

Πανεπιστήμιο Κρήτης, Ηράκλειο

Τμήμα Επιστήμης Υπολογιστών

Υλοποίηση Συνδυαστικών  
Κυκλωμάτων Πολλαπλής  
Ράγας, Μεταβλητής  
Καθυστέρησης, με  
Συστηματική Κωδικοποίηση

Μεταπτυχιακή Εργασία

Ιορδάνης Ασλανίδης

<iordanis@csd.uoc.gr>

Ηράκλειο, 6 Νοεμβρίου 2007



*Υλοποίηση Συνδυαστικών Κυκλωμάτων Πολλαπλής Ράγας,  
Μεταβλητής Καθυστερήσης, με Συστηματική Κωδικοποίηση*

Εργασία που υποβλήθηκε από τον Ιορδάνη Ασλανίδη, ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης

---

Ιορδάνης Ασλανίδης, Τμήμα Επιστήμης Υπολογιστών

---

Μανόλης Κατεβαίνης, Καθηγητής, Επόπτης, Μέλος της Εξεταστικής Επιτροπής

---

Απόστολος Τραχανίτης, Καθηγητής, Μέλος της Εξεταστικής Επιτροπής

---

Διονύσης Πνευματικάτος, Αναπλ. Καθηγητής Πολυτεχνείου Κρήτης,  
Μέλος της Εξεταστικής Επιτροπής

---

Πάνος Τραχανιάς, Καθηγητής, Διευθυντής Μεταπτυχιακών Σπουδών



Πανεπιστήμιο Κρήτης, Σχολή Θετικών και Τεχνολογικών Επιστημών  
Τμήμα Επιστήμης Υπολογιστών



# Περίληψη

Τα τελευταία χρόνια, η ασύγχρονη σχεδίαση επανέρχεται στο προσκήνιο με προτάσεις για την επίλυση μιας σειράς προβλημάτων της σχεδίασης ψηφιακών κυκλωμάτων. Τέτοια προβλήματα είναι η κλιμάκωση της απόδοσης και η μείωση της κατανάλωσης. Η πιο διαδεδομένη τεχνική για την υλοποίηση των συνδυαστικών μερών των ασύγχρονων κυκλωμάτων, π.χ. κυκλώματα QDI (Quasi Delay Insensitive), είναι ο μετασχηματισμός διπλής-ράγας. Τα κυκλώματα διπλής-ράγας, επιβάλλουν μία σημαντική επιβάρυνση στο εμβαδό των κυκλωμάτων, που ξεπερνάει σε ποσοστό το 100%, σε σχέση με τα συμβατικά μη κωδικοποιημένα ψηφιακά κυκλώματα. Αυτό είναι ένα σημαντικό μειονέκτημα για τη χρήση του μετασχηματισμού σε ολόκληρα ψηφιακά κυκλώματα, και για το λόγο αυτό περιορίζεται σε υποκυκλώματα.

Στην παρούσα εργασία, παρουσιάζεται ένας νέος μετασχηματισμός για την υλοποίηση συνδυαστικών DI κυκλωμάτων, που κάνει χρήση συστηματικών DI κωδικών. Η νέα αυτή τεχνική οδηγεί στο σχηματισμό κυκλωμάτων πολλαπλής-ράγας, και έχει ως στόχο τη μείωση της επιβάρυνσης σε εμβαδό σε σχέση με τα κυκλώματα της διπλής-ράγας. Παρουσιάζονται μεθοδολογίες για την υλοποίηση DI κυκλωμάτων, που είναι γενικές και δεν εξαρτώνται από συγκεκριμένους κώδικες. Παράλληλα, εξερευνάται η υλοποίηση κυκλωμάτων με χρήση του κώδικα Berger και ενός υποσυνόλου του κώδικα Sperner, με συστηματικές ιδιότητες. Η μεθοδολογία υλοποίησης έχει αυτοματοποιηθεί, και υποστηρίζεται από το σύστημα λογικής σύνθεσης SIS.

Τα αποτελέσματα της υλοποίησης για ένα εύρος των δειγματοληπτικών κυκλωμάτων του IWLS, έδειξαν ότι τουλάχιστον τα μισά κυκλώματα είχαν μείωση εμβαδού, καθώς και κατανάλωσης ισχύος. Επίσης, η πειραματική διαδικασία έδειξε ότι τα κυκλώματα που υλοποιήθηκαν με χρήση του, μεγαλύτερου από πλευράς αριθμού ψηφίων, κώδικα Sperner είχαν μεγαλύτερο κέρδος από αυτά που έκαναν χρήση του κώδικα Berger. Συ-

νεπώς, συμπεραίνουμε ότι ένας κώδικας με βέλτιστο αριθμό ψηφίων δεν είναι απαραίτητα η βέλτιστη επιλογή όταν χρησιμοποιείται για την υλοποίηση DI κυκλωμάτων.

# Abstract

During the past years, asynchronous design has resurfaced as a proposed solution for a number of difficult problems of digital circuits's design. These problems include performance scaling and power consumption. The most common technique for implementing the combinational parts of the asynchronous circuits i.e. quasi delay insensitive (DI) circuits, is the Dual-Rail transformation. The Dual-Rail circuits impose significant area overhead, *i.e.* 100% when compared to the area of the conventional unencoded digital circuits. This is the significant drawback for their use in a complete digital circuit, thus it is only applied on subcircuits.

In this master's thesis, a novel transformation is presented for the implementation of combinational DI circuits, which uses systematic codes. This new technique, leads to Multi-Rail encoded circuits, whose aim is to reduce the area overhead when compared to the conventional dual-rail encoding. We introduce methodologies for DI circuits which are general and are not tie to any specific DI code. In our work, we have explored systematic code implementations by the Berger code and a systematic subset of the Sperner code. The implementation methodology has been automated and is now supported by the SIS logic synthesis system.

The results of implementing the methodology on a subset of the IWLS circuits demonstrated that at least half of the benchmark circuits experienced area and power reduction. Results also showed, that circuits implemented using the longer, in bit length, Sperner code had higher benefits than those using the, sorter, Berger codes. Thus, we have concluded that a DI code that uses the optimum number of bits may not necessarily be the optimal choice when used to implement DI logic circuits.



# Ευχαριστίες

Θα ήθελα να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου της Κρήτης, για την αποδοχή της αίτησης μου για το Μεταπτυχιακό Πρόγραμμα Σπουδών του, η οποία μου έδωσε την ευκαιρία να γνωρίσω νέα πεδία έρευνας αλλά και να έρθω σε επαφή με πλήθος αξιόλογων καθηγητών και φοιτητών. Επίσης το ΙΤΕ-III για την οικονομική του υποστήριξη, αλλά και την υλικοτεχνική του προσφορά, η οποία ήταν απαραίτητη για την περάτωση της παρούσας εργασίας.

Σε πιο προσωπικό επίπεδο, θέλω να ευχαριστήσω τον Επιβλέποντά μου Κ. Χρήστο Σωτηρίου για τη διαρκή του καθοδήγηση και υποστήριξη με στόχο την τελική διαμόρφωση της εργασίας. Επιπροσθέτως, θα ήθελα να ευχαριστήσω τον Καθηγητή και Επόπτη μου Κ. Μανόλη Κατεβαίνη για τις πολύτιμες συμβουλές του σε όλη τη διάρκεια των σπουδών μου, και δεν θα μπορούσα να ξεχάσω τους Καθηγητές Κ. Απόστολο Τραγανίτη και Κ. Διονύσιο Πνευματικάτο που δέχτηκαν να είναι μέλη της επιτροπής που αξιολόγησε την παρούσα εργασία.

Επίσης, θέλω να ευχαριστήσω όσους ήταν δίπλα μου σε αυτή τη πορεία των δύο χρόνων παραμονής μου στην Κρήτη, τόσο στην Ομάδα Σχεδίασης Ασύγχρονων Κυκλωμάτων και Συστημάτων του ΙΤΕ, Ε. Κουναλάκη, Σ. Λυμπέρη, Μ. Παντελιά, Β. Κασαπάκη, Β. Ζεμπίλη, Α. Μιχαλά και ιδιαίτερα τον Π. Ματθαϊάκη, όσο και γενικότερα στους κόλπους του Πανεπιστημίου, Ε. Μάγγα, Μ. Παπαμιχαήλ, Γ. Μιχελογιαννάκη, Β. Παπαευσταθίου, Μ. Κατσαμάνη, Μ. Μουδάτσο, Σ. Καββαδία, Γ. Νικηφόρο, Μ. Λυγεράκη, Μ. Μαραζάκη.

Τέλος, για ξεχωριστούς αλλά και πολύ ιδιαίτερους λόγους θεωρώ υποχρέωσή μου να ευχαριστήσω τους φίλους μου, Λεωνίδα, Μάρκο, Σπύρο, Νίκο, Γιώργο, Νίκο, Κώστα, Θοδωρή, Κώστα, Ευαγγελία, Ρίτα, Κώστα, και φυσικά την οικογένεια μου για την αμέριστη συμπαράστασή τους.



Στους φίλους μου και τον αδερφό μου,  
που έκαναν το σπίτι να μοιάζει πιο κοντά



# Περιεχόμενα

Περίληψη	v
Abstract	vii
Κατάλογος Σχημάτων	xv
Κατάλογος Πινάκων	xvii
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Σύγχρονες και Ασύγχρονες Τεχνικές Σχεδίασης Ψηφιακών Κυκλωμάτων . . . . .	1
1.1.1 Απόδοση Κυκλωμάτων . . . . .	2
1.1.2 Κατανάλωση Ενέργειας . . . . .	3
1.1.3 Διασύνδεση Υποσυστημάτων . . . . .	4
1.2 Στόχοι της Εργασίας . . . . .	4
<b>2 Βασικές Αρχές Υλοποίησης Κυκλωμάτων Αναίσθητων στην Καθυστέρηση</b>	<b>7</b>
2.1 Κώδικες Αναίσθητοι στην Καθυστέρηση . . . . .	9
2.2 Υλοποίηση Κυκλωμάτων με DI Κώδικες . . . . .	15
2.2.1 Μονοτονία Συναρτήσεων . . . . .	16
2.2.2 Διφασική Λειτουργία . . . . .	19
2.2.3 Αναγνώριση Ολοκλήρωσης Υπολογισμού Εξόδων . . . . .	21
<b>3 Μεθοδολογία - Υλοποίηση</b>	<b>25</b>
3.1 Καθολική Λειτουργία . . . . .	25

3.2	Υλοποίηση του Επιπρόσθετου Δικτύου Boole . . . . .	32
3.2.1	Διεπίπεδη Υλοποίηση του Επιπρόσθετου Δικτύου . . . . .	33
3.2.1.1	Δυαδικές Συσχετίσεις ως Μεθοδολογία Απλο- ποίησης . . . . .	33
3.2.1.2	Διάσπαση Κυκλώματος σε Επίπεδα . . . . .	38
3.2.2	Πολυεπίπεδη Υλοποίηση του Επιπρόσθετου Δικτύου . . . . .	40
3.2.3	Ομαδοποίηση Εξόδων . . . . .	41
3.2.3.1	Μεθοδολογία . . . . .	41
3.2.3.2	Υλοποίηση . . . . .	42
3.3	Μονοτονία Δικτύων Boole . . . . .	44
<b>4</b>	<b>Αποτελέσματα - Κόστος Υλοποίησης</b>	<b>47</b>
4.1	Αναγνώριση Ολοκλήρωσης . . . . .	48
4.1.1	Αναγνώριση Ολοκλήρωσης με Βάση την Απαρίθμηση, ED . . . . .	48
4.1.2	Αναγνώριση Ολοκλήρωσης Βάση Σύγκρισης . . . . .	50
4.1.3	Μερική Αναγνώριση Ολοκλήρωσης . . . . .	51
4.2	Κυκλώματα Διεπίπεδης Προσέγγισης Υλοποίησης του Επιπρό- σθετου Δικτύου Boole . . . . .	53
4.3	Κυκλώματα με Πολυεπίπεδη Υλοποίηση του Επιπρόσθετου Δι- κτύου Boole . . . . .	65
<b>5</b>	<b>Συμπεράσματα - Μελλοντική Εργασία</b>	<b>77</b>
5.1	Μελλοντική Εργασία . . . . .	78
	<b>Βιβλιογραφία</b>	<b>82</b>

# Κατάλογος Σχημάτων

2.1	Ένα Συνδυαστικό Κύκλωμα Μεταβλητής Καθυστέρησης με Αναγνώριση Ολοκλήρωσης . . . . .	9
2.2	Αλυσίδες Εξαρτήσεων Μεταξύ Δεδομένων Τριών Ψηφίων . . . . .	11
2.3	Δημιουργία Λέξης Συστηματικού Κώδικα . . . . .	12
2.4	Ο Κώδικας Berger για Κυριολεκτικά Δεδομένα Τριών Ψηφίων . . . . .	13
2.5	Ο Συστηματικός Κώδικας Sperner για Κυριολεκτικά Δεδομένα Τριών Ψηφίων . . . . .	14
2.6	Η Παράκαμψη των Εισόδων κατά τη Φάση της Αρχικοποίησης . . . . .	20
2.7	Αναγνώριση Ολοκλήρωσης με Βάση τη Σύγκριση των Sync . . . . .	23
3.1	Σχεδιάγραμμα Υλοποίησης Κυκλώματος με Χρήση DI Κωδικών . . . . .	26
3.2	Σήματα Ελέγχου της Διφασικής Λειτουργίας . . . . .	26
3.3	Η Τελική Κατάσταση της Φάσης Αρχικοποίησης με Χρήση του Κώδικα Berger . . . . .	27
3.4	Η Τελική Κατάσταση της Φάσης Υπολογισμού με Χρήση του Κώδικα Berger . . . . .	29
3.5	Η Φάση Αρχικοποίησης με τη Μέθοδο του Συγκριτή και Χρήση του Κώδικα Sperner . . . . .	31
3.6	Η Φάση Υπολογισμού με τη Μέθοδο του Συγκριτή και Χρήση του Κώδικα Sperner . . . . .	32
3.7	Η Ροή Υλοποίησης Ενός Επιπρόσθετου Δικτύου Boole με Χρήση Δυαδικών Συσχετίσεων . . . . .	36

3.8	Η Σύνδεση των Λογικών Επιπέδων του Δικτύου 3.3 μετά το Μετασχηματισμό τους σε Υποκυκλώματα με Αναγνώριση Ολοκλήρωσης . . . . .	39
3.9	Η Ροή Υλοποίησης Ενός Επιπρόσθετου Δικτύου Boole . . . . .	40
4.1	Μονότονη Χρήση της Πύλης XNOR . . . . .	51
4.2	Σύγκριση Κυκλωμάτων Διπλής και Πολλαπλής Ράγας με Βάση τον Αριθμό των Όρων . . . . .	73
4.3	Σύγκριση Κυκλωμάτων Διπλής και Πολλαπλής Ράγας με Βάση το Μέσο Όρο των Μεταβάσεων . . . . .	74
4.4	Σύγκριση Των Όρων του Επιπρόσθετου Δικτύου Boole, με Χρήση Sprenger και Berger . . . . .	75

# Κατάλογος Πινάκων

2.1	Ο Κώδικας Sperner για $n = 1, 2, 3$ . . . . .	13
2.2	Οι Κώδικες Sperner και Berger για 1, 2 και 3 Ψηφία Κυριολεκτικών Δεδομένων . . . . .	15
2.3	Ο Κώδικας Berger για Δεδομένα 2 Ψηφίων . . . . .	21
2.4	Ο Κώδικας Sperner για Δεδομένα 2 Ψηφίων . . . . .	23
3.1	Ο Συστηματικός Κώδικας Sperner για Κυριολεκτικά Δεδομένα 3 Ψηφίων . . . . .	30
3.2	Ο Πίνακας Αληθείας του Δικτύου 3.1 . . . . .	34
3.3	Η Δυαδική Συσχέτιση που Προκύπτει από το Δίκτυο 3.1 . . . . .	35
3.4	Κάλυψη Ορθογωνίων για το Δίκτυο 3.4 . . . . .	43
4.1	Κόστος ED Κώδικα Berger . . . . .	49
4.2	Κόστος ED Συστηματικού Κώδικα Sperner . . . . .	49
4.3	Κόστος Αναγνώρισης Ολοκλήρωσης με τη Χρήση Συγκριτή . . . . .	52
4.4	Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole . . . . .	56
4.5	Παραδείγματα Υλοποίησης Κυκλωμάτων, με Χρήση DI Κωδίκων . . . . .	69



# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Σύγχρονες και Ασύγχρονες Τεχνικές Σχεδίασης Ψηφιακών Κυκλωμάτων

Οι τεχνικές σχεδίασης των ψηφιακών κυκλωμάτων μπορούν να κατηγοριοποιηθούν σε δύο κατηγορίες, τις σύγχρονες και τις ασύγχρονες [1]. Ως σύγχρονα, μπορούμε να ορίσουμε τα κυκλώματα των οποίων η συμπεριφορά κατευθύνεται και συντονίζεται από ένα ή περισσότερα περιοδικά σήματα, που διαδίδονται σε όλο το κύκλωμα και ονομάζονται ρολόγια.

Τα ασύγχρονα κυκλώματα αποτελούν μία ευρύτερη τάξη κυκλωμάτων από αυτή των σύγχρονων, στην οποία γενικά ανήκουν όσα κυκλώματα δεν χρησιμοποιούν ρολόι. Δεν υπάρχει μία και μόνο αρχιτεκτονική για τα κυκλώματα αυτά, για το λόγο αυτό και συναντούνται σε πολλές παραλλαγές. Τέτοια είναι τα Delay-Insensitive (DI), με τα οποία ασχολείται η παρούσα εργασία, τα Quasi-Delay-Insensitive (QDI), τα οποία χρησιμοποιούν υποθέσεις όσον αφορά την ισόχρονη διάδοση σημάτων σε κάποιες διασυνδέσεις τους, τα Speed-Independent (SI), που θεωρούν τις καθυστερήσεις των διασυνδέσεων μηδενικές ή αμελητέες, και τα Self-Timed (ST), που χρησιμοποιούν φραγμένες τιμές για τις καθυστερήσεις των μοντέλων τους για να εξασφαλίσουν την ορθή τους λειτουργία. Τα ασύγχρονα κυκλώματα απαρτίζονται από τον Έλεγχο (Control) και τα Δεδομένα (Datapath). Ο Έλεγχος είναι υπεύθυνος για την παρακολούθηση των αλλαγών στο κύκλωμα προκαλώντας για κάθε δράση ή γε-

γονός μία αντίδραση, κάτι που τον καθιστά υπεύθυνο για το σωστό χρονισμό του κυκλώματος. Τα Δεδομένα αποτελούν το συνδυαστικό κομμάτι του κυκλώματος, το οποίο δέχεται την πληροφορία εισόδου και γεννά την έξοδο του κυκλώματος, ενώ διαθέτει τη δυνατότητα να ειδοποιεί τον Έλεγχο για το πότε έχει ολοκληρωθεί η λειτουργία επί της εισόδου, και υπάρχει διαθέσιμη μία έγκυρη έξοδος.

Τα πρώτα κυκλώματα που κατασκευάστηκαν ήταν ασύγχρονα. Παρόλα αυτά, η εισαγωγή της έννοιας του ρολογιού έκανε πολύ προσιτή την ιδέα των σύγχρονων κυκλωμάτων, λόγω της απλούστερης δομής τους, και οδήγησε στην ευρεία διάδοσή τους. Η εμπειρία στη σχεδίαση και η εμπορική χρήση των σύγχρονων κυκλωμάτων ξεπερνάει αυτή των ασύγχρονων, όμως η ασύγχρονη σχεδίαση παραμένει μία ελκυστική τεχνική με πλήθος πλεονεκτημάτων. Μάλιστα, σημαντικό μέρος των κινήτρων για ασύγχρονη σχεδίαση πηγάζει από προβλήματα που αποδίδονται στη φύση των σύγχρονων κυκλωμάτων, και τα οποία γίνονται ολοένα πιο αισθητά με την εξέλιξη της τεχνολογίας.

### 1.1.1 Απόδοση Κυκλωμάτων

Ένα σημαντικό ζήτημα είναι οι επιπτώσεις της ύπαρξης ενός κοινού, περιοδικού σήματος ρολογιού. Η περίοδος του ρολογιού που ανατίθεται στα κυκλώματα είναι αποτέλεσμα χρονικής ανάλυσης που λαμβάνει υπόψη τις χειρότερες δυνατές συνθήκες λειτουργίας ενός κυκλώματος δηλαδή ως προς την κατασκευαστική διεργασία, την τάση και τη θερμοκρασία (P, V, T). Όμως οι συνθήκες στις οποίες λειτουργούν τα κυκλώματα δεν είναι πάντα οι χειρότερες δυνατές αφήνοντας ένα περιθώριο λειτουργίας με μεγαλύτερη απόδοση το οποίο δεν μπορούν να εκμεταλλευτούν τα σύγχρονα αλλά μπορούν τα ασύγχρονα κυκλώματα.

Εκτός των συνθηκών λειτουργίας, το ρολόι των σύγχρονων κυκλωμάτων έχει τέτοια περίοδο ώστε να μπορεί να συμπεριλάβει την καθυστέρηση για την χειρότερη δυνατή αλληλουχία δεδομένων. Για παράδειγμα αν θεωρήσουμε ένα Ripple-Carry αθροιστή, τότε η χειρότερη καθυστέρησή του είναι ανάλογη των αριθμών των ψηφίων που χειρίζεται, και με βάση αυτήν καθορίζεται το ρολόι. Όμως η καθυστέρηση υπολογισμού του αθροίσματος διαφέρει κάθε φορά ανά-

λογα με τα δεδομένα εισόδου. Για τα σύγχρονα κυκλώματα κάθε υπολογισμός θα γίνεται στον ίδιο, χειρότερο δυνατό, χρόνο ενώ για τα ασύγχρονα υπάρχει η δυνατότητα η συνδυαστική τους λογική (Δεδομένα) να προσαρμόζεται ανάλογα με την είσοδο, μειώνοντας έτσι το χρόνο υπολογισμού σε κάποιες περιπτώσεις.

Η περίοδος του σήματος ρολογιού των σύγχρονων κυκλωμάτων πρέπει επίσης να λαμβάνει υπόψη τις καθυστερήσεις διάδοσης του ρολογιού σε όλα τα σημεία του κυκλώματος (απόκλιση). Με τις σημερινές κατασκευαστικές διεργασίες να φτάνουν τα 45nm, άλλα ακόμα και σε διεργασίες των 65nm, οι καθυστερήσεις αυτές είναι συγκρίσιμες με την περίοδο του ρολογιού, καθότι η κλιμάκωση των τρανζίστορ δεν συνοδεύεται από αντίστοιχη κλιμάκωση των διασυνδέσεων. Αυτό έχει ως αποτέλεσμα την αύξηση της περιόδου ώστε να καλυφθούν και αυτές οι καθυστερήσεις.

### 1.1.2 Κατανάλωση Ενέργειας

Ένας ακόμα στόχος της ασύγχρονης σχεδίασης κυκλωμάτων είναι η κατανάλωση ενέργειας. Το δίκτυο του ρολογιού σε ένα σύγχρονο κύκλωμα καταναλώνει ένα σημαντικό μέρος της συνολικής ενέργειας που καταναλώνει το κύκλωμα. Ακόμα και σε περιόδους ηρεμίας όπου το κύκλωμα δεν εκτελεί καμία λειτουργία και δεν αλλάζει κατάσταση, το σήμα του ρολογιού συνεχίζει να ανεβοκατεβαίνει με σταθερή συχνότητα καταναλώνοντας ενέργεια. Έχουν αναπτυχθεί τεχνικές για την αποφυγή αυτού του φαινομένου οι οποίες φράζουν το ρολόι στα τμήματα του κυκλώματος που δε θα αλλάξουν κατάσταση (clock gating). Ακόμα και έτσι όμως, υπάρχει κατανάλωση σε περιόδους ηρεμίας δεδομένου ότι πρέπει να υπάρχει ένας μηχανισμός υπεύθυνος για τα τμήματα του κυκλώματος που είναι ανενεργά, και ο οποίος παρακολουθεί τις αλλαγές που συμβαίνουν στο κύκλωμα. Ο μηχανισμός αυτός απαιτεί προσθήκη υλικού, καθώς και επιπλέον χρονική ανάλυση για την εισαγωγή των προσθηκών στο χρονικό μοντέλο. Τα ασύγχρονα κυκλώματα, ακριβώς λόγω της απουσίας ρολογιού, εκμεταλλεύονται καλύτερα την ύπαρξη περιόδων ηρεμίας αφού από τη φύση τους δεν καταναλώνουν ενέργεια όταν δεν λειτουργούν.

### 1.1.3 Διασύνδεση Υποσυστημάτων

Ένα επιπλέον πλεονέκτημα της ασύγχρονης σχεδίασης, είναι η ευκολία στη διασύνδεση μεταξύ ασύγχρονων υποσυστημάτων σε ένα ευρύτερο ασύγχρονο σύστημα. Κάθε υποσύστημα είναι σε θέση να γνωρίζει το πότε ολοκληρώνει τη λειτουργία που του ανατίθεται και μπορεί να επικοινωνήσει με τα υπόλοιπα. Αντίθετα, λόγω του ότι ο χρονισμός των σύγχρονων κυκλωμάτων βασίζεται στην ύπαρξη του ρολογιού, υπάρχει η ανάγκη για συγχρονισμό μεταξύ κυκλωμάτων που χρησιμοποιούν ρολόι με διαφορετική ή ακόμα και ίδια περίοδο. Ο συγχρονισμός αυτός δεν είναι μία τετριμμένη διαδικασία και απαιτεί μία σειρά από χρονικούς περιορισμούς και υποθέσεις.

## 1.2 Στόχοι της Εργασίας

Στο παρόν σύγγραμμα θα εστιάσουμε στα συνδυαστικά κυκλώματα με αναγνώριση ολοκλήρωσης που ανήκουν στη τάξη των ασύγχρονων κυκλωμάτων. Η τρέχουσα αποδοτικότερη τεχνική υλοποίησης τέτοιων κυκλωμάτων είναι ο μετασχηματισμός της διπλής-ράγας, [6], ο οποίος βασίζεται στον DI κώδικα dual-rail. Με την τεχνική αυτή τα κυκλώματα που προκύπτουν έχουν εμβαδό περίπου διπλάσιο σε σχέση με τα συμβατικά ψηφιακά κυκλώματα. Αυτό έχει ως αποτέλεσμα η χρήση τους να είναι ασύμφορη για μεγάλα κυκλώματα. Στόχος της εργασίας που παρουσιάζεται στο σύγγραμμα, είναι να αναπτυχθεί μια νέα τεχνική μετασχηματισμού που θα καταπολεμήσει τη μεγάλη επιβάρυνση σε εμβαδό, που εμφανίζεται στα κυκλώματα διπλής-ράγας.

Με χρήση δύο DI κωδίκων, του κώδικα Berger και ενός υποσυνόλου του κώδικα Sperner, και ξεκινώντας από την περιγραφή ενός συμβατικού ψηφιακού συνδυαστικού κυκλώματος, είναι δυνατόν να προκύψουν κυκλώματα πολλαπλής-ράγας, με πλάτος ανάλογο με τα ψηφία εξόδου αυτών. Η μεθοδολογία που προτείνεται δεν περιορίζεται στους δύο αυτούς κώδικες, αλλά δίνει τις κατευθύνσεις για την υλοποίηση κυκλωμάτων με χρήση οποιουδήποτε συστηματικού DI κώδικα. Παράλληλα, με τεχνικές που πηγάζουν από την έννοια των Δυαδικών Συσχετίσεων (Boolean Relations, [15]) επιχειρείται η όσο το δυνατόν καλύτερη εκμετάλλευση

τις ελευθερίας για βελτιστοποίηση (don't cares) των κυκλωμάτων που προκύπτουν από το μετασχηματισμό, με στόχο να μειωθεί το εμβαδό τους. Με την εφαρμογή του νέου μετασχηματισμού σε δειγματοληπτικά κυκλώματα δίνεται μια ρεαλιστική εικόνα για τις επιδόσεις του σε σχέση με τη διπλή-ράγα τόσο όσον αφορά το εμβαδό όσο και την κατανάλωση ισχύος.

