε ότι η ψευδοτυχαία ακολουθία ελέγχου φορτώνεται σε μία αλυσίδα ελέγχου. Έστω ότι η αλυσίδα ελέγχου παίρνει δεδομένα από το κελί  $c_3$  του σχήματος 2.2. Στον πρώτο κύκλο  $t_0$  εισάγεται η τιμή  $c_3(t_0) = a_3$  στην αλυσίδα ελέγχου, στο δεύτερο κύκλο η τιμή  $c_3(t_1) = a_2 \oplus a_3$ , στον τρίτο κύκλο η τιμή  $c_3(t_2) = a_1 \oplus a_2 \oplus a_3$  κτλ. Έστω ότι πρέπει υπολογιστεί μια αρχική κατάσταση που κωδικοποιεί τους παρακάτω κύβους ελέγχου:  $\delta = (x1xx, 11xx, x1x0)$ . Ας δούμε αναλυτικά πως θα γίνει αυτό με την κλασσική μέθοδο για παράθυρο  $L=1$ . Για το  $\delta_0$  σχηματίζεται το σύστημα  $a_0 \oplus a_1 \oplus a_2 = x$ ,  $a_1 \oplus a_2 \oplus a_3 = x$ ,  $a_2 \oplus a_3 = 1$  και  $a_3 = x$  για το  $\delta_1$   $a_0 \oplus a_1 \oplus a_2 = x$ ,  $a_1 \oplus a_2 \oplus a_3 = x$ ,  $a_2 \oplus a_3 = 1$  και  $a_3 = 1$ , και για το  $\delta_2$   $a_0 \oplus a_1 \oplus a_2 = 0$ ,  $a_1 \oplus a_2 \oplus a_3 = x$ ,  $a_2 \oplus a_3 = 1$  και  $a_3 = x$ . Οι σχέσεις που οδηγούν σε  $x$  δεν χρειάζονται επίλυση γιατί τα  $x$  δηλώνουν 0 ή 1 οπότε είναι πάντα αληθής. Θέλοντας να βρούμε μια κοινή λύση για όλα τα συστήματα έχουμε το εξής σύστημα:

$$\begin{array}{ll} a_2 \oplus a_3 = 1 & a_2 = 0 \\ a_3 = 1 & a_3 = 1 \\ a_0 \oplus a_1 \oplus a_2 = 0 & \text{άρα} \\ & a_0 \oplus a_1 = 0 \end{array}$$

Άρα μπορούν να κωδικοποιηθούν όλοι οι κύβους ελέγχου στο πρώτο ψευδοτυχαίο διάνυσμα του LFSR χρησιμοποιώντας όλες τις μεταβλητές του. Παρατηρώντας καλύτερα τους κύβους ελέγχου θα δούμε ότι έχουν όλοι ένα κοινό χαρακτηριστικό. Το δεύτερο ψηφίο τους είναι 1. Αυτό θα είναι και το μοτίβο επανάληψης που το κύκλωμα ανάδρασης θα παράγει. Το κύκλωμα ανάδρασης θα αναλαμβάνει σε ένα κύκλο ολίσθησης να αντικαταστήσει το χαρακτηριστικό πολυώνυμο και να προκαλέσει τέτοιες γραμμικές αλλαγές στις εσωτερικές καταστάσεις του LFSR, ώστε την κατάλληλη χρονική στιγμή (δηλ. στην περίπτωσή μας μετά από ένα κύκλο) η επόμενη τιμή που θα εισαχθεί στην αλυσίδα ελέγχου θα είναι 1. Από το μοτίβο  $xx1x$  προκύπτει το σύστημα  $a_2 \oplus a_3 = 1$  δηλαδή  $a_2 = a_3 \oplus 1$ . Συνεπώς, μπορούμε με μια νέα νέα ανάδραση να κρατάμε σε όλα τα κελιά την ίδια τιμή εκτός από το κελί  $c_2$ , στο οποίο, αν υποθέσουμε ότι ο κύκλος εκτέλεσης είναι  $t$ , θα θέτει το αντίθετο ψηφίο από αυτό του τρίτου κελιού  $c_3(t) \oplus 1$  (σχήμα 3-1). Με τον

τρόπο αυτό θα προκαλέσουμε την εμφάνιση του 1 δύο κύκλους μετά. Παρακάτω φαίνεται σχηματικά αυτή η αρχιτεκτονική. Στον κύκλο που εγκλωβίζονται οι τιμές (Capture Cycle) το LFSR τίθεται σε κατάσταση λειτουργίας εκτέλεσης μοτίβου. Η γραμμική πράξη που προκαλεί την αποθήκευση στο κελί  $c_2$  το αντίθετο του κελιού  $c_3$  εκτελείται, γεγονός που έχει σαν αποτέλεσμα την αλλαγή των εξισώσεων στην κανονική κατάσταση λειτουργίας. Ας προσπαθήσουμε να κωδικοποιήσουμε πάλι το προηγούμενο σύνολο από κύβους ελέγχου. Το  $\delta_0$  κωδικοποιείται χωρίς λύση συστήματος, το  $\delta_1$  δίνει την εξίσωση  $a_3=1$  και το  $\delta_2$  την εξίσωση  $a_0 \oplus a_1 \oplus a_2 \oplus 1=1$ . Οι εξισώσεις με πρώτη ματιά φαίνονται πιο λίγες, αλλά δεν πρέπει να ξεχνάμε την ενσωματωμένη εξίσωση  $a_3 = a_2 \oplus 1$  οπότε προκύπτει το ίδιο παραπάνω σύστημα. Την ίδια διαδικασία μπορούμε να εφαρμόσουμε σε οποιοδήποτε κύκλο σύλληψης και σε περίπτωση όπου  $L \geq 0$ . Φυσικά η παραπάνω ενσωμάτωση εξισώσεων μπορεί να καλύψει διαφορετικά μοτίβα.



Σχήμα 3-1: Αρχιτεκτονική Ανάδρασης για Μοντελοποίηση Μοτίβων

Αν επεκτείνουμε την μέθοδο με χρήση παραθύρου  $L$  τότε είναι δυνατό εφαρμόζοντας σε κάθε κύκλο σύλληψης την κατάσταση μοτίβου και να έχουμε σε όλα τα

διανύσματα μέσα στο παράθυρο το επιθυμητό μοτίβο. Τότε είναι που έχουμε πραγματικό κέρδος. Ας δούμε το παρακάτω σχήμα:

cycle type		capture	shift	shift	shift	shift	capture	shift	shift	shift	shift
cycle		1	2	3	4	5	6	7	8	9	10
shell	lfsr coef	cell contents									
4	0	ε	δ	γ	β	αε	εδ	γδ	βγ	αβε	αδ
3	0	δ	γ	β	αε	εδ	γδ	βγ	αβε	αδ	εγ
2	0	γ	β	αε	εδ	γδ	βγ	αβε	αδ	εγ	βδ
1	1	β	αε	εδ	γδ	βγ	αβε	αδ	εγ	βδ	αγε
0	0	α	ε	δ	γ	β	αε	εδ	γδ	βγ	αβε
		$\delta_0$	1	1	x	1	$\delta_1$	0	x	0	1

↳	$\delta=1$	$\gamma\delta=0$	άρα	$\delta=1$	$\epsilon=0$
	$\gamma=1$	$\alpha\beta\epsilon=0$		$\gamma=1$	$\beta=0$
	$\alpha\epsilon=1$	$\alpha\delta=1$		$\alpha=0$	

Στο παραπάνω σχήμα φαίνεται η κλασική μέθοδος κωδικοποίησης για  $L=2$  πως κωδικοποιεί τα διανύσματα  $\delta_0$  και  $\delta_1$   $11 \times 1$  και  $0x01$  αντίστοιχα, που έχουν 6 καθορισμένα ψηφία, με μία αλυσίδα ελέγχου από το 4<sup>ο</sup> κελί του LFSR. Στο σχήμα φαίνονται τα περιεχόμενα των κελιών σε κάθε κύκλο. Κατά την επίλυση των εξισώσεων για την ενσωμάτωση των δύο διανυσμάτων παρατηρούμε ότι χρειάστηκαν 5 μεταβλητές από την αρχική κατάσταση ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  και  $\epsilon$ ) και ο λόγος συμπίεσης που επιτεύχθηκε είναι  $1-5/6=1/6$  ( $1$  - πλήθος μεταβλητών που χρειάστηκαν προς πλήθος καθορισμένων ψηφίων που κωδικοποιήθηκαν). Όπως παρατηρούμε τα δύο διανύσματα που ενσωματώθηκαν εμφανίζουν μοτίβο στο τελευταίο ψηφίο.

Εφαρμόζοντας το μοτίβο  $xxx1$  στο παραπάνω LFSR παίρνουμε τον εξής αντίστοιχο πίνακα:

pattern		x	x	X	1						
cycle type	capture	shift	shift	shift	shift	capture	shift	shift	shift	shift	
cycle	1	2	3	4	5	6	7	8	9	10	
shell	lfsr coef	cell contents									
4	0	ε	δ	γ	β	1	εδ	γδ	βγ	β'	1
3	0	δ	γ	β	1	εδ	γδ	βγ	β'	1	εγ
2	0	γ	β	1	εδ	γδ	βγ	β'	1	εγ	βδ
1	1	β	1	εδ	γδ	βγ	β'	1	εγ	βδ	γ
0	0	ε'	ε	δ	γ	β	(εδ)'	εδ	γδ	βγ	β'
		<b>δ<sub>0</sub></b>	1	1	x	1	<b>δ<sub>1</sub></b>	0	x	0	1

δ=1	γδ=0	άρα	δ=1	
γ=1	β'=0		γ=1	β=1

Όπως παρατηρούμε για την ενσωμάτωση των  $\delta_0$  και  $\delta_1$  χρειάστηκαν 3 μεταβλητές οι  $\beta$ ,  $\gamma$  και  $\delta$  με τη χρήση του μοτίβου. Άρα ο λόγος συμπίεσης που επιτεύχθηκε έγινε  $1-3/6=3/6$ .

Το πρόβλημα, λοιπόν, ανάγεται στην επιλογή του κατάλληλου μοτίβου το οποίο θα έχει μεγάλη απόδοση σε συμπίεση και μικρό κόστος σε υλικό.

### 3.2. Επιλογή Μοτίβου

Θα ορίσουμε σαν πίνακα διανυσμάτων ένα δυδιάστατο πίνακα από ψηφία 0, 1, x. Κάθε γραμμή του πίνακα είναι ένα διάνυσμα εισόδου. Οι στήλες του πίνακα αντιστοιχίζονται στη θέση ενός ψηφίου μέσα σε ένα διάνυσμα.

Παράδειγμα: Για τα διανύσματα  $X_1=x1x11xx$ ,  $X_2=x1xx0x0$ ,  $X_3=1xxx1x0$ ,  $X_4=x10xx1x$ ,  $X_5=xxxxx0$  και  $X_6=x0xx0xx$  προκύπτει ο παρακάτω πίνακας:

	1	2	3	4	5	6	7
$X_1$ :	x	1	x	1	1	x	x
$X_2$ :	x	1	x	x	0	x	0
$X_3$ :	1	x	x	x	1	x	0
$X_4$ :	x	1	0	x	x	1	x
$X_5$ :	x	x	x	x	x	x	0
$X_6$ :	x	0	x	X	0	x	x

Πίνακας 3-1: Παράδειγμα πίνακα

Το πρόβλημα της επιλογής μοτίβου είναι πρόβλημα επιλογής  $N$  στηλών από τις  $p$  στήλες που ορίζονται από το μήκος των διανυσμάτων ελέγχου. Κάθε στήλη βέβαια μπορεί να έχει την τιμή 0 ή 1 γεγονός που αντιμετωπίζεται κατάλληλα όπως θα φανεί παρακάτω.

Πρέπει αρχικά να διατυπωθούν κάποιοι ορισμοί για τη συμβατότητα μοτίβου και διανύσματος και για την απόδοση μιας αρχικής κατάστασης.

*Ορισμός:* Ένα διάνυσμα είναι συμβατό με ένα μοτίβο όταν για κάθε καθορισμένο ψηφίο του μοτίβου το αντίστοιχο ψηφίο του διανύσματος είναι α) καθορισμένο και ίδιο ή β) μη καθορισμένο.

Αντίστοιχα προκύπτει και ο ορισμός για την μη συμβατότητα:

*Ορισμός:* Ένα διάνυσμα είναι μη συμβατό με ένα μοτίβο όταν για ένα τουλάχιστον καθορισμένο ψηφίο του μοτίβου το αντίστοιχο ψηφίο του διανύσματος είναι καθορισμένο και διαφορετικό αυτού του μοτίβου.

*Παράδειγμα:* Για τον πίνακα διανυσμάτων 3-1 και το μοτίβο xxxxlxx, τα  $X_1, X_3, X_4$  και  $X_5$  είναι συμβατά με το μοτίβο ενώ τα  $X_2$  και  $X_6$  είναι μη συμβατά.

*Ορισμός:* Η δύναμη κωδικοποίησης μιας αρχικής κατάστασης φαίνεται από την απόδοση  $y$  που είναι ο λόγος συμπίεσης των ορισμένων ψηφίων που κωδικοποιήθηκαν προς το πλήθος των ψηφίων της αρχικής κατάστασης που καθορίστηκαν.

Τα ψηφία που καθορίστηκαν από μια αρχική κατάσταση δεν είναι πάντα ίσα με το μήκος του ολισθητή. Είναι δυνατό μια αρχική κατάσταση μετά μην κωδικοποίηση διανυσμάτων να έχει ακόμα ελεύθερες μεταβλητές, δηλαδή κελιά στον ολισθητή, των οποίων η αρχική κατάσταση είναι μη καθορισμένη. Αυτό το φαινόμενο όλοι οι αλγόριθμοι ενσωμάτωσης προσπαθούν να το αποφύγουν ή να το μειώσουν όσο γίνεται περισσότερο. Στην περίπτωση του ολισθητή παραγωγής μοτίβων, επειδή πολλά ψηφία δεν θα κωδικοποιούνται με τη χρήση μεταβλητών αλλά με τη χρήση των

ενσωματωμένων εξισώσεων πρέπει να φαίνεται αυτή η σχέση στην απόδοση μιας αρχικής κατάστασης. Αν εξαιρέσουμε την αρχική γνώση της ύπαρξης του μοτίβου που ενεργοποιεί την επιλεκτική ανάδραση ένα ψηφίο που κωδικοποιήθηκε από ένα μοτίβο δεν έκανε χρήση αρχικής μεταβλητής. Είναι σαν να κωδικοποιήθηκε εγγενώς από τον ολισθητή.

### 3.2.1. Αλγόριθμος Επιλογής

Ο σκοπός του αλγορίθμου αυτού είναι να βρει το μοτίβο με τη μεγαλύτερη απόδοση κωδικοποίησης. Ο κορμός όλων των αλγορίθμων που εφαρμόστηκαν είναι ο εξής αλγόριθμος απληστίας:

- A. Δίδεται ένα βάρος σε κάθε στήλη.
- B. Επιλέγεται η στήλη με το μεγαλύτερο βάρος.
- Γ. Αφαιρούνται από το σύνολο τα διανύσματα που είναι μη συμβατά με το μοτίβο που προκύπτει με την επιλογή της στήλης.
- Δ. Εάν δεν έχουν επιλεγθεί όσες στήλες χρειάζονται συνέχεια από το βήμα A, αλλιώς τέλος.

Όπως είναι κατανοητό το βάρος είναι αυτό που καθορίζει την δύναμη κωδικοποίησης του μοτίβου και η επιλογή του είναι κρίσιμη διαδικασία. Σαν αρχική ιδέα επιλέχθηκε το βάρος μιας στήλης να είναι ίσο με το πλήθος των διανυσμάτων που εμφανίζονται να έχουν ορισμένο ψηφίο (0 ή 1) σε αυτή τη στήλη. Ένα τέτοιο βάρος θα οδηγήσει σε ένα μοτίβο με το οποίο ο ολισθητής με μία αρχική κατάσταση θα επιλύσει τα περισσότερα διανύσματα. Εκ' πρώτης όψεως φαίνεται σωστός τρόπος. Όμως, αυτό δεν αυξάνει αναγκαστικά την απόδοση της κωδικοποίησης μιας αρχικής κατάστασης. Χρειάζεται, λοιπόν, το βάρος που θα επιλεγθεί να είναι στενά συνδεδεμένο με την απόδοση αυτή. Η απόδοση  $y$  κωδικοποίησης μιας αρχικής κατάστασης αυξάνει όταν τα διανύσματα που κωδικοποιούνται είναι *δύσκολα διανύσματα*. Με τον όρο αυτό εννοούνται τα διανύσματα με μεγάλο αριθμό από ορισμένα ψηφία.

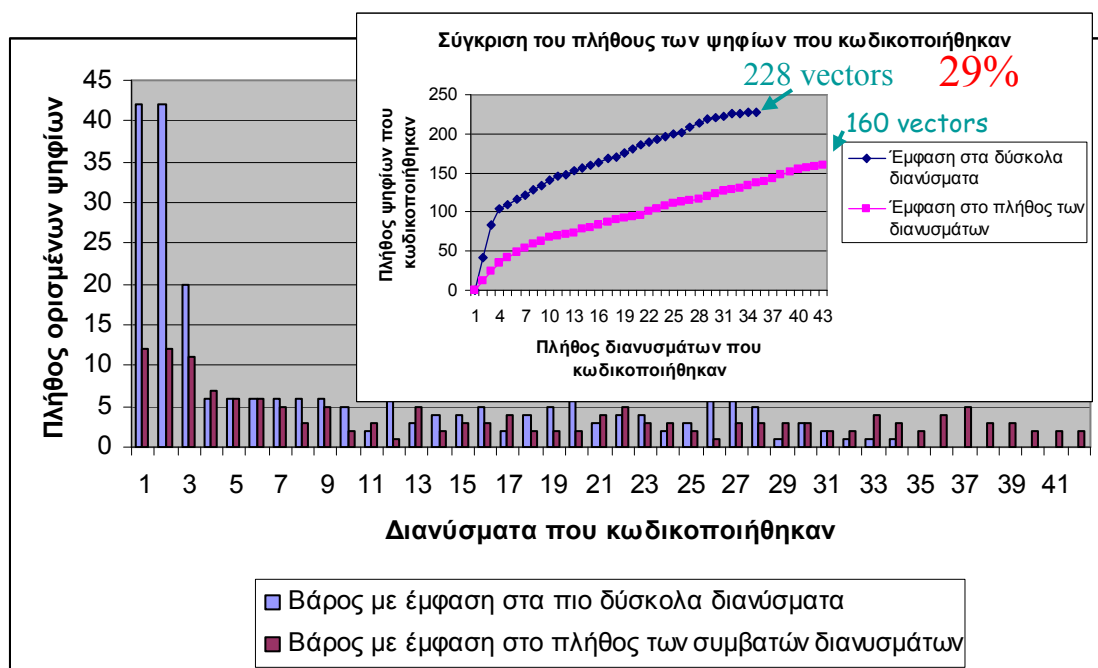


Στο πειράματα έγινε η χρήση δύο αλγόριθμων επιλογής, οι οποίοι ορίζουν διαφορετικά το βάρος στον παραπάνω αλγόριθμο επιλογής. Στον πρώτο αλγόριθμο επιλογής δίδεται έμφαση στα δύσκολα διανύσματα, δίδοντας στις στήλες ένα βάρος από το άθροισμα των προσθετέων  $2^{d-m}$  για κάθε διάνυσμα που έχει ορισμένο ψηφίο σε αυτή τη στήλη. Το  $d$  είναι το πλήθος των ορισμένων ψηφίων του διανύσματος και το  $m$  το ο αριθμητικός μέσος του πλήθους των ορισμένων ψηφίων όλων των διανυσμάτων. Το βάρος αυτό δίδει έμφαση στα δύσκολα διανύσματα.

