



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ ΣΧΟΛΗ
ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Αρχιτεκτονικές προγραμματιζόμενων κυκλωμάτων για δίκτυα
υψηλών ταχυτήτων**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτριος Σ. Νικολαΐδης

Αθήνα, Δεκέμβριος 2025



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ ΣΧΟΛΗ
ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Αρχιτεκτονικές προγραμματιζόμενων κυκλωμάτων για δίκτυα υψηλών ταχυτήτων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτριος Σ. Νικολαΐδης

Συμβουλευτική Επιτροπή : **Ηρακλής Αβραμόπουλος (Επιβλέπων), Καθηγητής ΕΜΠ**

Εμμανουήλ Βαρβαρίγος, Καθηγητής ΕΜΠ

Γεώργιος Ματσόπουλος, Καθηγητής ΕΜΠ

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 1^η Δεκεμβρίου 2025.

.....
Ηρακλής Αβραμόπουλος
Καθηγητής ΕΜΠ

.....
Εμμανουήλ Βαρβαρίγος
Καθηγητής ΕΜΠ

.....
Γεώργιος Ματσόπουλος
Καθηγητής ΕΜΠ

.....
Δημήτριος Σούντρης
Καθηγητής ΕΜΠ

.....
Αθανάσιος Παναγόπουλος
Καθηγητής ΕΜΠ

.....
Τσανάκας Παναγιώτης
Καθηγητής ΕΜΠ

.....
Αποστολόπουλος Δημήτριος
Επίκουρος Καθηγητής ΕΜΠ

Αθήνα, Δεκέμβριος 2025

.....

Δημήτριος Σ. Νικολαΐδης

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

Copyright © Δημήτριος Σ. Νικολαΐδης, 2025
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η ραγδαία αύξηση του αριθμού των δικτυακών χρηστών οδήγησε στην συνεχή αναζήτηση βιώσιμων λύσεων κλιμάκωσης των χαρακτηριστικών των δικτύων όπως η χωρητικότητα, η ταχύτητα και η επεξεργαστική ισχύ. Μία λύση που έχει αναδειχθεί για αυτό το πρόβλημα είναι η εισαγωγή προγραμματιζόμενης λογικής (reprogrammable logic) σε καίρια σημεία επικοινωνιακών υποδομών λόγω των συνεχώς αυξανόμενων δυνατοτήτων της. Στο πλαίσιο αυτό, το θέμα της διατριβής είναι η χρησιμοποίηση πλατφόρμων προγραμματιζόμενου υλικού (Field Programmable Gate Array) για την υλοποίηση ψηφιακών αρχιτεκτονικών εστιασμένων στην τέλεση λειτουργιών δικτύων σε μέγιστη απόδοση. Ο τίτλος της διατριβής είναι «Αρχιτεκτονικές προγραμματιζόμενων κυκλωμάτων για δίκτυα υψηλών ταχυτήτων» και οργανώνεται σε τρία κεφάλαια. Κάθε κεφάλαιο είναι αφιερωμένο στις διαφορετικές αρχιτεκτονικές που παρουσιάζονται. Οι λειτουργίες τις οποίες επιτελούν οι προτεινόμενες αρχιτεκτονικές είναι η δρομολόγηση δικτύων Benes, η αναγνώριση μεγάλου μήκους αλληλουχιών bit με αλλοιώσεις και ως απόρροια αυτού ο συγχρονισμός πλαισίων δεδομένων σε ψηφιακές επικοινωνίες. Οι αρχιτεκτονικές είναι πρωτότυπες και προσαρμοσμένες στις ιδιότητες του υλικού ώστε να εκμεταλλεύονται στο έπακρο τις δυνατότητές του.