Επίσης, αναλύονται οι λόγοι για τους οποίους η υλοποίηση κυκλωμάτων πολλαπλής-ράγας με χρήση DI κωδίκων, παρά τη μεγάλη προοπτική για βελτίωση των κυκλωμάτων, είναι υπολογιστικά ακριβή, και προτείνονται λύσεις για την μείωση του υπολογιστικού κόστους. Προτείνονται, και υλοποιήθηκαν, η διάσπαση των κυκλωμάτων σε επίπεδα και η ομαδοποίηση των εξόδων, η οποία κάνει χρήση ενός αλγορίθμου κάλυψης ορθογωνίων. Ειδικά για την τελευταία τεχνική, τα αποτελέσματα δείχνουν ότι μπορούν να προκύψουν πολύ καλύτερα κυκλώματα με τη χρήση τής.

Όλα τα απαραίτητα εργαλεία για το μετασχηματισμό υλοποιήθηκαν σε μία ροή με μεγάλο βαθμό αυτοματοποίησης, και ενσωματώθηκαν σε ένα από τα πιο γνωστά ακαδημαϊκά προγράμματα λογικής σύνθεσης, το SIS, [13]. Σε αυτό το περιβάλλον έγινε και η πειραματική διαδικασία με τους ίδιους αλγορίθμους βελτιστοποίησης να εφαρμόζονται τόσο στα κυκλώματα διπλής όσο και στα πολλαπλής ράγας, για μεγαλύτερη αξιοπιστία των μετρήσεων.

## Οργάνωση της Εργασίας

Στο δεύτερο κεφάλαιο γίνεται μία εισαγωγή στους DI κώδικες και παραθέτονται οι ιδιότητες που απαιτούνται για να διασφαλίσουν την ορθή λειτουργία ενός συνδυαστικού κυκλώματος με αναγνώριση ολοκλήρωσης, καθώς και οι μέθοδοι για τον μετασχηματισμό των κυκλωμάτων, ώστε να αποκτήσουν αυτές τις ιδιότητες. Στο τρίτο κεφάλαιο δίνονται οι λεπτομέρειες της υλοποίησης ασύγχρονων κυκλωμάτων με χρήση των DI κωδίκων Berger και Sperner, ενώ παρατίθεται αναλυτικά η λειτουργία ενός τέτοιου κυκλώματος. Το τέταρτο κεφάλαιο περιέχει τα αποτελέσματα της εφαρμογής της τεχνικής που παρουσιάζεται σε εύρος δειγματοληπτικών κυκλωμάτων, και γίνεται σύγκρισή της ως προς το εμβαδό και

την ισχύ που καταναλώνουν με τα αντίστοιχα κυκλώματα διπλής-ράγας. Τέλος, στο πέμπτο κεφάλαιο περιέχονται τα συμπεράσματα που αφορούν στη χρήση της τεχνικής, καθώς και προτάσεις για μελλοντική εργασία με σκοπό τη βελτίωση της τεχνικής.

## Κεφάλαιο 2

# Βασικές Αρχές Υλοποίησης Κυκλωμάτων Αναίσθητων στην Καθυστερήση

Στο κεφάλαιο ακολουθεί γίνεται μία εισαγωγή της έννοιας των κωδίκων που είναι αναίσθητοι στην καθυστέρηση, των πλεονεκτημάτων τους, καθώς και των συνθηκών που διασφαλίζουν την ασφαλή τους χρήση. Έπειτα προτείνεται ένας νέος τρόπος ενσωμάτωσης των κωδίκων σε κυκλώματα τα οποία κληρονομούν τις επιθυμητές ιδιότητες αυτών, τα DI κυκλώματα.

**Ορισμός 2.1.** Ως *Αναίσθητο Καθυστερήσεων (Delay Insensitive, DI)* κύκλωμα ορίζουμε αυτό του οποίου η ορθότητα λειτουργίας είναι ανεξάρτητη από τις καθυστερήσεις των καλωδίων και των πυλών που το απαρτίζουν. Το πέρας των υπολογισμών που επιτελεί μπορεί να καθοριστεί πλήρως, και η κυριολεκτική έξοδος του κυκλώματος να ανακτηθεί, χωρίς να υπόκειται σε οποιουδήποτε χρονικούς περιορισμούς.

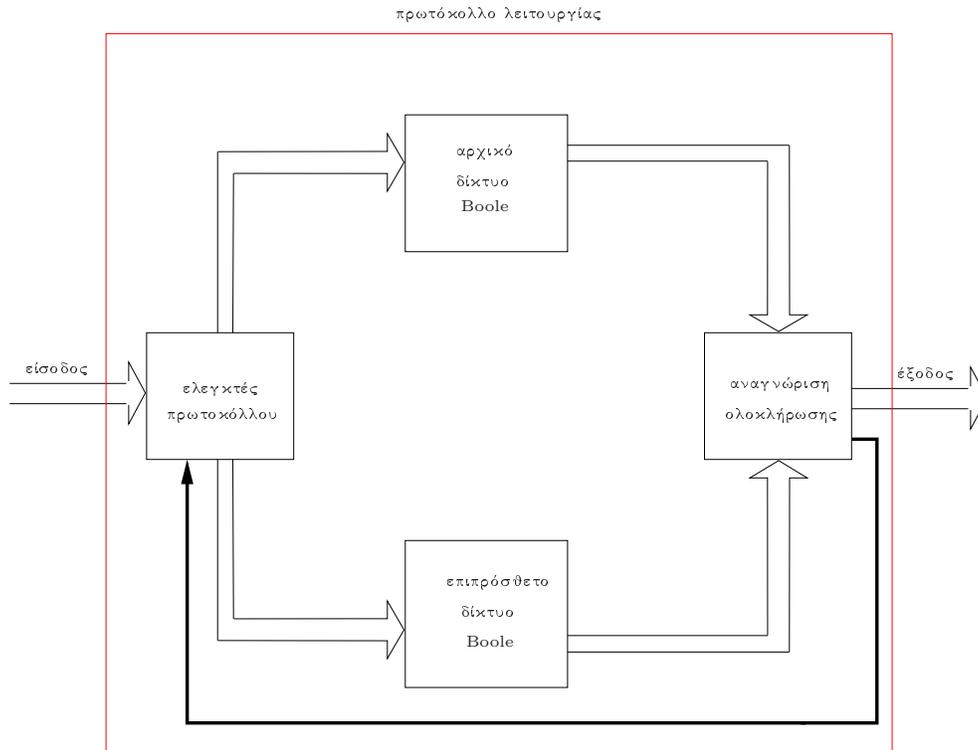
**Ορισμός 2.2.** Ορίζουμε ως κυριολεκτικά δεδομένα τα αυτούσια δυαδικά δεδομένα που διαρρέουν, ένα κύκλωμα, ένα διάπλο, και ούτω καθεξής, προτού αυτό υποστεί οποιοδήποτε είδους μετασχηματισμό που θα του επιβάλλει DI συμπεριφορά. Επίσης κυριολεκτικές εξόδους ονομάζουμε τις κυριολεκτικές τιμές των εξόδων πριν το μετασχηματισμό

Στην παρούσα εργασία, δίνεται έμφαση στο χειρισμό συνδυαστικής λογικής, με σκοπό αυτή να μετασχηματιστεί σε συνδυαστικά κυκλώματα με αναγνώριση ολοκλήρωσης, τα οποία έχουν DI συμπεριφορά. Γενικά, ένα DI κύκλωμα σχετίζεται με τα παρακάτω μέρη και ιδιότητες:

1. Ένα αρχικό δίκτυο Boole. Ως δίκτυο Boole ορίζουμε ένα σύνολο από κόμβους οι οποίοι με τη σειρά τους αντιστοιχούν σε μία εξίσωση Boole, πολλαπλών εισόδων και μίας εξόδου. Ένας κόμβος είναι δυνατόν να συνδέεται με ένα δεύτερο, όταν η έξοδός του αποτελεί είσοδο αυτού ή έχει ως είσοδό του την έξοδο του δεύτερου. Οι συσχετίσεις μεταξύ των κόμβων σχηματίζουν ένα γράφο, στον οποίο για ένα συνδυαστικό κύκλωμα επιτρέπονται όλες οι συνδέσεις μεταξύ κόμβων που δεν προκαλούν κύκλο στο γράφο. Οι εισοδοί που προέρχονται πέρα από το σύνολο των κόμβων ονομάζονται πρωτεύουσες εισοδοί και οι έξοδοί που εκτείνονται πέρα από τα όρια του συνόλου ονομάζονται πρωτεύουσες έξοδοί
2. Κάποιος μετασχηματισμός στο αρχικό δίκτυο Boole, είτε σε επίπεδο λογικής, είτε σε επίπεδο λειτουργίας-συμπεριφοράς
3. Ένα νέο δίκτυο Boole, το οποίο σχετίζεται με το αρχικό δίκτυο Boole, βάση του μετασχηματισμού, το επιπρόσθετο δίκτυο Boole
4. Ένα μηχανισμό αναγνώρισης ολοκλήρωσης των κυριολεκτικών εξόδων του μετασχηματισμένου δικτύου, ο οποίος βασίζεται σε κάποιον αριθμητικό κώδικα αναίσθητο στην καθυστέρηση, και συνήθως χρησιμοποιεί πληροφορίες από το αρχικό και το επιπρόσθετο δίκτυο Boole
5. Ένα πρωτόκολλο μονότονης λειτουργίας, συνήθως δύο φάσεων που είναι η φάση αρχικοποίησης και η φάση υπολογισμού
6. Στοιχεία ελεγκτών που εκτελούν το πρωτόκολλο λειτουργίας

Γραφικά τα παραπάνω φαίνονται στο Σχήμα 2.1.

Για παράδειγμα ο μετασχηματισμός διπλής-ράγας, [6], διατηρεί το αρχικό δίκτυο Boole ως έχει, αλλά του επιβάλλει μονότονη λειτουργία μετασχηματίζοντας



Σχήμα 2.1: Ένα Συνδυαστικό Κύκλωμα Μεταβλητής Καθυστερήσης με Αναγνώριση Ολοκλήρωσης

τις εισόδους του και κάθε κόμβο του σε δύο, έναν ενεργό με θετικό πρόσημο και έναν ενεργό με αρνητικό πρόσημο. Το επιπρόσθετο δίκτυο Boole είναι το δυικό του αρχικού και το πρωτόκολλο μονότονης λειτουργίας είναι η διφασική λειτουργία.

## 2.1 Κώδικες Αναίσθητοι στην Καθυστερήση

Ένα DI κύκλωμα χρησιμοποιεί αριθμητικούς DI κώδικες, που είναι αναίσθητοι στις καθυστερήσεις των σημάτων που χρησιμοποιούν. Αυτό σημαίνει ότι η αναγνώριση μιας έγκυρης λέξης αυτών των κωδίκων είναι πάντα μία σαφής πράξη. Το θεωρητικό υπόβαθρο για τους κώδικες αυτούς παρουσιάζεται λεπτομερώς στο [2]. Στο εν λόγω κεφάλαιο θα παρουσιάσουμε τις θεμελιώδεις έννοιες. Οι DI κώδικες εφαρμόζονται κυρίως σε τηλεπικοινωνιακούς διαύλους, όπου προσφέρουν έγκυρη ανίχνευση της ολοκλήρωσης της μετάδοσης ενός μηνύματος

αλλά και για τον εντοπισμό λαθών τα οποία οφείλονται σε λανθασμένη μετάδοση ψηφίων και οδηγούν στη λήψη μηνυμάτων που δεν είναι έγκυρες κωδικές λέξεις.

### Λειτουργία DI Κώδικα

Ως παράδειγμα θα χρησιμοποιήσουμε ένα διάυλο με τρία ψηφία για τι δεδομένα. Επιπλέον, θεωρούμε ότι στο διάυλο η ηρεμία αντιστοιχεί με την κωδικοποίηση 0 για όλα τα ψηφία, και η μετάδοση με την τιμή 1 για κάθε ένα από τα ψηφία. Έτσι, ο διάυλος σε ηρεμία θα αντιστοιχεί στην τιμή 000. Όμως, ένα μήνυμα με την τιμή 000 είναι διαφορετικό με την τιμή ηρεμίας του διαύλου. Εκτός αυτού, εάν ληφθεί ένα μήνυμα 001 δεν είναι σαφές εάν αυτό είναι το πραγματικό προς μετάδοση μήνυμα ή αποτελεί μέρος του μηνύματος 101, ή του 011, ή και του 111, όπου λόγω των μεταβλητών καθυστερήσεων των ψηφίων, είτε το συνολικό μήνυμα δεν έχει πάρει την τελική του κατάσταση, είτε λόγω ενός σφάλματος μετάδοσης δεν παραλήφθηκε σωστά. Για έναν ενσύρματο διάυλο, οι μεταβλητές καθυστερήσεις των ψηφίων μπορεί να οφείλονται στη μεταβλητότητα των χαρακτηριστικών του υλικού των διασυνδέσεων, με αποτέλεσμα τη διαφορετική ταχύτητα διάδοσης των σημάτων, ειδικά σε περιπτώσεις διαύλων μεγάλου μήκους, ενώ για έναν ασύρματο διάυλο σε διαφορετικές επιδράσεις θορύβου ή άλλων ηλεκτρομαγνητικών φαινομένων που επιδρούν στη μετάδοση των ψηφίων. Διαφορούμενες εναλλαγές, όπως αυτές που αναφέρθηκαν παραπάνω, παρουσιάζονται γραφικά στο Σχήμα 2.2. Κάθε μονοπάτι του Σχήματος 2.2 από το μήνυμα 000 στο 111, π.χ. το μονοπάτι με τα κόκκινα βέλη, αποτελεί μία αλυσίδα εξάρτησης κατά μήκος της οποίας δε μπορούμε να είμαστε βέβαιοι για την τελική κατάσταση λήψης του μεταδιδόμενου μηνύματος.

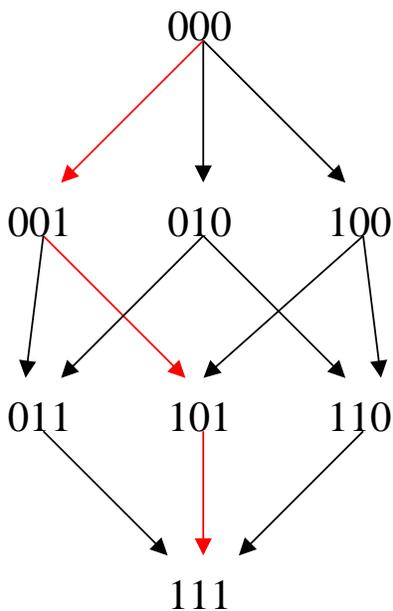
Ο στόχος των DI κωδίκων είναι να απαλείψουν τις λεγόμενες αλυσίδες εξάρτησης του Σχήματος 2.2, μέσω κωδικοποίησης, καθιστώντας απόλυτα σαφές το κάθε κωδικοποιημένο μήνυμα στις εξόδους και κάνοντας δυνατή την αναγνώριση ολοκλήρωσης (ή την ανίχνευση λαθών όταν αυτή είναι η επιθυμητή χρήση). Υπάρχουν δύο κατηγορίες DI κωδίκων, οι μη-συστηματικοί και οι συστηματικοί.

Οι κώδικες της πρώτης κατηγορίας, στην οποία δε θα εστιάσουμε στο παρόν σύγγραμμα, κωδικοποιούν τα κυριολεκτικά δεδομένα παράγοντας λέξεις που σχη-

ματίζουν έναν DI κώδικα. Τα κυριολεκτικά δεδομένα μπορούν να εξαχθούν από τον αντίστροφο μετασχηματισμό. Συνεπώς, οι κώδικες αυτοί εκτός από έναν κωδικοποιητή, έχουν την ανάγκη και ενός αποκωδικοποιητή για την ανάκτηση των κυριολεκτικών εξόδων καθιστώντας τη χρήση τους ακριβή.

Οι συστηματικοί κώδικες διατηρούν τα δεδομένα αυτούσια και προσθέτουν ένα επίθεμα που βοηθά στην απαλοιφή των αλυσίδων εξάρτησης και ονομάζεται Sync. Με τον τρόπο αυτό, το μόνο που απαιτείται για τη δημιουργία των κωδικών λέξεων είναι η παραγωγή του επιθέματος Sync. Για την ανάκτηση των κυριολεκτικών εξόδων χρειάζεται μόνο η ένδειξη του μηχανισμού της αναγνώρισης ολοκλήρωσης, και όχι κάποιος αποκωδικοποιητής, μιας και τα κυριολεκτικά δεδομένα δε μετασχηματίζονται. Η παραγωγή μιας κωδικής λέξης ενός συστηματικού κώδικα από τα κυριολεκτικά δυαδικά δεδομένα φαίνεται γραφικά στο Σχήμα 2.3. Η συνάρτηση  $f$  προκύπτει από τα κυριολεκτικά δεδομένα και μπορεί να έχει οποιαδήποτε αντιστοίχιση αρκεί να απαλείφει τις αλυσίδες εξάρτησης των κυριολεκτικών δεδομένων.

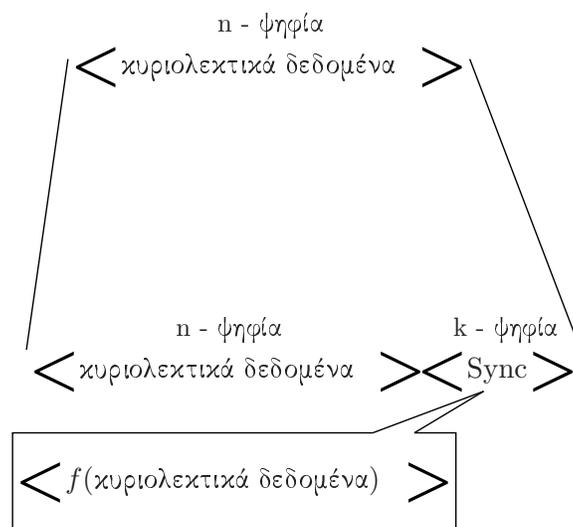
Ένας βέλτιστος συστηματικός κώδικας, με την έννοια ότι κανένας άλλος κώδικας με τα ίδια ψηφία δεν περιέχει μεγαλύτερο αριθμό λέξεων, είναι ο κώδικας



Σχήμα 2.2: Αλυσίδες Εξαρτήσεων Μεταξύ Δεδομένων Τριών Ψηφίων

Berger ο οποίος, ως επίθεμα Sync ορίζει τον αριθμό των μηδενικών, σε δυαδική αναπαράσταση, του διανύσματος των κυριολεκτικών δεδομένων. Επομένως, με βάση το Σχήμα 2.3, στον κώδικα Berger η συνάρτηση  $f$  δίνει ως αποτέλεσμα τον αριθμό των ψηφίων με τιμή 0 των κυριολεκτικών δεδομένων σε  $k$  ψηφία, όπου  $k = \lceil \log_2(n + 1) \rceil$  και  $n$  τα ψηφία των κυριολεκτικών δεδομένων. Ο κώδικας Berger για τρία ψηφία, καθώς και ο τρόπος που απαλείφονται οι αλυσίδες εξάρτησης φαίνεται στο Σχήμα 2.4. Όπως είναι φανερό, καμία από τις κωδικές λέξεις που σχηματίζονται δεν είναι υποσύνολο μιας άλλης, έτσι ώστε να υπάρχει κίνδυνος ασάφειας, ενώ η συνθήκη για τη λήψη ενός μηνύματος είναι τα κόκκινα ψηφία, του επιθέματος, να αναπαριστούν ορθώς τον αριθμό των μηδενικών του μέρους του διανύσματος που περιέχει αυτούσια τα κυριολεκτικά δεδομένα.

Όπως γίνεται κατανοητό μπορούν να υπάρχουν αρκετές υλοποιήσεις ενός συστηματικού DI κώδικα. Ως μέτρο απόδοσης ενός DI κώδικα, [2], θεωρείται το μέτρο των επιπρόσθετων ψηφίων που εισάγει, δηλαδή τα  $k$  ψηφία του Sync, αλλά και η εν γένει πολυπλοκότητα της υλοποίησης της συνάρτησης Sync. Ο κώδικας Sperner, [2], είναι άλλος ένας DI κώδικας που μπορεί να χρησιμοποιηθεί, όπως θα δείξουμε παρακάτω για την υλοποίηση DI κυκλωμάτων. Ο κώδικας Sperner ανήκει στην κατηγορία των  $n$  - από -  $2n$  κωδίκων, που σημαίνει ότι οι κωδικές του λέξεις έχουν μήκος  $2n$  ψηφίων και αποτελούνται από  $n$  ψηφία με την τιμή 0

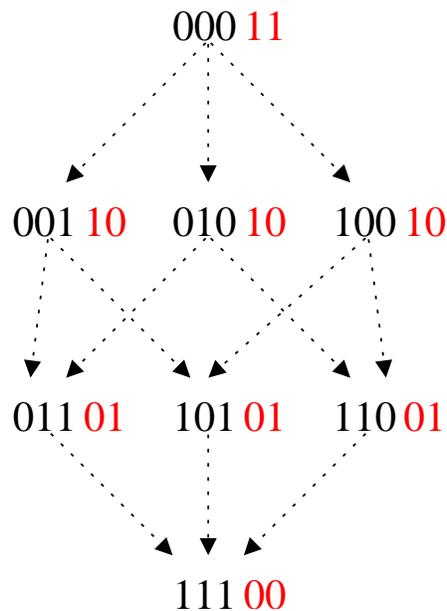


Σχήμα 2.3: Δημιουργία Λέξης Συστηματικού Κώδικα

$n$	κωδικές λέξεις
1	01, 10
2	0011, 0101, 0110, 1001, 1010, 1100
3	000111, 001011, 001101, 001110, 010011, 010101, 010110, 011001, 011010, 011100, 100011, 100101, 100110, 101001, 101010, 101100, 110001, 110010, 110100, 111000

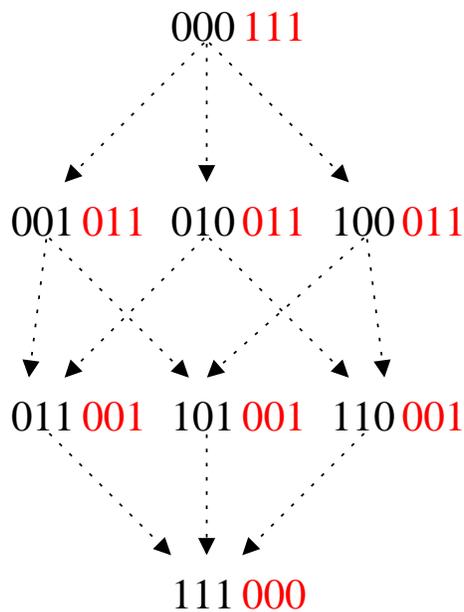
Πίνακας 2.1: Ο Κώδικας Sperner για  $n = 1, 2, 3$

και  $n$  ψηφία με την τιμή 1. Επίσης είναι βέλτιστος καθώς δεν υπάρχει άλλος DI κώδικας με τα ίδια ψηφία που να κωδικοποιεί με σαφήνεια περισσότερες λέξεις. Παρόλα αυτά, δεν είναι συστηματικός κώδικας όπως φαίνεται στον Πίνακα 2.1 που παραθέτει τους κώδικες Sperner για τιμή του  $n$  1, 2 και 3. Εάν επιλέξουμε ως αριθμό των ψηφίων των κυριολεκτικών δεδομένων την τιμή του  $n$ , τότε για  $n = 1$  βλέπουμε ότι έχουμε δύο κωδικές λέξεις όσες και οι διαφορετικές τιμές που μπορούμε να αναπαραστήσουμε με ένα ψηφίο. Από εκεί και πέρα όμως, για  $n > 2$ , οι κωδικές λέξεις είναι περισσότερες από τις τιμές που μπορούν να ανα-



Σχήμα 2.4: Ο Κώδικας Berger για Κυριολεκτικά Δεδομένα Τριών Ψηφίων

παρασταθούν με  $n$  ψηφία κυριολεκτικών δεδομένων, κάτι που είναι αντίθετο με την έννοια των συστηματικών κωδικών που απαιτούν 1 - προς - 1 απεικόνιση των κυριολεκτικών δεδομένων και των κωδικών λέξεων. Στη παρούσα εργασία, θα χρησιμοποιήσουμε συγκεκριμένα υποσύνολα του κώδικα Sperner που έχουν DI συμπεριφορά και ταυτόχρονα οι λέξεις τους μπορούν να σχηματιστούν με συστηματικό τρόπο. Τα αντίστοιχα υποσύνολα για 1, 2 και 3 ψηφία κυριολεκτικών δεδομένων φαίνονται στον Πίνακα 2.2, με την αντιστοίχιση των κυριολεκτικών δεδομένων και του απαραίτητου επιθέματος (Sync), ενώ στο Σχήμα 2.5 απεικονίζεται ο τρόπος με τον οποίο ο κώδικας Sperner απαλείφει τις αλυσίδες εξάρτησης για 3 ψηφία κυριολεκτικών δεδομένων. Σε αντιστοιχία με το Σχήμα 2.3, η συ-



Σχήμα 2.5: Ο Συστηματικός Κώδικας Sperner για Κυριολεκτικά Δεδομένα Τριών Ψηφίων

νάρτηση  $f$  για το συστηματικό κώδικα Sperner δίνει ως έξοδο ένα διάνυσμα, με  $k = n$  ψηφία, στο οποίο όσα 0 έχει το διάνυσμα των κυριολεκτικών δεδομένων, τόσα 1 έχει στη δεξιά του άκρη. Αυτή είναι και η τιμή του Sync για το συστηματικό κώδικα Sperner.

κυριολεκτικά δεδομένα	Sync Berger	Sync Sperner
1 ψηφίο κυριολεκτικών δεδομένων		
0	1	1
1	0	0
2 ψηφία κυριολεκτικών δεδομένων		
00	11	10
01	01	01
10	01	01
11	00	00
3 ψηφία κυριολεκτικών δεδομένων		
000	111	11
001	011	10
010	011	10
011	001	01
100	011	10
101	001	01
110	001	01
111	000	00

Πίνακας 2.2: Οι Κώδικες Sperner και Berger για 1, 2 και 3 Ψηφία Κυριολεκτικών Δεδομένων

## 2.2 Υλοποίηση Κυκλωμάτων με DI Κώδικες

Με βάση την παραπάνω ανάλυση, ένα συνδυαστικό κύκλωμα με αναγνώριση ολοκλήρωσης, όπως αυτό του Σχήματος 2.1, θα μπορούσε να κάνει χρήση του κώδικα Berger ή του συστηματικού κώδικα Sperner απλώς βάζοντας στη θέση του επιπρόσθετου δικτύου Boole τη λογική συνάρτηση Sync.

Μια τέτοια υλοποίηση αποτελεί ουσιαστικά γενίκευση του μετασχηματισμού της διπλής-ράγας. Ο μετασχηματισμός αυτός βασίζεται στο μικρότερο συστηματικό κώδικα, [2], τον Dual-Rail. Ο κώδικας αυτός συμπίπτει με αυτούς των Berger και Sperner για 1 ψηφίο κυριολεκτικών δεδομένων. Στη διπλή-ράγα, κά-

θε κόμβος ενός δικτύου Boole που υπόκειται το μετασχηματισμό αντιμετωπίζεται σαν ξεχωριστό κύκλωμα. Μπορούμε δηλαδή να θεωρούμε ότι ο κόκκος επεξεργασίας του αρχικού δικτύου Boole είναι 1 ψηφίο. Στο μετασχηματισμό που θα παρουσιάσουμε με τη χρήση συστηματικών κωδίκων, ο κόκκος επεξεργασίας είναι μεταβλητός και ανάλογος με τον αριθμό των ψηφίων της εξόδου του αρχικού δικτύου, οδηγώντας σε κυκλώματα πολλαπλής-ράγας. Η συμπεριφορά της πολλαπλής-ράγας περιορίζεται έξω από τα όρια του κυκλώματος, στον τρόπο με τον οποίο γίνεται αντιληπτό από το περιβάλλον στο οποίο βρίσκεται. Εσωτερικά το κύκλωμα συνεχίζει να χρησιμοποιεί τη συμβατική ψηφιακή κωδικοποίηση, όπου κάθε ψηφίο είναι ανεξάρτητο των υπολοίπων.