Παράδειγμα: Για τον πίνακα 3.1 και την στήλη 7,  $w = 2^{3-3} + 2^{3-3} + 2^{1-3} = 1 + 1 + 0.25 = 2.25$ . Επιπλέον για το παραπάνω σύνολο διανυσμάτων  $m=3$ .

Στον δεύτερο αλγόριθμο δίδεται ένα βάρος σε κάθε στήλη από το άθροισμα των ορισμένων ψηφίων κάθε διανύσματος που έχει ορισμένο ψηφίο σε αυτή τη στήλη.

Παράδειγμα: Για τον πίνακα 3.1 και την στήλη 7,  $w = 3 + 3 + 1 = 7$ .



Σχήμα 3-2: Δύσκολα διανύσματα σε μια αρχική κατάσταση σύμφωνα με δύο αλγόριθμους επιλογής

Στο παραπάνω σχήμα βλέπουμε στον άξονα των  $x$  τη σειρά των διανυσμάτων που κωδικοποιήθηκαν με τη χρήση των δύο αλγορίθμων επιλογής μοτίβου. Στον άξονα των  $y$  φαίνεται το μήκος καθενός διανύσματος που κωδικοποιήθηκε. Στην γραφική παράσταση φαίνεται το συνολικό πλήθος των καθορισμένων ψηφίων που κωδικοποιούνται, από κάθε ένα από τους δύο αλγορίθμους επιλογής, καθώς εκτελείται η κωδικοποίηση (αθροιστική κατανομή του πλήθους των καθορισμένων ψηφίων που κωδικοποιούνται). Φαίνεται ότι το πλήθος των ψηφίων που κωδικοποιούνται με χρήση αλγορίθμου επιλογής με έμφαση στα δύσκολα διανύσματα είναι μεγαλύτερη κατά 29% από αυτό του αλγορίθμου με έμφαση στο πλήθος των διανυσμάτων. Παρατηρούμε ότι είναι δυνατό, με το διαφορετικό ορισμό του βάρους, να κατευθυνθεί ο τύπος των διανυσμάτων που θα κωδικοποιηθεί και μάλιστα να υπάρχει και διαφορετική απόδοση στη συμπίεση.

Ένα ακόμη πρόβλημα που υπάρχει είναι η εναλλαγή των μοτίβων. Είναι αποδοτική η εφαρμογή πολλών μοτίβων; Πρέπει ένα μοτίβο να εφαρμόζεται σε ολόκληρο το παράθυρο; Να γίνεται η κωδικοποίηση μόνο με χρήση μοτίβων ή να δίδεται ένα πλήθος από αρχικές καταστάσεις την κλασική μέθοδο; Αυτά τα ερωτήματα μόνο με παρατήρηση των πειραματικών αποτελεσμάτων μπορούν να απαντηθούν, κάτι που δεν είναι εύκολο, αφού το πρόβλημα έχει πολλές παραμέτρους. Για να είναι δυνατή η μελέτη της συμπεριφοράς του μοτίβου στην κωδικοποίηση θα παρουσιαστούν αρχικά αποτελέσματα μόνο για ένα μοτίβο.

### 3.3. Κωδικοποίηση Σε Παράθυρο L Με Χρήση Μοτίβου

Πρέπει να τεθούν κάποιες ιδέες πάνω στις οποίες θα συνεχιστεί η μελέτη. Με την χρήση της επιλεκτικής ανάδρασης για παραγωγή μοτίβων η γραμμική εξάρτηση των αντίστοιχων μεταβλητών (κάποιες εξισώσεις) είναι ενσωματωμένη στο LFSR. Έτσι, μέχρι να εκτελεστεί ένας κύκλος σύλληψης, ο οποίος ενεργοποιεί το κύκλωμα μοτίβου δεν υπάρχει καμία τέτοια εξάρτηση στο LFSR (δεν χρειάζεται η αρχική κατάσταση να έχει αυτή την πληροφορία). Μια τακτική που ακολουθήθηκε με βάση αυτή την ιδέα είναι να μην εφαρμόζεται το κύκλωμα μοτίβου σε όλους τους κύκλους σύλληψης αλλά σε

κάποιους, κυρίως στο τέλος κάθε παραθύρου που η αρχική κατάσταση έχει χάσει την δύναμη κωδικοποίησής της, λόγω αντικατάστασης μεγάλου αριθμού μεταβλητών.

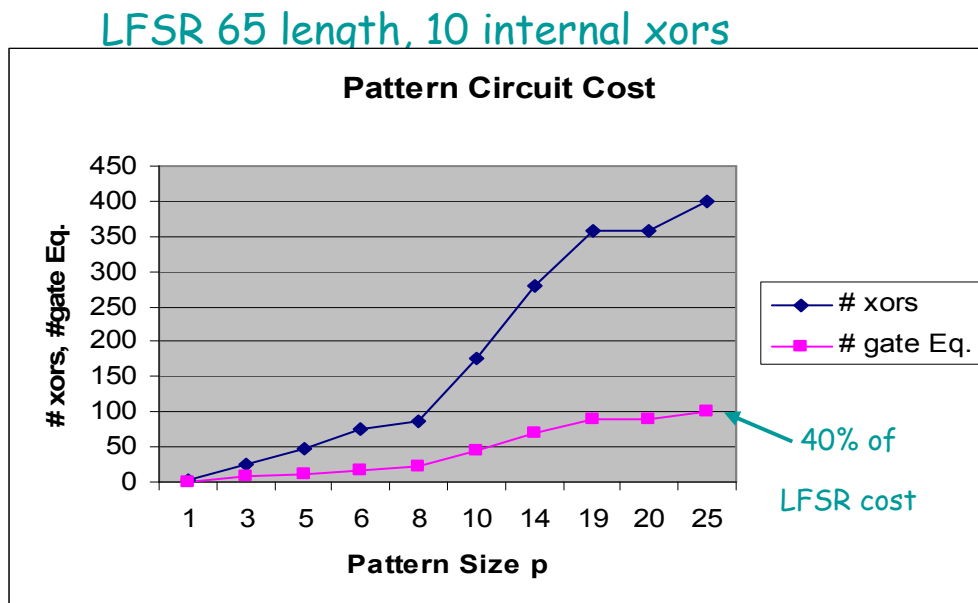
Τα κριτήρια που εφαρμόστηκαν για την κωδικοποίηση των διανυσμάτων είναι αυτά του πίνακα 2.1. Τα διανύσματα χωρίζονται σε δύο σύνολα. Στο σύνολο των συμβατών με το μοτίβο διανυσμάτων και σε αυτό των μη συμβατών με το μοτίβο διανυσμάτων. Τα συμβατά διανύσματα κωδικοποιούνται πριν από τα μη συμβατά με τα ίδια κριτήρια με μόνη τροποποίηση το τέταρτο κριτήριο. Έτσι όταν ένα γραμμικό σύστημα αφορά διάνυσμα που είναι συμβατό με το μοτίβο τότε αν γίνει χρήση του τέταρτου κριτηρίου δεν θα δοθεί έμφαση να βρίσκεται πιο κοντά αλλά πιο μακριά από την αρχή του παραθύρου.

Στη μέθοδο κωδικοποίησης με χρήση μοτίβων γίνεται έλεγχος της απόδοση μιας αρχικής κατάστασης. Έπειτα από την εύρεση κάθε αρχικής κατάστασης υπολογίζεται η απόδοσή της και αν αυτή είναι μικρότερη από μια παραμετρική απόδοση  $\gamma$  τότε η κωδικοποίηση γίνεται χωρίς τη χρήση μοτίβων.

### 3.4. Αρχιτεκτονική Αποκωδικοποίησης

Η αρχιτεκτονική αποκωδικοποίησης είναι πολύ απλή. Αρχικά γίνεται αρίθμηση στα διανύσματα ενός παραθύρου με έναν δείκτη από 1 έως  $L$ . Όταν ο δείκτης που αντιστοιχεί σε ένα τμήμα είναι περιττός δεν γίνεται χρήση μοτίβου. Αντίστοιχα όταν ο δείκτης είναι άρτιος ενεργοποιείται κατά τον κύκλο σύλληψης το κύκλωμα παραγωγής μοτίβου. Το πρώτο τμήμα που έχει δείκτη 1 δεν ενεργοποιείται το κύκλωμα παραγωγής μοτίβου. Ακόμα ένας μετρητής είναι απαραίτητος για να ελέγχει πόσες αρχικές καταστάσεις αρχικά κάνουν χρήση του κυκλώματος παραγωγής μοτίβου και πότε θα χρειαστεί να έρθει ο ολισθητής στην κλασική του λειτουργία σαν απλό LFSR. Έτσι με επιβάρυνση ενός μόνο Flip Flop, ενός μετρητή (από 2 έως 5 ψηφίων), μιας πύλης NOT και φυσικά του κυκλώματος παραγωγής μοτίβου ελέγχεται η αποκωδικοποίηση των διανυσμάτων.

Στο παρακάτω σχήμα φαίνεται το κόστος σε πύλες XOR του κυκλώματος παραγωγής μοτίβου ανάλογα με το μέγεθος του μοτίβου  $p$ . Το σχήμα αφορά ένα LFSR μήκους 65 και ένα μοτίβο με 10 ορισμένα ψηφία.



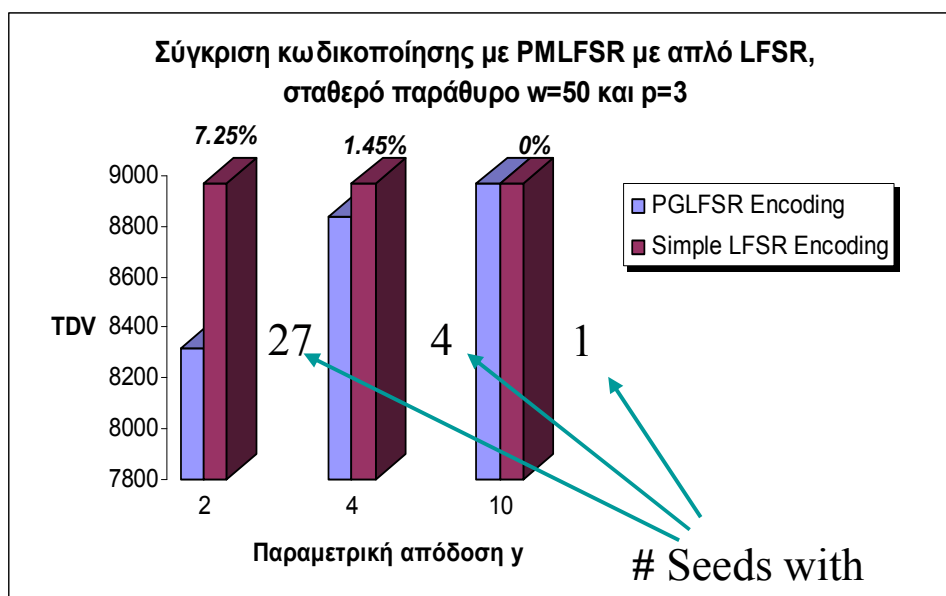
Σχήμα 3-3: Κόστος υλικού συγκριτικά με το μέγεθος του μοτίβου  $p$

Καθώς το  $p$  αυξάνει, αυξάνεται και το πλήθος από πύλες που απαιτούνται για την ενσωμάτωσή του. Στο σχήμα φαίνονται δύο τιμές. Η μια είναι το πλήθος των XOR πυλών που απαιτούνται για την ενσωμάτωση αυτή, ενώ η άλλη είναι το πλήθος από ισοδύναμες πύλες (NAND δύο εισόδων). Φαίνεται ότι το κόστος σε υλικό για 26 μέγεθος μοτίβου  $p$  (μέγεθος που είναι αρκετά μεγάλο) είναι το 40% του συνολικού κόστους του LFSR μαζί με τους ολισθητές φάσης για 32 αλυσίδες ελέγχου.

### 3.5. Πειραματικά Αποτελέσματα

Η προτεινόμενη τεχνική υλοποιήθηκε σε γλώσσα προγραμματισμού C και τα πειράματα έγιναν σε Pentium PC. Χρησιμοποιήθηκαν δύο κυκλώματα ελέγχου Atalanta τα s9234 και s13207. Για κάθε κύκλωμα χρησιμοποιήθηκαν δύο διαφορετικοί ολισθητές ανάδρασης με πρώτα (primitive) χαρακτηριστικά πολυώνυμα και αποκλειστικό-Η πύλες (ένα για κάθε σύνολο ελέγχου). Θα παρουσιαστούν και θα σχολιαστούν πειραματικά αποτελέσματα τα οποία θα συγκρίνουν την κωδικοποίηση με χρήση PGLFSR έναντι της χρήσης κλασικού LFSR.

#### 3.5.1. Σταθερό Παράθυρο L και Μέγεθος Μοτίβου p - Μεταβλητή Παραμετρική Απόδοση y



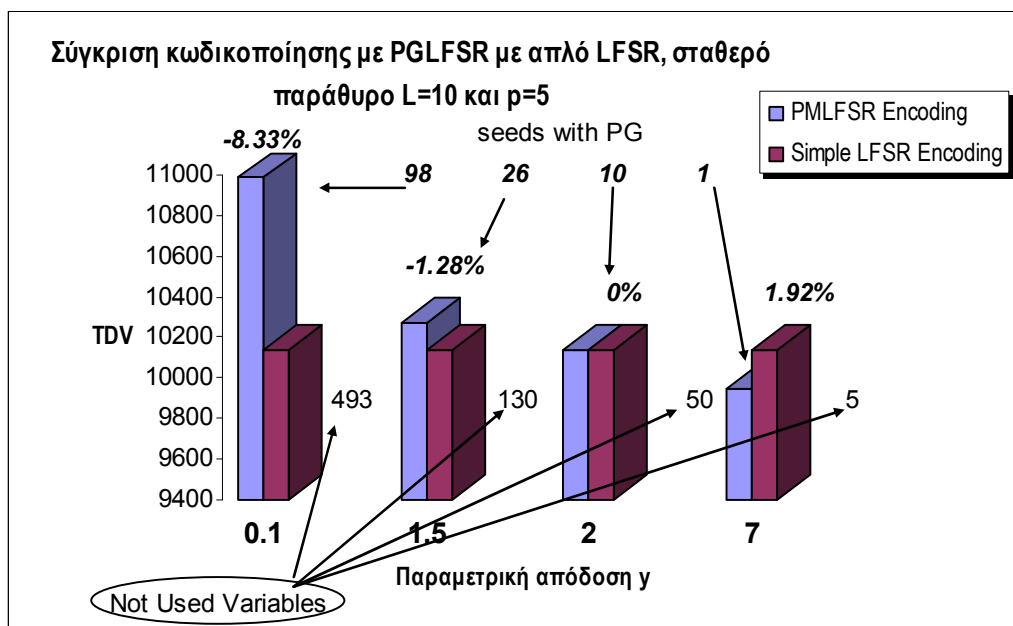
Σχήμα 3-4: Σύγκριση TDV για μεταβλητή παραμετρική απόδοση y με την κλασική μέθοδο

Στο παρακάτω διάγραμμα φαίνονται τα αποτελέσματα του όγκου των δεδομένων ελέγχου με χρήση PGLFSR και LFSR για σταθερό παράθυρο L=50 και μέγεθος μοτίβου

$p=3$ . Στον άξονα των  $\chi$  μεταβάλλεται η παραμετρική απόδοση  $y$  από 2 σε 4 και τελικά σε 10 ( $y$  είναι ο λόγος των καθορισμένων ψηφίων που κωδικοποιήθηκαν προς το πλήθος των μεταβλητών που επιλύθηκαν). Σε κάθε πείραμα φαίνεται ο όγκος δεδομένων ελέγχου της κλασικής κωδικοποίησης και της κωδικοποίησης με χρήση μοτίβου. Η παράμετρος  $y$  επηρεάζει μόνο την μέθοδο με χρήση μοτίβων και καθορίζει το πλήθος των αρχικών καταστάσεων που θα εφαρμοσθεί η επιλεκτική ανάδραση.

Το παραπάνω διάγραμμα γίνεται η αφορμή για να αναλυθεί ένα γεγονός που με την πρώτη ματιά δεν είναι σαφές. Η απόδοση της μεθόδου με μοτίβο, παρόλο που μειώνεται κατά την αύξηση της παραμετρικής απόδοσης στο παράδειγμα μας, δεν ικανοποιεί γενικά κάποιο τέτοιο κανόνα.

Η παραμετρική απόδοση καθορίζει τον έλεγχο για το πότε θα σταματήσει η κωδικοποίηση με μοτίβο κατά τον υπολογισμό μιας αρχικής κατάστασης. Αν βρεθεί αρχική κατάσταση με μοτίβο μικρότερης απόδοσης από το  $y$  τότε σταματάει η χρήση του μοτίβου και συνεχίζεται η κωδικοποίηση με την κλασική μέθοδο (δηλ. χωρίς μοτίβο). Μεγάλο  $y$  σημαίνει λιγότερη χρήση της κωδικοποίησης μοτίβων και το γεγονός ότι η λιγότερή τους χρήση οδηγεί σε χειρότερη κωδικοποίηση είναι ενθαρρυντικό αποτέλεσμα. Καταλήγει τελικά για  $y=10$  να μην συμπεριληφθεί καμία αρχική κατάσταση με μοτίβο και οι κωδικοποιήσεις να είναι πανομοιότυπες. Ωστόσο δεν σημαίνει ότι μικραίνοντας το παραμετρικό  $y$  μεγαλώνει πάντα και η συμπίεση.