Στο Κεφάλαιο 1 παρουσιάζεται η αρχιτεκτονική δρομολόγησης (routing) δικτύου Benes και εξηγείται η σημαντικότητα του στη λειτουργία ηλεκτροοπτικών δικτύων μέσα στα σύγχρονα κέντρα δεδομένων (data centres). Η μέθοδος που χρησιμοποιείται βασίζεται στη σταδιακή δρομολόγηση κάθε επιπέδου του δικτύου που λόγω της αναδρομικής φύσης του δικτύου Benes μπορεί να υλοποιηθεί με πολύ απλές μεθόδους και λιγοστούς πόρους. Η δρομολόγηση γίνεται με ταξινόμηση των ζευγαριών εισόδου-εξόδου του πίνακα δρομολόγησης που δίνεται ως είσοδος και περιγράφει ποια είσοδος πρέπει να συνδεθεί σε ποια έξοδο. Η θέση που καταλαμβάνουν τα ζευγάρια εισόδου-εξόδου σε αυτή την ταξινόμηση δηλώνει την κατάσταση στην οποία πρέπει να τεθεί κάθε διακόπτης ώστε να επιτευχθεί σωστή δρομολόγηση. Μετά το τέλος της δρομολόγησης του πρώτου επιπέδου έχουν παραχθεί οι πίνακες εισόδου-εξόδου των εσωτερικών επιπέδων τους οποίους χρησιμοποιεί ο αλγόριθμος για να συνεχίσει τη δρομολόγηση. Η διαδικασία αυτή ακολουθείται έως ότου όλοι οι διακόπτες να έχουν τεθεί στη σωστή κατάσταση. Λόγω της απλής κατασκευής του το κύκλωμα πετυχαίνει την σωστή δρομολόγηση δικτύων Benes μεγέθους 64×64 σε 3 μs με λιγότερο από 15% των πόρων ενός μέσου εμπορικού FPGA.

Στο Κεφάλαιο 2 περιγράφεται η προτεινόμενη μέθοδος ανίχνευσης ακολουθιών bit με αλλοιώσεις και αναπτύσσεται καινοτόμος αρχιτεκτονική για την υλοποίησή της. Η ιδιαιτερότητα της μεθόδου έγκειται στο ότι απαιτεί μόνο πύλες ισοδυναμίας (XNOR), συγκριτές και αθροιστές μικρού μήκους (μέχρι 4 bit). Η ανίχνευση επιτυγχάνεται μέσω κυκλωματικών δομών που αντιστοιχούν σε δέντρα

αθροιστών τα οποία είναι δέντρα δυαδικής μορφής (binary tree) και παρέχουν το άθροισμα ενός πλήθους αριθμών σε $\log_2 n$ κύκλους. Η χρήση δέντρων αθροιστών και απλών συγκριτών έχει ως αποτέλεσμα την υλοποίηση της αρχιτεκτονικής σε περιβάλλον υλικού (FPGA) πολύ αποδοτικά, δίνοντας τη δυνατότητα ανίχνευσης ακολουθιών μεγάλου μήκους (>1000 bit) και καταναλώνοντας μόνο το $\sim 1\%$ των πόρων ενός μέσου FPGA. Η αρχιτεκτονική πετυχαίνει ακρίβεια λάθους 10^{-4} σε επίπεδο αλλοιώσεων 5×10^{-2} BER.