Η επιβάρυνση του μετασχηματισμού της διπλής-ράγας είναι η προσάρτηση στο αρχικό δίκτυο Boole του δυϊκού του. Το γεγονός ότι στην πλειοψηφία των περιπτώσεων η δυϊκή λογική ενός δικτύου έχει παραπλήσιο μέγεθος και πολυπλοκότητα οδηγεί τα κυκλώματα διπλής-ράγας σε επιβάρυνση σε δυαδικούς όρους και εμβαδό που βρίσκεται κοντά στο 100% του αρχικού κυκλώματος. Στα κυκλώματα πολλαπλής-ράγας, η προσδοκία, την οποία εξερευνά η παρούσα εργασία, είναι ότι το σύννεφο λογικής που υλοποιεί τη συνάρτηση Sync, θα έχει μικρότερο εμβαδό από το αρχικό δίκτυο Boole οδηγώντας σε συνολικά μικρότερη επιβάρυνση.

Για την ορθή λειτουργία ενός DI κυκλώματος υπάρχουν προϋποθέσεις οι οποίες συμπληρώνουν και εξασφαλίζουν την ορθή χρήση των DI κωδίκων. Οι προϋποθέσεις αυτές αναφέρονται παρακάτω και είναι κατά σειρά:

- Η μονοτονία των συναρτήσεων
- Η διφασική λειτουργία
- Η ύπαρξη συνάρτησης αναγνώρισης ολοκλήρωσης

### 2.2.1 Μονοτονία Συναρτήσεων

Στα συμπεράσματα του [2] αναφέρεται ότι οι συνθήκες που εξασφαλίζονται από τη χρήση DI κωδίκων συμπίπτουν με αυτές που απαιτούνται για την ανίχνευση λαθών μιας κατεύθυνσης (unidirectional). Έτσι, τα μηνύματα που μεταδίδονται

οδηγούνται στην τελική τους κατάσταση με μεταβάσεις των ψηφίων προς μία μόνο κατεύθυνση, δηλαδή είτε από 0 σε 1, είτε από 1 σε 0, με τις μεταβάσεις  $0 \rightarrow 1 \rightarrow 0$  και  $1 \rightarrow 0 \rightarrow 1$  να απαγορεύονται. Στην άλγεβρα Boole αυτό σημαίνει ότι οι εξισώσεις των δικτύων που αποτελούν το κύκλωμα πρέπει να είναι μονότονες το οποίο με τη σειρά του σημαίνει ότι πρέπει να είναι είτε μονότονα αύξουσες είτε μονότονα φθίνουσες.

**Ορισμός 2.3.** Μία εξίσωση είναι μονότονα αύξουσα ως προς μία μεταβλητή  $x$  όταν με αύξηση αυτής της μεταβλητής, δηλαδή μετάβαση της τιμής της από 0 σε 1, δύναται να υπάρξουν μόνο αύξουσες μεταβάσεις, δηλαδή από 0 σε 1, και στην τιμή της εξόδου της. Επιπροσθέτως για μία μετάβαση από 1 σε 0 της μεταβλητής  $x$  η εξίσωση δύναται να έχει μόνο αντίστοιχη μετάβαση στην έξοδό της.

**Ορισμός 2.4.** Μία εξίσωση είναι μονότονα φθίνουσα ως προς μία μεταβλητή  $x$  όταν με αύξηση αυτής της μεταβλητής, δηλαδή μετάβαση της τιμής της από 0 σε 1, δύναται να υπάρξουν μόνο φθίνουσες μεταβάσεις, δηλαδή από 1 σε 0, στην τιμή της εξόδου της. Επιπροσθέτως για μία μετάβαση από 1 σε 0 της μεταβλητής  $x$  η εξίσωση δύναται να έχει μόνο μετάβαση από 0 σε 1 στην έξοδό της.

**Πόρισμα 2.1.** Μία εξίσωση είναι μονότονα αύξουσα, όταν είναι μονότονα αύξουσα ως προς όλες τις μεταβλητές που την απαρτίζουν. Αντιστοίχως είναι μονότονα φθίνουσα, όταν είναι μονότονα φθίνουσα ως προς όλες τις μεταβλητές που την απαρτίζουν.

Φυσικά, η μονότονη συμπεριφορά δεν είναι ιδιότητα οποιασδήποτε εξίσωσης και δικτύου, όμως υπάρχει η δυνατότητα να μετασχηματιστούν ώστε να είναι μονότονα.

### Μετασχηματισμός για Μονοτονία

Θα δούμε πρώτα πως εξασφαλίζεται η μονοτονία μιας λογικής εξίσωσης.

**Ορισμός 2.5.** Μία εξίσωση είναι μονότονα αύξουσα ως προς μία μεταβλητή εάν σε αυτήν εμφανίζεται μόνο με την κανονική της μορφή και όχι με τη

συμπληρωματική (Θετικά Μονόσημη, [15]). Αντιστοίχως μία εξίσωση είναι μονότονα φθίνουσα ως προς μία μεταβλητή εάν σε αυτήν εμφανίζεται μόνο με τη συμπληρωματική της μορφή και όχι με την κανονική (Αρνητικά Μονόσημη).

Συνεπώς, για να μπορούμε να μετασχηματίζουμε οποιαδήποτε εξίσωση σε μονότονη θα πρέπει να μπορούμε να τη φέρουμε σε μία μορφή που χρησιμοποιεί μόνο μία συγκεκριμένη μορφή μεταβλητών, την κανονική, ή την συμπληρωματική. Θεωρούμε την επόμενη εξίσωση:

$$f = a' \cdot b + c' \cdot d' + e \quad (2.1)$$

Η εξίσωση 2.1 είναι μονότονα αύξουσα ως προς τις μεταβλητές  $b$  και  $e$ , και μονότονα φθίνουσα ως προς τις μεταβλητές  $a$ ,  $c$  και  $d$ , όμως δεν είναι μονότονη, αφού συμμετέχουν σε αυτή μεταβλητές τόσο με την κανονική όσο και με τη συμπληρωματική τους μορφή. Για να έχουμε μία ενιαία εμφάνιση μεταβλητών χρησιμοποιούμε δύο νέες μεταβλητές, λ.χ. για την  $a$  χρησιμοποιούμε τις  $a_+$  και  $a_-$  όπου  $a_+ = a$  και  $a_- = a'$ . Ομοίως πράττουμε και για τις υπόλοιπες μεταβλητές  $b, c, d$  και  $e$ . Με αυτές τις συμβάσεις η εξίσωση  $f$  παίρνει την παρακάτω μορφή:

$$f_{\uparrow} = a_- \cdot b_+ + c_- \cdot d_- + e_+ \quad (2.2)$$

Η λειτουργικότητα της εξίσωσης διατηρείται και ταυτόχρονα δεν υπάρχει καμία μεταβλητή με τη συμπληρωματική της μορφή εφοδιάζοντας την με μονότονα αύξουσα συμπεριφορά. Αντιστοίχως εάν θέλουμε να προσδώσουμε στην εξίσωση  $f$  μονότονα φθίνουσα συμπεριφορά χρησιμοποιούμε μόνον μεταβλητές που μπορούν να εμφανιστούν με τη συμπληρωματική τους μορφή αφήνοντας ανέπαφη τη λογική λειτουργία της εξίσωσης:

$$f_{\downarrow} = a'_+ \cdot b'_- + c'_+ \cdot d'_+ + e'_- \quad (2.3)$$

Με τη χρήση μονότονων εξισώσεων μπορούμε να είμαστε σίγουροι για ορθότητα χρήσης των DI κωδίκων, όμως έχουμε τον περιορισμό ότι θα πρέπει για κάθε μεταβλητή που συμμετέχει σε οποιοδήποτε κόμβο ενός δικτύου να έχουμε διαθέσιμες και τις δύο μορφές μιας μεταβλητής. Ταυτόχρονα, οι δύο μορφές

θα πρέπει να εφαρμόζονται ανεξάρτητα στο κύκλωμα. Ο λόγος της ανεξάρτητης εφαρμογής θα φανεί καλύτερα μέσα από την περιγραφή της, απαραίτητης για την ορθή λειτουργία του κυκλώματος, διφασικής λειτουργίας στην ενότητα που ακολουθεί.

### 2.2.2 Διφασική Λειτουργία

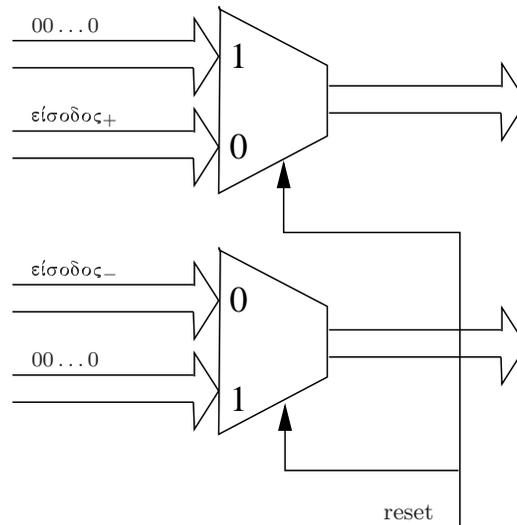
Στην Ενότητα 2.1 έγινε αναφορά στους τηλεπικοινωνιακούς διαύλους. Εκεί φάνηκε ότι για την ορθή χρήση των κωδικών ο δίαυλος πρέπει να βρίσκεται σε κατάσταση ηρεμίας πριν τη μετάδοση ενός μηνύματος. Το χαρακτηριστικό αυτό μεταφέρεται και στην τεχνική που παρουσιάζεται με τη μορφή της διφασικής λειτουργίας του κυκλώματος. Οι δύο φάσεις στις οποίες λειτουργεί το συνολικό κύκλωμα είναι (i) η φάση της αρχικοποίησης και (ii) η φάση του υπολογισμού.

Σκοπός της φάσης αρχικοποίησης είναι στο τέλος αυτής η κυριολεκτική έξοδος του κυκλώματος, καθώς και η έξοδος του Sync που έχει πάρει τη θέση του επιπρόσθετου δικτύου Boole, να έχουν σταθεροποιηθεί στο μηδενικό διάνυσμα, δηλαδή  $00\dots 0$ . Η κατάσταση αυτή αντιστοιχεί στην ηρεμία ενός επικοινωνιακού διαύλου. Παράλληλα, το ζεύγος,  $\langle \text{κυριολεκτική έξοδος, Sync} \rangle = \langle 00\dots 0, 00\dots 0 \rangle$ , δεν αποτελεί έγκυρη κωδική λέξη, ούτε του κώδικα Berger, ούτε του συστηματικού κώδικα Sperner, αποφεύγοντας τον κίνδυνο να θεωρηθεί η κυριολεκτική έξοδος  $00\dots 0$  στη φάση της αρχικοποίησης λανθασμένα ως έγκυρη.

**Πόρισμα 2.2.** *Εάν όλες οι μεταβλητές εισόδου μιας μονότονα αύξουσας εξίσωσης μεταβούν μονότονα στην τιμή 0, τότε και η τιμή της εξίσωσης οδηγείται με μονότονο τρόπο στο 0.*

Με βάση το Πόρισμα 2.2, για οδηγήσουμε τις εξόδους του DI κυκλώματος στο 0, θα πρέπει να μετασχηματίσουμε όλες τις εξισώσεις που το αποτελούν σε μονότονα αύξουσες και να τροφοδοτήσουμε με την τιμή 0 όλες τις μεταβλητές εισόδου. Αν μία μεταβλητή εισόδου, λ.χ. η  $a$  χρησιμοποιείται σε μία εξίσωση με την κανονική της μορφή,  $a_+$ , και σε μία άλλη με τη συμπληρωματική της,  $a_-$ , τότε θα πρέπει και οι δύο να λάβουν την τιμή 0 στη φάση αρχικοποίησης. Αυτό ισοδυναμεί στην αναληθή συνθήκη  $a_+ = a_- = 0$  και είναι ο λόγος για τον οποίο

οι δύο μορφές των μεταβλητών εισόδου πρέπει να εφαρμόζονται με ανεξάρτητο έλεγχο στο κύκλωμα. Ένας τρόπος επιβολής της ειδικής τιμής αρχικοποίησης, με ένα σήμα ελέγχου reset, φαίνεται στο Σχήμα 2.6, στο οποίο το σήμα reset



Σχήμα 2.6: Η Παράκαμψη των Εισόδων κατά τη Φάση της Αρχικοποίησης

έχει την τιμή 1 όταν το σύστημα βρίσκεται στη φάση της αρχικοποίησης και το κύκλωμα τροφοδοτείται με τα διανύσματα  $00 \dots 0$  μέσω των πολυπλεκτών.

Στη φάση υπολογισμού το σήμα reset του Σχήματος 2.6 παίρνει την τιμή 0 με αποτέλεσμα το κύκλωμα να τροφοδοτείται ταυτόχρονα, σε κυκλωματικό επίπεδο, με τις δύο μορφές των εισόδων, οι οποίες θα πρέπει να έχουν σταθεροποιηθεί πριν την πτώση του σήματος reset. Με την αλλαγή επιλογής του πολυπλέκτη κάποιες από τις μεταβλητές εισόδου στο κύκλωμα θα μεταβούν από την τιμή 0 της φάσης αρχικοποίησης στην τιμή 1, τροφοδοτώντας το κύκλωμα με έγκυρα δεδομένα εισόδου. Για την ακρίβεια ακριβώς οι μισές μεταβλητές εισόδου θα έχουν την τιμή 1. Αυτό προκύπτει από το γεγονός ότι κάθε είσοδος αντιστοιχεί σε δύο μεταβλητές, εάν η είσοδος έχει την τιμή 1 τότε η μεταβλητή που αντιστοιχεί στην κανονική μορφή της εισόδου θα έχει την τιμή 1 ενώ εάν η είσοδος έχει την τιμή 0 τότε η μεταβλητή που αντιστοιχεί στη συμπληρωματική της μορφή θα έχει την τιμή 1.

Λόγω της μονοτονίας των εξισώσεων των δικτύων Boole, οι έξοδοι του κυκλώματος μπορούν να μεταβούν μονάχα από 0 σε 1, με την έξοδο του αρχικού

δικτύου Boole να είναι η κυριολεκτική έξοδος του κυκλώματος και την έξοδο του επιπρόσθετου δικτύου Boole να είναι το Sync του εκάστοτε DI κώδικα, δηλαδή είτε Berger, είτε Sperner. Όταν ο συνδυασμός των δύο αυτών εξόδων αποτελεί μια έγκυρη λέξη του κώδικα, η φάση υπολογισμού θεωρείται λήξασα, κάτι που πρέπει να ανιχνεύσει ο μηχανισμός της αναγνώρισης ολοκλήρωσης.

### 2.2.3 Αναγνώριση Ολοκλήρωσης Υπολογισμού Εξόδων

Η έως τώρα κατασκευή του κυκλώματος εξασφαλίζει ότι στην τελική κατάσταση της φάσης υπολογισμού η έξοδος του κυκλώματος, <κυριολεκτική έξοδος, Sync>, θα είναι μία λέξη του κώδικα Berger χωρίς να υπάρχει ενδεχόμενο παρερμηνείας έως ότου το διάνυσμα φτάσει στην τελική του κατάσταση. Η ανίχνευση της τελικής κατάστασης είναι αποστολή του μηχανισμού της αναγνώρισης ολοκλήρωσης. Στο [9] αναφέρεται ένας τέτοιος μηχανισμός ως Απαριθμητικός Αποκωδικοποιητής (Enumeration-based Decoder, ED). Η δομή του ED είναι μία εξίσωση που έχει ως είσοδο τα ψηφία ενός διανύσματος που είναι μία υποψήφια λέξη ενός DI κώδικα και παίρνει την τιμή 1 όταν και μόνο όταν το διάνυσμα είναι πράγματι μια τέτοια λέξη. Στον Πίνακα 2.3 φαίνεται ο κώδικας Berger για κυριολεκτικά δεδομένα δύο ψηφίων. Τα  $d1d0$  είναι τα ψηφία των κυριολεκτικών δεδομένων και τα  $c1c0$  τα ψηφία του Sync. Συνεπώς, ο ED για αυτόν τον κώδικα θα είναι μία

d1	d0	c1	c0
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

Πίνακας 2.3: Ο Κώδικας Berger για Δεδομένα 2 Ψηφίων

συνάρτηση τεσσάρων εισόδων, η οποία για κάθε ελαχιστόρο του Πίνακα 2.3, θα παίρνει την τιμή 1. Αν για τους υπόλοιπους δυνατούς ελαχιστόρους η συνάρτηση δίνει 0 τότε η υλοποίηση αυτής της συνάρτησης οδηγεί σε έναν αποκωδικοποιητή που μπορεί να παράγει απρόβλεπτες μεταβάσεις, όπως  $0 \rightarrow 1 \rightarrow 0$  ή  $1 \rightarrow 0 \rightarrow 1$

στην έξοδο του, λόγω της μεταβλητότητας στις καθυστερήσεις των εισόδων του, των πυλών και των καλωδίων που τον αποτελούν [3, 12]. Κάτι τέτοιο είναι ανεπιθύμητο σε μονότονα κυκλώματα γιατί περιμένουμε από τον ED να έχει έξοδο 1 όταν και μόνο όταν λάβει μία έγκυρη κωδική λέξη. Παρόλα αυτά μπορούμε να εκμεταλλευτούμε τη διφασική λειτουργία και την μονοτονία των εξισώσεων του κυκλώματος για να υλοποιήσουμε την εξίσωση του ED ως εξής:

$$f = c1 + d1 \cdot d0 + d0 \cdot c0 + d1 \cdot c0 \quad (2.4)$$

Γενικότερα, ο προτεινόμενος από το [9] τρόπος υλοποίησης του ED είναι μία AND πύλη για κάθε κωδική λέξη, με τις εισόδους της πύλης να είναι τα ψηφία που έχουν την τιμή 1 στην κωδική λέξη. Οι έξοδοι των πυλών AND είναι εισοδοί μιας πύλης OR της οποίας η έξοδος είναι αυτή του ED. Η εν λόγω συνάρτηση έχει τα εξής χαρακτηριστικά:

- Είναι εκ φύσεως μονότονα αύξουσα χωρίς να χρειάζεται κανενός είδους προσαρμογή όπως αυτές που είδαμε σε προηγούμενες ενότητες.
- Δεν προκαλεί ανεπιθύμητες μεταβάσεις στην έξοδό της με δεδομένο ότι οι εισοδοί της ακολουθούν διφασική λειτουργία, δηλαδή διαδοχή του μηδενικού διανύσματος και έγκυρων δεδομένων.
- Είναι η βέλτιστη διεπίπεδη απεικόνιση, μιας και όλοι οι όροι που την αποτελούν είναι πρωτεύοντες.

Όλα τα παραπάνω ισχύουν και για το μηχανισμό αναγνώρισης ολοκλήρωσης του συστηματικού κώδικα Spener με χρήση του ED. Στον Πίνακα 2.4 φαίνεται ο κώδικας Spener για κυριολεκτικά δεδομένα δύο ψηφίων. Τα  $d1d0$  είναι τα ψηφία των κυριολεκτικών δεδομένων και τα  $c1c0$  τα ψηφία του Sync. Η υλοποίηση του ED στην περίπτωση αυτή είναι:

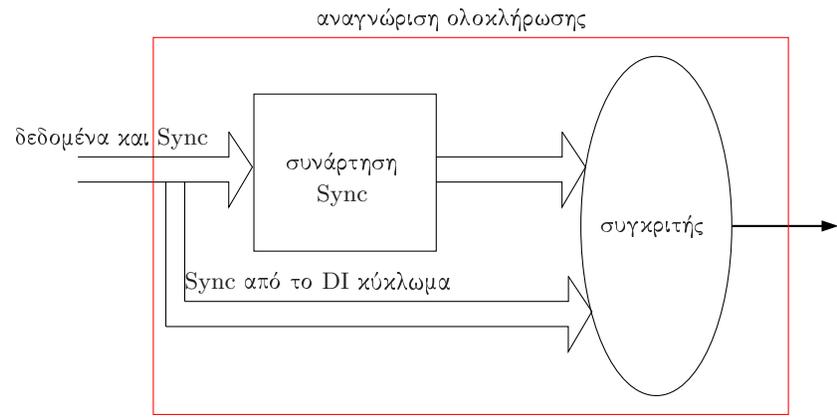
$$f = c1 \cdot c0 + d1 \cdot d0 + d0 \cdot c0 + d1 \cdot c0 \quad (2.5)$$

Στο [9] αναφέρεται ένας ακόμα δημοφιλής τρόπος για να επιτευχθεί αναγνώριση ολοκλήρωσης όταν χρησιμοποιούνται DI κώδικες. Με την μέθοδο αυτή, η οποία εφαρμόζεται επίσης σε συστηματικούς κώδικες, δεν επεξεργάζονται άμεσα

τα δεδομένα μαζί με το αποτέλεσμα της συνάρτησης Sync για να ελεγχθεί αν ικανοποιούν τον εκάστοτε κώδικα, αλλά εφαρμόζονται ξεχωριστά τα κυριολεκτικά δεδομένα σε υλοποίηση της συνάρτησης Sync και συγκρίνονται με αυτά που συνοδεύουν τη λέξη όπως δείχνει το Σχήμα 2.7. Όταν ο συγκριτής διαπιστώσει τα δύο διανύσματα ως ίσα, τότε η λέξη που έχουμε εισάγει ανήκει στον κώδικα. Δυστυχώς, αυτή η μέθοδος αναγνώρισης ολοκλήρωσης δε μπορεί να χρησιμοποιηθεί για κώδικες των οποίων τα πλεονάζοντα ψηφία δεν είναι τουλάχιστον όσα αυτά των κυριολεκτικών δεδομένων, λόγω του ότι στην υλοποίηση προκύπτουν κυκλώματα που παράγουν μη-μονότονες μεταβάσεις στην έξοδό τους, [9]. Κάτι τέτοιο αποκλείει τον κώδικα Berger όμως ταιριάζει στο συστηματικό κώδικα Sperner, προσφέροντας μια εναλλακτική υλοποίηση για την αναγνώριση ολοκλήρωσης του κώδικα αυτού.

d1	d0	c1	c0
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

Πίνακας 2.4: Ο Κώδικας Sperner για Δεδομένα 2 Ψηφίων



Σχήμα 2.7: Αναγνώριση Ολοκλήρωσης με Βάση τη Σύγκριση των Sync

Στο Κεφάλαιο αυτό αναλύσαμε τις βασικές αρχές που πρέπει να διέπουν ένα DI κύκλωμα. Είδαμε τους απαραίτητους μετασχηματισμούς που γίνονται στο αρχικό δίκτυο Boole και τον τρόπο της ορθής λειτουργίας του κυκλώματος. Επίσης παρουσιάσαμε δύο DI κώδικες, τον Berger και τον Sperner, τους οποίους θα χρησιμοποιήσουμε στην κατασκευή των DI κυκλωμάτων. Το επόμενο κεφάλαιο ξεκινά με μία λεπτομερέστερη παρουσίαση της καθολικής λειτουργίας ενός DI κυκλώματος, και έπειτα γίνεται αναφορά σε θέματα υλοποίησης που αφορούν κυρίως το επιπρόσθετο δίκτυο Boole, δηλαδή την εκάστοτε συνάρτηση Sync.

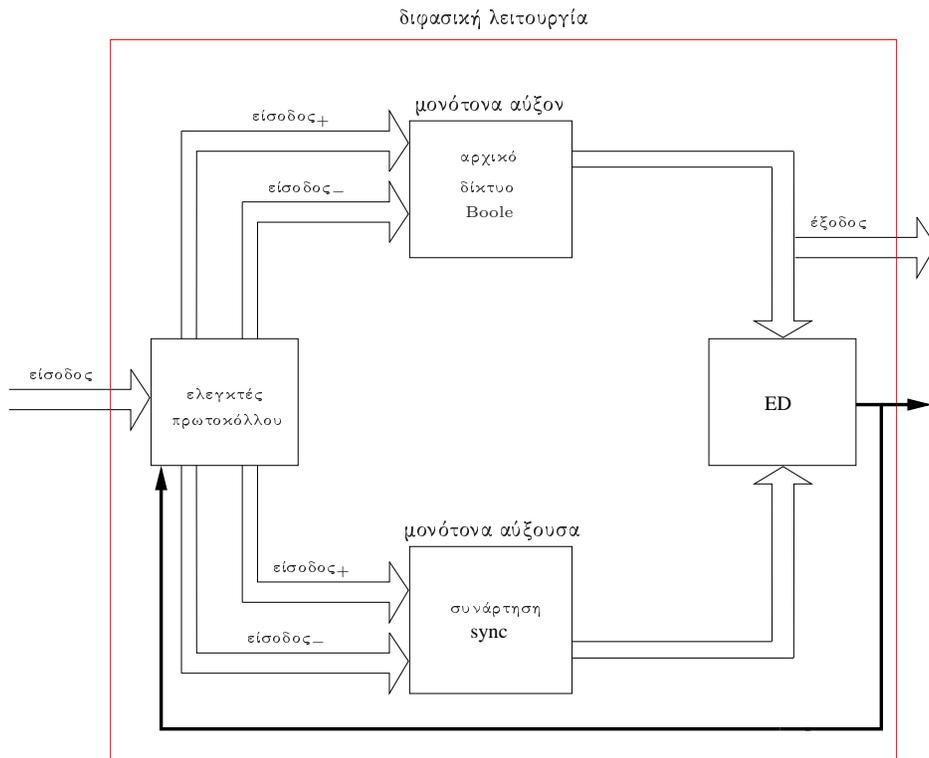
## Κεφάλαιο 3

# Μεθοδολογία - Υλοποίηση

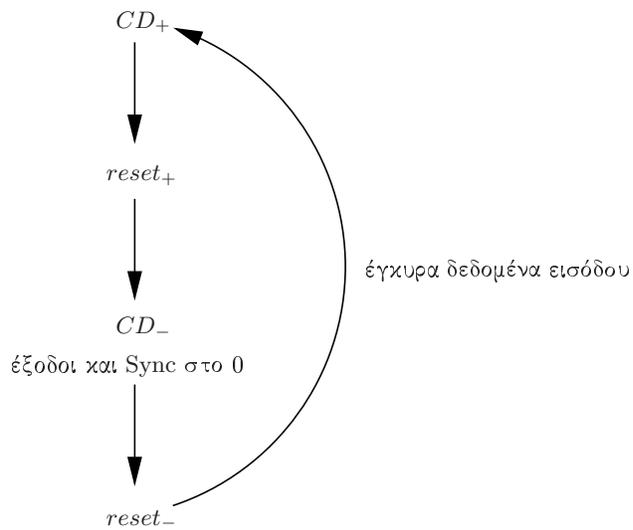
Στο προηγούμενο κεφάλαιο παρουσιάσαμε τις βασικές αρχές για τη χρήση των DI κωδίκων και τις συνθήκες λειτουργίας στις οποίες μπορούν να εφαρμόζονται. Έπειτα συσχετίσαμε τις συνθήκες αυτές με ιδιότητες συνδυαστικών κυκλωμάτων σε μια προσπάθεια να μετασχηματιστούν αυτά σε κυκλώματα με αναγνώριση ολοκλήρωσης. Στο παρόν κεφάλαιο βρίσκεται το μεγαλύτερο μέρος των όσων υλοποιήθηκαν στη διάρκεια της εργασίας που παρουσιάζεται, καθώς ενώνουμε όλα τα παραπάνω δείχνοντας τη καθολική λειτουργία ενός τέτοιου κυκλώματος. Επίσης, θα δώσουμε τις λεπτομέρειες για τη ροή υλοποίησης του κυκλώματος και θα παραθέσουμε μια σειρά από τεχνικές βελτιστοποίησης αυτής.