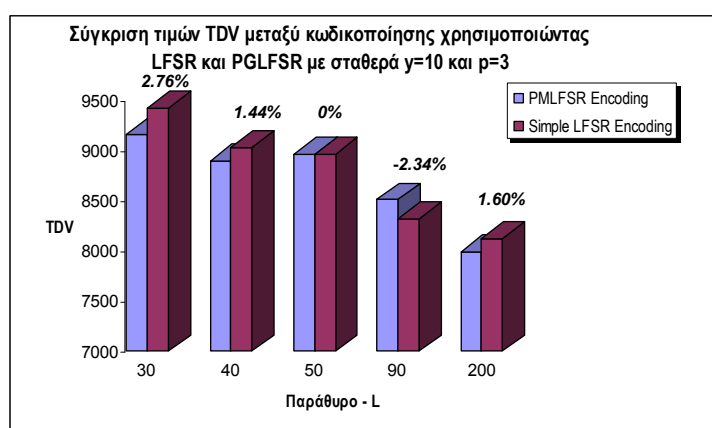
Σχήμα 3-5: TDV σταθερό L, p μεταβλητό  $y$

Στο παραπάνω διάγραμμα βλέπουμε την συμπεριφορά της παραμετρικής απόδοσης καθώς μειώνεται από 7 σε 0.1 σταδιακά. Ας δούμε το διάγραμμα από δεξιά προς τα αριστερά. Αρχικά για  $y=7$  παρουσιάζεται μια μικρή βελτίωση στη συμπίεση έχοντας πάρει μια αρχική κατάσταση με χρήση του μοτίβου – οι υπόλοιπες αρχικές καταστάσεις υπολογίστηκαν χωρίς μοτίβο και με εφαρμογή των κριτηρίων επίλυσης του πίνακα 2.1. Στην συνέχεια ξανακάνουμε το πείραμα με  $y=2$  και παρόλο που έχει σαν αποτέλεσμα να πάρουμε 10 αρχικές καταστάσεις με τη χρήση του μοτίβου δεν παρουσιάζεται καμία βελτίωση. Αυτό οφείλεται στο ότι το μοτίβο που έχει επιλεγεί δεν είναι και ιδιαίτερα «ικανό». Με τον όρο αυτό σημαίνει ότι το πλήθος των διανυσμάτων που είναι συμβατά με το μοτίβο δεν είναι αρκετό, ώστε να στηρίζει πολλές αρχικές καταστάσεις έχοντας θετική απόδοση (ενσωματώνονται εξισώσεις χωρίς να κωδικοποιούν ψηφία). Έτσι, παρόλο που πετυχαίνει την παραμετρική απόδοση, τα διανύσματα που κωδικοποιεί δεν κάνουν χρήση του μοτίβου, γεγονός που φαίνεται από τις πολλές μεταβλητές που μένουν αχρησιμοποίητες σε κάθε αρχική κατάσταση. Αρχικά για  $y=7$  έμεναν 7 μεταβλητές αχρησιμοποίητες από όλες τις αρχικές καταστάσεις, ενώ

στη συνέχεια για  $y=2$  μένουν 50. Καθώς μειώνουμε και άλλο το  $y$  στις τιμές 1.5 και 0.1 το πλήθος των αρχικών καταστάσεων που επιλύονται με τη χρήση του μοτίβου είναι 26 και 98 αντίστοιχα. Και στις δύο περιπτώσεις πολύ μεγάλο, μια και το σύνολο δεν έχει τόσα πολλά διανύσματα που παρουσιάζουν αυτό το μοτίβο, αφήνοντας ανεκμετάλλευτες 130 και 493 μεταβλητές αντίστοιχα.

### 3.5.2. Σταθερή Απόδοση $y$ και Μέγεθος Μοτίβου $p$ – Μεταβλητό Παράθυρο $L$

Στο παρακάτω διάγραμμα παρουσιάζεται ο όγκος των δεδομένων ελέγχου μετά την κωδικοποίησή τους με χρήση PGLFSR και LFSR. Για τα πειράματα αυτά χρησιμοποιήθηκε σταθερή παραμετρική απόδοση  $y$ , σταθερό μέγεθος μοτίβου και μεταβλητό παράθυρο. Τα πειράματα αφορούν το s9234 Atalanta σύνολο ελέγχου με σταθερό μέγεθος μοτίβου  $p=3$ , το οποίο, λόγω της μεγάλης απόδοσης 10 που απαιτήθηκε, αναγκάστηκε να εφαρμοσθεί για μία μόνο αρχική κατάσταση σε όλα τα πειράματα. Έτσι αυτό που βλέπουμε στην ουσία είναι η σύγκριση της κλασικής μεθόδου με την ίδια αν την πρώτη αρχική κατάσταση της την υπολογίσουμε με χρήση ενός μοτίβου μεγέθους 3 για παράθυρα 30, 40, 50, 90 και 200.



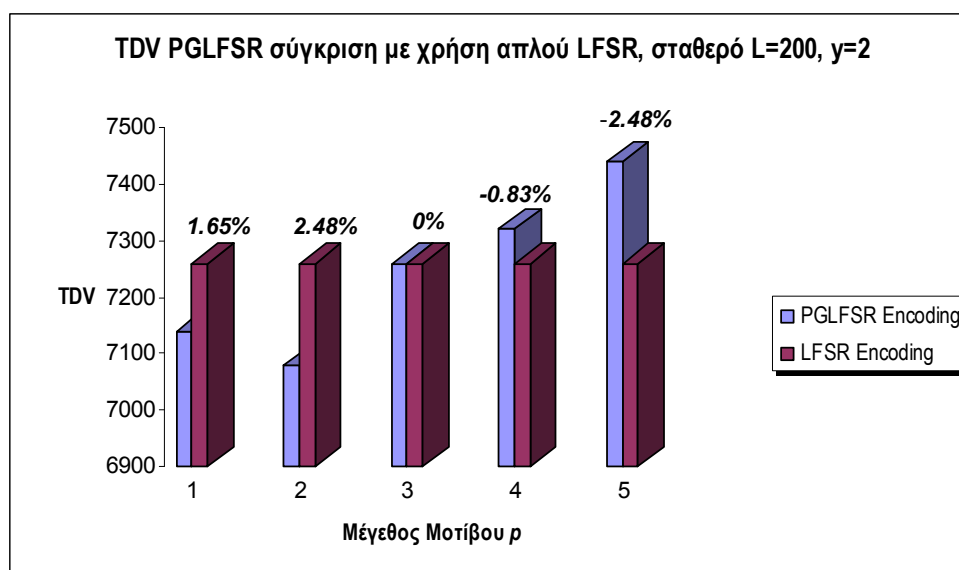
Σχήμα 3-6: Σταθερό  $y$ ,  $p$  μεταβλητό παράθυρο  $L$

Παρατηρούμε ότι για μικρά παράθυρα παρουσιάζεται σημαντική βελτίωση (2.76%), αν αναλογιστούμε ότι εφαρμόστηκε για μία μόνο αρχική κατάσταση πρόκειται για ένα καλό



αποτέλεσμα. Στην πορεία η βελτίωση μειώνεται μέχρι που για μεγαλύτερο του 50 παράθυρο, όπου και δεν παρουσιάζει καμία βελτίωση, η βελτίωση είναι αρνητική. Αυτό συμβαίνει παρόλο που στα μεγάλα παράθυρα επιλύονται πολλά διανύσματα. Όμως, επιλύονται πολλά εύκολα, τα οποία παρόλο που είναι συμβατά με το μοτίβο δεν έχουν κανένα καθορισμένο ψηφίο στις θέσεις του μοτίβου και έτσι προκαλούν ενσωμάτωση εξισώσεων χωρίς κέρδος.

### 3.5.3. Σταθερό Παράθυρο L και Απόδοση $\gamma$ – Μεταβλητό Μέγεθος Μοτίβου p



Σχήμα 3-7: Σταθερό L και  $\gamma$ , μεταβλητό p

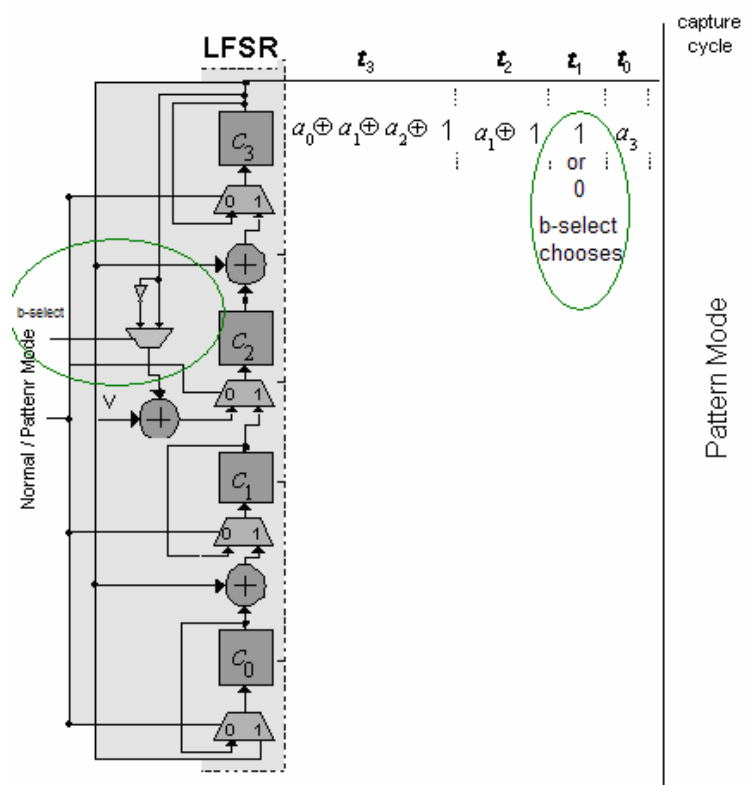
Στο παραπάνω διάγραμμα φαίνεται η τιμή του όγκου δεδομένων ελέγχου της κωδικοποίησης με χρήση μοτίβων του συνόλου διανυσμάτων ελέγχου για το s13207 Atalanta όταν μένει σταθερό το παράθυρο L=200 και η παραμετρική απόδοση  $\gamma=2$  και μεταβάλλεται το μέγεθος του μοτίβου p στις τιμές 1,2,3,4 και 5. Επίσης τα αποτελέσματα συγκρίνονται με την κωδικοποίηση με χρήση του κλασικού αλγορίθμου ενσωμάτωσης διανυσμάτων με μέγιστη συμπίεση. Στα πειράματα αυτά αρχικά για p=1 υπάρχει μια μικρή βελτίωση της κωδικοποίησης 1.65%. Στη συνέχεια, παρόλο που για μέγεθος

μοτίβου  $p=2$  η βελτίωση αυξάνει στην τιμή  $p=3$  μηδενίζεται. Ακολουθούν οι τιμές 4 και 5 κατά τις οποίες η κωδικοποίηση παρουσιάζει αρνητικά αποτελέσματα χειροτερεύοντας τη συμπίεση κατά 0.83% και 2.48% αντίστοιχα. Τα αρνητικά αυτά αποτελέσματα οφείλονται στην τιμή της παραμετρικής απόδοσης, η οποία, καθώς μεγαλώνει το μέγεθος του μοτίβου  $p$ , αφήνει περισσότερες από όσες χρειάζονται αρχικές καταστάσεις να υπολογιστούν με τη χρήση του μοτίβου.

Στο σημείο αυτό πιθανή αιτία της ασυνέπειας των παραμετρικών δεδομένων με τα αποτελέσματα είναι να μην έχει οριστεί κατάλληλα ο υπολογισμός της απόδοσης μιας αρχικής κατάστασης με χρήση μοτίβων μια και με αλλαγή του  $y$  σε όλες τις παραπάνω τιμές του  $p$  είναι δυνατό να υπάρξει βελτίωση.

### 3.6. Αρχιτεκτονική για Πολλαπλά Μοτίβα

Είναι μια ιδέα για σχετικά οικονομική εφαρμογή πολλαπλών μοτίβων με τη χρήση μιας αρχιτεκτονικής PGLFSR. Ας θεωρήσουμε την προηγούμενη αρχιτεκτονική PGLFSR. Είχαμε ενσωματώσει την εξίσωση  $a_3 = a_2 \oplus 1$  για να επιτύχουμε την εμφάνιση του 1 σε κάθε δεύτερο κύκλο μετά την εφαρμογή του κύκλου σύλληψης.



Ας υποθέσουμε με βάση το παραπάνω σχήμα τον πολυπλέκτη που οδηγείται από το  $b$ -select και μια πύλη NOT που δέχεται το  $C_3$ . Με τον πολυπλέκτη μπορούμε κατά επιλογή να ενσωματώνουμε την  $a_3 = a_2 \oplus 1$  ή την  $a_3 = \text{NOT}(a_2 \oplus 1) = a_2 \oplus 1 \oplus 1$ . Συνεπώς, μπορούμε να παράγουμε και το μοτίβο που έχει 0 στο δεύτερο κύκλο μετά τον κύκλο σύλληψης ανάλογα με την τροφοδότηση που θα δώσουμε στο  $b$ -select χρησιμοποιώντας την ίδια αρχιτεκτονική.

### 3.6.1. Αλγόριθμος Επιλογής

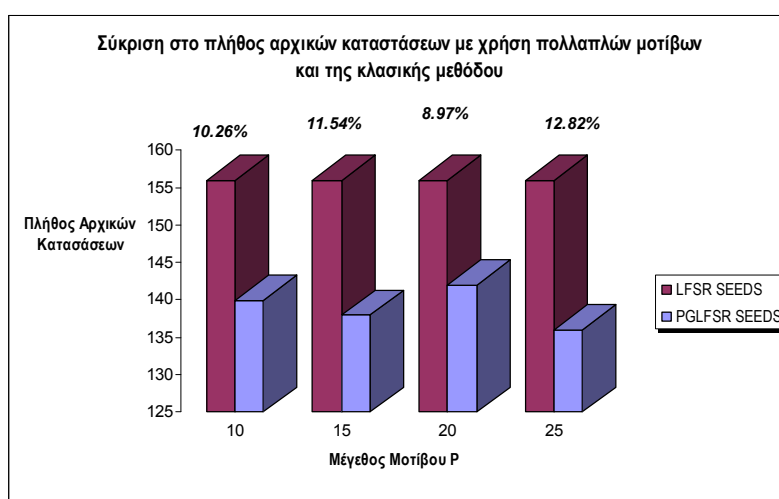
Ο αλγόριθμος επιλογής μοτίβου για την αρχιτεκτονική με πολλαπλά μοτίβα παραμένει ο ίδιος, χωρίς να γίνεται διαχωρισμός στα βάρη σε κάθε στήλη σε βάρος που αντιστοιχεί σε 1 και σε βάρος που αντιστοιχεί σε 0, αλλά με χρήση του αθροίσματος αυτών των βαρών.

### 3.6.2. Κωδικοποίηση

Η κωδικοποίηση γίνεται με τον εξής τρόπο. Αν υποθέσουμε ότι έχουμε ένα μοτίβο με μέγεθος  $p=3$  τότε υπάρχουν  $2^3=8$  πιθανά μοτίβα. Το νούμερο είναι εκθετικό, οπότε δεν είναι δυνατό να επιλεχθούν όλα. Επιλέγονται τα  $n$  με το μεγαλύτερο βάρος. Το βάρος ενός μοτίβου ορίζεται σαν το άθροισμα των βαρών των συμβατών διανυσμάτων του. Για βάρος ενός διανύσματος έχουν δοκιμαστεί πολλές τιμές, αλλά η πιο αποδοτική είναι η  $2^k$ , όπου  $k$  είναι το πλήθος των ορισμένων ψηφίων ενός διανύσματος. Στη συνέχεια ξεκινά ο αλγόριθμος της κωδικοποίησης κυκλικά από το πιο «βαρύ» μοτίβο προς το πιο «ελαφρύ» από αυτά που έχει επιλέξει για να καταλήξει στην κωδικοποίηση χωρίς μοτίβα. Σε κάθε μοτίβο (και στην περίπτωση που εφαρμόζει την κλασική μέθοδο) βρίσκει μόνο μια αρχική κατάσταση και εφόσον δεν έχει κωδικοποιήσει όλα τα διανύσματα επαναλαμβάνει τον κύκλο.

### 3.6.3. Πειραματικά Αποτελέσματα

Τα πειραματικά αποτελέσματα της διαδικασίας αυτής για  $n=5$  και εφαρμογή της στο s9234 Atalanta και παράθυρο  $L=10$  φαίνονται στο παρακάτω διάγραμμα.



Σχήμα 3-8: Βελτίωση στο πλήθος αρχικών καταστάσεων με χρήση πολλαπλών μοτίβων σε μια αρχιτεκτονική

Η τιμή που υπολογίζεται σε κάθε πείραμα είναι το πλήθος των αρχικών καταστάσεων, το οποίο είναι ανάλογο το TDV. Το μέγεθος του LFSR που χρησιμοποιήθηκε είναι 65. Όπως είναι φανερό δεν υπάρχουν δραματικές μεταβολές καθώς μεταβάλλεται το  $p$ . Η μέθοδος αυτή έχει μια σταθερή βελτίωση.

Ωστόσο, η μέθοδος αυτή δεν έχει αποκωδικοποιητή με μικρό κόστος υλικού για να διατηρεί την βελτίωση που προκαλεί.

### 3.7. Συμπεράσματα

Η τακτική αυτή είχε λίγο καλύτερα αποτελέσματα από την κλασική διαδικασία σε πλήθος από αρχικές καταστάσεις (περίπου 1-5%, υπήρχαν και περιπτώσεις που είχε χειρότερα αποτελέσματα και μεμονωμένες με πολύ καλύτερα έως και 12% - η σύγκρισή πάντα γίνεται με την καλύτερη μέθοδο ενσωμάτωσης διανυσμάτων ελέγχου [22]).

Circuit	Par. $y$	L	$p$	proposed			classical method		TDV impr %	
				#xors	seeds	TDV	seeds	TDV		
s13207	10	30	3	89	141	9165	145	9425	2.76%	
	10	200	3	89	123	7995	125	8125	1.60%	
	2	50	3	89	128	8320	138	8970	7.25%	
	4	50	3	89	136	8840	138	8970	1.45%	
s9234	2	200	1	2	119	7140	121	7260	1.65%	
	2	200	2	15	118	7080	121	7260	2.48%	
	multiple patterns	10	10	10	175	140	9100	156	10140	10.26%
		10	15	10	282	138	8970	156		11.54%
		10	20	10	355	142	9230	156		8.97%
		10	25	10	401	136	8840	156		12.82%

Πίνακας 3-2: Αποτελέσματα της μεθόδου παραγωγής μοτίβου.

Στον πίνακα 3-2 φαίνονται κάποια ενδεικτικά αποτελέσματα από τα πειράματα που έγιναν. Στην 1<sup>η</sup> στήλη του πίνακα φαίνεται το κύκλωμα για το οποίο αναφέρεται το κάθε πείραμα, στην 2<sup>η</sup> στήλη η παραμετρική απόδοση, στην 3<sup>η</sup> στήλη το παράθυρο L και

στην 4<sup>η</sup> το μέγεθος του μοτίβου. Στην 5<sup>η</sup>, 6<sup>η</sup> και 7<sup>η</sup> στήλη φαίνονται για την προτεινόμενη μέθοδο το πλήθος σε XOR πύλες του κυκλώματος παραγωγής μοτίβου που απαιτείται, ο αριθμός από αρχικές καταστάσεις που υπολογίζονται και ο τελικός όγκος των δεδομένων ελέγχου αντίστοιχα. Στις επόμενες δύο στήλες (8<sup>η</sup> και 9<sup>η</sup>) φαίνεται το πλήθος των αρχικών καταστάσεων και ο όγκος των δεδομένων για την κλασσική μέθοδο κωδικοποίησης με χρήση LFSR. Τέλος, στην τελευταία στήλη έχει υπολογιστεί η βελτίωση του όγκου των δεδομένων ελέγχου που επιτυγχάνεται από την προτεινόμενη μέθοδο. Το μικρό μέγεθος της βελτίωσης εξηγείται αφενός από το γεγονός ότι εφαρμόζεται όταν η αρχική κατάσταση έχει χάσει τη δύναμη κωδικοποίησης του και θα κωδικοποιήσει κύβους με μικρό αριθμό από ψηφία 0 ή 1, αφετέρου από την πολύ μεγάλη συμπίεση που επιτυγχάνεται από την κλασσική μέθοδο ενσωμάτωσης διανυσμάτων ελέγχου [22] που πλησιάζει σε ορισμένες περιπτώσεις το 95%.

### 3.8. Μελλοντική Μελέτη

Ας προσέξουμε τον πίνακα 3.1 με το παράδειγμα διανυσμάτων ελέγχου. Θεωρήστε επίσης ένα PGLFSR που παράγει το παραπάνω μοτίβο και κωδικοποιεί όλα τα διανύσματα σε ένα παράθυρο  $w=2$ . Στο πρώτο τμήμα του παραθύρου κάνει χρήση του κυκλώματος μοτίβου ενώ στο δεύτερο όχι. Έτσι στο πρώτο τμήμα κωδικοποιεί τα διανύσματα  $X_1, X_3, X_4$  και  $X_5$  και στο δεύτερο τα  $X_2$  και  $X_6$ . Για χάρη του παραδείγματος θεωρούμε ότι δεν επιλύθηκαν όλα τα ψηφία τυχαία και ότι για όλα χρειάστηκε να δαπανηθεί ένα ψηφίο της αρχικής κατάστασης. Όμως, για πολλά ψηφία που κωδικοποιήθηκαν από το μοτίβο δεν χρειάστηκαν αυτές οι μεταβλητές. Τα ψηφία αυτά δεν χρειάστηκαν μεταβλητή αλλά μια γραμμική σχέση μιας μεταβλητής με κάποια άλλη. Αν λυθεί λοιπόν η μια μεταβλητή λύνεται και η άλλη αυτομάτως με αποτέλεσμα να προκαλείται διπλό κόστος για κάποια ψηφία που ισοσταθμίζει τελικά την απόδοση. Όμως, πολλές φορές πολλές μεταβλητές παραμένουν αχρησιμοποίητες ακόμα και όταν πια μια αρχική κατάσταση δεν έχει τη δυνατότητα να κωδικοποιήσει άλλα διανύσματα. Ίσως να ήταν δυνατό να βρεθούν αυτά τα μοτίβα, τα οποία προκαλούν την επίλυση των μεταβλητών αυτών και να εφαρμοσθούν στα τμήματα που πρέπει ώστε να επιτευχθεί

πλήρης χρήση των μεταβλητών και μέγιστη απόδοση. Επίσης, θα έπρεπε να βρεθεί μια πιο κατάλληλη παραμετρική απόδοση  $y$  που θα συμπεριλαμβάνει τη θέση μέσα στο παράθυρο στην οποία εφαρμόζεται ένα μοτίβο π.χ. αν το μοτίβο εμφανίζεται στο πρώτο τμήμα ενός παραθύρου τότε θα χρεώνει με ένα ψηφίο την απόδοση, αν βρίσκεται στη μέση με μισό κτλ. Ίσως, έτσι η παραμετρική απόδοση να συμβαδίζει περισσότερο με τα πειραματικά αποτελέσματα.

## ΚΕΦΑΛΑΙΟ 4. ΟΛΙΣΘΗΤΗΣ ΠΑΡΑΒΛΕΨΗΣ ΚΑΤΑΣΤΑΣΕΩΝ

---

4.1. Κύκλωμα Παράβλεψης

4.2. Μέθοδος Μείωσης της Ακολουθίας Ελέγχου

4.3. Αρχιτεκτονική αποκωδικοποίησης

---

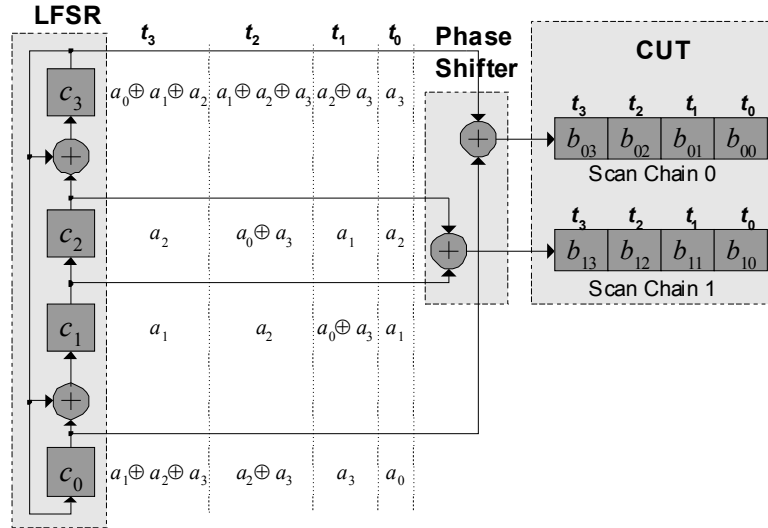
Στο κεφάλαιο αυτό παρουσιάζεται άλλη μία εφαρμογή της επιλεκτικής ανάδρασης στους γραμμικούς ολισθητές με δυνατότητα παράβλεψης καταστάσεων που έχει πολύ καλά αποτελέσματα μείωσης του μήκους της ακολουθίας ελέγχου. Η εφαρμογή που θα παρουσιαστεί ονομάζεται Ολισθητής Γραμμικής Ανάδρασης με δυνατότητα Παράβλεψης Καταστάσεων (State Skip LFSR - SSLFSR) και παράλληλα θα παρουσιαστεί μια συστηματική μέθοδος μείωσης της ακολουθίας ελέγχου. Θα γίνει λεπτομερή περιγραφή της αρχιτεκτονικής του αποκωδικοποιητή και θα παρουσιαστούν πολλά πειραματικά αποτελέσματα που αποδεικνύουν την ανωτερότητα της μεθόδου αυτής σε χρόνο αποκωδικοποίησης των διανυσμάτων ελέγχου συγκριτικά με τις καλύτερες υπάρχουσες μεθόδους.

### 4.1. Κύκλωμα Παράβλεψης

Ο ολισθητής παράβλεψης χρησιμοποιεί το κύκλωμα παράβλεψης αντί του κυκλώματος που εκφράζει το χαρακτηριστικό πολυώνυμο του ολισθητή, για υπολογισμό μελλοντικής κατάστασης του ολισθητή. Έτσι ο ολισθητής μπορεί να παραβλέπει προκαθορισμένο σταθερό αριθμό από καταστάσεις μέσα σε ένα κύκλο, υπολογίζοντας την κατάσταση που τις ακολουθεί.



Για να γίνει αντιληπτή η λειτουργία του ολισθητή παράβλεψης ή SSLFSR, θα παρουσιαστούν αναλυτικά τα περιεχόμενα των κελιών του LFSR κατά τους κύκλους του ρολογιού  $t_0$  και  $t_2$  σύμφωνα με το σχήμα 2.3:



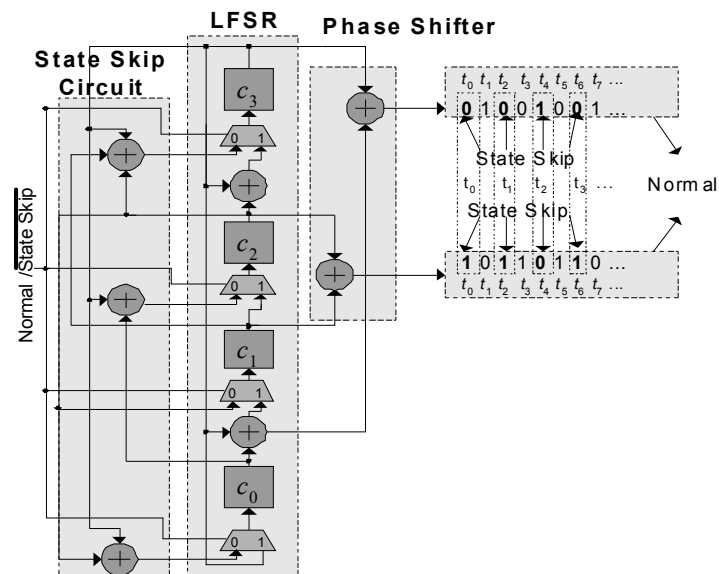
Σχήμα 2-3: Παράδειγμα Γραμμικών Συστημάτων LFSRs

Παρατηρούμε ότι η τιμή του  $c_0$  στον κύκλο  $t_2$  είναι ίση με το αποτέλεσμα της πράξης αποκλειστικό-Η των τιμών στα κελιά  $c_2$ ,  $c_3$  κατά τον κύκλο  $t_0$ , δηλαδή  $c_0(t_2) = c_2(t_0) \oplus c_3(t_0)$ , όπου  $c_i(t_j)$  είναι η τιμή στο κελί  $c_i$  κατά τον κύκλο  $t_j$ . Για τα υπόλοιπα κελιά μπορούμε να εξάγουμε παρόμοιες σχέσεις:  $c_1(t_2) = c_2(t_0)$ ,  $c_2(t_2) = c_0(t_0) \oplus c_3(t_0)$ ,  $c_3(t_2) = c_1(t_0) \oplus c_2(t_0) \oplus c_3(t_0)$ . Αυτές οι σχέσεις εξαρτώνται μόνο από το χαρακτηριστικό πολυώνυμο και τους συγκεκριμένους κύκλους και δεν εξαρτώνται από την κατάσταση του LFSR. Έτσι ικανοποιούνται για οποιοδήποτε ζεύγος κύκλων ρολογιού  $t_{i+2}$ ,  $t_i$  π.χ.:  $c_0(t_{i+2}) = c_2(t_i) \oplus c_3(t_i)$ ,  $c_1(t_{i+2}) = c_2(t_i)$ ,  $c_2(t_{i+2}) = c_0(t_i) \oplus c_3(t_i)$  και  $c_3(t_{i+2}) = c_1(t_i) \oplus c_2(t_i) \oplus c_3(t_i)$ . Γενικά μπορούμε να υποθέσουμε πως για ένα LFSR μήκους  $n$  και για κάθε  $k \geq 1$ , υπάρχουν  $n$  γραμμικές εκφράσεις που ικανοποιούν τις παρακάτω σχέσεις:

$$\begin{aligned}
 c_0(t_{i+k}) &= F_0^k(c_0(t_i), c_1(t_i), \dots, c_{n-1}(t_i)) \\
 c_1(t_{i+k}) &= F_1^k(c_0(t_i), c_1(t_i), \dots, c_{n-1}(t_i)) \\
 &\dots \\
 c_{n-1}(t_{i+k}) &= F_{n-1}^k(c_0(t_i), c_1(t_i), \dots, c_{n-1}(t_i))
 \end{aligned} \tag{1}$$

Στην περίπτωση που  $k=1$  αναπαριστούν το χαρακτηριστικό πολυώνυμο του LFSR  $(F_0, F_1, \dots, F_{n-1})$ . Οι παράγοντες  $F$  μπορούν εύκολα να υπολογιστούν θέτοντας  $i=0$  και έπειτα προσομοιώνοντας συμβολικά το LFSR (προσέξτε ότι οι σχέσεις (1) ικανοποιούνται για κάθε τιμή του  $i$ ). Συγκεκριμένα το LFSR αρχικοποιείται στην συμβολική κατάσταση  $(c_0(t_0), c_1(t_0), \dots, c_{n-1}(t_0)) = (a_0, a_1, \dots, a_{n-1})$  και ενεργοποιείται για  $k$  κύκλους. Έπειτα από τον  $k$ -οστό κύκλο ρολογιού, τα περιεχόμενα των κελιών του LFSR  $c_0(t_k), c_1(t_k), \dots, c_{n-1}(t_k)$  περιέχουν γραμμικές εκφράσεις των αρχικών περιεχομένων  $c_0(t_0), c_1(t_0), \dots, c_{n-1}(t_0)$  οι οποίες είναι οι ζητούμενες  $F$  εκφράσεις.

Στην εργασία αυτή προτείνεται η ενσωμάτωση των πράξεων στη  $F_0^k, F_1^k, \dots, F_{n-1}^k$  δομή του LFSR. Το τροποποιημένο LFSR, το οποίο θα ονομάζεται από εδώ και στο εξής Ολισθητής Γραμμικής Ανάδρασης με δυνατότητα παράβλεψης καταστάσεων (State Skip LFSR), λειτουργεί σε δύο διαφορετικές καταστάσεις την κανονική (Normal Mode) και την παράβλεψης (State Skip Mode). Στην κανονική κατάσταση λειτουργίας η ακολουθία των καταστάσεων του LFSR παράγεται από το χαρακτηριστικό πολυώνυμο, ενώ στην κατάσταση παράβλεψης από το κύκλωμα παράβλεψης καταστάσεων. Όταν το LFSR λειτουργεί σε κατάσταση παράβλεψης σε κάθε κύκλο ρολογιού πραγματοποιεί μια αναπήδηση μήκους  $k$  μπροστά στην ακολουθία των καταστάσεών του, παραβλέποντας με αυτό τον τρόπο τις  $k-1$  ενδιάμεσες καταστάσεις που θα παράγονταν εάν το LFSR λειτουργούσε σε κανονική κατάσταση λειτουργίας (Normal Mode). Συνεπώς, στην κατάσταση λειτουργίας παράβλεψης καταστάσεων η ψευδοτυχαία ακολουθία διανυσμάτων τμηματοποιείται και μειώνεται με παράγοντα το  $k$ , ο οποίος θα λέγεται *επιτάχυνση ανάδρασης* από εδώ και στο εξής. Όπως θα φανεί και στα πειραματικά αποτελέσματα, οι γραμμικές εκφράσεις αποτελούνται από ένα μικρό πλήθος από αποκλειστικό-Η πύλες, όταν το  $k$  είναι μικρότερο του 25. Ωστόσο, τιμές του  $k$  μικρότερες του 25, όπως θα φανεί από τα πειραματικά αποτελέσματα, είναι ικανές να επιτύχουν δραστική μείωση του μήκους της ακολουθίας ελέγχου.



Σχήμα 4-1: Γραμμικός Ολισθητής Παράβλεψης Καταστάσεων (SSLFSR)

*Παράδειγμα:* Στο σχήμα 4.1 φαίνεται η έκδοση του LFSR τους σχήματος 2.3 με πρόσθεση της δυνατότητας παράβλεψης καταστάσεων για  $k = 2$ . Στην είσοδο κάθε κελιού του ολισθητή έχει προστεθεί ένας πολυπλέκτης 2 σε 1 για την επιλογή είτε της λογικής τιμής που παράγεται από το πολυώνυμο (Normal Mode) είτε της τιμής από το κύκλωμα παράβλεψης (State Skip Mode). Θεωρώντας ότι η αρχική κατάσταση του LFSR είναι  $(c_0, c_1, c_2, c_3) = (1011)$  στο σχήμα φαίνονται όλες οι λογικές τιμές που παράγονται στις εξόδους των ολισθητών φάσης τόσο σε κανονική κατάσταση λειτουργίας (όλες οι λογικές τιμές μέσα στην γκριζαρισμένη περιοχή) όσο και σε λειτουργία παράβλεψης (έντονοι, τυπωμένοι με μαύρους χαρακτήρες). Όπως φαίνεται σε κατάσταση παράβλεψης με  $k = 2$  μόνο οι μισές λογικές τιμές παράγονται στις εξόδους των ολισθητών φάσης γεγονός που έχει σαν αποτέλεσμα τη την παράβλεψη της μισής ακολουθίας διανυσμάτων ελέγχου.

#### 4.2. Μέθοδος Μείωσης της Ακολουθίας Ελέγχου

Με την ιδιότητα που έχει το κύκλωμα παράβλεψης είναι δυνατό να ελαχιστοποιηθεί το μήκος της ψευδοτυχαίας ακολουθίας που παράγεται κατά την επέκταση κάθε αρχικής κατάστασης σε παράθυρο από  $L$  ψευδοτυχαία διανύσματα. Αρχικά, κάθε παράθυρο διαμερίζεται σε  $L/S$  τμήματα των  $S$  διανυσμάτων (το  $S$  είναι παράμετρος που δίδεται από τον σχεδιαστή). Κάθε τμήμα χαρακτηρίζεται σαν *χρήσιμο* όταν ενσωματώνει ένα τουλάχιστον κύβο ελέγχου, ο οποίος δεν ενσωματώνεται σε κανένα άλλο χρήσιμο τμήμα, *άχρηστο* σε αντίθετη περίπτωση. Τα χρήσιμα τμήματα παράγονται από το SSLFSR χρησιμοποιώντας την κανονική κατάσταση λειτουργίας, ενώ τα άχρηστα χρησιμοποιώντας τη λειτουργία παράβλεψης. Σαν αποτέλεσμα, το πρόβλημα της ελαχιστοποίησης της ακολουθίας ελέγχου ανάγεται σε πρόβλημα ελαχιστοποίησης του πλήθους των χρήσιμων τμημάτων.

Για να επιτευχθεί αυτό διαμερίζονται οι κύβοι ελέγχου σε δύο ομάδες: (α) κύβοι ελέγχου που ενσωματώνονται σε ένα μοναδικό τμήμα σε όλα τα παράθυρα και (β) κύβοι ελέγχου που ενσωματώνονται σε περισσότερα του ενός τμήματα. Αρχικά, όλα τα τμήματα που ενσωματώνουν κύβους της ομάδας (α) επιλέγονται και χαρακτηρίζονται σαν χρήσιμα και την ίδια στιγμή όλοι οι κύβοι της ομάδας (β) που ενσωματώνονται σε αυτά τα τμήματα απομακρύνονται από την ομάδα (β). Στους υπόλοιπους κύβους ελέγχου της ομάδας (β) και τα τμήματα που τα ενσωματώνουν εφαρμόζεται ο παρακάτω αλγόριθμος απληστίας:

- α. Από τα τμήματα που ενσωματώνουν ένα τουλάχιστον κύβο ελέγχου που δεν έχει ήδη ενσωματωθεί (ομάδα β) επέλεξε αυτό που ενσωματώνει το μεγαλύτερο αριθμό από κύβους ελέγχου. Εάν υπάρχουν περισσότερα του ενός τέτοια τμήματα επέλεξε αυτό που είναι πιο κοντά στην αρχή του παραθύρου.
- β. Αφαίρεσε τους κύβους ελέγχου που ενσωματώνονται στο τμήμα που επιλέχθηκε από την ομάδα (β).
- γ. Εάν υπάρχουν κύβοι ελέγχου που δεν έχουν ενσωματωθεί πήγαινε στο βήμα α.