### 3.1 Καθολική Λειτουργία

Παραθέτουμε εδώ ξανά, στο Σχήμα 3.1, το Σχήμα 2.1 με τα μέρη του DI κυκλώματος, όπου το επιπρόσθετο δίκτυο Boole έχει αντικατασταθεί από την κατάλληλη συνάρτηση Sync και η αναγνώριση ολοκλήρωσης από τον ED. Το κεντρικό χαρακτηριστικό του συστήματος είναι οι ελεγκτές του πρωτοκόλλου. Εκεί βρίσκονται οι πολυπλέκτες του Σχήματος 2.6, αλλά και η λογική που συντονίζει το πέρασμα από τη μία φάση λειτουργίας στην άλλη. Η λογική αυτή λειτουργεί ως μία απλή μηχανή καταστάσεων που οδηγεί τα σήματα ελέγχου του Σχήματος 3.2. Το σήμα CD είναι η έξοδος της Αναγνώρισης Ολοκλήρωσης (Completion Detection, CD) ενώ τα + και - δίπλα στα ονόματα των σημάτων υποδηλώνουν



Σχήμα 3.1: Σχεδιάγραμμα Υλοποίησης Κυκλώματος με Χρήση DI Κωδικών



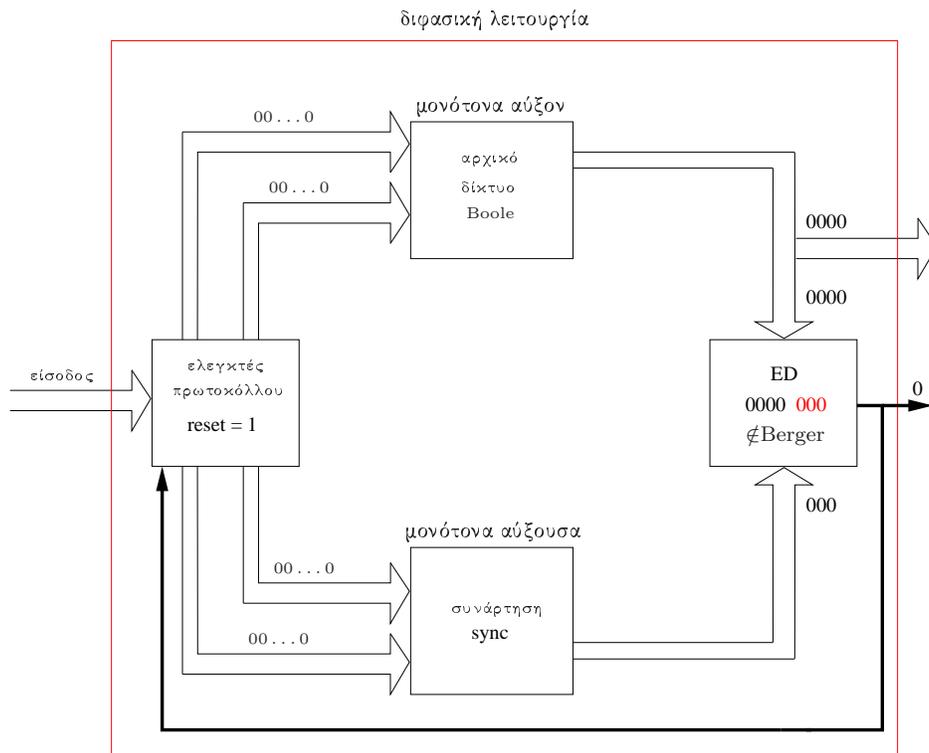
Σχήμα 3.2: Σήματα Ελέγχου της Διφασικής Λειτουργίας

ότι το σήμα εκτελεί τη μετάβαση  $0 \rightarrow 1$  ή  $1 \rightarrow 0$ , δηλαδή είναι μονότονα αύξον ή φθίνον αντίστοιχα.

Ακολουθώντας ένα πλήρη κύκλο λειτουργίας με χρήση του DI κώδικα Berger,

Ξεκινώντας από τη φάση της αρχικοποίησης, το σήμα reset ξεκινά με τιμή 1, δηλαδή βρισκόμαστε στην κατάσταση  $reset_+$  του Σχήματος 3.2. Οι πολυπλέκτες του Σχήματος 2.6, οδηγούν τα διανύσματα  $00\dots 0$  στις εισόδους των κυκλωμάτων, μηδενίζοντας τις εξόδους τους καθώς και το σήμα CD αφού η λέξη με κυριολεκτικά δεδομένα  $00\dots 0$  και Sync  $00\dots 0$  δεν ανήκει στον κώδικα Berger. Η φάση αυτή απεικονίζεται στο Σχήμα 3.3 όπου παραθέτουμε ένα παράδειγμα με έξοδο τεσσάρων ψηφίων, και συνεπώς συνάρτηση Sync με τρία ψηφία, δηλαδή αρκετά για να αναπαραστήσουμε τον αριθμό τέσσερα σε δυαδική αναπαράσταση. Με χρήση του συστηματικού κώδικα Sperner, η διάταξη δε διαφοροποιείται ιδιαίτερα, με τη μόνη αλλαγή ότι η αντίστοιχη έξοδος Sync έχει τώρα τέσσερα ψηφία, όσα δηλαδή και τα δεδομένα, ενώ και η εσωτερική συνάρτηση του ED προσαρμόζεται κατάλληλα.

Όταν οι έξοδοι σταθεροποιηθούν, το σήμα reset πέφτει και η επόμενη έγκυρη είσοδος στο κύκλωμα περνάει μέσα από τον πολυπλέκτη για να τροφοδοτήσει τα



Σχήμα 3.3: Η Τελική Κατάσταση της Φάσης Αρχικοποίησης με Χρήση του Κώδικα Berger

δύο υποκυκλώματα. Η σταθεροποίηση των σημάτων μπορεί να ελεγχθεί προσθέτοντας μία επιπλέον λειτουργία στον ED που να ενημερώνει τους ελεγκτές με ένα σήμα ότι όλες οι εξόδους έχουν μηδενιστεί. Εναλλακτικά μπορούμε να κάνουμε χρήση ενός στοιχείου καθυστέρησης και να πέφτει το σήμα reset σε τόσο χρόνο όσο κάνει το κύκλωμα να αρχικοποιηθεί με βάση ένα εργαλείο χρονικής ανάλυσης. Η τιμές των έγκυρων δεδομένων εισόδου θα πρέπει να έχουν σταθεροποιηθεί προτού πέσει το σήμα reset, λ.χ. μπορεί να είναι αποθηκευμένες σε μανταλωτές, αφού για να προφυλαχθεί η μονότονη λειτουργία του κυκλώματος δεν επιτρέπεται να έχουμε παράνομες μεταβάσεις στις εισόδους. Κάποιες από τις εισόδους θα μεταβούν από 0 σε 1, όλα τα εσωτερικά σήματα συμπεριφέρονται μονότονα, με κάθε ψηφίο να αλλάζει τιμή το πολύ μία φορά, και με την πάροδο του χρόνου το κύκλωμα θα ισορροπήσει με τις εξόδους του να έχουν πάρει τις τελικές τους τιμές.

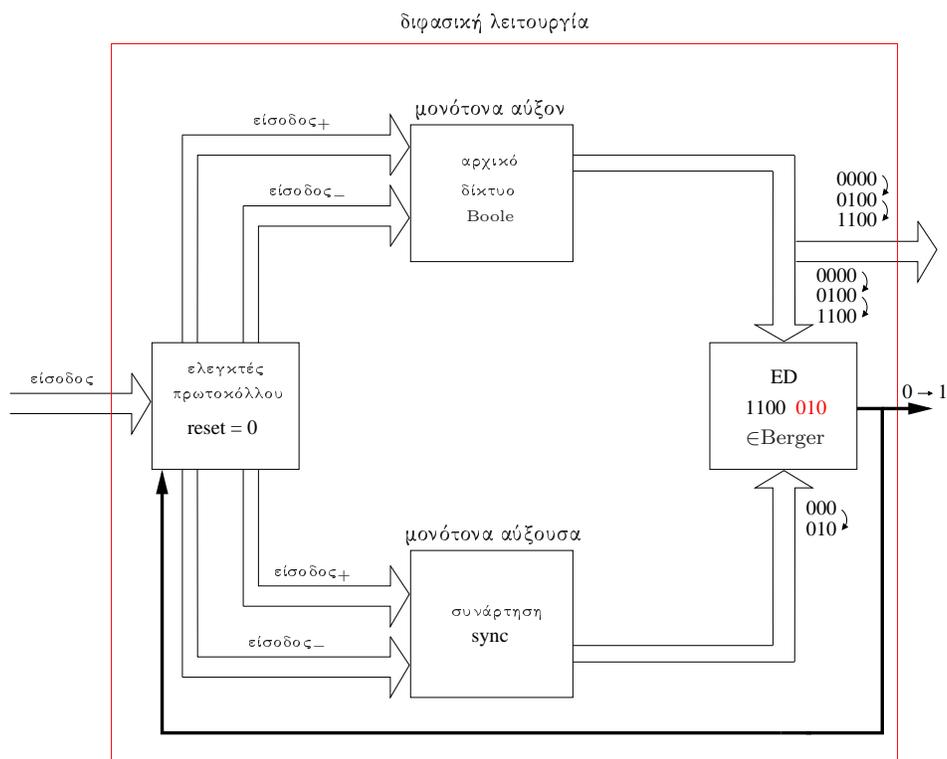
Στο Σχήμα 3.4 παρουσιάζεται η τελική κατάσταση της φάσης υπολογισμού για έξοδο των τεσσάρων ψηφίων, και για κάποια συγκεκριμένα δεδομένα εισόδου που παράγουν την τελική κατάσταση του διανύσματος εξόδου 1100. Η τελική τιμή της συνάρτησης Sync θα είναι 010, δηλαδή η δυαδική αναπαράσταση του αριθμού 2, μια και τόσα είναι τα μηδενικά του διανύσματος εξόδου. Επίσης, στο Σχήμα 3.4 φαίνονται πιθανές, μονότονες, μεταβάσεις των εξόδων μέχρι αυτές να λάβουν την τελική τους τιμή. Μόλις καταλήξουν τα διανύσματα εξόδου στις τελικές τους τιμές, το σήμα της αναγνώρισης ολοκλήρωσης (CD) θα γίνει 1 και έτσι δηλώνεται ότι έχει ολοκληρωθεί ο υπολογισμός του κυκλώματος. Την αλλαγή αυτή ανιχνεύει ο μηχανισμός της μηχανής καταστάσεων του Σχήματος 3.2, με συνέπεια το σήμα reset να ξαναπάρει την τιμή 1 και να ξεκινήσει η όλη διαδικασία από την αρχή.

### **Αναγνώριση Ολοκλήρωσης με Βάση τη Σύγκριση**

Στο τέλος του 2<sup>ου</sup> κεφαλαίου αναφέρθηκε ένας δεύτερος τρόπος για την αναγνώριση ολοκλήρωσης, όταν χρησιμοποιούνται DI κώδικες με πλεονασμό τουλάχιστον όσα είναι και τα ψηφία των δεδομένων, με δομή που είναι διαφορετική από αυτή του ED. Μπορεί συνεπώς να χρησιμοποιηθεί με το συστηματικό κώδι-

κα Sperner. Στον Πίνακα 3.1 φαίνεται ο κώδικας για τρία ψηφία κυριολεκτικών δεδομένων. Τα  $d2d1d0$  είναι τα ψηφία των δεδομένων και τα  $c2c1c0$  τα ψηφία του Sync. Η συνάρτηση Sync του Πίνακα 3.1, και κατ' επέκταση η συγκεκριμένη συνάρτηση Sync για αυθαίρετο αριθμό κυριολεκτικών δεδομένων, είναι εκ του ορισμού της μονότονα φθίνουσα. Το γεγονός ότι είναι μονότονη μας επιτρέπει να υλοποιήσουμε σωστά ένα συνδυαστικό κύκλωμα με αναγνώριση ολοκλήρωσης εκμεταλλευόμενοι τη διφασική λειτουργία και τη μονοτονία που αναλύσαμε στο προηγούμενο κεφάλαιο.

Όπως και στην προηγούμενη υλοποίηση στη φάση αρχικοποίησης τροφοδοτούμε το κύκλωμα με μηδενικά διανύσματα. Η κυριολεκτική έξοδος του αρχικού δικτύου Boole και και αυτή του επιπρόσθετου δικτύου Boole του κυκλώματος έχουν παρομοίως μηδενικές εξόδους, λόγω της μονότονα αύξουσας λειτουργίας τους. Αντιθέτως η συνάρτηση Sync του μηχανισμού της αναγνώρισης ολοκλήρωσης, ως μονότονα φθίνουσα συνάρτηση, έχει έξοδο  $11\dots 1$ . Γραφικά η



Σχήμα 3.4: Η Τελική Κατάσταση της Φάσης Υπολογισμού με Χρήση του Κώδικα Berger

d2	d1	d0	c2	c1	c0
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	0

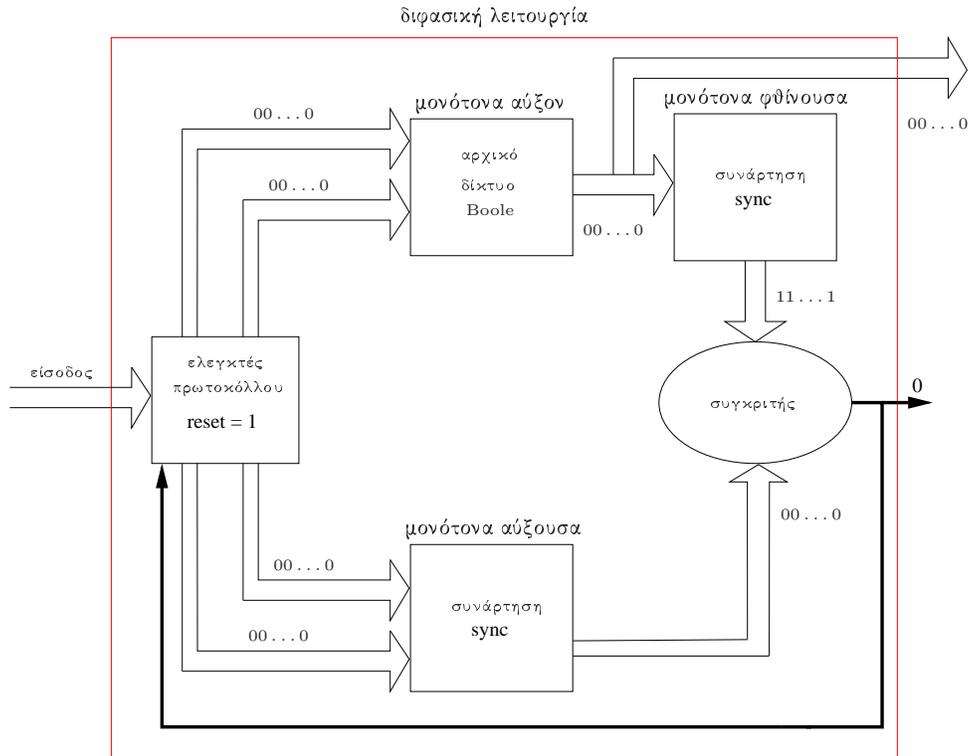
Πίνακας 3.1: Ο Συστηματικός Κώδικας Sperner για Κυριολεκτικά Δεδομένα 3 Ψηφίων

κατάσταση αυτή απεικονίζεται στο Σχήμα 3.5.

Όταν το σύστημα εισέρχεται στη φάση του υπολογισμού συμβαίνουν μεταβάσεις  $0 \rightarrow 1$  στις εισόδους, οι οποίες με τη σειρά τους προκαλούν αντίστοιχες μεταβάσεις στις εξόδους του αρχικού και του επιπρόσθετου δικτύου Boole. Η μονότονα φθίνουσα συνάρτηση  $Syn_c$  του μηχανισμού της αναγνώρισης ολοκλήρωσης βλέπει στην είσοδο της αύξηση και συνεπώς μειώνει την έξοδό της, επιτρέποντας μόνον μεταβάσεις  $1 \rightarrow 0$ . Τελικά, οι δύο εισοδοί του συγκριτή εξισώνονται και η έξοδός του παίρνει την τιμή 1, σηματοδοτώντας το τέλος της φάσης υπολογισμού. Στο Σχήμα 3.6, φαίνεται ένα παράδειγμα λειτουργίας της φάσης αυτής όπου παραδείγματα χάριν θεωρούμε μια έξοδο δεδομένων των πέντε ψηφίων και ότι το διάνυσμα της κυριολεκτικής εξόδου για κάποια συγκεκριμένη είσοδο έχει σαν τελική κατάσταση την τιμή 10110. Απεικονίζονται επίσης και πιθανές μονότονες μεταβάσεις των διανυσμάτων έως ότου αποκτήσουν τις τελικές τους τιμές.

Αυτό που απομένει είναι να αποδειχθεί ότι οι εισοδοί του συγκριτή εξισώνονται όταν και μόνον όταν έχουν λάβει την τελική τους τιμή, η οποία και ακολουθεί.

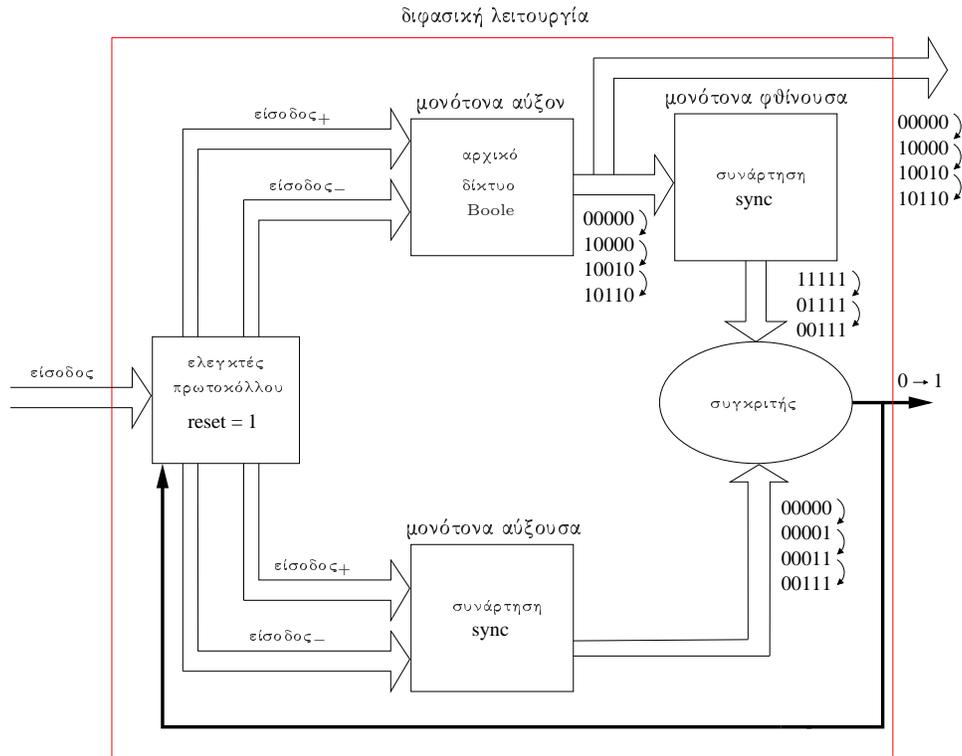
**Απόδειξη 3.1.** Ορίζουμε ως  $Syn_c \uparrow$  την έξοδο του επιπρόσθετου δικτύου Boole του κυκλώματος, λόγω του ότι είναι μονότονα αύξουσα, και ως  $Syn_c \downarrow$  την έξοδο της συνάρτησης  $Syn_c$  του μηχανισμού της αναγνώρισης ολοκλήρωσης, λόγω



Σχήμα 3.5: Η Φάση Αρχικοποίησης με τη Μέθοδο του Συγκριτή και Χρήση του Κώδικα Sperner

του ότι είναι μονότονα φθίνουσα. Έστω ότι τα δύο αυτά διανύσματα έχουν  $n$  ψηφία, και ότι στην τελική τους κατάσταση έχουν τα  $k$  λιγότερο σημαντικά τους ψηφία με τιμή 1. Υποθέτουμε ότι τα δύο διανύσματα γίνονται ίσα σε κάποιο άλλο στιγμιότυπο της φάσης υπολογισμού, όχι δηλαδή στην τελική τους κατάσταση, ενώ δε θα έπρεπε και ότι ο αριθμός των ψηφίων τους που έχουν την τιμή 1 είναι  $m$ , όπου  $m \neq k$ .

- Εάν  $m < k$  τότε το διάνυσμα  $Sync \downarrow$  θα πρέπει να φτάσει σε μία κατάσταση με λιγότερα από  $k$  ψηφία με τιμή 1, ξεκινώντας από  $11 \dots 1$ , το οποίο είναι αδύνατο αφού λόγω της μονότονης λειτουργίας του δεν επιτρέπονται μεταβάσεις  $1 \rightarrow 0 \rightarrow 1$  σε κανένα από τα ψηφία του. Έτσι όταν αρχικά έχει  $n$  ψηφία με τιμή 1 και στην τελική του κατάσταση  $k$ , δεν μπορεί να λάβει τιμή με λιγότερα από  $k$  ψηφία με τιμή 1.
- Εάν  $m > k$  τότε το διάνυσμα  $Sync \uparrow$  θα πρέπει να φτάσει σε μία τιμή με περισσότερα από  $k$  ψηφία με τιμή 1, ξεκινώντας από  $00 \dots 0$ , κάτι που είναι



Σχήμα 3.6: Η Φάση Υπολογισμού με τη Μέθοδο του Συγκριτή και Χρήση του Κώδικα Sperner

αδύνατο λόγω της μονότονης λειτουργίας του αφού θα έπρεπε να επιτρέπονται οι μεταβάσεις  $0 \rightarrow 1 \rightarrow 0$ .

Συνεπώς, υπό μονότονες συνθήκες και διφασική λειτουργία, τα  $Sync\downarrow$  και  $Sync\uparrow$  μπορούν να εξισωθούν μόνο στην τελική τους κατάσταση και να ειδοποιήσουν ορθά για τη λήξη της φάσης του υπολογισμού.

### 3.2 Υλοποίηση του Επιπρόσθετου Δικτύου Boole

Το επιπρόσθετο δίκτυο Boole του κυκλώματος είναι αυτό που κάνει δυνατή την αναγνώριση ολοκλήρωσης, καθότι παρέχει την απαραίτητη πληροφορία, η οποία σε συνδυασμό με τις κυριολεκτικές εξόδους του αρχικού δικτύου Boole διαγιγνώσκει εάν το κύκλωμα έχει φτάσει στην τελική του κατάσταση. Το επιπρόσθετο δίκτυο Boole είναι η εκάστοτε συνάρτηση Sync εξεφρασμένη ως προς τις πρωτεύουσες εισόδους του κυκλώματος. Δηλαδή, για κάθε συνδυα-

σμό εισόδων το υποκύκλωμα αυτό δίνει στην έξοδο το αντίστοιχο Sync χωρίς να χρειάζεται να κοιτάζει την τελική κατάσταση των κυριολεκτικών εξόδων του αρχικού δικτύου Boole. Μέχρι στιγμής έχουμε σκιαγραφήσει το δίκτυο αυτό αναφέροντας τη λειτουργία και τη χρήση του, όμως δεν έχει γίνει αναφορά στη διαδικασία σχηματισμού του, η οποία κάνει χρήση του αρχικού δικτύου Boole. Για το σχηματισμό του επιπρόσθετου δικτύου Boole ακολουθήθηκαν δύο προσεγγίσεις, η μία με χρήση διεπίπεδης λογικής και η άλλη με χρήση πολυεπίπεδης, με τη δεύτερη να είναι η επικρατούσα τόσο λόγω πολυπλοκότητας όσο και λόγω αποτελεσμάτων όπως θα δούμε στη συνέχεια.

### 3.2.1 Διεπίπεδη Υλοποίηση του Επιπρόσθετου Δικτύου

Η προσέγγιση αυτή θεωρείται διεπίπεδη όχι γιατί το δίκτυο που παράγεται είναι διεπίπεδο<sup>1</sup>, αλλά επειδή κάνει χρήση μιας διεπίπεδης τεχνικής βελτιστοποίησης, αυτή των Δυαδικών Συσχετίσεων (Boolean Relations) [8]. Θα δούμε τις λεπτομέρειες των δυαδικών συσχετίσεων μέσα από τη χρήση τους στην υλοποίηση του επιπρόσθετου δικτύου Boole ενός συνδυαστικού κυκλώματος με αναγνώριση ολοκλήρωσης.

#### 3.2.1.1 Δυαδικές Συσχετίσεις ως Μεθοδολογία Απλοποίησης

Θεωρούμε ένα απλό παράδειγμα, όπως το παρακάτω:

$$\begin{aligned} e &= a + b + c + d \\ f &= a' + b + c + d \\ g &= a + b' + c + d \end{aligned} \tag{3.1}$$

όπου έχουμε ένα δίκτυο Boole με τέσσερις εισόδους και τρεις εξόδους. Ο πίνακας αληθείας για το δίκτυο αυτό φαίνεται στον Πίνακα 3.2. Εξ' ορισμού, η συνάρτηση Sync για το Δίκτυο 3.1, έχει την ίδια έξοδο για εισόδους που έχουν τον ίδιο αριθμό 1. Συνεπώς, αν στον Πίνακα αληθείας 3.2 αντικαθιστούσαμε ένα ή μερικά

<sup>1</sup>Αλλωστε ακόμα και αυτό να συνέβαινε υπάρχει πάντα η δυνατότητα μετατροπής του σε πολυεπίπεδο μέσω αλγορίθμων λογικής σύνθεσης

ή και όλα τα διανύσματα εξόδου με άλλα τα οποία έχουν το ίδιο αριθμό 1 αυτό δε θα προκαλούσε καμία αλλαγή στη λειτουργία της συνάρτησης Sync. Η ιδέα αυτή απεικονίζεται στον Πίνακα 3.3. Εκεί βλέπουμε ότι για τρεις ελαχιστόρους έχουμε τη δυνατότητα να διαλέξουμε μεταξύ τριών εξόδων και να έχουμε την ίδια συμπεριφορά για το κύκλωμα. Ο Πίνακας 3.3 απεικονίζει μία δυαδική συσχέτιση. Δηλαδή, είναι η περιγραφή ενός κυκλώματος με πλεονάζουσα πληροφορία η οποία για το ίδιο διάνυσμα εισόδου δύναται να έχει πάνω από μία ισοδύναμες εξόδους.