Πίνακας 4-1: Αλγόριθμος απληστίας για επιλογή χρησίων τμημάτων

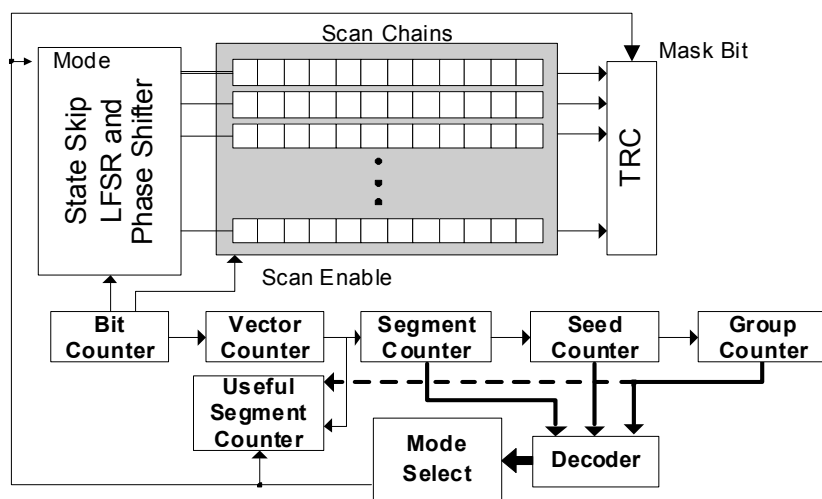
Ο παραπάνω αλγόριθμός ελαχιστοποιεί το πλήθος των χρησίων τμημάτων, όταν υπάρχουν πολλοί κύβοι ελέγχου, που τυχαία ενσωματώνονται σε πολλά τμήματα.

Αφού επιλεγθούν τα χρήσιμα τμήματα, οι αρχικές καταστάσεις ομαδοποιούνται ανάλογα με το πλήθος των χρησίων τμημάτων που περιέχουν και οι ομάδες των αρχικών καταστάσεων ταξινομούνται σε αύξουσα σειρά ανάλογα με βάση το πλήθος των χρησίων τμημάτων που περιέχουν: η ομάδα 1 περιέχει όλες τις αρχικές καταστάσεις με 1 χρήσιμο τμήμα, η ομάδα 2 όλες με 2 χρήσιμα τμήματα και στη γενική μορφή η ομάδα  $j$  όλες τις αρχικές καταστάσεις που έχουν  $j$  χρήσιμα τμήματα. Η ταξινόμηση γίνεται γιατί η παραγωγή των ψευδοτυχαίων ακολουθιών για κάθε αρχική κατάσταση μπορεί να σταματήσει μετά την παραγωγή της τελευταίας χρήσιμης και με αυτό τον τρόπο να μειωθεί η ακολουθία ελέγχου ακόμα περισσότερο.

Είναι φανερό ότι η απόδοση της παραπάνω διαδικασίας εξαρτάται σημαντικά από την τιμή του  $S$ . Όπως θα φανεί και από τα πειραματικά αποτελέσματα μικρά σε μήκος τμήματα επιτυγχάνουν μεγαλύτερη μείωση στο μήκος της ακολουθίας ελέγχου από μεγαλύτερα τμήματα αλλά απαιτούν λίγο παραπάνω υλικό. Ωστόσο ο χρόνος εκτέλεσης της διαδικασίας αυτής είναι πολύ μικρός, επομένως ο σχεδιαστής μπορεί γρήγορα να εντοπίσει την καλύτερη λύση που ταιριάζει στην περίπτωσή του. Μια άλλη σχεδιαστική λεπτομέρεια που αφορά το  $S$  είναι ότι εάν το  $S$  δεν διαιρείται ακριβώς από το  $k$ , τότε το τελευταίο διάνυσμα από κάθε άχρηστο τμήμα θα είναι πιο μικρό από τα άλλα και δεν θα

γεμίζει ακριβώς της αλυσίδες ελέγχου. Η περίπτωση αυτή μπορεί να αντιμετωπιστεί από την λογική που θα ελέγχει τον αποκωδικοποιητή με μηδαμινό κόστος. Όμως, εάν η παραγωγή του κύκλου σύλληψης-ολίσθησης (Shift-Capture Cycle) πρέπει να παραμείνει άθικτη τότε τα  $S$  και  $k$  πρέπει να επιλεγθούν κατάλληλα. Πιο συγκεκριμένα εάν  $\mu$  είναι το μέγεθος μιας αλυσίδας ελέγχου τότε για  $S = 2$  οι καταστάσεις που πρέπει να παραβλεθούν είναι  $S \times (\mu+1) = 2 \times (\mu+1)$ , με  $\mu+1$  δηλώνεται και ο κύκλος σύλληψης. Έτσι για να μπορεί να παραβλέπει καταστάσεις το κύκλωμα παράβλεψης χωρίς να υπάρχει περίπτωση να προσπεράσει κάποιο χρήσιμο ψηφίο, πρέπει το  $k$  να είναι διαιρέτης του  $S \times (\mu+1)$ . Για τα πειράματα που έχουν γίνει, για όλα τα κυκλώματα και για όλα τα  $S$  που δοκιμάστηκαν, βρέθηκαν αρκετά  $k < 25$  να ικανοποιούν τις παραπάνω προϋποθέσεις για όλα τα κυκλώματα (τουλάχιστον έξι για κάθε κύκλωμα).

#### 4.3. Αρχιτεκτονική αποκωδικοποίησης



Σχήμα 4-2: Προτεινόμενη αρχιτεκτονική αποκωδικοποιητή

Η προτεινόμενη αρχιτεκτονική αποκωδικοποίησης φαίνεται στο σχήμα 4.2. Ο μετρητής ψηφίων (Bit Counter) και ο μετρητής διανυσμάτων (Vector Counter) ελέγχουν την σωστή «φόρτωση» των ψευδοτυχαίων διανυσμάτων στις αλυσίδες ελέγχου ενώ ο

μετρητής τμημάτων (Segment Counter) και ο μετρητής χρήσιμων τμημάτων (Useful Segment Counter) ελέγχουν μετρούν το πλήθος των τμημάτων και των χρήσιμων τμημάτων αντίστοιχα για κάθε αρχική κατάσταση. Τελικά, ο μετρητής καταστάσεων (Seed Counter) ελέγχει το πλήθος των αρχικών καταστάσεων για κάθε ομάδα και ο μετρητής ομάδων (Group Counter) μετράει τις ομάδες. Στην αρχή κάθε νέας ανατροφοδότησης με μια νέα κατάσταση, ο μετρητής χρήσιμων τμημάτων φορτώνεται με την τιμή του μετρητή ομάδων μια και είναι ίσος με το πλήθος των χρήσιμων τμημάτων κάθε αρχικής κατάστασης που ανήκει σε αυτή την ομάδα (δείτε τη διαδικασία της ομαδοποίησης στην προηγούμενη ενότητα). Τότε, για κάθε χρήσιμο τμήμα που παράγεται ο μετρητής χρήσιμων τμημάτων μειώνεται κατά ένα και όταν φτάσει την τιμή 0 ο μετρητής καταστάσεων αυξάνει κατά ένα για να επιτρέψει σε μια νέα αρχική κατάσταση να φορτωθεί από τον ελεγκτή.

Όπως αναφέρθηκε στην προηγούμενη ενότητα, όταν το S δεν διαιρείται ακριβώς από το k το τελευταίο διάνυσμα ενός άχρηστου τμήματος είναι ελλιπές. Έτσι στον τελευταίο κύκλο κάθε άχρηστου τμήματος, ο μετρητής ψηφίων μηδενίζεται και προκαλείται επανεκκίνηση της διαδικασίας φόρτωσης των αλυσίδων (Scan in/out) για το πρώτο διάνυσμα του επόμενου τμήματος.

Η μονάδα επιλογής (Mode Select) της αρχιτεκτονικής είναι ένα συνδυαστικό κύκλωμα, που καθορίζει αν το επόμενο τμήμα είναι χρήσιμο ή άχρηστο. Λαμβάνει τα αποκωδικοποιημένα δεδομένα των μετρητών τμήματος, αρχικής κατάστασης και ομάδας και παράγει το σήμα επιλογής (Mode signal), το οποίο οδηγεί το κύκλωμα παράβλεψης και είναι ίσο με το 1 μόνο όταν το τμήμα που θα ξεκινήσει είναι χρήσιμο. Το μέγεθος της μονάδας αυτής σε υλικό εξαρτάται, κυρίως, από το συνολικό αριθμό χρήσιμων τμημάτων. Όμως, τα χρήσιμα τμήματα είναι μόνο ένα πολύ μικρό ποσοστό του συνολικού αριθμού των τμημάτων. Επιπλέον, ένα χαρακτηριστικό της διαδικασίας επιλογής αρχικής κατάστασης που πρέπει να αναφερθεί εδώ είναι ότι το πρώτο τμήμα κάθε κατάστασης είναι πάντα χρήσιμο, επειδή πάντα στην πρώτη θέση πιάνεται ο πιο δύσκολος κύβος και συνεπώς πάντα υπάρχει διάνυσμα που κωδικοποιείται σε αυτή τη

θέση (δείτε τη διαδικασία επιλογής χρήσιμων τμημάτων του προηγούμενου υποκεφαλαίου). Σαν αποτέλεσμα δεν χρειάζεται ειδική λογική για την αποκωδικοποίηση της χρησιμότητάς του, με αποτέλεσμα να μειώνεται σημαντικά το μέγεθος της λογικής που αποφασίζει για την χρησιμότητα των τμημάτων. Επιπρόσθετα, σε ένα περιβάλλον πολλών υπό έλεγχο πυρήνων μόνο η μονάδα επιλογής χρειάζεται να επαναυλοποιηθεί για κάθε πυρήνα, αφού οι υπόλοιπες μονάδες της αρχιτεκτονικής παραμένουν οι ίδιες για κάθε πυρήνα.

Ένα ακόμα σημαντικό χαρακτηριστικό του σήματος επιλογής, που παράγεται από την μονάδα επιλογής, είναι ότι μπορεί να χρησιμοποιηθεί για να προβλέπονται και να περιορίζονται όλες τις απαντήσεις των άχρηστων τμημάτων, εάν αδιάφοροι όροι παράγονται στις εξόδους ή στις αλυσίδες του κυκλώματος που υποβάλλεται υπό έλεγχο εξαιτίας των ψευδοτυχαίων ακολουθιών. Με τον τρόπο αυτό, η υπογραφή στον ελεγκτή αποκρίσεων δεν θα καταστραφεί. Ωστόσο, εάν ο πυρήνας που εξετάζεται δεν παράγει αδιάφορους όρους, τότε τα άχρηστα τμήματα μπορούν να μείνουν ελεύθερα και έτσι να αυξάνεται η πιθανότητα τυχαίας ανίχνευσης μη μοντελοποιημένων σφαλμάτων.



## ΚΕΦΑΛΑΙΟ 5. ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΣΥΓΚΡΙΣΕΙΣ ΟΛΙΣΘΗΤΗ ΠΑΡΑΒΛΕΨΗΣ ΚΑΤΑΣΤΑΣΕΩΝ

- 
- 5.1. Βασικά Δεδομένα για Συγκρίσεις
  - 5.2. Επίδραση k, S, L στη Βελτίωση της Ακολουθίας Ελέγχου
  - 5.3. Κόστος Υλικού του Κυκλώματος Παράβλεψης
  - 5.4. Αποτελέσματα Προτεινόμενης Μεθόδου
  - 5.5. Συγκρίσεις
  - 5.6. Κόστος Υλικού
- 

Η προτεινόμενη τεχνική υλοποιήθηκε σε γλώσσα προγραμματισμού C και τα πειράματα έγιναν σε Pentium PC για τα μεγαλύτερα ISCAS 89 κυκλώματα συγκρίσεων, θεωρώντας 32 και 64 αλυσίδες ελέγχου. Χρησιμοποιήθηκαν δύο διαφορετικά σύνολα ελέγχου, για σφάλματα μόνιμης τιμής, που παράχθηκαν από Atalanta [30] και Mintest [12]. Για κάθε κύκλωμα χρησιμοποιήθηκαν δύο διαφορετικοί ολισθητές ανάδρασης με πρώτα χαρακτηριστικά πολυώνυμα και αποκλειστικό-Η πύλες (ένα για κάθε σύνολο ελέγχου). Έγιναν πολλά πειράματα για να επιτρέπουν τη σύγκριση της με όλες σχεδόν τις αντίστοιχες μεθόδους στη βιβλιογραφία.

### 5.1. Βασικά Δεδομένα για Συγκρίσεις

Στον πίνακα 5.1 παρουσιάζεται το συνολικό πλήθος των κύβων ελέγχου και το συνολικό πλήθος των ψηφίων που είναι 0 ή 1 για τα Atalanta (“Atal.”) και για τα Mintest (“Mint.”) σύνολα ελέγχου, καθώς και το μήκος των ολισθητών που χρησιμοποιήθηκαν για τα πειράματα τόσο για 32 όσο και για 64 αλυσίδες ελέγχου. Ο χρόνος εκτέλεσης, της

προτεινόμενης τεχνικής μείωσης του μήκους της ακολουθίας ελέγχου, είναι μόνο μερικά λεπτά.

Circuit	# Test Cubes		Spec. Bits		LFSR Length 32 sc		LFSR Length 64 sc	
	Atal.	Mint.	Atal.	Mint.	Atal.	Mint.	Atal.	Mint.
s9234	1190	1928	17567	25559	44	38	46	39
s13207	2217	3237	20017	26851	24	26	25	27
s15850	2391	3920	21412	30563	39	33	40	35
s38417	6322	10771	93123	142184	85	87	56	56
s38584	8317	13468	67590	92200	56	51	87	89

Πίνακας 5-1: Σύνολα Ελέγχου και LFSRs

Στον παρακάτω πίνακα 5.2 παρουσιάζονται τα αποτελέσματα της πρωτότυπης τεχνικής με ανατροφοδότηση και χρήση παραθύρων (χρήση των 4 κριτηρίων που αναφέρθηκαν στο κεφάλαιο 2, πίνακας 2.1) για 32 αλυσίδες ελέγχου για τα δύο σύνολα ελέγχου και για παράθουρα L=1, 50, 100, 200, 500 και 1000.

Circuit	L=1		L=50		L=100		L=200		L=500		L=1000		
	TDV	TSL	TDV	TSL	TDV	TSL	TDV	TSL	TDV	TSL	TDV	TSL	
Atalanta	s9234	10692	243	8008	9100	7656	17400	7128	32400	6688	76000	6512	148000
	s13207	8856	369	5328	11100	4632	19300	3816	31800	2688	56000	2160	90000
	s15850	11622	298	7410	9500	6981	17900	6669	34200	6201	79500	5811	149000
	s38417	58225	685	50660	29800	49215	57900	48110	113200	47005	276500	46070	542000
	s38584	22680	405	10584	9450	8400	15000	7056	25200	5152	46000	4088	73000
Mintest	s9234	11742	309	8930	11750	8436	22200	8018	42200	7562	99500	7258	191000
	s13207	9386	361	5980	11500	5018	19300	4342	33400	3276	63000	2548	98000
	s15850	12672	384	7689	11650	6930	21000	6402	38800	5775	87500	5379	163000
	s38417	74385	855	64728	37200	63771	73300	62379	143400	61161	351500	59943	689000
	s38584	23664	464	11730	11500	9537	18700	7548	29600	5559	54500	4233	83000

Πίνακας 5-2: Αποτελέσματα της πρωτότυπης μεθόδου ανατροφοδότησης για 32 αλυσίδες ελέγχου

Στον πίνακα 5.2 παρουσιάζεται επίσης και ο όγκος των δεδομένων ελέγχου (Test Data Volume - TDM) που υπολογίζεται ως το γινόμενο του μήκους του ολισθητή και του πλήθους των αρχικών καταστάσεων καθώς και το μήκος της ακολουθίας ελέγχου (Test

Sequence Length - TSL), που είναι το πλήθος των διανυσμάτων που εφαρμόζεται σε κάθε πυρήνα και ισούται με το γινόμενο του παράθυρου L και του πλήθους των αρχικών καταστάσεων, για 32 αλυσίδες ελέγχου.

		L=1		L=50		L=100		L=200		L=500		L=1000	
circuit		TDV	TSL	TDV	TSL	TDV	TSL	TDV	TSL	TDV	TSL	TDV	TSL
Atalanta	s9234	11362	247	8280	9000	7820	17000	7636	33200	7038	76500	6532	142000
	s13207	8875	355	5325	10650	4575	18300	3950	31600	2975	59500	2125	85000
	s15850	11800	295	7400	9250	7000	17500	6720	33600	6160	77000	5720	143000
	s38417	58899	677	50286	28900	49329	56700	48111	110600	46632	268000	45762	526000
	s38584	22512	402	10528	9400	8736	15600	6832	24400	5264	47000	4144	74000
Mintest	s9234	12012	308	8736	11200	8424	21600	8190	42000	7644	98000	7215	185000
	s13207	9450	350	5967	11050	5184	19200	4401	32600	3429	63500	2673	99000
	s15850	12985	371	7770	11100	7000	20000	6580	37600	5810	83000	5250	150000
	s38417	75116	844	65237	36650	63902	71800	62478	140400	60609	340500	60164	676000
	s38584	23856	426	11760	10500	9968	17800	7952	28400	5824	52000	4480	80000

Πίνακας 5-3: Αποτελέσματα της πρωτότυπης μεθόδου ανατροφοδότησης για 64 αλυσίδες ελέγχου

Τα αντίστοιχα δεδομένα για 64 αλυσίδες ελέγχου φαίνονται στον παραπάνω πίνακα. Πρέπει να παρατηρηθεί, ότι η προτεινόμενη μέθοδος δεν επηρεάζει την ποσότητα του όγκου των δεδομένων ελέγχου της πρωτότυπης μεθόδου. Από τους πίνακα 5.2 και 5.3 γίνεται φανερό ότι όταν το L αυξάνει, ο όγκος των δεδομένων μειώνεται. Ωστόσο, την ίδια στιγμή το μήκος της ακολουθίας ελέγχου αυξάνει ραγδαία. Θα εφαρμοστεί η προτεινόμενη μέθοδος στις ακολουθίες ελέγχου που φαίνονται στον πίνακα 5.2 για  $L \geq 50$ , με σκοπό να μειωθεί το μήκος τους. Πρέπει να σημειωθεί ότι η βελτίωση του μήκους της ακολουθίας ελέγχου υπολογίζεται με βάση τον παρακάτω τύπο:

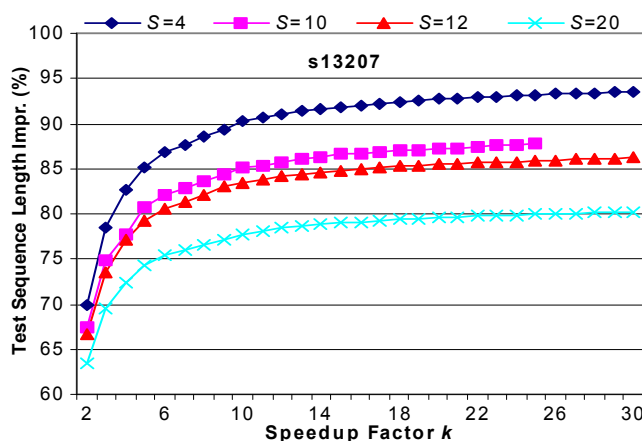
$$TSL \text{ Improvement}(\%) = \left( 1 - \frac{TSL \text{ of prop. method}}{TSL \text{ of orig. windowbased method}} \right) \cdot 100 \quad (2)$$

## 5.2. Επίδραση $k$ , $S$ , $L$ στη Βελτίωση της Ακολουθίας Ελέγχου

Σαν πρώτο βήμα μελετήθηκε η επίδραση στην βελτίωση του μήκους της ακολουθίας ελέγχου, που επιτυγχάνεται από την προτεινόμενη μέθοδο, του παράγοντα επιτάχυνση ανάδρασης  $k$ , του μεγέθους  $S$  ενός τμήματος και του παράθουρου  $L$ . Εξαιτίας του μεγάλου όγκου των πειραμάτων που έγιναν, θα επικεντρωθούν τα αποτελέσματα στα πειράματα που έγιναν στο κύκλωμα s13207 (στο παράρτημα υπάρχουν αναλυτικά όλα τα αποτελέσματα) και στα διανύσματα ελέγχου που παρέχονται από το Atalanta. Ωστόσο, τα υπόλοιπα κυκλώματα παρουσιάζουν παρόμοια συμπεριφορά και για τα δύο είδη διανυσμάτων ελέγχου.

### 5.2.1. Επίδραση $k$ , $S$ σε Σταθερό $L$

Στο πρώτο σύνολο πειραμάτων (σχήμα 5.2) φαίνεται η επίδραση του παράγοντα επιτάχυνσης ανάδρασης  $k$  στο μήκος της ακολουθίας ελέγχου για διάφορες τιμές του μεγέθους των τμημάτων  $S$ .



Σχήμα 5-1: Βελτίωση του μήκους της ακολουθίας ελέγχου για διάφορες τιμές  $k$ ,  $S$

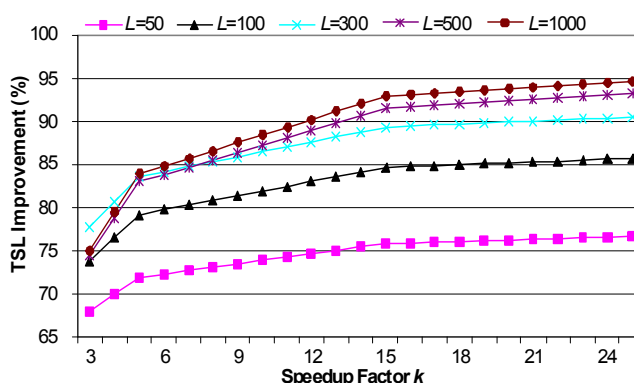
Παρουσιάζονται αποτελέσματα για  $2 \leq k \leq 30$ , και  $S = 4, 10, 12$  και  $20$  θεωρώντας παράθουρο  $L = 300$  διανύσματα. Από το σχήμα 6 γίνεται αντιληπτό ότι η βελτίωση του

μήκους της ακολουθίας ελέγχου είναι τεράστια αρχίζοντας από 63-70% για τον μικρότερο δυνατό παράγοντα επιτάχυνσης ( $k = 2$ ) και καταλήγοντας σε 80-94% για  $k = 30$  για τα διάφορα μεγέθη των τμημάτων που εξετάστηκαν.

Παρατηρείστε ότι η βελτίωση αυξάνεται, όταν ο παράγοντας επιτάχυνσης  $k$  αυξάνεται και/ή όταν το μέγεθος των τμημάτων  $S$  μειώνεται. Όταν το  $k$  αυξάνεται, ο αριθμός των κύκλων, που χρειάζονται για να παραβληθούν τα άχρηστα τμήματα, μειώνεται και έτσι μειώνεται και το TSL. Όταν μειώνεται το  $S$ , τότε η τμηματοποίηση των παραθύρων έχει καλύτερα αποτελέσματα μια και το ποσοστό των χρήσιμων τμημάτων προς το συνολικό αριθμό των τμημάτων μειώνεται. Αυτό εξηγείται και από το γεγονός ότι ένα χρήσιμο τμήμα μπορεί να περιέχει και διανύσματα που δεν κωδικοποιούν κάποιο κύβο έλεγχου, γεγονός που εξαρτάται από το μέγεθος του τμήματος  $S$ . Μειώνοντας το  $S$  λιγότερα άχρηστα διανύσματα ενσωματώνονται στα χρήσιμα τμήματα. Αν αναλογιστεί κανείς ότι ένα άχρηστο τμήμα παράγεται γρηγορότερα από ένα χρήσιμο, μια και το μεγαλύτερό του τμήμα παραβλέπεται, καταλήγει στο συμπέρασμα ότι το μήκος της ακολουθίας ελέγχου μειώνεται. Πρέπει να σημειωθεί, ότι για τιμές του  $k$  μεγαλύτερες του 25, δεν παρατηρήθηκαν ιδιαίτερα μεγαλύτερες βελτιώσεις.

### 5.2.2. Επίδραση $k$ , $L$ σε Σταθερό $S$

Στα επόμενα πειράματα (σχήμα 5.2), θα μελετηθεί η επίδραση του παράγοντα επιτάχυνσης  $k$  στο μήκος της ακολουθίας ελέγχου (TSL) για διάφορες τιμές παράθυρου ( $L$ ).



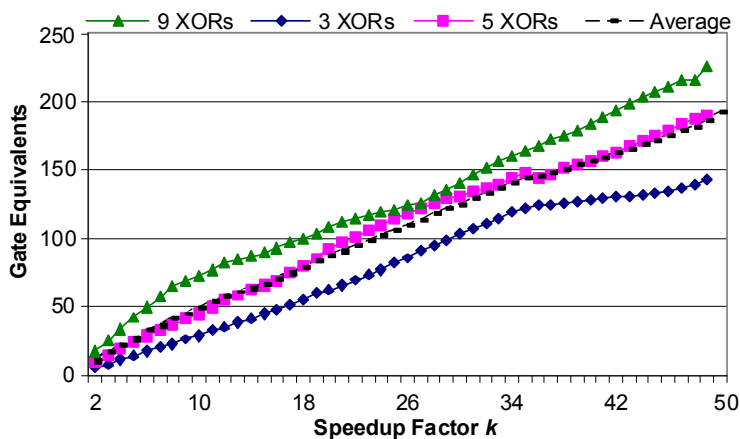
Σχήμα 5-2: Βελτίωση του μήκους της ακολουθίας ελέγχου για διάφορες τιμές του  $k$ ,  $L$  (s13207)

Στο σχήμα φαίνεται η βελτίωση του μήκους της ακολουθίας ελέγχου για  $3 \leq k \leq 25$  και  $L=50, 100, 300, 500, 1000$  ( το μέγεθος των τμημάτων  $S$  ήταν 5 για όλα τα πειράματα). Παρατηρούμε ότι όσο το  $L$  αυξάνεται, τόσο αυξάνεται και η βελτίωση του TSL. Αυτό εξηγείται από το γεγονός ότι τα μεγαλύτερα παράθυρα περιέχουν και περισσότερα άχρηστα τμήματα και το μήκος των άχρηστων τμημάτων μειώνεται δραστικά από την προτεινόμενη τεχνική. Πρέπει να σημειωθεί ότι για  $k \geq 20$  η βελτίωση που παρατηρείται φθάνει σε κορεσμό.

### 5.3. Κόστος Υλικού του Κυκλώματος Παράβλεψης

Όπως έχει ήδη αναφερθεί, ο παράγοντας της επιτάχυνσης  $k$  είναι σημαντικός για την μείωση του μήκους της ακολουθίας ελέγχου TSL. Όμως, ο παράγοντας  $k$  επηρεάζει και το κόστος σε υλικό του κυκλώματος παράβλεψης (State Skip Circuit). Για τον υπολογισμό του κόστους αυτού χρησιμοποιήθηκε ένα εμπορικό εργαλείο για την σύνθεση κυκλωμάτων παράβλεψης που προκύπτουν από τρία διαφορετικά πρώτα χαρακτηριστικά πολυώνυμα LFSR μεγέθους 65 με 3, 5 και 9 εσωτερικές πύλες αποκλειστικό-Η (internal XORs) και εξετάζοντας όλες τις τιμές του  $k$  στο διάστημα [2,50]. Προσέξτε ότι στα πειράματα αυτά, τα χαρακτηριστικά πολυώνυμα δεν είναι ίδια με αυτά που παρουσιάστηκαν στον πίνακα 5.1.

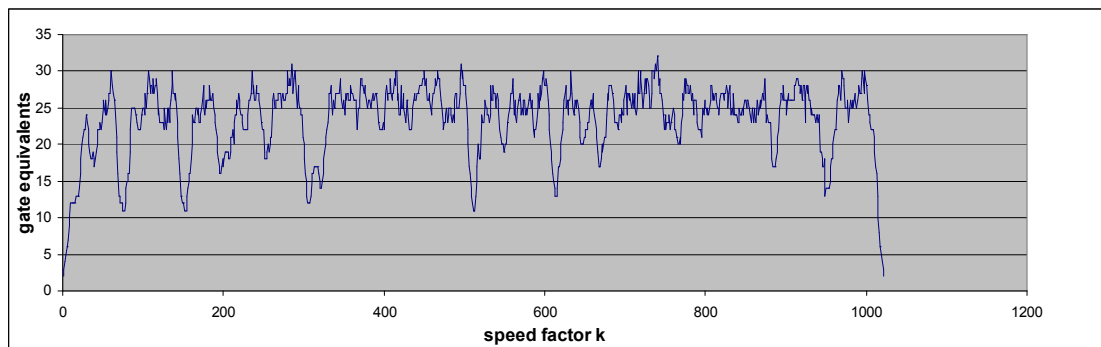
Στο σχήμα 5.3, παρουσιάζεται το κόστος του υλικού του κυκλώματος παράβλεψης καταστάσεων σε ισοδύναμες πύλες (gate equivalents – μια ισοδύναμη πύλη είναι το κόστος μιας δύο εισόδων όχι-και πύλης). Φαίνεται ότι το κόστος αυτό αυξάνει α) σχεδόν γραμμικά με το  $k$  και β) όταν αυξάνεται το πλήθος των εσωτερικών αποκλειστικό-Η πυλών. Και οι δύο αυτές παρατηρήσεις εξηγούνται από το γεγονός ότι σε κάθε κύκλο ρολογιού, σε κανονική κατάσταση λειτουργίας, εκτελείται ένα πλήθος από αποκλειστικό-Η πράξεις, το οποίο είναι ίσο με το πλήθος των εσωτερικών αποκλειστικό-Η πυλών του LFSR. Στην κατάσταση λειτουργίας παράβλεψης καταστάσεων, το κύκλωμα παράβλεψης καταστάσεων εκτελεί σε ένα κύκλο τις γραμμικές πράξεις που απαιτούν  $k$  κύκλους για την εκτέλεσή τους, σε κανονική κατάσταση λειτουργίας. Έτσι, όταν το πλήθος των εσωτερικών πυλών αποκλειστικό-Η και/ή ο παράγοντας επιτάχυνσης  $k$  αυξάνονται, το πλήθος των γραμμικών πράξεων αυξάνεται και σαν αποτέλεσμα το κόστος σε υλικό του κυκλώματος παράβλεψης γίνεται μεγαλύτερο.



Σχήμα 5-3: Κόστος υλικού του κυκλώματος παράβλεψης κατάστασης

Ωστόσο, όπως έχει φανεί και από τα προηγούμενα πειράματα, η μεγιστοποίηση της βελτίωσης του μήκους της ακολουθίας ελέγχου επιτυγχάνεται από τιμές του  $k$  μέσα στο διάστημα  $[10, 25]$ . Μέσα σε αυτό το διάστημα, όπως φαίνεται και από το σχήμα, το

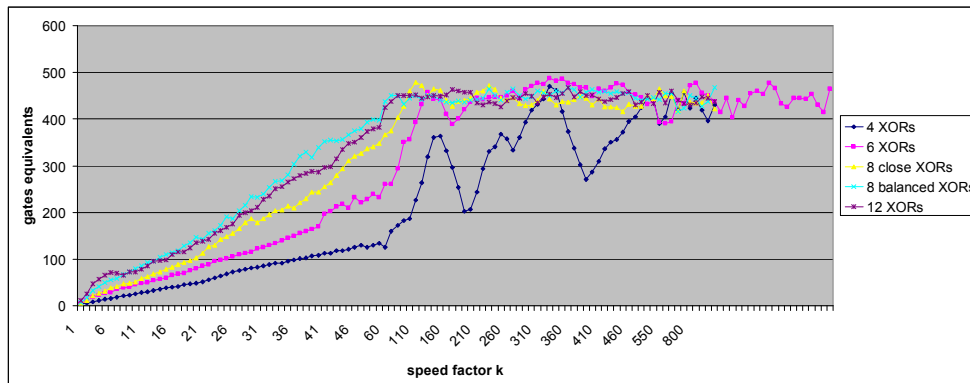
κόστος υλικού του αντίστοιχου κυκλώματος παράβλεψης είναι μικρό για όλα τα χαρακτηριστικά πολυώνυμα (μεταξύ 50 με 100 ισοδύναμες πύλες είναι η συνήθεις τιμές). Συμπεραίνουμε λοιπόν, ότι μια τιμή του  $k$  στο διάστημα  $[10, 25]$  είναι μια πολύ καλή επιλογή (πρέπει να σημειωθεί ότι το ίδιο κόστος σε υλικό παρουσιάζει και το κύκλωμα παραγωγής μοτίβων της προηγούμενης μεθόδου).



Στο παραπάνω σχήμα βλέπουμε το μέγεθος ενός XOR δένδρου που παράγεται από ένα LFSR μήκους 10 για όλα τα  $2^{10}-1 = 1023$  πιθανά  $k$ . Παρατηρούμε ότι υπάρχουν πολύ μεγάλα  $k$  για τα οποία παρουσιάζεται πτώση του κόστους σε υλικό του κύκλωμα παράβλεψης. Το γεγονός αυτό, επειδή το ίδιο κύκλωμα παράβλεψης μπορεί να χρησιμοποιηθεί για τον έλεγχο οποιουδήποτε πυρήνα, αφήνει περιθώρια για ακόμα μεγαλύτερες βελτιώσεις στο μήκος των ακολουθιών ελέγχου με κόστος επιπλέον υλικού.

Στο παρακάτω σχήμα φαίνεται το μέγεθος των XOR δένδρων για LFSR μήκους 50 με 5 διαφορετικά χαρακτηριστικά πολυώνυμα, οι γραμμικές πράξεις των οποίων είναι 4, 6, 8, 8 και 12. Όπως φαίνεται όσο πληθαίνουν οι XORs του χαρακτηριστικού πολυωνύμου αυξάνεται και το πλήθος από πύλες που απαιτούνται για τον παράγοντα από το κύκλωμα παράβλεψης. Η διαφορά που έχουν τα δύο πειράματα που έγιναν με ίδιο αριθμό από XORs (8 dense, 8 sparse) είναι ότι το ένα είχε 8 XORs σε κοντινούς παράγοντες στο χαρακτηριστικό πολυώνυμο ενώ το άλλο ήταν ομαλά κατανεμημένες οι XORs στην τάξη του πολυωνύμου.





### 5.4. Αποτελέσματα Προτεινόμενης Μεθόδου

Στον πίνακα 5.4 παρουσιάζονται οι τιμές του μήκους της ακολουθίας ελέγχου για  $L=50, 100, 200, 500$  και  $1000$ ,  $S=2, 5, 10, 20$  και  $50$ , και  $2 \leq k \leq 23$  για 32 αλυσίδες ελέγχου.

	S	Atalanta										Mintest									
		L=50		L=100		L=200		L=500		L=1000		L=50		L=100		L=200		L=500		L=1000	
		k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL
e9234	2	18	1082	18	1348	18	1784	18	3055	18	5192	18	1292	18	1533	18	1983	18	3411	18	5704
	5	15	2006	15	2395	15	2969	15	4468	15	6940	15	2522	15	2775	15	3360	15	4926	15	7421
	10			18	3752	18	4348	18	5666	18	7739			18	4524	18	4982	18	6244	18	8057
	20					20	7150	20	8415	20	10143					20	8079	20	9288	20	10743
	Impr.	78%-88%		78%-92%		78%-94%		89%-96%		93%-96%		79%-89%		80%-93%		81%-95%		91%-97%		94%-97%	
e13207	2	23	1309	23	1510	23	1756	23	2701	23	3809	23	1314	23	1475	23	1892	23	2792	23	4230
	5	23	2604	23	2779	23	3063	23	3929	23	4962	23	2594	23	2755	23	3151	23	3983	23	5221
	10			23	4629	23	4987	23	5766	23	6793			23	4433	23	5020	23	5831	23	7121
	20					23	8141	23	9014	23	9846					23	8257	23	8995	23	10473
	Impr.	77%-88%		76%-92%		74%-94%		84%-95%		89%-96%		77%-89%		77%-92%		75%-94%		86%-96%		89%-96%	
e15850	2	21	1129	21	1349	21	1740	21	2791	21	4901	21	1404	21	1666	21	2162	21	3416	21	5643
	5	21	2131	21	2378	21	2757	21	3752	21	5389	21	2711	21	3053	21	3508	21	4666	21	6654
	10			21	3937	21	4361	21	5267	21	7008			21	5049	21	5568	21	6520	21	8473
	20					21	7224	21	8218	21	9863					21	9241	21	9930	21	11866
	Impr.	78%-88%		78%-92%		79%-95%		90%-96%		93%-97%		77%-88%		76%-92%		76%-94%		89%-96%		93%-97%	
e38417	2	2	8414	2	14888	2	28685	2	62257	2	128918	2	9619	2	17494	2	34335	2	79860	2	155787
	5	5	7626	5	10289	5	13113	5	27728	5	51892	5	9035	5	12158	5	18584	5	36390	5	79378
	10			10	12952	10	16138	10	21865	10	34067			10	15470	10	19016	10	27791	10	41933
	20					20	23606	20	26936	20	33043					20	28528	20	34180	20	41568
	Impr.	72%-74%		74%-82%		75%-88%		77%-92%		76%-94%		74%-76%		76%-83%		76%-87%		77%-92%		77%-94%	
e38584	2	2	4146	2	6693	2	11155	2	20884	2	34254	2	4818	2	7719	2	12210	2	24545	2	36928
	5	5	3805	5	4991	5	6639	5	10556	5	15685	5	4526	5	5616	5	7363	5	11866	5	17281
	10					10	7082	10	9054	10	11218					10	7686	10	10103	10	12844
	20					20	10967	20	12101							20	12131	20	12131	20	13945
	Impr.	56%-60%		55%-67%		56%-74%		55%-80%		53%-85%		58%-61%		59%-70%		59%-75%		55%-81%		56%-85%	

Πίνακας 5-4: Αποτελέσματα της παρούσας μεθόδου για 32 αλυσίδες ελέγχου

Ο όγκος των δεδομένων ελέγχου καθώς και το μήκος των ακολουθιών ελέγχου της πρωτότυπης τεχνικής πριν την παρουσίαση της εργασίας αυτής φαίνονται στον

πίνακα 5.2. Στο αριστερό (δεξί) μέρος, του πίνακα 5.2, φαίνονται οι τιμές του  $k$  και του TSL για τα διανύσματα ελέγχου Atalanta (Mintest). Στην τελευταία γραμμή, σε κάθε κύκλωμα, παρουσιάζεται το διάστημα της βελτίωσης του μήκους της ακολουθίας ελέγχου που επιτεύχθηκε. Αντίστοιχα στον πίνακα 5.5 φαίνονται τα αποτελέσματα για 64 αλυσίδες ελέγχου που μπορούν να συγκριθούν με αυτά του 5.3. Παρατηρούμε ότι για 64 αλυσίδες ελέγχου παρουσιάζεται μεγαλύτερη βελτίωση στη μείωση της ακολουθίας ελέγχου. Η βελτίωση για τις 64 αλυσίδες ελέγχου είναι λίγο καλύτερη από τις 32 για τα μικρά κυκλώματα (s9234, s13207, s15850). Για τα μεγάλα κυκλώματα (s38417, s38584), όπως μπορείτε να δείτε, είναι αρκετά μεγαλύτερη (ενδεικτικά για το s38417 είναι 97% και για το s38584 είναι 94%).