Στόχος της επίλυσης μιας δυαδικής συσχέτισης είναι γίνουν οι κατάλληλες επιλογές των εξόδων ώστε να προκύψει ένα δίκτυο με το μικρότερο δυνατό εμβαδό. Για παράδειγμα αν στη συσχέτιση του Πίνακα 3.3 επιλέξουμε για όλες τις αμφιλεγόμενες εξόδους το διάνυσμα 110, το Δίκτυο 3.1 παίρνει τη μορφή που ακολουθεί:

$$\begin{aligned}
 e &= 1 \\
 f &= 1 \\
 g &= a \cdot b + c + d
 \end{aligned}
 \tag{3.2}$$

Το δίκτυο αυτό έχει σαφώς μικρότερο εμβαδό από το αρχικό.

a	b	c	d	e	f	g
0	0	0	0	0	1	1
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	0	1	1	1	1	1
0	1	0	0	1	1	0
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

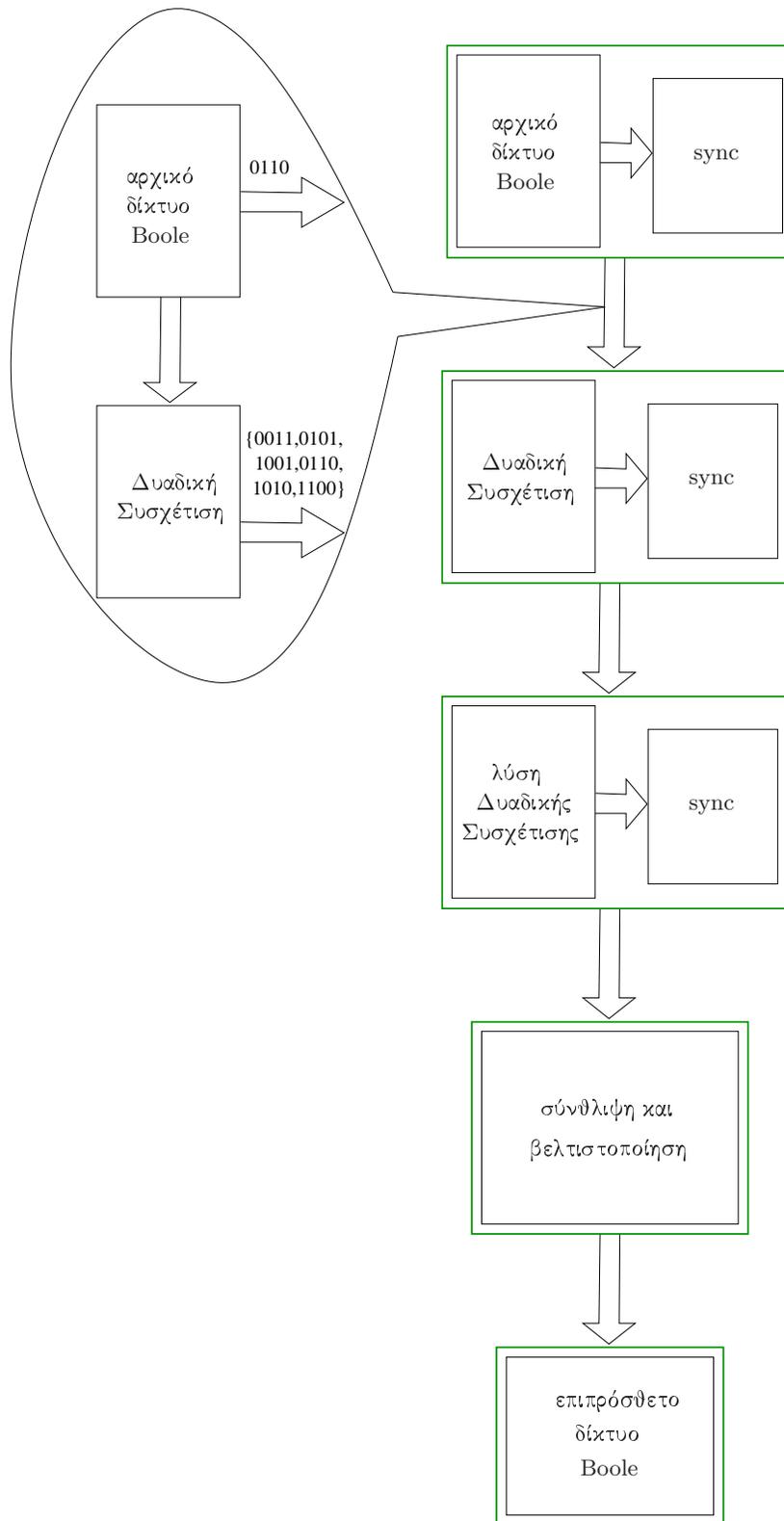
Πίνακας 3.2: Ο Πίνακας Αληθείας του Δικτύου 3.1

a	b	c	d	e f g
0	0	0	0	{011,101,110}
0	0	0	1	111
0	0	1	0	111
0	0	1	1	111
0	1	0	0	{011,101,110}
0	1	0	1	111
0	1	1	0	111
0	1	1	1	111
1	0	0	0	{011,101,110}
1	0	0	1	111
1	0	1	0	111
1	0	1	1	111
1	1	0	0	111
1	1	0	1	111
1	1	1	0	111
1	1	1	1	111

Πίνακας 3.3: Η Δυαδική Συσχέτιση που Προκύπτει από το Δίκτυο 3.1

Η έννοια των δυαδικών συσχετίσεων είναι διεπίπεδη γιατί για να περιγραφεί η συσχέτιση είναι απαραίτητη μία αντιστοίχιση όπως αυτή ενός πίνακα αληθείας, δηλαδή εξετάζει όλο το κύκλωμα συνολικά εκφράζοντάς το μονάχα σε ένα λογικό επίπεδο. Επίσης, η βελτιστότητα της λύσης μιας δυαδικής συσχέτισης έγκειται στο να χρησιμοποιεί το μικρότερο αριθμό πρωτευουσών συνιστωσών, [8]. Οι πρωτεύοντες συνιστώσες είναι ο βασικός άξονας των αλγορίθμων της διεπίπεδης βελτιστοποίησης.

Η παραπάνω θεώρηση δείχνει και γιατί υπάρχει η προσδοκία ότι η συνάρτηση Sync που αντιστοιχεί στο επιπρόσθετο δίκτυο Boole του κυκλώματος θα έχει εμβადό μικρότερο από αυτό του αρχικού δικτύου Boole. Η ροή υλοποίησης φαίνεται στο Σχήμα 3.7. Εκεί φαίνεται ότι το επιπρόσθετο δίκτυο Boole μπορεί να είναι ένα αντίγραφο του αρχικού δικτύου το οποίο ακολουθείται από ένα κύκλωμα που υλοποιεί τη συνάρτηση Sync. Αυτό όμως εισάγει πλεονάζουσα πολυπλοκότητα καθώς δεν απαιτείται η ακριβής υλοποίηση του αρχικού δικτύου Boole παρά μονάχα του Sync. Για το λόγο αυτό δημιουργείται μία δυαδική συσχέτιση παρόμοια με αυτήν του Πίνακα 3.3, όπου δηλαδή σε κάθε αντιστοίχιση εισόδου-εξόδου



Σχήμα 3.7: Η Ροή Υλοποίησης Ενός Επιπρόσθετου Δικτύου Boole με Χρήση Δυαδικών Συσχετίσεων

προστίθενται οι ισοδύναμες εξόδοι, προκειμένου να επιτευχθεί μείωση του εμβαδού σε σχέση με το αντίγραφο του αρχικού δικτύου Boole. Ένα παράδειγμα προσθήκης φαίνεται μέσα στο σύννεφο του Σχήματος 3.3 όπου θεωρούμε δεδομένα εξόδου 4 ψηφίων και απεικονίζονται οι προσθήκες που αντιστοιχούν σε διάνυσμα εξόδου με τιμή 0110. Η σύνθλιψη και η βελτιστοποίηση της λύσης της συσχέτισης με τη συνάρτηση *Sync* γίνεται για την ενδεχόμενη απαλοιφή περιττών ενδιάμεσων κόμβων και το αποτέλεσμα δίνει το επιπρόσθετο δίκτυο Boole. Η βελτιστοποίηση κάνει χρήση κλασικών αλγορίθμων λογικής σύνθεσης.

**Ορισμός 3.1.** Ως σύνθλιψη ορίζεται η έκφραση των πρωτεύουσών εξόδων ενός κυκλώματος, ως προς τις πρωτεύουσες εισόδους αυτού, απαλείφοντας όλα τα ενδιάμεσα λογικά επίπεδα, όπως ενδιάμεσους λογικούς κόμβους, πυρήνες, κοινούς όρους κ.α.

Η διεπίπεδη φύση της τεχνικής των δυαδικών συσχετίσεων δημιουργεί δύο προβλήματα στη ροή του Σχήματος 3.7:

- Πρακτικά είναι δύσκολο να παραχθεί ολόκληρος ο πίνακας αληθείας ακόμα και για πολύ μικρά πραγματικά κυκλώματα.
- Η λύση μιας δυαδικής συσχέτισης μπορεί να είναι η βέλτιστη διεπίπεδη, αλλά δεν είναι εξασφαλισμένο ότι δίνει κύκλωμα με μικρότερο εμβαδό από το αρχικό δίκτυο Boole σε πολυεπίπεδη λογική.

Στα παραπάνω έρχεται να προστεθεί και το γεγονός ότι ο επακριβής αλγόριθμος επίλυσης δυαδικών συσχετίσεων, που υλοποιήθηκε [8], έχει μεγάλη υπολογιστική πολυπλοκότητα και αδυνατεί να ολοκληρωθεί για μεγάλα παραδείγματα. Για την έως ένα βαθμό αντιμετώπιση των προβλημάτων αυτών, η ροή του Σχήματος 3.7 δεν εφαρμόζεται σε όλο το αντίγραφο του αρχικού δικτύου Boole αλλά σε κάθε επίπεδο της πολυεπίπεδης μορφής αυτού.

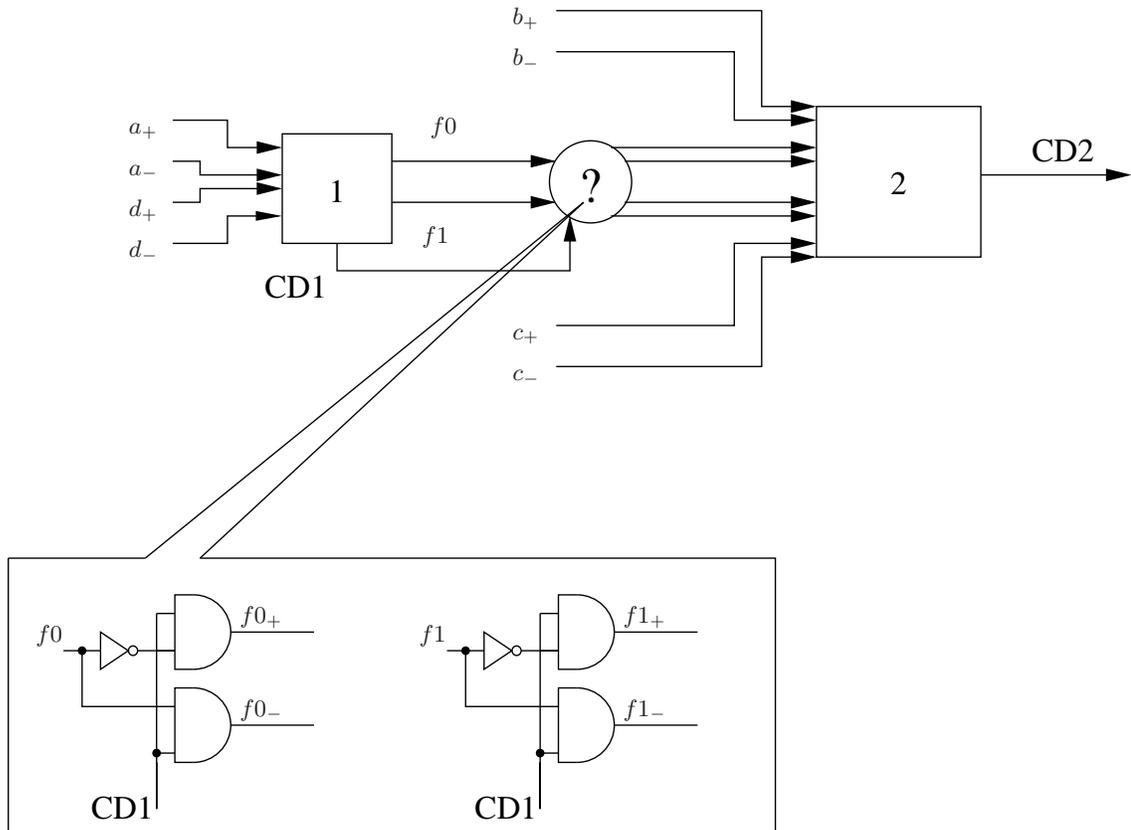
### 3.2.1.2 Διάσπαση Κυκλώματος σε Επίπεδα

Θεωρούμε το παρακάτω δίκτυο, με πρωτεύουσες εισόδους τα  $a, b, c$  και  $d$  και πρωτεύουσες εξόδους τις  $s_0, s_1, s_2$  και  $s_3$ :

$$\begin{aligned} s_0 &= f_0' + b' + c \\ s_1 &= f_0 + b + c' \\ s_2 &= f_1 + b' + c' \\ s_3 &= f_1' + b + c \\ f_0 &= a + d' \\ f_1 &= a' + d \end{aligned} \tag{3.3}$$

Όπως είναι εμφανές, το Δίκτυο 3.3 αποτελείται από δύο επίπεδα. Ένα που απαρτίζεται από τους εσωτερικούς κόμβους  $f_0$  και  $f_1$ , και ένα δεύτερο που το αποτελούν οι τέσσερις εξισώσεις των πρωτευουσών εξόδων του δικτύου. Αντί να συνθλιψουμε όλο το δίκτυο, να παράγουμε το πίνακα αληθείας αυτού, να σχηματίσουμε και να λύσουμε τις αντίστοιχες δυαδικές συσχετίσεις, με τις συνέπειες και τα προβλήματα που αναφέραμε στην προηγούμενη ενότητα, μπορούμε να εφαρμόσουμε τη ροή του Σχήματος 3.7 σε κάθε επίπεδο του δικτύου. Κάθε λογικό επίπεδο ενός πολυεπίπεδου κυκλώματος μπορεί να απεικονισθεί ως ένα διεπίπεδο κύκλωμα. Συνεπώς, στο Δίκτυο 3.3 θα γίνει ξεχωριστή υλοποίηση για τις τέσσερις πρώτες εξισώσεις και για τις δύο τελευταίες.

Η διάσπαση του κυκλώματος σε επίπεδα και η ξεχωριστή εφαρμογή της υλοποίησης σε αυτά δημιουργεί την ανάγκη για διασύνδεση μεταξύ των επιπέδων μετά το μετασχηματισμό τους σε ανεξάρτητα υποκυκλώματα με αναγνώριση ολοκλήρωσης. Το γιατί χρειάζεται αυτό φαίνεται στο Σχήμα 3.8. Όπως είδαμε στο Κεφάλαιο 2 για να λειτουργήσει ένα κύκλωμα με αναγνώριση ολοκλήρωσης σωστά θα πρέπει να είναι μονότονο. Η μονοτονία απαιτεί τα κυκλώματα να εκφραστούν χρησιμοποιώντας τόσο την κανονική μορφή όσο και την αντιστροφή των εισόδων τους ως ξεχωριστές μεταβλητές. Στην περίπτωση του Δικτύου 3.3, οι εσωτερικοί κόμβοι  $f_0$  και  $f_1$  που είναι είσοδοι του δεύτερου λογικού επιπέδου



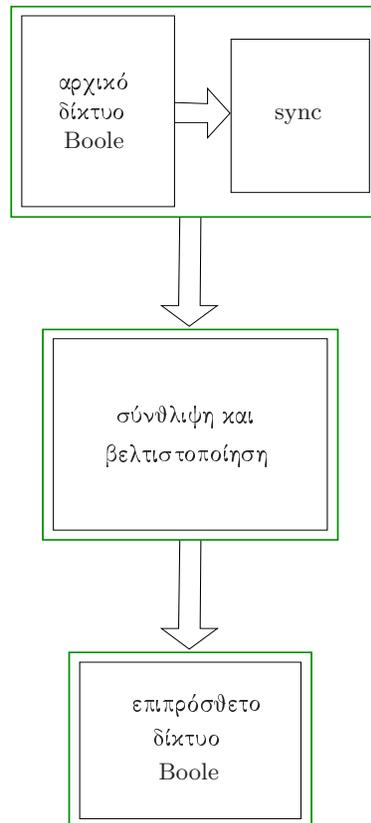
Σχήμα 3.8: Η Σύνδεση των Λογικών Επιπέδων του Δικτύου 3.3 μετά το Μετασχηματισμό τους σε Υποκυκλώματα με Αναγνώριση Ολοκλήρωσης

παρέχονται μόνο στην κανονική τους μορφή. Εάν η αντιστροφή των κόμβων αυτών παραχθεί απευθείας από έναν αντιστροφέα τότε δημιουργείται πρόβλημα κατά τη φάση αρχικοποίησης των κυκλωμάτων, όπου και οι δύο μορφές των μεταβλητών πρέπει να έχουν την τιμή 0. Για το σκοπό αυτό η αρχική μεταβλητή και η έξοδος του αντιστροφέα φράσσονται από δύο πύλες AND δύο εισόδων, όπως φαίνεται στο Σχήμα 3.8, με το σήμα ολοκλήρωσης (CD) του κυκλώματος από το οποίο προέρχονται. Έτσι, στη φάση αρχικοποίησης όπου το σήμα αναγνώρισης ολοκλήρωσης έχει την τιμή 0 οδηγεί και τις δύο μορφές των εσωτερικών κόμβων στην τιμή αυτή. Αντίστοιχα, στη φάση του υπολογισμού, μόλις το αντίστοιχο επίπεδο λογικής φτάσει στην τελική του κατάσταση, το σήμα της αναγνώρισης ολοκλήρωσης θα πάρει την τιμή 1 επιτρέποντας τότε να περάσουν οι τελικές τιμές των σημάτων. Το ενδεχόμενο οι έξοδοι του συστήματος να μην έχουν λάβει τις τελικές τους τιμές ενώ το σήμα της αναγνώρισης ολοκλήρωσης παίρνει την τιμή

1 είναι αδύνατο μια και η αναγνώριση ολοκλήρωσης ενεργοποιείται μόνον όταν όλες οι έξοδοι έχουν καταλήξει.

### 3.2.2 Πολυεπίπεδη Υλοποίηση του Επιπρόσθετου Δικτύου

Η πολυεπίπεδη προσέγγιση παρακάμπτει την διεπίπεδη τεχνική των δυαδικών συσχετίσεων και κάνει χρήση μόνο χρήση της σύνθλιψης και της πολυεπίπεδης βελτιστοποίησης, όπως απεικονίζεται στην ροή του Σχήματος 3.9. Με τον τρό-



Σχήμα 3.9: Η Ροή Υλοποίησης Ενός Επιπρόσθετου Δικτύου Boole

πο αυτό δεν είναι απαραίτητη η διάσπαση του αρχικού δικτύου Boole σε επίπεδα (αλλά εξακολουθεί να υπάρχει αυτή η δυνατότητα) και εξαλείφεται ο υπολογιστικός φόρτος από το σχηματισμό και την επίλυση των δυαδικών συσχετίσεων. Η συγκεκριμένη υλοποίηση είναι απόρροια αυτού ακριβώς του υπολογιστικού φόρτου αφού καθιστούσε την διεπίπεδη προσέγγιση αναποτελεσματική για πρακτικά δίκτυα.

Η σύνθλιψη απαλείφει τους εσωτερικούς κόμβους ανάμεσα στο αντίγραφο

του αρχικού δικτύου Boole και της συνάρτησης Sync, μεταξύ των οποίων και οι έξοδοι του αρχικού δικτύου Boole. Έπειτα, οι αλγόριθμοι βελτιστοποίησης επιχειρούν να φέρουν το δίκτυο σε μία απεικόνιση κοντά σε αυτήν που θα επιτυγχανόταν με χρήση δυαδικών συσχετίσεων, με τη διαφορά ότι οι αλγόριθμοι αυτοί είναι ευριστικοί σε αντίθεση με την επακριβή μέθοδο επίλυσης των συσχετίσεων, κάτι που τους κάνει πολύ πιο ταχείς και όχι εκθετικούς σε πολυπλοκότητα.

### 3.2.3 Ομαδοποίηση Εξόδων

#### 3.2.3.1 Μεθοδολογία

Ένα κοινό πρόβλημα που αντιμετωπίζουν και οι δύο προσεγγίσεις που αναφέραμε, είναι όταν χρειάζεται να χειριστούν δίκτυα ή υποδίκτυα με μεγάλο αριθμό εξόδων. Για τη μεν διεπίπεδη προσέγγιση αυτό σημαίνει πολύπλοκες δυαδικές συσχετίσεις, [8], για τη δε πολυεπίπεδη σημαίνει μεγαλύτερα δίκτυα, πιο πολύπλοκη συνάρτηση Sync, κάτι που καθιστά την σύνθλιψη υπολογιστικά ακριβή<sup>2</sup>. Επίσης, όπως θα δούμε στο επόμενο κεφάλαιο, το κόστος για την αναγνώριση ολοκλήρωσης αυξάνεται ραγδαία όταν αυξάνονται οι έξοδοι του κυκλώματος στο οποίο εφαρμόζεται. Για το σκοπό αυτό είναι επιθυμητή η ομαδοποίηση των εξόδων δικτύων ή υποδικτύων σε ομάδες αρκούντως μικρές ώστε να είναι εφικτή η υλοποίηση. Κάθε ομάδα θα έχει τη δική της αναγνώριση ολοκλήρωσης και όλες μαζί θα οδηγούν ένα κοινό σήμα ολοκλήρωσης του κυκλώματος ή υποκυκλώματος. Όπως γίνεται αντιληπτό, ο βαθμός βελτιστοποίησης του επιπρόσθετου δικτύου Boole ενδέχεται να είναι μικρότερος, λόγω του ότι θα υπάρχουν λιγότερες εξισώσεις ανά ομάδες που να μπορούν να ανταλλάξουν μεταξύ τους κύβους και να μοιραστούν κοινούς όρους. Συνεπώς, η ιδανική ομαδοποίηση είναι αυτή στην οποία οι εξισώσεις μεταξύ διαφορετικών ομάδων δε θα έχουν καμία ή πολύ λίγες κοινές μεταβλητές, ελαχιστοποιώντας την πιθανότητα να υπάρχουν μεταξύ τους κοινοί όροι. Αν μεγάλος αριθμός εξισώσεων έχει πολλές κοινές μεταβλητές τότε θα πρέπει με τεχνητό, και σε μεγάλο βαθμό τυχαίο, τρόπο να διασπαστούν

---

<sup>2</sup>Η σύνθλιψη ως λογική πράξη είναι υπολογιστικά ακριβή σε πολύπλοκα δίκτυα γιατί πρέπει να αντικατασταθούν όλοι οι ενδιάμεσοι κόμβοι ενός δικτύου και να γίνουν όλες οι λογικές πράξεις που θα οδηγήσουν σε μία διεπίπεδη έκφραση του δικτύου

σε μικρότερες. Η υλοποίηση της ομαδοποίησης εξόδων, κάνει χρήση ενός αλγορίθμου Κάλυψης Ορθογωνίων (Rectangle Covering) όπως αναφέρεται στην επόμενη ενότητα όπου δίνεται και ένα παράδειγμα.

### 3.2.3.2 Υλοποίηση

Θεωρούμε το παρακάτω δίκτυο:

$$\begin{aligned}
 o0 &= h' + i \\
 o1 &= h \cdot i + h' \cdot i' \\
 o2 &= d \cdot e + d' \cdot e' \\
 o3 &= d + e' \\
 o4 &= h + i' \\
 o5 &= d' + e
 \end{aligned} \tag{3.4}$$

Το επιπρόσθετο δίκτυο Boole που αντιστοιχεί στο Δίκτυο 3.4, με χρήση του κώδικα Berger, είναι το παρακάτω:

$$\begin{aligned}
 sync0 &= (d' \cdot e + d \cdot e') \cdot (h' \cdot i + h \cdot i') \\
 sync1 &= (d' \cdot e + d \cdot e') \cdot (h' \cdot i' + h \cdot i) + (h' \cdot i + h \cdot i') \cdot (d' \cdot e' + d \cdot e) \\
 sync2 &= 0
 \end{aligned} \tag{3.5}$$

Το διάνυσμα  $\{sync2, sync1, sync0\}$  είναι η έξοδος του επιπρόσθετου δικτύου Boole, που δίνει τον αριθμό των 0 της κυριολεκτικής εξόδου του Δικτύου 3.4.

Η ανάμιξη εξισώσεων με διαφορετικές μεταξύ τους μεταβλητές δεν επιτρέπει τη μέγιστη βελτιστοποίηση και δημιουργεί προβλήματα πολυπλοκότητας στο επιπρόσθετο δίκτυο Boole. Αυτό συμβαίνει γιατί κάθε ψηφίο του διανύσματος Sync εξαρτάται και από τις 4 μεταβλητές  $d, e, h, i$ , χωρίς όμως να υπάρχει η ελευθερία για απλοποίηση δεδομένου ότι υπάρχουν μονάχα εξισώσεις όπου χρησιμοποιούνται είτε οι  $d, e$ , είτε οι  $h, i$ . Οι εξισώσεις αυτές δεν είναι δυνατό να ανταλλάξουν κοινούς όρους, οδηγώντας σε πολύπλοκα δίκτυα με μεγάλο αριθμό όρων.

Στην περίπτωση του Δικτύου 3.4 είναι επιθυμητό να ομαδοποιηθούν οι εξισώσεις  $o0, o1, o4$  και οι  $o2, o3, o5$ . Αυτό είναι δυνατό να εξαχθεί με έναν αλγόριθμο κάλυψης ορθογωνίων. Πρώτα σχηματίζεται ένας διδιάστατος πίνακας που έχει στον οριζόντιο άξονα τις μεταβλητές του δικτύου και στον κατακόρυφο άξονα τις εξισώσεις του δικτύου. Ένα στοιχείο αυτού του πίνακα  $(i, j)$  θα έχει την τιμή 1 όταν η μεταβλητή  $j$  συμμετέχει στην εξίσωση  $i$ . Ο πίνακας του Δικτύου 3.4 παραθέτεται στο Πίνακα 3.4. Το κίτρινο και το κυανό ορθογώνιο του Πίνακα

	h	d	e	i
o0	1	-	-	1
o1	1	-	-	1
o2	-	1	1	-
o3	-	1	1	-
o4	1	-	-	1
o5	-	1	1	-

Πίνακας 3.4: Κάλυψη Ορθογωνίων για το Δίκτυο 3.4

3.4 είναι το αποτέλεσμα του αλγορίθμου κάλυψης ορθογωνίων. Στον ίδιο πίνακα φαίνεται επίσης ότι η έννοια του ορθογωνίου στον αλγόριθμο ορίζεται είτε ως ένα σύνολο γραμμών που έχουν τις ίδιες στήλες, είτε εναλλακτικά ως ένα σύνολο στηλών που έχουν τις ίδιες γραμμές.

Τα δύο ξεχωριστά δίκτυα που προκύπτουν μαζί με τα αντίστοιχα επιπρόσθετα δίκτυα Boole είναι τα 3.6 και 3.7, όπου είναι προφανές ότι και η πολυπλοκότητα και τα κόστη είναι πολύ μικρότερα.

$$o0 = h' + i$$

$$o1 = h \cdot i + h' \cdot i'$$

$$o4 = h + i' \tag{3.6}$$

$$sync0 = h' \cdot i + h \cdot i'$$

$$sync1 = 0$$

$$\begin{aligned}
o2 &= d \cdot e + d' \cdot e' \\
o3 &= d + e' \\
o5 &= d' + e \\
sync0 &= d' \cdot e + d \cdot e' \\
sync1 &= 0
\end{aligned} \tag{3.7}$$

Συνοπτικά, ο αλγόριθμος κάλυψης ορθογωνίων, [14], παίρνει τη μεγαλύτερη γραμμή ενός πίνακα και βρίσκει το μεγαλύτερο δυνατό τετράγωνο, με μέτρο τον αριθμό των γραμμών επί των στηλών, στο οποίο αυτή ανήκει. Το ίδιο κάνει και με τη μεγαλύτερη στήλη. Το μεγαλύτερο ορθογώνιο από αυτά είναι αυτό που επιστρέφει ο αλγόριθμος, ενώ υπάρχουν και κάποια ευριστικά τα οποία μπορούν ανιχνεύσουν μεγαλύτερα ορθογώνια στα οποία δεν ανήκουν οι μεγαλύτερες στήλες ή γραμμές. Όταν αποφασιστεί ένα ορθογώνιο οι εξισώσεις που αντιστοιχούν σε αυτό σχηματίζουν μια ξεχωριστή ομάδα και ο αλγόριθμος συνεχίζει με τις υπόλοιπες.