		Atalanta										Mintest										
		L-50		L-100		L-200		L-500		L-1000		L-50		L-100		L-200		L-500		L-1000		
S	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL	k	TSL
s9234	2	10	1242	10	1736	10	2471	10	4727	10	8944	10	1535	10	1907	10	2711	10	5054	10	9259	
	5	25	2048	25	2332	25	2695	25	3576	25	4998	25	2516	25	2709	25	2969	25	3848	25	5731	
	10			25	3995	25	4411	25	5247	25	6669			25	4456	25	4763	25	5629	25	7142	
	20					25	7340	25	8296	25	9630					25	6006	25	8993	25	10662	
	Impr.																					
s13207	2	24	1261	24	1430	24	1761	24	2465	24	3584	24	1265	24	1461	24	1794	24	2610	24	3842	
	5	20	2521	20	2729	20	3182	20	3997	20	5207	20	2417	20	2751	20	3140	20	3964	20	5512	
	10			24	4533	24	4982	24	5454	24	6622			24	4412	24	4792	24	5403	24	6897	
	20					24	7945	24	8474	24	9868					24	7920	24	8691	24	9864	
	Impr.																					
s15850	2	22	1164	22	1331	22	1703	22	2837	22	4663	22	1409	22	1619	22	2031	22	3326	22	5376	
	5	11	2359	11	2716	11	3415	11	5668	11	9249	11	2832	11	3288	11	4066	11	6559	11	10783	
	10			22	3962	22	4337	22	5419	22	7090			22	4796	22	5208	22	6247	22	8303	
	20					22	7112	22	8393	22	9919					22	8569	22	9432	22	11527	
	Impr.																					
s38417	2	18	3336	18	4092	18	5544	18	9623	18	15906	18	3972	18	5007	18	6673	18	12501	18	20457	
	5	15	6505	15	7513	15	9209	15	14096	15	21015	15	7668	15	9206	15	11140	15	18101	15	27112	
	10			18	12220	18	13720	18	17889	18	23523			18	15056	18	16873	18	22813	18	30109	
	20					20	22668	20	27267	20	31977					20	28368	20	34882	20	41292	
	Impr.																					
s38584	2	24	1975	24	2210	24	2561	24	3346	24	4267	24	2077	24	2379	24	2775	24	3720	24	4781	
	5	24	3420	24	3761	24	4134	24	4900	24	5954	24	3481	24	3975	24	4478	24	5563	24	6564	
	10					24	6260	24	7027	24	8046					24	6800	24	7925	24	9065	
	20					24	10434	24	11503							24	11742	24	13209			
	Impr.																					

Πίνακας 5-5: Αποτελέσματα της παρούσας μεθόδου για 64 αλυσίδες ελέγχου.

### 5.5. Συγκρίσεις

Επόμενο βήμα είναι να συγκρίνουμε την προτεινόμενη μέθοδο με τις πιο αποδοτικές μεθόδους ενσωμάτωσης διανυσμάτων ελέγχου και συμπίεσης διανυσμάτων στη βιβλιογραφία, οι οποίες είναι για άγνωστες δομές πυρήνων (IP Cores) με πολλαπλές αλυσίδες ελέγχου. Πρέπει να παρατηρηθεί, ότι δεν γίνονται συγκρίσεις με μεθόδους που χρησιμοποιούν γνώση της δομής των κυκλωμάτων που πρόκειται να εξεταστούν ή απαιτούν τη χρήση αυτόματης παραγωγής διανυσμάτων ελέγχου (ATPG) επειδή θέτουν πολλούς περιορισμούς. Επίσης, κωδικοποιούν συγχωνεύοντας δεδομένα στην υπάρχουσα κωδικοποίηση γεγονός που προκαλεί μεγάλη μείωση στον όγκο των δεδομένων που πρόκειται να συμπεστούν. Όμως, έχει ήδη αναφερθεί ότι σε πυρήνες άγνωστης δομής δεν είναι δυνατό να εφαρμοστεί ούτε αυτόματη παραγωγή μοτίβων ελέγχου, ούτε προσομοίωση λαθών μια και είναι διαδικασίες που απαιτούν γνώση της δομής του πυρήνα και δεν είναι κατάλληλες για IP πυρήνες. Επίσης, δεν υπάρχουν αποτελέσματα συγκρίσεων με την προσέγγιση στο [9] μια και χρειάζεται η απαίτηση διάφορων συνθηκών από έναν πυρήνα του υπό έλεγχο κυκλώματος για να χρησιμοποιηθεί σαν αποκωδικοποίητης.

Η βελτίωση του μήκους της ακολουθίας ελέγχου υπολογίζεται σύμφωνα με τη σχέση (2), αντικαθιστώντας την τιμή "*TSL of the orig. window based method*" με την τιμή "*TSL of compared method*" (μήκος ακολουθίας ελέγχου της προτεινόμενης τεχνικής).

Στον πίνακα 5.6 παρουσιάζονται αποτελέσματα σύγκρισης της προτεινόμενης αρχιτεκτονικής, συγκριτικά με την τεχνικής ενσωμάτωσης διανυσμάτων ελέγχου του [21].

Circuit	[21]		Proposed				TSL Impr.
	TDV	TSL	L	S	TDV	TSL	
s13207	2688	38976	500	2	2688	2701	<b>93.10%</b>
s15850	5616	44928	1000	2	5811	4901	<b>89.10%</b>
s38417	43860	154800	1000	20	46070	33043	<b>78.70%</b>
s38584	4312	38500	1000	10	4088	11218	<b>70.90%</b>

Πίνακας 5-6: Σύγκριση με [21]

Για να είναι όσο το δυνατό πιο δίκαιη η σύγκριση σε σχέση με το μήκος της ακολουθίας ελέγχου η τιμή του παραθύρου  $L$  για την προτεινόμενη μέθοδο επιλέχθηκε από τον πίνακα 5.2 (μόνο 32 αλυσίδες ελέγχου, στις 64 έχουμε καλύτερα αποτελέσματα) με τέτοιο τρόπο ώστε και οι δύο μέθοδοι να έχουν σχεδόν τον ίδιο όγκο δεδομένων ελέγχου (TDV). Γίνεται φανερό ότι για παρόμοιο όγκο δεδομένων ελέγχου, (TDV) η προτεινόμενη μέθοδος απαιτεί πολύ μικρότερο μήκος της ακολουθίας ελέγχου.

Circuit	L=300			L=500		
	[22]	Prop.	TSL Impr.	[22]	Prop.	TSL Impr.
	<b>Atalanta</b>					
s9234	24592	2163	<b>91.20%</b>	41552	3055	<b>92.60%</b>
s13207	24724	2072	<b>91.60%</b>	39664	2701	<b>93.20%</b>
s15850	27630	2138	<b>92.30%</b>	45495	2791	<b>93.90%</b>
s38417	85885	18512	<b>78.40%</b>	130100	21865	<b>83.20%</b>
s38584	29358	7489	<b>74.50%</b>	41370	9054	<b>78.10%</b>
<b>Mintest</b>						
s9234	27570	2428	<b>91.20%</b>	47120	3411	<b>92.80%</b>
s13207	27485	2210	<b>92.00%</b>	41589	2792	<b>93.30%</b>
s15850	32320	2591	<b>92.00%</b>	51422	3416	<b>93.40%</b>
s38417	109485	21964	<b>79.90%</b>	165636	27791	<b>83.20%</b>
s38584	34398	8885	<b>74.20%</b>	48894	10103	<b>79.30%</b>

Πίνακας 5-7: Σύγκριση με [22]

Στον πίνακα 5.7 παρουσιάζονται τα συγκριτικά αποτελέσματα της τεχνικής ενσωμάτωσης διανυσμάτων ελέγχου του [22] σε σχέση με αυτά της προτεινόμενης τεχνικής για μέγεθος παραθύρου 300 και 500. Επειδή και οι δύο μέθοδοι μειώνουν το μήκος των παραθύρων κάθε αρχικής κατάστασης, για να είναι δίκαιη η σύγκριση

εκτελέστηκε και η μέθοδος μείωση της ακολουθίας ελέγχου του [22] για τα χαρακτηριστικά πολυώνυμα των LFSR και των ολισθητών φάσεων που χρησιμοποιούνται στην εργασία αυτή (δηλαδή οι ίδιες ακολουθίες ελέγχου μειώθηκαν και από τις δύο μεθόδους). Πρέπει να σημειωθεί ότι η μέθοδος [22] απαιτεί λίγο μεγαλύτερο όγκο δεδομένων (TDV) από την προτεινόμενη μέθοδο, αφού επιβαρύνεται από τη χρήση ενός επιπλέον ψηφίου ανά αρχική κατάσταση για να μειώσει το μήκος του παραθύρου σε κάθε αρχική κατάσταση.

Από τον πίνακα 5.7 δεν υπάρχει καμία αμφιβολία για την πιο αποδοτική μείωση του μήκους της ακολουθίας ελέγχου της προτεινόμενης αρχιτεκτονικής συγκριτικά με την [22], προσέξτε ότι για 64 αλυσίδες ελέγχου τα αποτελέσματα της προτεινόμενης μεθόδου είναι ακόμα καλύτερα.

Circuit	[32]		Proposed (L=500)		TSL Impr.
	TDV	TSL	TDV	TSL	
s9234	648	135765	7562	3411	97.50%
s13207	162	152596	3276	2792	98.20%
s15850	396	222336	5775	3416	98.50%
s38417	5440	625273	61161	27791	95.60%
s38584	228	383009	5559	10103	97.40%

Πίνακας 5-8: Σύγκριση με [32]

Στον πίνακα 5.8 συγκρίνονται η προτεινόμενη μέθοδος ( $L = 500$ ) με την τεχνική ενσωμάτωσης διανυσμάτων ελέγχου του [32] για 32 αλυσίδες ελέγχου για τα Mintest σύνολα ελέγχου (δεν δίδονται αποτελέσματα για τα Atalanta από την [32]). Είναι φανερό ότι η μέθοδος [32] απαιτεί ελάχιστη μνήμη από τον ελεγκτή, αλλά το μήκος των ακολουθιών ελέγχου που πυροδοτεί είναι πολύ μεγάλο. Επίσης, έχει δειχθεί από το [22] ότι το κόστος σε υλικό, που απαιτεί η μέθοδος αυτή, είναι απαγορευτικά μεγάλο, ιδίως όταν το πλήθος των αλυσίδων ελέγχου μεγαλώνει (υπολογίστηκε μεταξύ 1300-9800 ισοδύναμων πυλών για 32 αλυσίδες ελέγχου και 4500-12500 ισοδύναμες πύλες για 64 αλυσίδες ελέγχου για τα μεγαλύτερα ISCAS 89 κυκλώματα). Η προτεινόμενη μέθοδος ξεκάθαρα προσπερνά την [32] σε αποτελέσματα που αφορούν το μήκος της ακολουθίας ελέγχου.

Circuit	[3]		[27]	[31]	[47]	[28]		L=1		Prop. L=200		
	TSL	TDV	TSL	TDV		TSL	TDV	TSL	TDV	TSL	TDV	
s9234	170	15092	205	12445	9767	-	-	243	10692	1784	7128	
s13207	229	12798	266	11859	9335	10810	266	14307	369	8856	1756	3816
s15850	244	15480	269	12663	10928	12405	226	15067	298	11622	1740	6669
s38417	376	37020	376	36430	28332	32154	376	49001	685	58225	13113	48110
S38584	296	31574	296	30355	28594	31000	296	28994	405	22680	6639	7056

Πίνακας 5-9: TSL και TDV αποτελέσματα για διάφορες μεθόδους

Στον πίνακα 5.9 παρουσιάζονται τα μήκη της ακολουθίας ελέγχου και ο όγκος των δεδομένων ελέγχου για διάφορες μεθόδους που χρησιμοποιούν ανατροφοδότηση LFSR και μπορούν να χρησιμοποιηθούν για έλεγχο πυρήνων άγνωστης δομής με πολλές αλυσίδες ελέγχου [3] στην στήλη 2 και 3, [27], [31] και [47] στις στήλες 4-7, [28] στη στήλη 8 και 9 η κλασική τεχνική ανατροφοδότησης με μέγεθος παραθύρου  $L = 1$  για τα Atalanta σύνολα ελέγχου στις στήλες 10 και 11 και η προτεινόμενη τεχνική για  $L = 300$  στις στήλες 12 και 13).

Προσέξτε πως για τις μεθόδους [27], [31] και [47] το μήκος της ακολουθίας ελέγχου είναι ίσο με τον πλήθος των διανυσμάτων ελέγχου, γι' αυτό και δηλώνονται στην ίδια στήλη. Παρόλο που είναι φανερό, ότι συγκριτικά με τις περισσότερες μεθόδους λιγότερα δεδομένα χρειάζονται από την προτεινόμενη μέθοδο με το κόστος μεγαλύτερων ακολουθιών ελέγχου, δεν μπορούν να βγουν συμπεράσματα όπως με τις προηγούμενες μεθόδους μια και χρησιμοποιούνται διαφορετικά σύνολα ελέγχου. Για παράδειγμα, με τις πληροφορίες που δίδονται στο [47] υπολογίστηκε ότι στα πειράματα τις μεθόδου αυτής τα διανύσματα ελέγχου είχαν 2 με 3 φορές μεγαλύτερο αριθμό από ορισμένα ψηφία (0 ή 1) από αυτά που χρησιμοποιήθηκαν στο [47].

Είναι γεγονός, ότι η απόδοση μιας τεχνικής κωδικοποίησης εξαρτάται σε μεγάλο βαθμό από τον όγκο των ορισμένων ψηφίων των διανυσμάτων ελέγχου και έτσι δεν είναι δυνατό να συγκριθούν τεχνικές που δεν εφαρμόζεται στο ίδιο σύνολο διανυσμάτων ελέγχου. Πιο συνεπής και ακριβής σύγκριση μπορεί να γίνει μεταξύ μεθόδων που

χρησιμοποιούν σύνολα ελέγχου, που παράγονται από το ίδιο εργαλείο αυτόματης δημιουργίας μοτίβων ελέγχου.

Circuit	TSL		TDV			
	Mintest-[33]-[40]	Prop. L=200	Mintest	[33]	[40]	Prop. L=200
s9234	159	1983	39273	30144	-	8018
s13207	236	1892	165200	20988	74423	4342
s15850	126	2162	76986	25140	26021	6402
s38417	99	18584	164736	85225	45003	62379
s38584	136	7363	199104	57120	73464	7548

Πίνακας 5-10: Συγκρίσεις με Mintest, [33] και [40]

Ο πίνακας 5-10 παρουσιάζει συγκριτικά αποτελέσματα με τα συμπιεσμένα Mintest διανύσματα ελέγχου και τις τεχνικές [33] και [40], που επιτυγχάνουν περαιτέρω συμπίεση των διανυσμάτων αυτών και είναι μέθοδοι για πυρήνες άγνωστης δομής με δυνατότητα χρήσης πολλών αλυσίδων ελέγχου. Όπως φαίνεται, σε όλες εκτός από μία περίπτωση (s38417 στο [40]), η προτεινόμενη τεχνική έχει καλύτερα αποτελέσματα σε σχέση με τον όγκο των δεδομένων ελέγχου που απαιτούνται αλλά απαιτεί μεγαλύτερο μήκος ακολουθιών ελέγχου (πρέπει να σημειωθεί ότι στο [40] ένα μεγάλο μέρος των δεδομένων που πρέπει να αποθηκευτούν στον ελεγκτή δεν δηλώνεται).

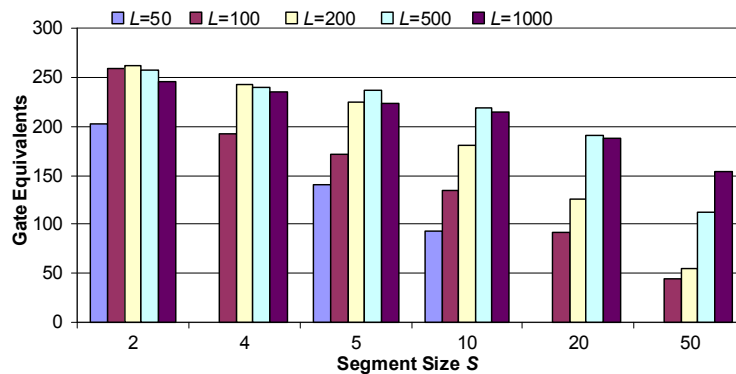
Στους πίνακες 5.9 και 5.10 φαίνονται οι δύο επιλογές που δίδονται για τον έλεγχο πυρήνων άγνωστης δομής: συμπίεση δεδομένων ελέγχου (μεγάλος όγκος δεδομένων ελέγχου – μικρού μήκους ακολουθίες ελέγχου) και ενσωμάτωση διανυσμάτων ελέγχου (μικρός όγκος δεδομένων – μεγάλου μήκους ακολουθίες ελέγχου). Πριν από αυτή την εργασία, το μήκος ακολουθιών της δεύτερης κατηγορίας ήταν απαγορευτικά μεγάλο. Ο Γραμμικός Ολισθητής Ανάδρασης και Παράβλεψης Κατάστασης (State Skip LFSR) γεφυρώνει το χάσμα των δύο τεχνικών, προσφέροντας τη μεγάλη συμπίεση του όγκου των δεδομένων που προσφέρουν οι τεχνικές ενσωμάτωσης διανυσμάτων ελέγχου με τις μικρές ακολουθίες ελέγχου.