### 3.3 Μονοτονία Δικτύων Boole

Στα σχήματα της Ενότητας 3.1, όπου παραθέεται η λειτουργία του συνδυαστικού κυκλώματος με αναγνώριση ολοκλήρωσης, αναφέρεται ότι όλα τα υποκυκλώματα έχουν μονότονη συμπεριφορά. Στην Ενότητα 2.2.1 αναλύθηκε πως μετασχηματίζεται μία λογική εξίσωση ώστε να έχει μονότονη συμπεριφορά. Για να είναι συνεπώς ένα δίκτυο Boole μονότονο, θα πρέπει όλες οι λογικές εξισώσεις που το απαρτίζουν να είναι μονότονες. Θεωρούμε το παρακάτω δίκτυο Boole, όπου τα  $a$ ,  $b$ ,  $c$ ,  $d$ ,  $e$  και  $o$  είναι οι πρωτεύουσες είσοδοι, τα  $f$  και  $g$  οι πρωτεύουσες έξοδοι και τα  $s0$  και  $s1$  ενδιάμεσοι κόμβοι του δικτύου:

$$\begin{aligned}
f &= s0 \cdot a + s0' \cdot a' \\
g &= s1 \cdot d + s1' \cdot d'
\end{aligned} \tag{3.8}$$

$$s0 = b \cdot c + b' \cdot c'$$

$$s1 = e \cdot o' + e' \cdot o$$

Το Δίκτυο 3.8 δεν είναι μονότονο καθώς πλήθος μεταβλητών εμφανίζονται με δύο πρόσημα, δηλαδή είναι αμφίσιμες. Με βάση την ανάλυση της Ενότητας 2.2.1, όπου αναλύθηκε ο μετασχηματισμός για μονοτονία, θα πρέπει να χρησιμοποιήσουμε δύο νέες μεταβλητές για κάθε πρωτεύουσα είσοδο. Όμως οι ενδιάμεσοι κόμβοι του δικτύου,  $s0$  και  $s1$ , χρησιμοποιούνται και αυτοί και με τις δύο μορφές τους<sup>3</sup>, κάνοντας απαραίτητη την ύπαρξη της άρνησης των εσωτερικών κόμβων. Για να παρακαμφθεί το πρόβλημα αυτό, εκφράζουμε όλες τις εξόδους μόνο ως προς τις πρωτεύουσες εισόδους του δικτύου, συνθλιβοντάς το. Συνθλιβοντάς λοιπόν το Δίκτυο 3.8 παίρνουμε το Δίκτυο 3.9.

$$f = a \cdot b \cdot c + a \cdot b' \cdot c' + a' \cdot b \cdot c' + a' \cdot b' \cdot c \quad (3.9)$$

$$g = d \cdot e \cdot o' + d \cdot e' \cdot o + d' \cdot e \cdot o + d' \cdot e' \cdot o'$$

Το δίκτυο 3.9 μπορεί πλέον να αποκτήσει μονότονη συμπεριφορά, μετασχηματιζόμενο στο 3.10.

$$f = a_+ \cdot b_+ \cdot c_+ + a_+ \cdot b_- \cdot c_- + a_- \cdot b_+ \cdot c_- + a_- \cdot b_- \cdot c_+ \quad (3.10)$$

$$g = d_+ \cdot e_+ \cdot o_- + d_+ \cdot e_- \cdot o_+ + d_- \cdot e_+ \cdot o_+ + d_- \cdot e_- \cdot o_-$$

Τέλος το δίκτυο 3.11 μπορεί να απεικονισθεί σε πολυεπίπεδο, όπως ήταν και το αρχικό δίκτυο, για φθηνότερη, από πλευράς όρων υλοποίησης. Με χρήση κλασικών αλγορίθμων βελτιστοποίησης, προκύπτει το δίκτυο 3.11, με εσωτερικούς κόμβους τους  $s2$ ,  $s3$ ,  $s4$  και  $s5$ .

$$f = s2 \cdot a_- + s3 \cdot a_+$$

$$g = s4 \cdot d_- + s5 \cdot d_+$$

$$s2 = b_- \cdot c_+ + b_+ \cdot c_- \quad (3.11)$$

$$s3 = b_- \cdot c_- + b_+ \cdot c_+$$

<sup>3</sup> Αν χρησιμοποιούνταν μόνο με την κανονική τους μορφή τότε δεν θα υπήρχε λόγος περαιτέρω επεξεργασίας.

$$s4 = e_- \cdot o_- + e_+ \cdot o_+$$

$$s5 = e_- \cdot o_+ + e_+ \cdot o_-$$

Στο κεφάλαιο αυτό, παρουσιάστηκε αναλυτικά η λειτουργία ενός συνδυαστικού κυκλώματος με αναγνώριση ολοκλήρωσης κάνοντας χρήση του D1 κώδικα Berger και του συστηματικού κώδικα Sperner. Ακόμα, δόθηκαν οι λεπτομέρειες για την υλοποίηση του επιπρόσθετου δικτύου Boole που είναι το κλειδί για την λειτουργία του κυκλώματος, αλλά και για τον μετασχηματισμό των δικτύων Boole σε μονότονα. Στο κεφάλαιο που ακολουθεί, παραθέτονται τα αποτελέσματα από τη χρήση της μεθοδολογίας που έχει περιγραφεί, σε μία ομάδα από γνωστά κυκλώματα. Επίσης, γίνεται σύγκριση με τον μετασχηματισμό της διπλής-ράγας, που αποτελεί μία από τις επικρατέστερες τεχνικές στην κατασκευή D1 κυκλωμάτων.

## Κεφάλαιο 4

# Αποτελέσματα - Κόστος Υλοποίησης

Στο κεφάλαιο αυτό παρουσιάζουμε την ποσοτική και ποιοτική ανάλυση των μεθοδολογιών που έχουμε εισάγει μέχρι στιγμής. Η όλη πειραματική διαδικασία έλαβε μέρος στο ακαδημαϊκό πρόγραμμα λογικής σύνθεσης SIS [13]. Σε αυτό ενσωματώθηκαν ρουτίνες και αλγόριθμοι περιγεγραμμένοι σε γλώσσα προγραμματισμού C οι οποίοι είναι τα εργαλεία για την υλοποίηση της τεχνικής που παρουσιάστηκε. Τα εργαλεία αυτά επιδρούν πάνω στη δομή του κυκλώματος που είναι αποθηκευμένη μέσα στο SIS, ώστε να επιτευχθούν οι απαραίτητες αλλαγές και μετασχηματισμοί. Συνοπτικά, στις ενότητες που ακολουθούν θα αναφερθούμε στις παρακάτω υλοποιήσεις:

- Αναγνώριση ολοκλήρωσης.
- Διεπίπεδη προσέγγιση υλοποίησης
- Πολυεπίπεδη προσέγγιση υλοποίησης

Οι λεπτομέρειες για τη πειραματική ροή των όσων υλοποιήθηκαν αναφέρονται ξεχωριστά σε κάθε ενότητα.

Τέλος, στα αποτελέσματα που ακολουθούν υπάρχουν μετρήσεις που αναφέρονται είτε στο εμβαδό είτε στην κατανάλωση ισχύος των κυκλωμάτων. Τόσο στο περιβάλλον του SIS, αλλά και γενικότερα στο πεδίο της λογικής σύνθεσης

κυκλωμάτων, το εμβασμό συνδέεται με τον αριθμό των όρων που χρειάζεται για να περιγραφεί το κύκλωμα, ο οποίος αποτελεί μία ευριστική ένδειξη. Κάτι ανάλογο ισχύει και για τη κατανάλωση ισχύος, όπου ο ευριστικός γνώμονας είναι ο αριθμός των μεταβάσεων που κάνουν τα σήματα που το απαρτίζουν, κατά τη λειτουργία του.

## 4.1 Αναγνώριση Ολοκλήρωσης

### 4.1.1 Αναγνώριση Ολοκλήρωσης με Βάση την Απαρίθμηση, ED

Στην Ενότητα 2.2.3, γίνεται αναφορά στη συνάρτηση του ED με την οποία επιτυγχάνεται η αναγνώριση ολοκλήρωσης χρησιμοποιώντας μία AND πύλη για κάθε κωδική λέξη και μία OR, η οποία ενώνει τις εξόδους των AND. Ο Πίνακας 4.1 παραθέτει τα κόστη σε όρους της συνάρτησης ED για αριθμό ψηφίων κυριολεκτικών δεδομένων από 1 έως 10 και χρήση του DI κώδικα Berger. Οι συναρτήσεις περιγράφηκαν σε διεπίπεδη λογική και βελτιστοποιήθηκαν με χρήση πολυεπίπεδων αλγορίθμων βελτιστοποίησης, οι οποίοι είναι ήδη υλοποιημένοι στο SIS. Η διεπίπεδη περιγραφή, υλοποιήθηκε ως μία αυτοματοποιημένη διαδικασία, στην οποία παράγονται όλες οι λέξεις του εκάστοτε κώδικα και για τον αριθμό ψηφίων που ζητείται, και από αυτές σχηματίζονται οι πύλες AND που συμμετέχουν στον ED. Τα αντίστοιχα κόστη όταν γίνεται χρήση του συστηματικού DI κώδικα Sperner, παρατίθενται στον Πίνακα 4.2.

Από τους Πίνακες 4.1 και 4.2 φαίνεται ότι το κόστος αυξάνεται με την αύξηση των ψηφίων των κυριολεκτικών δεδομένων, μάλιστα για αριθμό ψηφίων μεγαλύτερο του 9 αυξάνεται σε πολύ μεγάλο βαθμό<sup>1</sup>, κάτι που ενισχύει την ανάγκη για την ομαδοποίηση των εξόδων που αναπτύχθηκε στο προηγούμενο κεφάλαιο. Επίσης, όταν γίνεται χρήση του συστηματικού DI κώδικα Sperner το κόστος είναι μεγαλύτερο, λόγω των περισσότερων ψηφίων που απαιτεί η συνάρτηση Sync για τον κώδικα αυτό.

---

<sup>1</sup>πιθανώς αυτό να οφείλεται και εν μέρει στην πολυπλοκότητα και την αδυναμία των ευριστικών αλγορίθμων του SIS να οδηγήσουν το δίκτυο σε μία καλύτερη υλοποίηση.

ψηφία κυριολεκτικών δεδομένων	κόστος ED σε εμβαδό
1	2
2	7
3	14
4	26
5	39
6	52
7	66
8	84
9	100
10	3526

Πίνακας 4.1: Κόστος ED Κώδικα Berger

ψηφία κυριολεκτικών δεδομένων	κόστος ED σε εμβαδό
1	2
2	7
3	17
4	29
5	45
6	75
7	92
8	140
9	176
10	3532

Πίνακας 4.2: Κόστος ED Συστηματικού Κώδικα Sperner

#### 4.1.2 Αναγνώριση Ολοκλήρωσης Βάση Σύγκρισης

Με το συστηματικό κώδικα Sperner μπορεί να γίνει χρήση και ενός άλλου τρόπου αναγνώρισης ολοκλήρωσης, αυτού που είδαμε στη διάταξη του Σχήματος 2.7, ο οποίος δημιουργεί το  $Syn_c$  εκ νέου από τα ανακτηθείσα κυριολεκτικά δεδομένα εξόδου και το συγκρίνει με αυτό που προέρχεται από το επιπρόσθετο δίκτυο Boole. Ως κόστος της αναγνώρισης ολοκλήρωσης θεωρούμε το κόστος της συνάρτησης  $Syn_c$  και του συγκριτή, καθώς στα όρια αυτού του υποκυκλώματος εμφανίζεται η υποψήφια κωδική λέξη. Ο συγκριτής στο Σχήμα 2.7 δεν είναι τίποτα άλλο από τόσες πύλες XNOR 2 εισόδων όσα και τα ψηφία των συναρτήσεων  $Syn_c$  που συγκρίνονται, καθώς και ένα δέντρο από πύλες AND που συνδυάζουν της εξόδους των πυλών XNOR και δίνουν το τελικό σήμα της αναγνώρισης ολοκλήρωσης, καθότι θα πρέπει όλα τα αντίστοιχα ψηφία των δύο εκδοχών του  $Syn_c$  να είναι ίσα μεταξύ τους. Η δύο εισοδοί της XNOR είναι ένα ψηφίο του επιπρόσθετου δικτύου Boole, δηλαδή του  $Syn_c$ , του κυκλώματος και το αντίστοιχης τάξης ψηφίο από το  $Syn_c$  που παράγεται από την κυριολεκτική έξοδο του κυκλώματος.

#### Ασφαλής Χρήση Πύλης XNOR

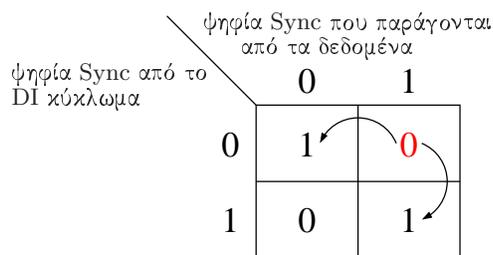
Οι πύλες XNOR, όπως και οι XOR, είναι θέλουν προσεκτικό χειρισμό όταν χρησιμοποιούνται σε DI κυκλώματα, γιατί προκαλούν παράνομες μεταβάσεις στις εξόδους τους, λ.χ. αύξηση στην μία είσοδό τους συνεπάγεται είτε αύξηση είτε μείωση στην έξοδό τους ανάλογα με την τιμή και το χρόνο άφιξης της άλλης εισόδου. Στην περίπτωση του συγκριτή που χρησιμοποιούμε, και λόγω του πρωτοκόλλου της διασικλής λειτουργίας, δείχνουμε ότι η πύλη μπορεί να χρησιμοποιηθεί με ασφάλεια. Το γιατί φαίνεται γραφικά στο Σχήμα 4.1. Στην κατάσταση της αρχικοποίησης, και με βάση το Σχήμα 3.5, όλες οι πύλες XNOR του συγκριτή θα έχουν έξοδο 0 η οποία στο Σχήμα 4.1 αντιστοιχεί στο πάνω δεξιά 0 του πίνακα αληθείας. Στη φάση υπολογισμού, είτε το εκάστοτε ψηφίο του  $Syn_c$  που παράγεται από την κυριολεκτική έξοδο του κυκλώματος θα μεταβεί από το 1 στο 0 παρασύροντας την έξοδο της XNOR κατά το αριστερό βελάκι του σχήματος

4.1, είτε το αντίστοιχό του που προέρχεται από το επιπρόσθετο δίκτυο Boole θα μεταβεί από το 0 στο 1 παρασύροντας την έξοδο της XNOR στο 1 κατά το δεξί βελάκι του Σχήματος 4.1. Και στις δύο περιπτώσεις η πύλη XNOR έχει μονότονα αύξουσα συμπεριφορά κατά τη φάση υπολογισμού. Συνεπώς, μπορεί να χρησιμοποιηθεί χωρίς να προκαλεί ανεπιθύμητα σκαμπανεβάσματα (hazards).

Στον Πίνακα 4.3 φαίνονται τα κόστη για τη συνάρτηση Sync, ο αριθμός των πυλών XNOR του συγκριτή, και μία εκτίμηση για τις πύλες AND που χρειάζονται για την δημιουργία του σήματος αναγνώρισης ολοκλήρωσης θεωρώντας ότι χρησιμοποιούνται πύλες AND δύο εισόδων. Για τις πύλες XNOR και AND θεωρούμε ότι προσθέτουν κόστος 2 όρων. Στις παρενθέσεις του Πίνακα 4.3 παραθέτονται τα εκτιμώμενα κόστη σε όρους. Αυτό που μπορεί να διαπιστωθεί συγκρίνοντας τους Πίνακες 4.2 και 4.3, είναι ότι για αριθμό ψηφίων κυριολεκτικών δεδομένων μεγαλύτερο του 7, η υλοποίηση της αναγνώρισης ολοκλήρωσης με τη χρήση συγκριτή είναι πιο συμφέρουσα.

#### 4.1.3 Μερική Αναγνώριση Ολοκλήρωσης

Οι παραπάνω πίνακες αφορούν κόστη χωρίς να εκμεταλλευόμαστε τη πληροφορία των εξόδων που αναμένουμε από το κύκλωμα. Για παράδειγμα, στην περίπτωση που γίνεται χρήση του ED μπορεί κάποια κωδική λέξη να μη χρησιμοποιείται ποτέ, συνεπώς δεν υπάρχει λόγος να υπάρχει η πύλη AND για αυτήν. Ομοίως, για τη δεύτερη τεχνική αναγνώρισης ολοκλήρωσης, μπορεί κάποιο από τα ψηφία του Sync να μην παίρνει ποτέ ή να παίρνει μονίμως την τιμή 1 και ενδεχομένως



Σχήμα 4.1: Μονότονη Χρήση της Πύλης XNOR

ψηφία κυριολεκτικών δεδομένων	κόστος Sync	πύλες XNOR	AND 2 εισόδων
1	1	1 (2)	0
2	4	2 (4)	1 (2)
3	12	3 (6)	2 (4)
4	22	4 (8)	3 (6)
5	33	5 (10)	4 (8)
6	48	6 (12)	5 (10)
7	65	7 (14)	6 (12)
8	82	8 (16)	7 (14)
9	102	9 (18)	8 (16)
10	124	10 (20)	9 (18)

Πίνακας 4.3: Κόστος Αναγνώρισης Ολοκλήρωσης με τη Χρήση Συγκριτή

να μη χρειάζεται να υλοποιηθεί. Η απαιτούμενη διασφάλιση είναι για όλες τις πιθανές εξόδους να είναι δυνατή η αναγνώριση ολοκλήρωσης.

Η διάγνωση της πιθανότητας ελάττωσης του κόστους μπορεί να γίνει με πλήρη προσομοίωση ή σύνθλιψη, κάτι που το καθιστά ακριβό. Για παράδειγμα, το Δίκτυο 3.1 στη Σελίδα 33 μπορεί να πάρει μόνο τις τιμές 011, 101, 110 και 111 στην έξοδό του. Συνεπώς, η συνάρτηση για την αναγνώριση ολοκλήρωσης αρκεί να έχει μόνο 4 πύλες<sup>2</sup> AND, όσες και οι χρησιμοποιούμενες κωδικές λέξεις {011 01, 101 01, 110 01, 111 00}, με χρήση του κώδικα Berger, μειώνοντας το κόστος από 14, με βάση τον Πίνακα 4.1, σε 11 όρους.

Εκτός από την προσομοίωση, η διάγνωση για την ελάττωση του κόστους της αναγνώρισης ολοκλήρωσης μπορεί να γίνει και εξετάζοντας το επιπρόσθετο δίκτυο Boole που προκύπτει. Για παράδειγμα το Δίκτυο 4.1:

$$sync0 = 0$$

$$sync1 = 0 \tag{4.1}$$

$$sync2 = (a' + b') \cdot c' \cdot d'$$

<sup>2</sup>αντί για 8 όσες είναι και οι κωδικές λέξεις για δεδομένα των 3 ψηφίων

είναι το επιπρόσθετο δίκτυο Boole του Δικτύου 3.1, με βάση το συστηματικό κώδικα Sperner. Μπορούμε να διαγνώσουμε από αυτό ότι 2 ψηφία της κυριολεκτικής εξόδου θα έχουν σίγουρα την τιμή 1. Συνεπώς, το κύκλωμα θα έχει ολοκληρώσει τη φάση υπολογισμού, είτε όταν έχουν σταθεροποιηθεί 2 ψηφία, είτε όταν έχουν σταθεροποιηθεί 3 ψηφία στην τιμή 1. Υλοποιώντας την αναγνώριση ολοκλήρωσης με χρήση συγκριτή, συνεπάγεται 2 ή 3 μεταβάσεις της εξόδου Sync, που παράγεται από την κυριολεκτική έξοδο του κυκλώματος, από το 1 στο 0. Η μία από αυτές τις μεταβάσεις, που αντιστοιχεί στη σταθεροποίηση ενός 1 στα δεδομένα και θα συμβαίνει πάντα, δε χρειάζεται να ανιχνευθεί. Αυτό οδηγεί στη μείωση του κόστους του Sync από 12, με βάση τον Πίνακα 4.3, σε 9.

Και τα δύο είδη βελτιστοποιήσεων μπορούν να εφαρμοστούν σε οποιοδήποτε κώδικα και οποιαδήποτε τεχνική αναγνώρισης ολοκλήρωσης από αυτές που παρουσιάστηκαν. Τέλος, η εφαρμογή της μερικής αναγνώρισης ολοκλήρωσης δεν υλοποιήθηκε ώστε να προσφέρεται αυτοματοποιημένα μέσα από το SIS, όμως χρησιμοποιήθηκε με ξεχωριστή επέμβαση σε κάθε κύκλωμα το οποίο παράχθηκε για να συμπεριληφθεί στα αποτελέσματα που θα ακολουθήσουν.

## 4.2 Κυκλώματα Διεπίπεδης Προσέγγισης Υλοποίησης του Επιπρόσθετου Δικτύου Boole

Στη διεπίπεδη προσέγγιση υλοποίησης σχηματισμός και επίλυση των δυαδικών συσχετίσεων με βάση τα όσα αναπτύχθηκαν στην Ενότητα 3.2.1. Στον Πίνακα 4.4 που ακολουθεί παρουσιάζονται τα αποτελέσματα από την χρήση των δυαδικών συσχετίσεων σε κάποια από τα δειγματοληπτικά κυκλώματα του IWLS. Η διαδικασία σχηματισμού και επίλυσης των δυαδικών συσχετίσεων υλοποιήθηκε και ενσωματώθηκε στο περιβάλλον του SIS. Η ροή που ακολουθείται για κάθε κύκλωμα είναι η ακόλουθη:

- Επεξεργάζεται η δομή του κυκλώματος που βρίσκεται στο SIS. Η επεξεργασία αυτή αφορά στη διάσπαση του κυκλώματος στα λογικά επίπεδα που αποτελούν και στην ομαδοποίηση των εξόδων για κάθε λογικό επίπεδο, με χρήση του αλγόριθμου κάλυψης ορθογωνίων.

- Έπειτα, για κάθε ομάδα που έχει προκύψει γίνεται πλήρης προσομοίωση μέσα στο SIS και γίνεται καταγραφή του πίνακα αληθείας για την ομάδα εξισώσεων.
- Προσθέτοντας στον πίνακα αληθείας τις πιθανές εξόδους που αντιστοιχούν σε κάθε ελαχιστόρο, όπως δείξαμε στην ενότητα των δυαδικών συσχετίσεων, φτιάχνουμε την περιγραφή των συσχετίσεων για κάθε ομάδα.
- Ένας επακριβής επιλυτής δυαδικών συσχετίσεων που ενσωματώθηκε στο SIS, παίρνει τις περιγραφές των συσχετίσεων και παράγει την λύση για κάθε ομάδα εξισώσεων.

Τόσο η διάσπαση στα λογικά επίπεδα και η ομαδοποίηση εξόδων, όσο και ο επακριβής επιλυτής των δυαδικών συσχετίσεων αποτελούν ξεχωριστές υλοποιήσεις που δεν περιλαμβάνονταν στο περιβάλλον του SIS. Ο λόγος για τον οποίο η δημιουργία των ομάδων και η επίλυση των δυαδικών τους συσχετίσεων γίνεται ξεχωριστά και όχι σε ένα βήμα, είναι το ότι ο επακριβής επιλυτής δεν έδινε πάντα αποτέλεσμα και αυτό απεικονίζεται στον Πίνακα 4.4, που ακολουθεί, με μία '-' στο πεδίο του αποτελέσματος της δυαδικής συσχέτισης. Τα πεδία του Πίνακα 4.4 είναι κατά σειρά:

- Το όνομα του κυκλώματος στο οποίο ανήκουν οι δυαδικές συσχετίσεις (κύκλωμα).
- Το λογικό επίπεδο στο οποίο αντιστοιχούν οι δυαδικές συσχετίσεις (λογικό επίπεδο). Όπως παρουσιάστηκε στην Ενότητα 3.2.1.2, η διάσπαση ενός κυκλώματος σε επίπεδα παράγει ευκολότερα επιλύσιμες δυαδικές συσχετίσεις. Το λογικό επίπεδο με αύξοντα αριθμό 1 είναι αυτό που έχει ως εισόδους μόνον πρωτεύουσες εισόδους του κυκλώματος.
- Ο αύξων αριθμός των ομάδων που σχηματίζονται, με βάση τα όσα αναφέρθηκαν στην Ενότητα 3.2.3 για την ομαδοποίηση εξόδων, και στις οποίες αντιστοιχούν οι δυαδικές συσχετίσεις που βρίσκονται στην ίδια γραμμή του Πίνακα (ομάδες εξόδων).

- Το κόστος, σε δυαδικούς όρους, της ομάδας από την οποία προκύπτει η δυαδική συσχέτιση (αρχικό κόστος δικτύου).
- Το κόστος της λύσης της δυαδικής συσχέτισης (λύση δυαδικής συσχέτισης).