Λαμβάνοντας υπόψη τόσο το μεγάλο πλήθος των αλυσίδων ελέγχου (λίγοι εσωτερικοί κύκλοι χρειάζονται για να παραχθεί ένα διάνυσμα) τόσο τον μικρό όγκο των δεδομένων (συγκριτικά με τις μεθόδους συμπίεσης δεδομένων ελέγχου) που χρειάζεται να μεταφερθούν μέσω της αργής σύνδεσης του ελεγκτή με το υπό έλεγχο κύκλωμα όσο και το μικρό μήκος της ακολουθίας ελέγχου καταλήγουμε στο συμπέρασμα πως η τεχνική που παρουσιάστηκε στην εργασία αυτή σηματοδοτεί μια νέα πιο δυνατή αντιμετώπιση του προβλήματος ελέγχου ορθής λειτουργίας ολοκληρωμένων κυκλωμάτων.

### 5.6. Κόστος Υλικού

Κλείνοντας το κεφάλαιο των πειραματικών αποτελεσμάτων, θα παρουσιαστούν πειράματα που αφορούν το κόστος σε υλικό της προτεινόμενης τεχνικής. Επίκεντρο των πειραμάτων θα είναι για μία ακόμη φορά το s13207 κύκλωμα (στο παράρτημα μπορούν να βρεθούν και πειράματα που αφορούν και άλλα κυκλώματα), αλλά και για τα υπόλοιπα κυκλώματα τα αποτελέσματα είναι παρεμφερή. Έχει ήδη δειχθεί, ότι το κόστος του κυκλώματος παράβλεψης είναι πολύ μικρό για τους παράγοντες επιτάχυνσης, που έχουν βέλτιστη απόδοση στη μείωση της ακολουθίας ελέγχου ( $k \leq 25$ ). Για παράδειγμα, στο κύκλωμα s13207 και για  $k = 15$  το κόστος σε υλικό του κυκλώματος παράβλεψης είναι 66 ισοδύναμες πύλες. Για το ίδιο κύκλωμα και για διάφορες τιμές των  $L$  και  $S$  ο μέσος όρος των υπόλοιπων τμημάτων (LFSR, ολισθητές φάσης, μετρητές, μονάδα ελέγχου και μονάδα αποκωδικοποίησης) χωρίς την μονάδα επιλογής ήταν γύρω στις 320 ισοδύναμες πύλες. Το κόστος αυτό είναι πολύ μικρό και παρόμοιο με αυτό που εμφανίζεται, και στις πιο γνωστές τεχνικές ενσωματωμένου ελέγχου, στη βιβλιογραφία. Επιπρόσθετα, η προαναφερθείσα λογική αποκωδικοποίησης, καθώς και το κύκλωμα παράβλεψης, χρειάζεται να υλοποιηθούν μόνο μια φορά και μπορούν να χρησιμοποιηθούν για κάθε πυρήνα. Από την άλλη μεριά, η μονάδα επιλογής (Mode Select) χρειάζεται να υλοποιηθεί για κάθε πυρήνα που πρόκειται να ελεγχθεί και εξαιτίας αυτού πρέπει να εξεταστεί διεξοδικά.





Σχήμα 5-4: Κόστος υλικού της Μονάδας Επιλογής

Στο σχήμα 5.4 παρουσιάζεται το κόστος υλικού της Μονάδας Επιλογής (Mode Select Unit) για το κύκλωμα s13207 και για  $L = 50, 100, 200, 500$  και  $1000$  και για  $S = 2, 4, 5, 10, 20$  και  $50$  για 32 αλυσίδες ελέγχου. Φαίνεται, ότι το κόστος υλικού της μονάδας αυτής μειώνεται καθώς το  $S$  μειώνεται και στις περισσότερες περιπτώσεις και όταν το  $L$  μειώνεται. Ωστόσο, σε όλες τις περιπτώσεις το κόστος της μονάδας αυτής είναι σχετικά μικρό. Για τις 64 αλυσίδες ελέγχου το κόστος υλικού της μονάδας αυτής είναι ελαφρώς μικρότερο μια και μειώνεται περισσότερο το μήκος της ακολουθίας ελέγχου.

Με σκοπό να υπολογιστεί το συνολικό κόστος της τεχνικής που παρουσιάστηκε στην εργασία αυτή, έγινε σύνθεση ενός υποθετικού συστήματος με πολλούς πυρήνες αποτελούμενου από τα πέντε κυκλώματα ISCAS 89. Για κάθε κύκλωμα τέθηκε  $L = 200$ ,  $S = 10$  και  $k = 10$ . Η μονάδα επιλογής ξεχωριστά υλοποιήθηκε για κάθε κύκλωμα. Ο συνολικός επιπλέον φόρτος σε υλικό του προτεινόμενου αποκωδικοποιητή ήταν μόλις το 6.6% της περιοχής που καταλαμβάνεται από τους πέντε πυρήνες.

## ΚΕΦΑΛΑΙΟ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ

---

Στην κωδικοποίηση διανυσμάτων ελέγχου οι ολισθητές γραμμικής ανάδρασης προσφέρουν την καλύτερη συμπίεση στην βιβλιογραφία. Ωστόσο, ως τώρα η συμπίεση αυτή προκαλεί την απαίτηση σε χρόνο για την αποκωδικοποίηση των διανυσμάτων ελέγχου. Ένας νέου τύπου ολισθητής γραμμικής ανάδρασης για ενσωμάτωση διανυσμάτων ελέγχου παρουσιάστηκε στην εργασία αυτή που εξαλείφει αυτό το μειονέκτημα. Ο ολισθητής ανάδρασης με δυνατότητα παράβλεψης καταστάσεων (State Skip LFSR) έχει τη δυνατότητα να παραβλέπει επιλεγμένα τμήματα των ψευδοτυχαίων ακολουθιών, το μέγεθος των οποίων εξαρτάται από τον παράγοντα επιτάχυνσης παράβλεψης  $k$ . Πειράματα έδειξαν ότι μια τιμή του  $k$  μέσα στο διάστημα  $[10, 25]$  είναι πάντα αποδοτική για την ελαχιστοποίηση των ακολουθιών ελέγχου και το αντίστοιχο κύκλωμα παράβλεψης έχει πολύ μικρό κόστος υλικού. Συμπερασματικά, η προτεινόμενη τεχνική ενσωμάτωσης διανυσμάτων ελέγχου συμπιέζει περισσότερο από τις κλασσικές μεθόδους ενσωμάτωσης διανυσμάτων ελέγχου, ενώ το μήκος των ψευδοτυχαίων ακολουθιών ελέγχου μειώνεται έως και 97% συγκρινόμενο με το μήκος των κλασσικών μεθόδων ενσωμάτωσης διανυσμάτων ελέγχου. Επιπρόσθετα, η επιλεκτική ανάδραση χρησιμοποιήθηκε για να βελτιώσει τη συμπίεση που προκαλείται από τους γραμμικούς ολισθητές ανάδρασης. Ο νέος ολισθητής που ονομάστηκε Ολισθητής Παραγωγής Μοτίβων έχει μικρό κόστος σε υλικό και με την εφαρμογή του για μια μόνο αρχική κατάσταση προκαλεί 5% αύξηση στην απόδοση της συμπίεσης της κλασσικής μεθόδου. Μια μέθοδος κωδικοποίησης με χρήση πολλαπλών μοτίβων που έχει 13% καλύτερη απόδοση παρουσιάστηκε. Η απόδοση του ολισθητή παραγωγής μοτίβων εξαρτάται από πολλές παραμέτρους και δίνει δυναμική στον γραμμικό ολισθητή ανάδρασης για μελλοντικές έρευνες. Το γεγονός αυτό υποσχεται ακόμη μεγαλύτερη συμπίεση με χρήση γραμμικών ολισθητών ανάδραση



## ΑΝΑΦΟΡΕΣ

---

- [1] N. Badereddine, *et. al.*, “Power-Aware Test Data Compression for Embedded IP Cores”, in *Proc. ATS*, 2006, pp. 5-10.
- [2] K. J. Balakrishnan and N. Touba, “Improving Linear Test Data Compression”, *IEEE Trans. on VLSI Systems*, vol. 14, No 11, pp. 1227-1237, Nov. 2006.
- [3] K. J. Balakrishnan, S. Wang, S. T. Chakradhar, “PIDISC: Pattern Independent Design Independent Seed Compression Technique”, in *Proc. VLSID*, 2006, pp. 811-817
- [4] C. Barnhart, *et. al.*, “OPMISR: the foundation for compressed ATPG vectors”, in *Proc. ITC*, 2001, pp. 748 – 757.
- [5] I. Bayraktaroglu and A. Orailoglu, “Concurrent application of compaction and compression for test time and data volume reduction in scan designs”, *IEEE Trans. on Comp.*, vol. 52, pp. 1480- 1489, Nov. 2003.
- [6] A. Chandra and K. Chakrabarty, “System-on-a-chip test-data compression and decompression architectures based on Golomb codes”, *IEEE Trans. on CAD*, vol. 20, pp. 355-368, Mar. 2001.
- [7] A. Chandra and K. Chakrabarty, “A unified approach to reduce SOC test data volume, scan power and testing time”, *IEEE Trans. on CAD*, vol. 22, pp. 352 – 363, Mar. 2003.
- [8] A. Chandra and K. Chakrabarty, “Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes”, *IEEE Trans. on Comp.*, vol. 52 , pp. 1076 – 1088, Aug. 2003
- [9] R. Dorsch and H. -J. Wunderlich, “Tailoring ATPG for Embedded Testing”, in *Proc. ITC*, 2001, pp. 530-537
- [10] P. T. Gonciari, B. Al-Hashimi and N. Nicolici, “Variable-length input Huffman coding for system-on-a-chip test”, *IEEE Trans. on CAD*, vol. 22 , pp. 783 – 796, June 2003.
- [11] I. Hamzaoglu and J. H. Patel, “Reducing test application time for full scan embedded cores”, in *Proc. FTCS*, June 1999, pp. 260-267.
- [12] I. Hamzaoglu and J. H. Patel, “Test set compaction algorithms for combinational circuits”, *IEEE Trans. on CAD*, vol. 19, pp. 957-963, Aug. 2000.
- [13] S. Hellebrand, *et. al.*, “Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers” *IEEE Trans. on Comp.*, vol. 44, pp. 223 – 233, Feb. 1995
- [14] S. Hellebrand, H. -G. Liang and H. -J. Wunderlich, “A mixed mode BIST scheme based on reseeding of folding counters”, *J. of Electronic Testing: Theory and Applications*, vol. 17, pp. 341 - 349, June 2001.
- [15] A. Jas and N. A. Touba, “Test vector decompression via cyclical scan chains and its application to testing core-based designs”, in *Proc. ITC*, 1998, pp. 458-464.
- [16] A. Jas, J. Ghosh-Dastidar, M. Ng., N. Touba, “An efficient test vector compression scheme using selective Huffman coding”, *IEEE Trans. on CAD*, vol. 22, pp. 797-806, June 2003
- [17] A. Jas, B. Pouya and N. Touba, “Test data compression technique for embedded cores using virtual scan chains”, *IEEE Trans. on VLSI Systems*, vol. 12, pp. 775-781, July 2004

- [18] D. Kagaris, S. Tragoudas and A. Majumdar, "On the Use of Counters for Reproducing Deterministic Test Sets", *IEEE Trans. on Comp.*, vol. 45, Dec. 1996, pp. 1405-1419.
- [19] D. Kagaris & S. Tragoudas, "On the design of optimal counter based schemes for test set embedding", *IEEE Trans. on CAD*, Feb. 1999, pp.219-230.
- [20] E. Kalligeros, X. Kavousianos and D. Nikolos, "Multiphase BIST: A New Reseeding Technique for High Test Data Compression", *IEEE Trans. on CAD*, vol. 23, pp. 1429-1446, Oct. 2004.
- [21] E. Kalligeros, X. Kavousianos and D. Nikolos, "Efficient Multiphase Test Set Embedding for Scan-based Testing", in *Proc. ISQED*, 2006, pp. 433-438.
- [22] D. Kaseridis, E. Kalligeros, X. Kavousianos and D. Nikolos, "An efficient test set embedding scheme with reduced test data storage and test sequence length requirements for scan-based testing", *Inf. Pap. Dig. IEEE ETS*, 2005, pp. 147-150.
- [23] X. Kavousianos, E. Kalligeros and D. Nikolos, "Efficient Test-Data Compression for IP Cores Using Multilevel Huffman Coding", in *Proc. DATE*, 2006, pp. 1033-1038.
- [24] B. Koenemann, "LFSR-coded Test Patterns for Scan Design", in *Proc ETC*, 1991, pp. 237-242.
- [25] B. Koenemann, *et al.*, "A SmartBIST variant with guaranteed encoding", in *Proc. ATS*, 2001, pp. 325-330.
- [26] C. Krishna, A. Jas, N. Touba, "Test Vector Encoding Using Partial LFSR Reseeding", *Proc. ITC*, 2001, pp. 885-893
- [27] C. V. Krishna and N. A. Touba, "Reducing test data volume using LFSR reseeding with seed compression", in *Proc. ITC*, 2002, pp. 321-330.
- [28] C. V. Krishna and N. A. Touba, "Adjustable width linear combinational scan vector decompression", in *Proc. IEEE/ACM ICCAD*, 2003, pp. 863-866.
- [29] C. V. Krishna and N. A. Touba, "3-Stage variable length continuous-flow scan vector decompression scheme", in *Proc. 22<sup>nd</sup> IEEE VTS*, 2004, pp. 79-86.
- [30] H. K. Lee and D. S. Ha, "Atalanta: An Efficient ATPG for Combinational Circuits", TR, 93-12, Dep't of Electrical Eng., Virginia Polytechnic Institute and State University, 1993.
- [31] J. Lee, N. Touba, "Low Power Test Data Compression Based on LFSR Reseeding", in *Proc. ICCD*, 2004, pp. 180-185.
- [32] L. Li and K. Chakrabarty, "Test set embedding for deterministic BIST using A reconfigurable interconnection network", *IEEE Trans. on CAD*, vol.23, pp. 1289-1305, Sept. 2004.
- [33] L. Li *et al.*, "Efficient space/time compression to reduce test data volume and testing time for IP cores", in *Proc. 18th Int. Conf. on VLSI Des.*, 2005, pp. 53-58.
- [34] H. -G. Liang, S. Hellebrand and H. -J. Wunderlich, "Two-Dimensional Test Data Compression for Scan-Based Deterministic BIST", *Proc. of ITC*, Oct.-Nov. 2001, pp. 894-902.
- [35] S. Mitra, K. Kim, "XPAND: An Efficient Test Stimulus Compression Technique", *IEEE Trans. on Comp.*, vol. 55, No. 2, Feb. 2006, pp. 163-173.
- [36] M. Nourani and M. H. Tehranipour, "RL-Huffman encoding for test compression and power reduction in scan applications", *ACM Trans. on Design Automation of Electronic Systems*, vol. 10, pp. 91-115, Jan. 2005
- [37] J. Rajski, J. Tyszer, N. Zacharia, "Test Data Decompression for Multiple Scan Designs with Boundary Scan", *IEEE Trans. on Comp.*, vol. 47, No 11, Nov. 1998, pp. 1188-1200.
- [38] J. Rajski, J. Tyszer, M. Kassab and N. Mukherjee, "Embedded deterministic test", *IEEE Trans. on CAD*, vol. 23, pp. 776-792, May 2004.

- [39] W. Rao, I. Bayraktaroglu, A. Orailoglu, "Test Application Time and Volume Compression through Seed Overlapping", in *Proc. DAC*, 2003, pp. 732-737.
- [40] S. Reda, A. Orailoglu, "Reducing Test Application Time Through Test Data Mutation Encoding", in *Proc. DATE*, 2002, pp. 1-5.
- [41] P. Rosinger, *et. al.*, "Simultaneous reduction in volume of test data and power dissipation for systems-on-a-chip", *Electronics Letters*, vol. 37, pp. 1434–1436, Nov. 2001.
- [42] S. Swaminathan & K. Chakrabarty, "On using twisted-ring counters for test set embedding in BIST", *JETTA*, vol. 17, no. 6, Dec. 2001, pp.529-542.
- [43] M. Tehranipour, M. Nourani and K. Chakrabarty, "Nine-coded compression technique for testing embedded cores in SoCs", *IEEE Trans. on VLSI Systems*, vol. 13, pp. 719-731, June 2005.
- [44] E. Volkerink, A. Khoche, S. Mitra, "Packet-based input test data compression techniques", *Proc. ITC*, 2002, pp. 154–163.
- [45] E. Volkerink, S. Mitra, "Efficient Seed Utilization for Reseeding based Compression", *Proc. VTS*, 2003, pp. 232-237.
- [46] Z. Wang, K. Chakrabarty, "Test Data Compression for IP Embedded Cores Using Selective Encoding of Scan Slices", in *Proc. ITC*, 2005, pp. 1-10
- [47] S. Ward, C. Schattauer, N. Touba, "Using Statistical Transformations to Improve Compression for Linear Decompressors", in *Proc. DFT*, 2005, pp. 42-50.
- [48] P. Wohl, *et. al.*, "Efficient Compression of Deterministic Patterns into Multiple PRPG Seeds", *Proc. ITC*, 2005, pp. 1-10.
- [49] N. Zacharia, J. Rajski, J. Tyszer and J. A. Waicukauski, "Two-Dimensional Test Data Decompressor for Multiple Scan Designs", in *Proc. ITC*, 1996, pp. 186-19

## **ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ ΣΗΜΕΙΩΜΑ**

---

Ο Τενέντες Βασίλειος του Ευαγγέλου γεννήθηκε στην Καλαμάτα το 1981. Αποφοίτησε από το Τμήμα Πληροφορικής του Πανεπιστημίου Πειραιά το 2003 και γράφτηκε στο Μεταπτυχιακό Πρόγραμμα Σπουδών του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων. Έχει εργαστεί σαν διαχειριστής συστήματος στο Πανεπιστήμιο Πειραιά, σαν προγραμματιστής δικτυακών εφαρμογών στην EPT και σαν Μηχανικός Λογισμικού στην Siemens Enterprise Networks. Τα ερευνητικά του ενδιαφέροντα προσανατολίζονται στην πιστοποίηση λειτουργίας ψηφιακών συστημάτων και στις αποδοτικές μεθόδους κωδικοποίησης και προγραμματισμού, ενώ η επαγγελματική του δραστηριότητα σχετίζεται με διαδικτυακές εφαρμογές πολυμέσων πραγματικού χρόνου.