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης	
9symml	1	1	6	6	
		2	7	7	
		3	4	4	
		4	8	8	
		5	4	4	
		6	4	4	
		7	4	4	
	2	2	1	9	9
			2	5	5
			3	7	7
4			6	6	
5			3	3	
6			6	6	
7			8	8	
8			2	2	
9			5	5	
10			2	2	
3	3	1	15	-	
		2	15	-	
		3	12	-	
		4	4	4	
		5	4	4	
		6	8	8	
4	4	1	17	-	
		2	10	-	
		3	19	-	
5	5	1	26	-	
		2	15	-	
		3	6	6	
6	6	1	8	8	
		7	16	16	
alu2	1	1	8	0	
		2	9	5	
		3	4	1	
		4	2	2	
		5	8	2	
		6	4	4	
	2	2	1	7	7
			2	4	4
			3	6	6
			4	4	0
		5	3	3	

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης
alu2	<b>2</b>	6	5	5
		7	5	3
		8	2	2
		9	7	7
		10	4	4
		11	7	7
	<b>3</b>	1	5	5
		2	9	-
		3	4	4
		4	4	4
		5	8	8
		6	9	9
	<b>4</b>	1	12	-
		2	5	5
	<b>5</b>	1	16	-
	<b>6</b>	1	13	-
		2	4	0
		3	3	3
	<b>7</b>	1	21	-
		2	5	5
		3	4	4
		4	12	-
	<b>8</b>	1	18	-
		2	9	9
		3	7	7
	<b>9</b>	1	22	-
		2	8	8
		3	10	10
	<b>10</b>	1	10	2
		2	4	4
		3	7	7
		4	8	8
	<b>11</b>	1	22	-
		2	8	8
		3	4	4
		4	8	8
		5	4	4
	<b>12</b>	1	6	6
		2	19	19
	<b>13</b>	1	12	12
	<b>14</b>	1	17	17
	<b>15</b>	1	14	-

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης	
alu2	<b>16</b>	1	23	-	
b9	<b>1</b>	1	7	5	
		2	7	7	
		3	6	6	
		4	5	3	
		5	5	5	
		6	9	9	
		<b>2</b>	1	6	6
		2	5	5	
		3	6	6	
		4	5	5	
		5	4	4	
		6	13	13	
		<b>3</b>	1	11	-
		2	8	8	
		3	4	0	
		4	13	13	
<b>4</b>	1	9	-		
2	8	8			
<b>5</b>	1	6	6		
cm42	<b>1</b>	1	4	0	
		2	2	2	
		<b>2</b>	1	9	3
		2	9	3	
		3	6	2	
4	4	4			
cm82	<b>1</b>	1	6	2	
		2	2	2	
		<b>2</b>	1	8	8
		<b>3</b>	1	8	8
<b>4</b>	1	4	4		
cm85	<b>1</b>	1	8	4	
		2	8	4	
		3	8	8	
		<b>2</b>	1	10	10
		2	10	10	
		3	3	3	
		<b>3</b>	1	6	6
		2	3	3	
		<b>4</b>	1	6	6
cm138	<b>1</b>	1	3	3	
		<b>2</b>	1	4	0

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης
cm138	3	1	12	0
		2	12	0
cm150	1	1	32	-
		2	16	16
		3	8	8
		4	5	5
cm151	1	1	8	8
		2	8	8
		2	8	8
		3	4	4
		4	2	2
		5	1	1
cm152	1	1	8	8
		2	8	8
		2	8	8
		3	4	4
cm162	1	1	6	6
		2	5	5
		3	6	2
		4	4	4
		2	4	4
		3	10	10
		4	5	5
		5	10	10
		6	2	2
		7	2	2
cm163	1	1	7	7
		2	4	4
		2	7	6
		2	4	4
		3	4	4
		3	7	6
		2	2	2
		4	7	6
		2	2	2
		5	6	6
		6	2	2
		7	2	2
		cmb	1	1
2	5			5
2	14			-

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης
cmb	3	1	2	2
	4	1	4	4
comp	1	1	6	6
		2	6	6
		3	6	6
		4	6	6
		5	48	48
	2	1	34	34
	3	1	16	16
	4	1	16	16
	5	1	13	-
	6	1	3	3
	7	1	3	3
8	1	2	2	
C17	1	1	2	2
		2	2	2
	2	1	6	6
cordic	1	1	6	6
		2	10	10
		3	28	28
	2	1	13	13
	3	1	16	-
	4	1	6	6
cu	1	1	8	8
		2	8	8
		3	8	8
		4	2	2
	2	1	12	0
		2	13	-
		3	5	5
	3	1	7	7
4	1	3	3	
frg1	1	1	13	13
		2	9	9
		3	7	-
		4	5	5
		5	4	4
		6	4	4
		7	4	4
	2	1	14	-
		2	12	-

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης
frg1	2	3	10	-
		4	6	6
		5	9	9
	3	1	16	-
		2	3	3
	4	1	8	8
	5	1	14	14
	6	1	4	4
	7	1	4	4
i2	1	1	172	172
	2	1	12	-
		2	26	26
	3	1	20	20
i3	1	1	132	132
	2	1	16	16
	3	1	56	56
mux	1	1	8	0
	2	1	16	-
		2	16	-
	3	1	8	8
	4	1	4	4
	5	1	2	2
pcle	1	1	6	6
		2	3	3
	2	1	8	8
		2	6	6
	3	1	6	6
		2	4	4
	4	1	6	2
		2	4	4
	5	1	6	6
		2	4	4
	6	1	6	6
		2	4	4
	7	1	7	6
		2	4	4
	8	1	4	4
pcle8	1	1	6	6
		2	6	6
		3	7	7
		4	6	6
	2	1	6	6

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης
pcler8	<b>3</b>	1	6	6
	<b>4</b>	1	6	2
	<b>5</b>	1	6	6
	<b>6</b>	1	6	6
	<b>7</b>	1	7	6
	<b>8</b>	1	40	-
term1	<b>1</b>	1	10	7
		2	6	6
		3	9	-
		4	7	7
		5	6	7
		6	8	8
		7	6	6
		8	4	4
		9	12	12
	<b>2</b>	1	6	6
		2	5	5
		3	13	-
		4	18	18
	<b>3</b>	1	12	12
		2	5	5
		3	16	16
	<b>4</b>	1	5	5
		2	13	13
		3	13	-
		4	8	8
<b>5</b>	1	14	-	
	2	15	15	
<b>6</b>	1	3	3	
	2	14	14	
<b>7</b>	1	15	-	
	2	7	7	
<b>8</b>	1	10	-	
	2	3	3	
x2	<b>1</b>	1	12	7
		2	4	4
	<b>2</b>	1	10	-
		2	10	10
<b>3</b>	1	18	-	
x3	<b>1</b>	1	8	7
		2	6	6
		3	26	-

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης		
x3	1	4	10	2		
		5	10	2		
		6	10	2		
		7	9	9		
		8	9	9		
		9	6	4		
		10	8	8		
		11	4	1		
		12	2	2		
		13	20	20		
		2	2	1	32	-
				2	32	32
				3	17	-
4	7			7		
5	22			-		
6	8			8		
7	6			5		
8	4			4		
9	13			13		
3	3	1	32	-		
		2	32	32		
		3	32	32		
		4	32	32		
		5	32	32		
		6	32	32		
		7	8	8		
		8	8	8		
		9	7	7		
		10	7	7		
		11	6	6		
		12	6	6		
		13	4	4		
		14	4	4		
		15	35	35		
4	4	1	75	-		
		2	6	2		
		3	7	7		
		4	8	8		
		5	6	6		
		6	6	6		
		7	4	4		
		8	18	18		

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

κύκλωμα	λογικό επίπεδο	ομάδες εξόδων	αρχικό κόστος δικτύου	λύση δυαδικής συσχέτισης
$x_3$	<b>5</b>	1	11	11
		2	12	12
		3	6	6
		4	7	6
		5	8	-
		6	6	6
		7	6	6
		8	58	58
	<b>6</b>	1	18	-
		2	8	8
		3	7	7
		4	8	-
		5	17	17
	<b>7</b>	1	8	-
		2	8	8
		3	12	12
	<b>8</b>	1	12	12

Πίνακας 4.4: Αποτελέσματα Δυαδικών Συσχετίσεων για τη Διεπίπεδη Προσέγγιση Υλοποίησης του Επιπρόσθετου Δικτύου Boole (συνέχεια)

Από τα αποτελέσματα του Πίνακα 4.4 είναι φανερό ότι ένα πολύ μικρό ποσοστό των δυαδικών συσχέτισεων που σχηματίζονται προκαλεί μείωση του κόστους. Εάν σε αυτό το γεγονός, συνυπολογιστεί το κόστος της αναγνώρισης ολοκλήρωσης για κάθε ομάδα εξόδων, καθώς και το κόστος για την σύνδεση των ομάδων μεταξύ τους, όπως το παράδειγμα του Σχήματος 3.8, τότε η διεπίπεδη προσέγγιση υλοποίησης δεν είναι δυνατό να οδηγήσει σε κυκλώματα με κόστος μικρότερο από το  $x_2$ , ως προς αυτό του αρχικού δικτύου Boole, το οποίο επιτυγχάνει ο μετασχηματισμός της διπλής-ράγας.

Η μεγάλη πολυπλοκότητα των δυαδικών συσχέτισεων, έγκειται στη φύση των DI κωδίκων. Η συμπεριφορά της συνάρτησης *SynC* που υλοποιεί το προσθετικό δίκτυο, είναι να ξεχωρίζει τα διανύσματα που δέχεται στην είσοδό της με βάση τον αριθμό των μηδενικών που έχουν. Ουσιαστικά αυτό λειτουργεί σαν ένας μηχανισμός μέτρησης των μηδενικών ή ανίχνευσης ισοτιμίας, που είναι κυκλώματα που δεν τυγχάνουν αποδοτικής υλοποίησης, ούτε σε διεπίπεδη, ούτε σε πολυεπίπεδη αναπαράσταση. Επίσης, ο μεγάλος χώρος εξερεύνησης των πιθανών λύσεων των δυαδικών συσχέτισεων, δυσκολεύει την εφαρμογή τους. Για

ένα διάνυσμα των  $n$  ψηφίων με  $k$  ψηφία με τιμή 0, οι ισοδύναμες αναπαραστάσεις που δίνουν ίδια έξοδο Sync, είναι  $n!/(k! \cdot (n - k)!)$ . Όσο αυξάνεται το  $n$ , ακόμα και για μικρούς αριθμούς, αυτή η ποσότητα είναι μεγάλη και οδηγεί σε πολλές επιλογές αυξάνοντας εκθετικά το χώρο εξερεύνησης. Πάντως, η χρήση των δυαδικών συσχετίσεων παραμένει μια εναλλακτική υλοποίηση.

### 4.3 Κυκλώματα με Πολυεπίπεδη Υλοποίηση του Επιπρόσθετου Δικτύου Boole

Στον Πίνακα 4.5 παρατίθενται τα αποτελέσματα για το κόστος υλοποίησης συνδυαστικών κυκλωμάτων με αναγνώριση ολοκλήρωσης, στα οποία το επιπρόσθετο δίκτυο Boole έχει παραχθεί με χρήση της πολυεπίπεδης υλοποίησης. Η υλοποίηση, από όπου προκύπτουν και τα κόστη, ενσωματώθηκε στο περιβάλλον του SIS και αποτελείται από τα παρακάτω βήματα.

- Δημιουργία αντίγραφου του αρχικού δικτύου Boole
- Μετασχηματισμός του αρχικού δικτύου Boole, ώστε να αποκτήσει μονότονα αύξουσα συμπεριφορά.
- Δημιουργία της συνάρτησης Sync με βάση τον αριθμό των εξόδων του δικτύου.
- Επισύναψη της συνάρτησης Sync στο αντίγραφο του αρχικού δικτύου Boole, με ταυτόχρονη αφαίρεση τις ιδιότητες των εξόδων αυτού ως πρωτευουσών. Η ένωση των δύο δικτύων είναι το επιπρόσθετο δίκτυο Boole.
- Εφαρμογή πολυεπίπεδων αλγορίθμων βελτιστοποίησης που βρίσκονται στο SIS.
- Μετασχηματισμός του επιπρόσθετου δικτύου Boole, ώστε να αποκτήσει μονότονα αύξουσα συμπεριφορά.

Όλα τα παραπάνω, εκτός από τους αλγορίθμους βελτιστοποίησης και την επιβολή της μονοτονίας, αποτελούν νέες υλοποιήσεις. Για την επιβολή της μονοτονίας

χρησιμοποιήθηκε, αφού πρώτα τροποποιήθηκε κατάλληλα, κομμάτι του κώδικα που υπάρχει για τον σκοπό αυτό, στο πακέτα της υλοποίησης της διπλής-ράγας. Τέλος, σε μερικές περιπτώσεις χρησιμοποιήθηκε η ομαδοποίηση των εξόδων του κυκλώματος, η οποία έλαβε μέρος πριν τα παραπάνω βήματα. Έπειτα ακολουθήθηκε η ίδια διαδικασία για κάθε ομάδα ξεχωριστά.

Εκτός από τις υλοποιήσεις των κυκλωμάτων και τα αποτελέσματα σε κόστος δυαδικών όρων, έγιναν μετρήσεις στο SIS για τον αριθμό των μεταβάσεων των κόμβων που αποτελούν τα κυκλώματα, με λειτουργία προσομοίωσης. Όπως έχει ήδη αναφερθεί αυτό είναι ένα ευριστικός μέτρο της κατανάλωσης της ισχύος των κυκλωμάτων. Για τη διαδικασία των μετρήσεων δημιουργήθηκαν τυχαία διανύσματα εισόδου, τα οποία τροφοδοτούσαν το κύκλωμα εναλλάξ με την λέξη αρχικοποίησης 00...0. Με τον τρόπο αυτό προσημειώθηκε η τυπική διφασική λειτουργία ενός DI κυκλώματος. Για κάθε προσομοίωση ενός διανύσματος, γινόταν μία εγγραφή για τις μεταβάσεις των σημάτων έως ότου η έξοδος λάβει την τελική της τιμή. Το γεγονός ότι τα κυκλώματα είναι μονότονα, μας επιτρέπει να μην χρειάζεται να λάβουμε υπόψη μας τη πιθανότητα μεταστάθειας στις τιμές των κόμβων. Ο μέσος όρος των μεταβάσεων για ζεύγη διανυσμάτων που αποτελούνταν από μία λέξη αρχικοποίησης, ακολουθούμενη από έγκυρα δεδομένα εισόδου, χρησιμοποιήθηκε ως ένα μέτρο εκτίμησης της κατανάλωσης της ισχύος των κυκλωμάτων. Για τον μετασχηματισμό της διπλής-ράγας, χρησιμοποιήθηκαν 512 ζεύγη διανυσμάτων για την εξαγωγή του μέσου όρου, ενώ για τα κυκλώματα πολλαπλής-ράγας, χρησιμοποιήθηκαν 2048 ζεύγη. Η διαφορά στον αριθμό των ζευγών που χρησιμοποιήθηκαν έγκειται στο γεγονός ότι ο μέσος όρος των μεταβάσεων για τον μετασχηματισμό της διπλής-ράγας, συνέκλινε πιο γρήγορα και για λιγότερα ζεύγη διανυσμάτων.

Τα πεδία του Πίνακα 4.5 είναι κατά σειρά:

- Το όνομα του κυκλώματος (*κύκλωμα*). Τα κυκλώματα που υλοποιήθηκαν ανήκουν στην ομάδα των δειγματοληπτικών κυκλωμάτων IWLS που περιγράφονται στο [16]. Σε μερικά κυκλώματα εφαρμόστηκε η τεχνική της ομαδοποίησης των εξόδων, που αναφέρθηκε στην Ενότητα 3.2.3. Οι ομάδες

που σχηματίστηκαν αναφέρονται με το όνομα του κυκλώματος ακολουθούμενο από τον αύξων αριθμό της ομάδας.

- Ο αριθμός των κυριολεκτικών εισόδων του κυκλώματος (είσοδοι).
- Ο αριθμός των κυριολεκτικών εξόδων του κυκλώματος (έξοδοι).
- Το κόστος, σε αριθμό όρων, του αρχικού δικτύου Boole, προτού αυτό υποστεί οποιοδήποτε είδους επεξεργασία (αρχικό).
- Το κόστος του αρχικού δικτύου Boole όταν αυτό μετασχηματισθεί, με σκοπό να αποκτήσει μονότονη συμπεριφορά (μονότονο).
- Το κόστος του επιπρόσθετου δικτύου Boole με χρήση DI κώδικα Berger (*Berger*). Το κόστος που παραθέτεται αφορά στο επιπρόσθετο δίκτυο Boole αφότου αυτό έχει μετασχηματισθεί ώστε να αποκτήσει μονότονη συμπεριφορά.
- Το κόστος αναγνώρισης ολοκλήρωσης με χρήση DI κώδικα Berger (*CD B*).
- Το κόστος του επιπρόσθετου δικτύου Boole με χρήση συστηματικού DI κώδικα Sperner (*Sperner*). Το κόστος που παρατίθεται αφορά στο επιπρόσθετο δίκτυο Boole αφότου αυτό έχει μετασχηματισθεί ώστε να αποκτήσει μονότονη συμπεριφορά.
- Το κόστος αναγνώρισης ολοκλήρωσης με χρήση συστηματικού DI κώδικα Sperner (*CD S*).
- Το κόστος υλοποίησης του αντίστοιχου κυκλώματος, με χρήση του μετασχηματισμού διπλής-ράγας (διπλή-ράγα).
- Ο μέσος όρος των μεταβάσεων των κόμβων που αποτελούν το κύκλωμα υλοποιημένο σε μορφή διπλής-ράγας, για ένα πλήρη κύκλο λειτουργίας που περιλαμβάνει την φάση αρχικοποίησης και την φάση υπολογισμού (μεταβάσεις *DR*).

- Το συνολικό κόστος της υλοποίησης του κυκλώματος (πολλαπλή-ράγα). Το κόστος που παρατίθεται είναι το μικρότερο μεταξύ αυτού που χρησιμοποιεί DI κώδικα Berger και αυτού που κάνει χρήση συστηματικού DI Sperner.
- Ο μέσος όρος μεταβάσεων των κόμβων που αποτελούν το κύκλωμα, για ένα πλήρη κύκλο λειτουργίας που περιλαμβάνει την φάση αρχικοποίησης και την φάση υπολογισμού (μεταβάσεις *MR*).

Τα κενά στον Πίνακα 4.5, που είναι χρωματισμένα μπλε, αφορούν τις περιπτώσεις που χρησιμοποιήθηκε η ομαδοποίηση εξόδων. Στις ομάδες δεν έγινε εφαρμογή του μετασχηματισμού διπλής-ράγας, γιατί η ομαδοποίηση δεν είναι μέρος της συγκεκριμένης τεχνικής, και έγινε μόνο εφαρμογή στο αρχικό κύκλωμα. Επίσης, όπου υπάρχει η '-', σημαίνει ότι η εκάστοτε διαδικασία για το συγκεκριμένο πεδίο δεν ολοκληρώθηκε. Πιο συγκεκριμένα, οι αλγόριθμοι βελτιστοποίησης του SIS δεν μπόρεσαν να δώσουν κάποιο αποτέλεσμα, σε εύλογο χρόνο εκτέλεσης.

Από τον Πίνακα 4.5 μπορούμε να συμπεράνουμε ότι το κόστος για το επιπρόσθετο δίκτυο Boole, υλοποιημένο με χρήση συστηματικού Sperner είναι για όλα τα κυκλώματα μικρότερο ή ίσο με αυτό που προκύπτει με χρήση κώδικα Berger. Αυτή η παρατήρηση είναι ιδιαίτερης σημασίας, αφού παρά το γεγονός ότι ο συστηματικός Sperner χρησιμοποιεί περισσότερα ψηφία για την κωδικοποίηση από τον Berger, οδηγεί σε καλύτερα κυκλώματα. Συνεπώς, η βελτιστότητα από πλευράς αριθμού ψηφίων δεν αντικατοπτρίζεται στα κυκλώματα που υλοποιούνται με χρήση αυτών. Επίσης, το κόστος της υλοποίησης των κυκλωμάτων πολλαπλής-ράγας είναι τις περισσότερες φορές μικρότερο από το άθροισμα των κοστών των μερών που το αποτελούν (μονότονο αρχικό δίκτυο Boole, μονότονο επιπρόσθετο δίκτυο Boole, αναγνώριση ολοκλήρωσης). Αυτό οφείλεται στο γεγονός ότι κατά την ένωση του μονότονου αρχικού δικτύου Boole με το επιπρόσθετο δίκτυο Boole, είναι πιθανό να βρεθούν κοινοί κύβοι ή και πυρήνες μεταξύ των δύο, κάτι που μειώνει το κόστος υλοποίησης.

Αναλύοντας τις τιμές του Πίνακα 4.5 και συγκρίνοντας τα αντίστοιχα μεγέθη μεταξύ κυκλωμάτων διπλής και πολλαπλής ράγας, βλέπουμε ότι υπάρχουν

κύκλωμα	είσοδοι	έξοδοι	αρχικά	μονότονο	Berger	CD B.	Sperner	CD S.	διπλή-ράγα	μεταβάσεις DR	πολλαπλή-ράγα	μεταβάσεις
9symml	9	1	267 (261)	266	102	2	102	2	710	822	355	317
alu2	10	6	478 (445)	447	1017	52	913	75	1217	1408	1364	788
alu2_1	10	3	431	437	504	14	391	17			829	612
alu2_2	4	3	11	11	20	14	14	17				
alu4	14	8	917 (1218)	1261	-	-	-	-	2057	2498	-	-
alu4_1	10	2	327	336	473	7	395	7			2803	1663
alu4_2	14	2	757	796	1070	7	952	7				
alu4_3	2	2	5	5	6	7	6	7				
alu4_4	14	2	286	293	822	7	381	7				
b1	3	4	12 (11)	10	6	15	6	15	29	39	31	36
C17	5	2	12 (12)	12	20	7	14	7	29	41	33	42
cm42	4	10	34 (34)	34	3	45	3	45	100	132	82	151
cm82	5	3	28 (28)	42	45	14	30	17	66	81	79	95
cm85	11	3	62 (66)	60	89	14	68	17	127	162	122	162
cm138	6	8	31 (31)	31	3	35	3	35	88	114	69	132
cm150	21	1	61 (53)	53	54	2	54	2	101	102	101	102
cm151	12	2	31 (30)	51	0	2	0	2	53	54	53	55
cm152	11	1	28 (28)	28	28	2	28	2	50	48	50	48
cm162	14	5	58 (52)	59	217	39	155	45	121	157	253	305
cm163	16	5	54 (48)	55	255	39	166	45	113	135	266	315
cmb	16	4	53 (37)	48	38	25	28	23	70	80	88	113
cmb_1	12	3	25	36	24	14	12	10			74	88
cmb_2	12	1	12	12	12	2	12	2				
cordic	23	2	83 (97)	141	19	6	19	6	148	173	152	190
cu	14	11	66 (63)	67	75	-	53	136	147	190	247	171
cu_1	6	4	23	28	13	15	13	15			141	173
cu_2	7	4	17	17	10	12	5	5				
cu_3	13	1	24	24	23	2	23	2				
cu_4	5	2	7	7	15	7	9	7				

Πίνακας 4.5: Παραδείγματα Υλοποίησης Κυκλωμάτων, με Χρήση DI Κωδίκων

κύκλωμα	είσοδοι	έξοδοι	αρχικά	μονότονο	Berger	CD B.	Sperner	CD S.	διπλή-ράγα	μεταβάσεις DR	πολλαπλή-ράγα	μεταβάσεις M
f51m	8	8	159 (159)	180	436	84	358	125	341	399	627	519
f51m_1	8	4	141	158	241	26	199	29			403	392
f51m_2	4	2	16	22	32	7	27	7				
f51m_3	2	2	5	4	2	7	4	7				
frg1	28	3	146 (146)	147	229	14	90	17	313	394	251	290
i2	201	1	230 (220)	220	222	2	222	2	446	588	444	584
majority	5	1	12 (12)	12	12	2	12	2	26	36	26	36
mux	21	1	54 (54)	54	53	2	53	2	127	158	101	99
pcl	19	9	78 (78)	87	-	-	240	143	188	250	469	436
pcl_1	16	6	57	65	195	52	115	71			320	373
pcl_2	9	3	25	26	41	14	38	17				
pcler8	27	17	102 (115)	123	-	-	-	-	261	355	-	-
pcler8_1	27	9	107	115	-	-	360	151			819	801
pcler8_2	9	8	16	16	169	84	90	125				
t481	16	1	1028 (52)	87	84	2	84	2	113	135	98	123
term1	34	10	271 (139)	228	-	-	-	-	324	406	-	-
term1_1	20	5	51	75	102	39	78	45			450	482
term1_2	19	2	63	98	112	7	116	7				
term1_3	11	2	31	54	96	7	94	7				
term1_4	1	1	1	0	0	2	0	2				
too_large	38	3	491 (395)	425	-	-	290	17	836	994	670	555
x2	10	7	54 (51)	55	57	66	43	78	129	176	174	230
x2_1	3	3	11	11	11	14	6	10			120	151
x2_2	7	2	10	10	11	6	11	6				
x2_3	10	2	37	37	37	7	32	7				
z4ml	7	4	42 (42)	66	117	26	63	29	96	115	138	161

Πίνακας 4.5: Παραδείγματα Υλοποίησης Κυκλωμάτων, με Χρήση DI Κωδίκων (συνέχεια)

αρκετά κυκλώματα για τα οποία η υλοποίηση πολλαπλής-ράγας δίνει μικρότερο κόστος. Ένας καθοριστικός παράγοντας για να συμβεί αυτό είναι το κόστος υλοποίησης του επιπρόσθετου δικτύου Boole να είναι μικρότερο από αυτό του αρχικού δικτύου. Ένας δεύτερος παράγοντας είναι η μορφή του επιπρόσθετου δικτύου Boole, ως προς το αν έχει ψηφία με τιμές σταθερές στο 0 ή το 1, το οποίο συνεισφέρει σημαντικά στην απλοποίηση της αναγνώρισης ολοκλήρωσης, όπως αναφέρθηκε στην Ενότητα 4.1.3, που περιγράφει την τεχνική για τη περαιτέρω ελαχιστοποίηση του κόστους αναγνώρισης ολοκλήρωσης, με τη βοήθεια προσομοίωσης ή εξερεύνησης της συμπεριφοράς του επιπρόσθετου δικτύου.

Για παράδειγμα, το επιπρόσθετο δίκτυο Boole του κυκλώματος *cm42* είναι το παρακάτω Δίκτυο, 4.2:

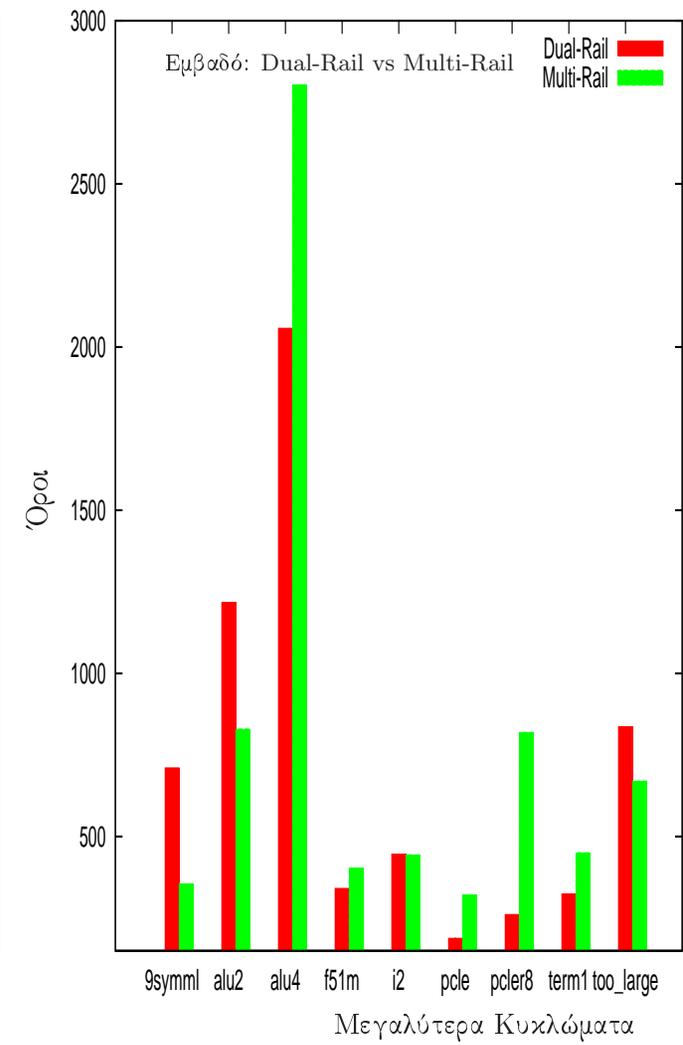
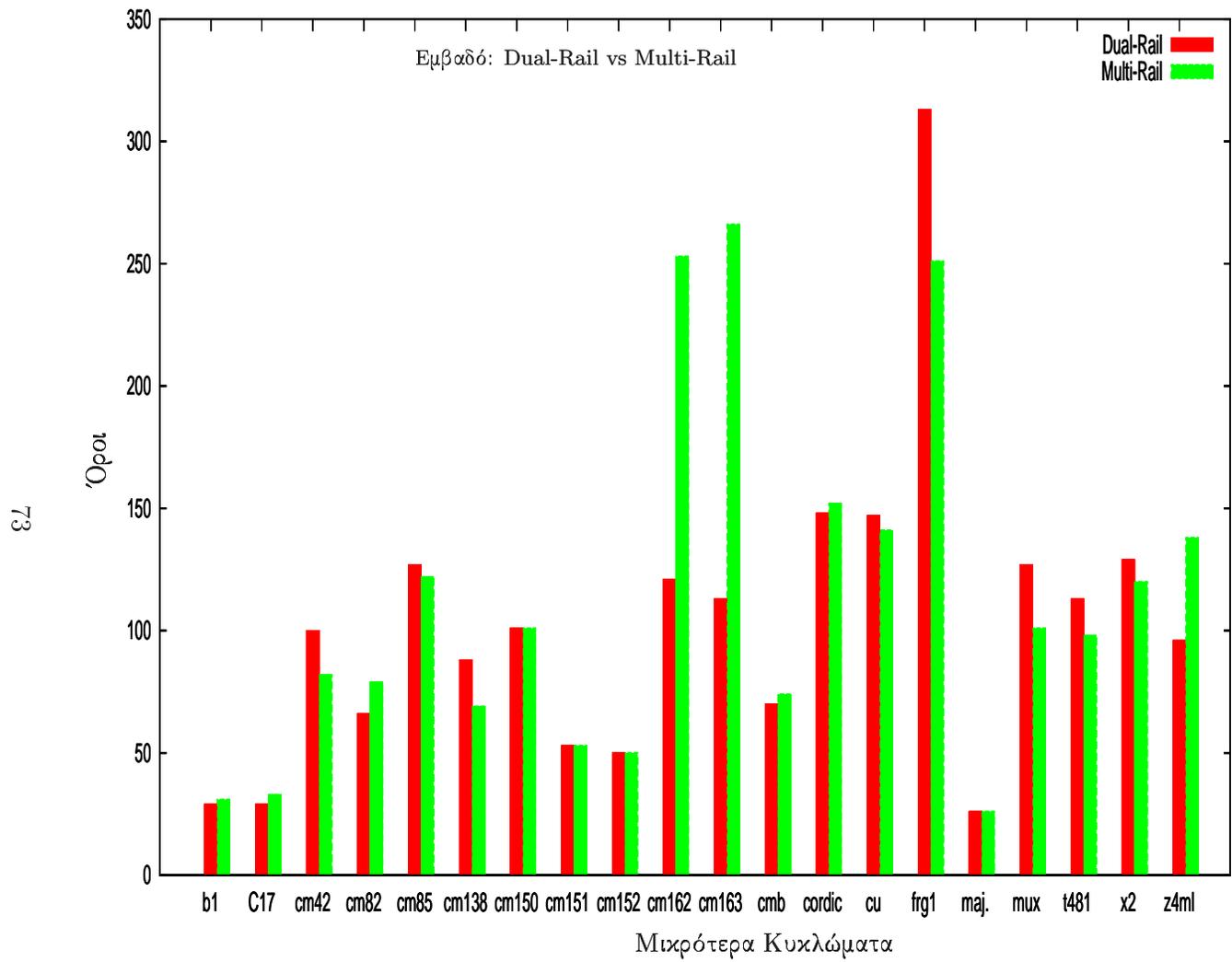
$$\begin{aligned}
 sync0 &= 0 \\
 sync1 &= 0 \\
 sync2 &= 0 \\
 sync3 &= 0 \\
 sync4 &= 0 \\
 sync5 &= 0 \\
 sync6 &= 0 \\
 sync7 &= 0 \\
 sync8 &= 0 \\
 sync9 &= b_ \cdot c_ + d_
 \end{aligned}
 \tag{4.2}$$

όπου  $b_$ ,  $c_$ ,  $d_$  είναι οι συμπληρωματικές μορφές τριών εκ των τεσσάρων εισόδων του κυκλώματος, που είναι απαραίτητες για την επιβολή της μονοτονίας. Τα σταθερά ψηφία των εξισώσεων του Δικτύου 4.2 κάνουν δυνατή τη μείωση του κόστους της αναγνώρισης ολοκλήρωσης, η οποία σε αντίθετη περίπτωση, και λόγω των πολλών εξόδων του κυκλώματος, θα καθιστούσε απαγορευτική, από πλευράς κόστους, την υλοποίηση.

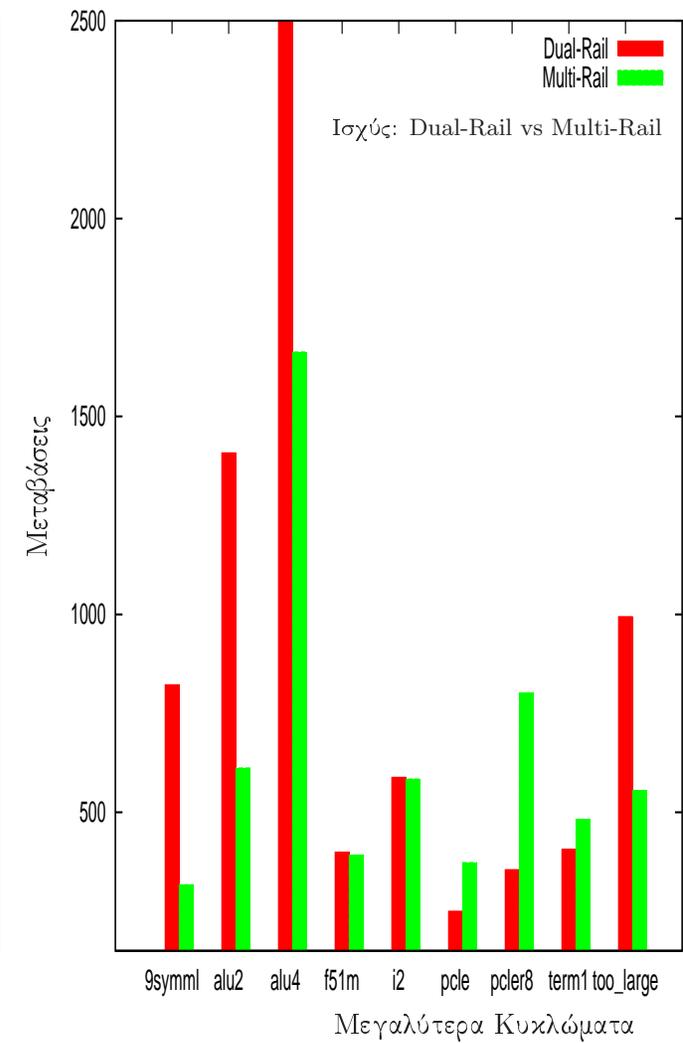
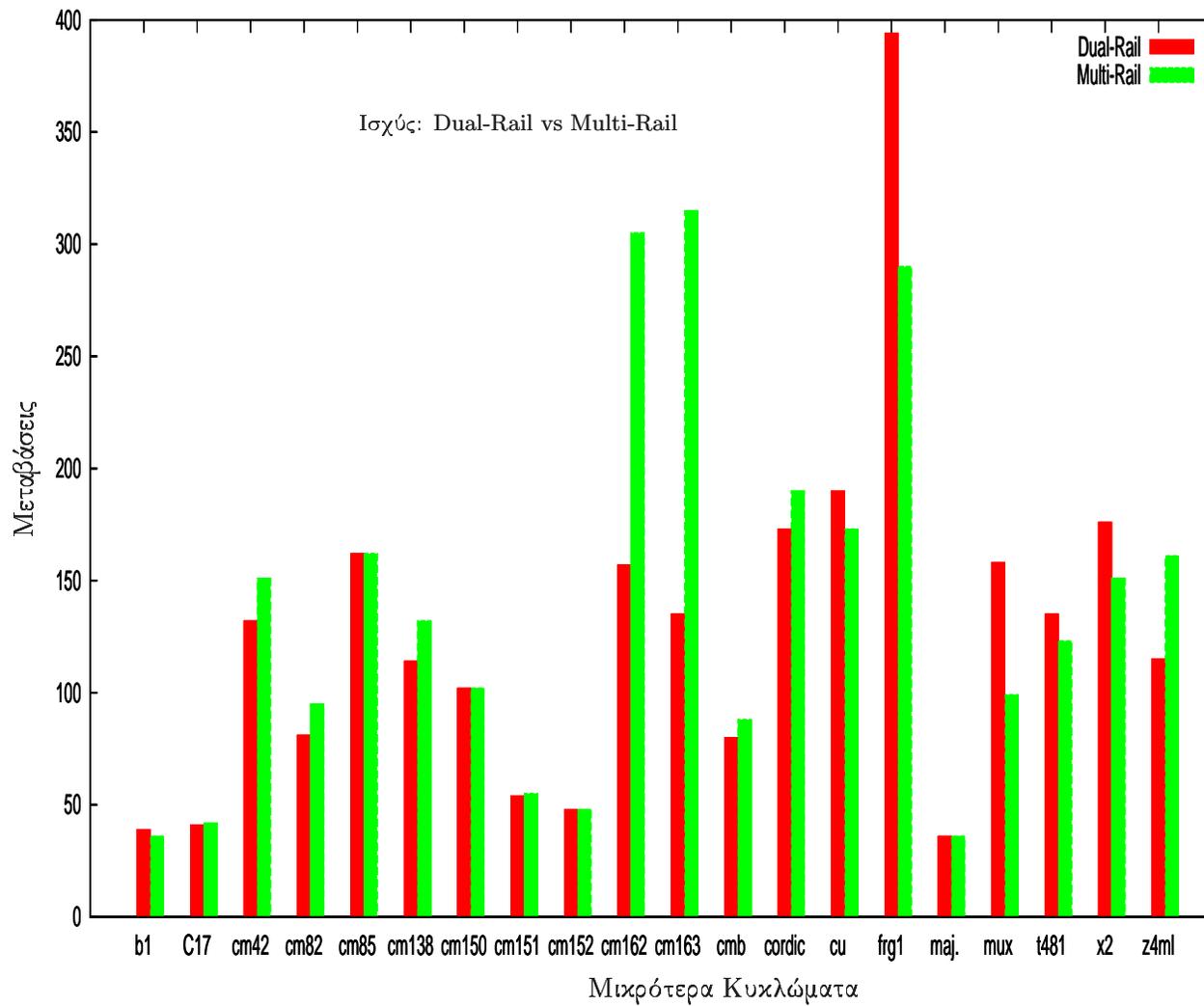
Σε αντίθεση με το προηγούμενο, κυκλώματα όπως το *frg1* ή το *too\_large* δεν έχουν δομή που να επιτρέπει τη μείωση του κόστους της αναγνώρισης ολοκλήρωσης, όμως ο μικρός αριθμός των πρωτευουσών εξόδων κρατά το κόστος αυτής χαμηλά. Αν παρόλα αυτά ο αριθμός των πρωτευουσών εξόδων είναι μεγαλύτερος, τότε κυκλώματα όπως τα *alu2*, *cu* και *x2*, δείχνουν πως το κόστος υλοποίησης μπορεί να μειωθεί ομαδοποιώντας τις.

Όσον αφορά το μέσο όρο των μεταβάσεων των κόμβων, στα περισσότερα παραδείγματα αυξάνεται ή μειώνεται μαζί με το κόστος υλοποίησης. Όμως σε κάποιες περιπτώσεις (*alu2*, *cm42*, *f51m*) αυτό δεν ισχύει, υποδηλώνοντας τη διαφορετική δομή των κυκλωμάτων πολλαπλής-ράγας σε σύγκριση με αυτά της διπλής-ράγας.

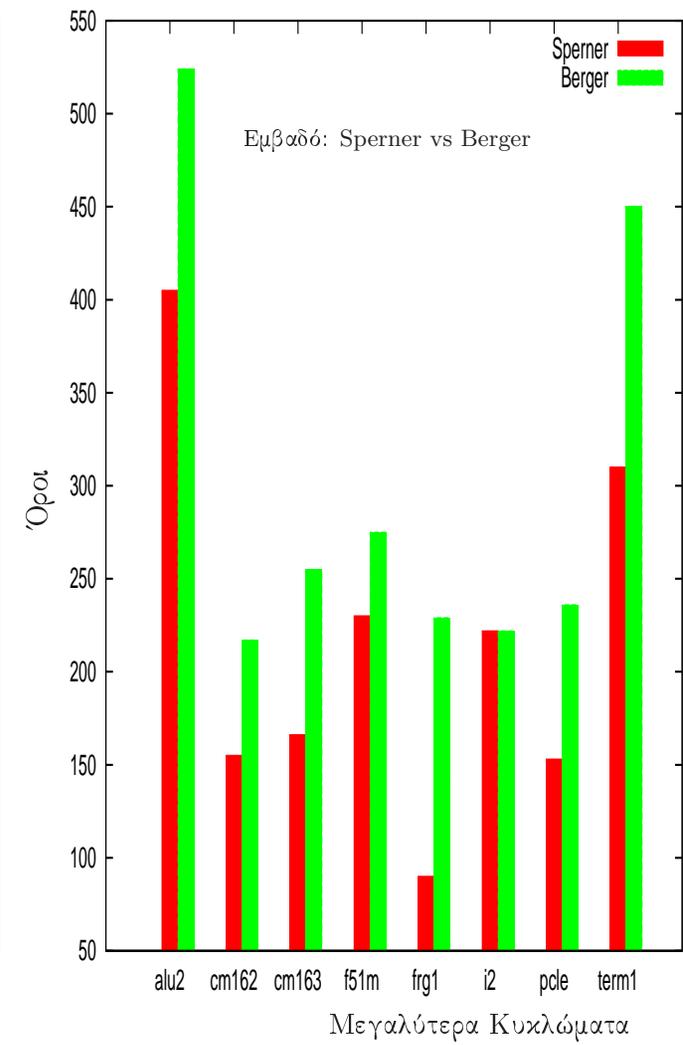
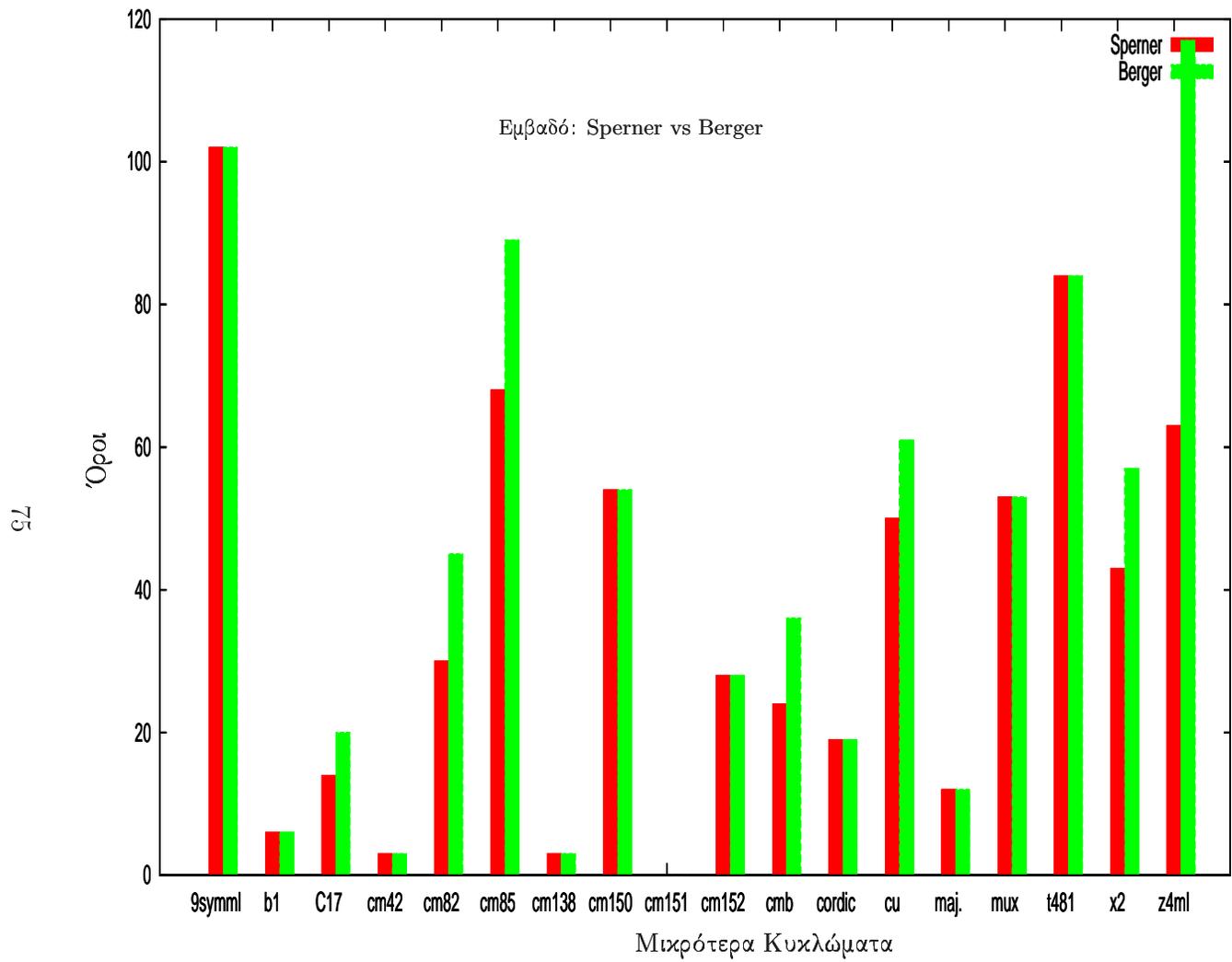
Στα Σχήματα 4.2 και 4.3 που ακολουθούν αναπαριστώνται γραφικά τα αποτελέσματα του Πίνακα 4.5 για τον αριθμό των όρων και το μέσο όρο των μεταβάσεων των κυκλωμάτων αντίστοιχα, για τις τεχνικές της διπλής και της πολλαπλής ράγας. Τέλος, στο Σχήμα 4.4 γίνεται η σύγκριση του εμβαδού των επιπρόσθετων δικτύων Boole για κάποια από τα κυκλώματα που υλοποιήθηκαν, για τους κώδικες Sperner και Berger. Για το κύκλωμα *cm151*, το ότι ο αριθμός των όρων του επιπρόσθετου δικτύου Boole είναι μηδέν, σημαίνει ότι προέκυψε σταθερή συνάρτηση. Η ερμηνεία αυτού του γεγονότος, είναι ότι κάθε κυριολεκτική έξοδος του *cm151* έχει τον ίδιο αριθμό ψηφίων με τιμή 0, και συνεπώς σταθερή τιμή Sync, που είναι η έξοδος του επιπρόσθετου δικτύου Boole.



Σχήμα 4.2: Σύγκριση Κυκλωμάτων Διπλής και Πολλαπλής Ράγας με Βάση τον Αριθμό των Όρων



Σχήμα 4.3: Σύγκριση Κυκλωμάτων Διπλής και Πολλαπλής Ράγας με Βάση το Μέσο Όρο των Μεταβάσεων



Σχήμα 4.4: Σύγκριση Των Όρων του Επιπρόσθετου Δικτύου Boole, με Χρήση Sperner και Berger



## Κεφάλαιο 5

# Συμπεράσματα - Μελλοντική

## Εργασία

Από τα αποτελέσματα του Κεφαλαίου 4 μπορούν να εξαχθούν δύο σημαντικά συμπεράσματα. Το πρώτο πηγάζει από τη διαπίστωση πως τα μισά περίπου από τα κυκλώματα τα οποία μετασχηματίστηκαν σε κυκλώματα πολλαπλής-ράγας έχουν μικρότερο ή ίδιο αριθμό όρων (literals) σε σχέση με αυτά στα οποία έγινε χρήση της τεχνικής της διπλής-ράγας. Αυτό δείχνει ότι πιθανώς υπάρχει περιθώριο βελτίωσης για τα κυκλώματα με χρήση DI κωδίκων και το οποίο χρήζει περαιτέρω διερεύνησης.

Το δεύτερο, εξίσου χρήσιμο, συμπέρασμα είναι η επιβεβαίωση του γεγονότος ότι η τεχνική της διπλής-ράγας είναι μία αποδοτική υλοποίηση κυκλωμάτων με αναγνώριση ολοκλήρωσης.

Εκτός των αποτελεσμάτων, η όλη πειραματική διαδικασία, ανέδειξε τη φύση των κυκλωμάτων που χρησιμοποιούν κώδικες όπως ο Berger και ο Sperner. Οι κώδικες αυτοί, για να δημιουργήσουν τις προϋποθέσεις για αναγνώριση ολοκλήρωσης χρειάζεται να κοιτάνε τον αριθμό των μηδενικών των κυριολεκτικών δεδομένων. Αυτό έχει ως αποτέλεσμα να λειτουργούν ως αθροιστές ή κυκλώματα ισοτιμίας, τα οποία είναι γνωστά για την μη αποδοτική υλοποίησή τους [17]. Συνεπώς, παρά τις επιθυμητές ιδιότητες των DI κωδίκων, τα κυκλώματα που κάνουν χρήση αυτών έχουν την τάση να είναι ιδιαίτερα πολύπλοκα στην υλοποίησή τους.

Ειδικά για τον κώδικα Berger του οποίου η συμπεριφορά του Sync είναι ταυτόσημη με αυτή ενός δυαδικού αθροιστή, φάνηκε ότι σε αρκετές περιπτώσεις, το επιπρόσθετο δίκτυο που προέκυπτε είχε πολύπλοκη δομή, με αποτέλεσμα οι ευριστικοί αλγόριθμοι βελτιστοποίησης να μην μπορούν να καταλήξουν σε μία αποδοτική υλοποίηση. Έτσι, παρά το γεγονός ότι αυτός ο κώδικας χρησιμοποιεί λιγότερα ψηφία από τον Sperner, οδηγεί σε χειρότερα κυκλώματα. Συμπερασματικά, η βελτιστότητα ενός κώδικα ως προς τον αριθμό των ψηφίων που τον απαρτίζουν, δεν αντικατοπτρίζεται, απαραίτητα, στην υλοποίηση των κυκλωμάτων που τον χρησιμοποιούν.

## 5.1 Μελλοντική Εργασία

Οι προεκτάσεις της παρούσας εργασίας μπορούν να έχουν δύο κατευθύνσεις. Μία προς τη βελτίωση των αρνητικών σημείων και τη λύση των προβλημάτων που αντιμετωπίζει η τεχνική της πολλαπλής-ράγας, και μία προς τη δοκιμή και αξιολόγηση νέων στοιχείων και ιδεών.

Το κυριότερο μειονέκτημα της τεχνικής είναι η πράξη της σύνθλιψης των κυκλωμάτων, που είναι απαραίτητη τόσο για την επιβολή της μονοτονίας των κυκλωμάτων, όσο και για την δημιουργία του επιπρόσθετου δικτύου Boole. Η εξάρτηση από την πράξη της σύνθλιψης κάνει γενικά δύσκολη την εφαρμογή της τεχνικής σε πολύ μεγάλα κυκλώματα, καθώς σύνθλιψη των εξόδων ως προς τις πρωτεύουσες εισόδους είναι απαγορευτική ως υπολογιστικά πολύ ακριβή. Όσον αφορά τη μονοτονία, η σύνθλιψη μπορεί αντικατασταθεί από μία λιγότερο ακριβή υπολογιστικά τακτική, η οποία θα συνθλίβει μονάχα εσωτερικούς κόμβους του κυκλώματος, οι οποίοι εμφανίζονται με την κανονική και τη συμπληρωματική τους μορφή. Με τον τρόπο αυτό μπορούν να αποφευχθούν περιττές αντικαταστάσεις, κάτι που, αναλόγως με τη δομή του κυκλώματος, μπορεί να οδηγήσει σε μεγάλη μείωση του χρόνου υπολογισμού.

Η χρήση της σύνθλιψης στη δημιουργία του επιπρόσθετου δικτύου Boole είναι λιγότερο ευέλικτη. Το πόσο θα απλοποιηθεί η έκφραση του επιπρόσθετου δικτύου Boole, μπορεί να εξαρτάται από τα πόσα επίπεδα λογικής θα συνθλιβούν.

Υπό την έννοια αυτή μπορεί να επιτευχθεί μία αντιστάθμιση του υπολογιστικού φόρτου και της απλοποίησης (trade-off) που θα προσπαθεί να επιτύχει το καλύτερο δυνατό αποτέλεσμα σε αποδεκτό χρόνο εκτέλεσης. Μία τεχνική η οποία μπορεί επίσης να βοηθήσει στην υλοποίηση του επιπρόσθετου δικτύου Boole είναι αυτή που αναπτύχθηκε στην Ενότητα 3.2.1.2, και αφορά τη διάσπαση των κυκλωμάτων σε λογικά επίπεδα. Με τον τρόπο αυτό μπορούν να ομαδοποιούνται λογικά επίπεδα και να υλοποιούνται επιπρόσθετα δίκτυα Boole που αντιστοιχούν σε κάθε ομάδα.

Μία βελτίωση της τεχνικής, αλλά με απαίτηση χρήσης νέων αλγορίθμων, αφορά την επίλυση των δυαδικών συσχετίσεων. Εάν υπήρχε ένας αλγόριθμος που να μπορούσε να επιλύσει σε λογικό χρόνο αρκετά μεγάλες σε μέγεθος δυαδικές συσχετίσεις, τότε θα μπορούσε να μειωθεί η ανάγκη της χρήσης της πράξης της σύνθλιψης. Νέες ιδέες θα μπορούσαν να αναζητηθούν και στον τομέα των DI κωδίκων. Στην παρούσα εργασία ερευνήθηκαν μονάχα οι ιδιότητες των Berger και Sperner. Είναι πιθανό κάποιοι άλλοι DI κώδικες να έχουν καλύτερα αποτελέσματα ή να έχουν κάποιες ιδιότητες που ταιριάζουν καλύτερα στην τεχνική.

Τέλος, κοιτάζοντας πιο κοντά στην φάση της υλοποίησης, μπορούμε να διαπιστώσουμε ότι σε κάποιες περιπτώσεις η μεγαλύτερη επιβάρυνση σε εμβαδό οφείλεται στην αναγνώριση ολοκλήρωσης. Οι μονάδες της αναγνώρισης ολοκλήρωσης είναι επαναχρησιμοποιούμενες σε κάθε κύκλωμα πολλαπλής-ράγας, χωρίς να αλλάζει η δομή τους. Συνεπώς, μπορούν να αναλυθούν και να τύχουν μιας πιο ειδικής επεξεργασίας για απλοποίηση, όπως να είναι έτοιμα κυκλώματα σε γρήγορη λογική λ.χ. σε επίπεδο τρανζίστορ CMOS, η οποία δεν είναι δυνατό να εφαρμόζεται με αυτόματο τρόπο στα κυκλώματα γενικού σκοπού. Σε ένα ευρύτερο πλαίσιο, η βελτιστοποίηση των σταθερών επιβαρύνσεων της ασύγχρονης σχεδίασης και η ενσωμάτωσή τους σε μία αυτοματοποιημένη υλοποίηση είναι ένα σημαντικό βήμα προς την αποδοχή των ασύγχρονων κυκλωμάτων από τους σχεδιαστές, και κάτι που έχει συνεισφέρει στην έως σήμερα μεγάλη διάδοση των σύγχρονων κυκλωμάτων.



# Βιβλιογραφία

- [1] A. Davis and S.M. Nowick, “An Introduction to Asynchronous Circuit Design”, Technical Report UUCS-97-013, Computer Science Department, University of Utah, Sep. 1997.
- [2] T.Verhoeff, “Delay-insensitive codes-an overview”, *Distributed Computing*, vol. 3, no. 1-8, 1988.
- [3] S. H. Unger, *Asynchronous Sequential Switching Circuit*. New York: Wiley-Interscience, John Wiley & Sons, Inc., 1969.
- [4] J. Sparsø and J. Staunstrup, “Delay-insensitive multi-ring structures”, *Integration, the VLSI journal*, vol. 15, no. 3, pp. 313-340, Oct. 1993.
- [5] A. Kondratyev and K. Lwin, “Design of asynchronous circuits by synchronous CAD tools”, in *Proc. ACM/IEEE Design Automation Conference*, June 2002.
- [6] J. Cortadella, A. Kondratyev, L. Lavagno, and C. Sotiriou, “Coping with the variability of combinational logic delays”, in *Proc. International Conf. Computer Design (ICCD)*, July 2004.
- [7] V. I. Varshavsky, Ed., *Self-Timed Control of Concurrent Processes: The Design of Aperiodic Logical Circuits in Computers and Discrete Systems*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1990.
- [8] R. Brayton and F. Somenzi, “An exact minimizer for boolean relations”, in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, pages 316-319, Nov. 1989.

- [9] Akella, V., Vaidya, N. H., Redinbo, G. R., “Limitations of VLSI implementations of delay-insensitive codes”, Proc. 26th Int. Symp. on Fault-Tolerant Computing, Sendai, June 1996, pp. 208-217.
- [10] Piestrak, S.J., Membership test logic for Delay-Insensitive codes, Proc Async ‘98, San Diego, California, April 1998, pp. 194-204.
- [11] D. Baneres, J. Cortadella, M. Kishinevsky, “A Recursive Paradigm To Solve Boolean Relations”, dac, pp. 416-421, Design Automation Conference, 41st Conference on (DAC’04), 2004.
- [12] E. B. Eichelberger *Hazard Detection in Combinational and Sequential Switching Circuits*. IBM Journal of Research, (9):90-99, mar 1965.
- [13] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, “SIS: A system for sequential circuit synthesis”, U.C. Berkeley, Tech. Rep., May 1992.
- [14] R. L. Rudell. “Logic Synthesis for VLSI Design”, PhD thesis, University of California at Berkeley, 1989.
- [15] Gary D. Hachtel , Fabio Somenzi, “Logic Synthesis and Verification Algorithms”, Kluwer Academic Publishers, Norwell, MA, 1996.
- [16] Saeyang Y., *Logic Synthesis and Optimization Benchmarks User guide*, MCNC International Workshop on Logic Synthesis, 1991.
- [17] S. Devadas, A. Ghosh, K. Keutzer, “Logic Synthesis”, McGraw-Hill, 1994.