Στο Κεφάλαιο 3 χρησιμοποιείται η κεντρική ιδέα του Κεφαλαίου 2 στην πιο άμεση και πρακτική εφαρμογή της, τον συγχρονισμό πλαισίων δεδομένων σε ψηφιακές ζεύξεις. Η εξεταζόμενη αρχιτεκτονική εντοπίζει την ακριβή θέση των πλαισίων δεδομένων μέσω της ανίχνευσης ακολουθιών που έχουν τοποθετηθεί χρονικά μπροστά τους και συλλαμβάνει το ωφέλιμο φορτίο. Ως σύλληψη νοείται η απόδοση του φορτίου στην έξοδο του κυκλώματος. Η αρχιτεκτονική έχει ως είσοδο την λαμβανόμενη από-διαμορφωμένη ροή πληροφορίας πάνω στην οποία εφαρμόζει ψηφιακή συσχέτιση και με γνώμονα την τιμή της συσχέτισης αυτής εντοπίζει το πλαίσιο δεδομένων. Αποτελείται από τρεις βασικές μονάδες, την μονάδα παραθύρου, την μονάδα συσχέτισης και την μονάδα σύλληψης ωφέλιμου φορτίου. Η μονάδα παραθύρου απομονώνει ένα κομμάτι της ροής πληροφορίας πάνω στο οποίο η μονάδα συσχέτισης εφαρμόζει την πράξη της συσχέτισης με την ακολουθία ανίχνευσης. Η μονάδα ωφέλιμου φορτίου χρησιμοποιεί τα δεδομένα που παράγονται από την παραπάνω διαδικασία για να συλλάβει το ωφέλιμο φορτίο του πλαισίου. Λόγω των μεγάλων σε μήκος ακολουθιών συσχέτισης που χρησιμοποιούνται, η αρχιτεκτονική πετυχαίνει συγχρονισμό με ποσοστό λάθους 10~100 φορές καλύτερο από τις υπάρχουσες μεθόδους. Η απλή μέθοδος και η μινιμαλιστική κυκλωματική δομή που χρησιμοποιούνται επιτρέπουν την εξ' ολοκλήρου υλοποίηση της αρχιτεκτονικής αυτής σε πολύ φθηνή εμπορική πλατφόρμα (NEXYS 4 DDR) με το 40% των πόρων και υποστηριζόμενη ροή πληροφορίας κοντά στα 15Gbps.

Λέξεις κλειδιά: Ψηφιακός σχεδιασμός, Αρχιτεκτονικές υψηλών επιδόσεων, Field Programmable Logic Arrays (FPGAs), Δίκτυα Benes, Αλγόριθμοι δρομολόγησης, Δίκτυα τηλεπικοινωνιών, Ψηφιακές επικοινωνίες, Ανίχνευση ακολουθιών bit, Συγχρονισμός πλαισίων δεδομένων

Abstract

The rapid increase of network users has led to the continuous search for sustainable solutions aimed at scaling network characteristics such as capacity, speed and processing power. A viable solution to this problem is the introduction of FPGA (Field Programmable Gate Array) technology to key network infrastructure points due its increasing capabilities. In that context, the subject of the thesis is the utilization of FPGAs for the implementation of hardware architectures that perform network functions with great efficiency. The title of this thesis is “Programmable Circuit Architectures for High-Speed Networks” and it concerns the development of three novel hardware architectures presented in three respective chapters. The first architecture implements a Benes Network routing algorithm, the second a novel scheme for long bit sequence matching and detection under heavy interference, and the third a sync word based accurate frame synchronization module with minimal resources at very high bit rates.

Chapter 1 presents the Benes network routing architecture and explains its importance in the operation of electro-optical networks within modern data centers. The method used is based on the gradual routing of each layer of the network, which due to the recursive nature of the Benes network can be implemented with very simple methods and few resources. Routing is done by sorting the input-output pairs of the routing table given as input. The position occupied by the input-output pairs in this post-sorting table indicates the state in which each switch should be placed in order to achieve correct routing. After the end of the first layer routing, the input-output tables of the inner layers have been generated, which the algorithm uses to continue the routing. This process is followed until all the switches have been set to the correct state. Due to its simple construction, the circuit achieves the correct routing of 64×64 Benes networks in $3 \mu s$ with less than 15% of the resources of an average commercial FPGA.

Chapter 2 describes the proposed architecture of bit sequence detection under heavy interference, analyzes its innovation and explains the method it uses for its operation. The importance of the method lies in the fact that it uses only equivalence gates (XNOR), comparators and short-length adders (up to 4 bits). Detection is achieved through circuit structures called adder trees, which are binary trees that provide the sum of a set of numbers in $\log_2 n$ cycles. Modification of the adder trees with equivalence gates and comparators results in the implementation of the architecture in a hardware environment (FPGA) very efficiently, enabling the detection of long sequences (>1000 bits) and consuming only $\sim 1\%$ of the FPGA resources. The architecture achieves an error accuracy of 10^{-4} at a 5×10^{-2} BER level of corruption.

Chapter 3 uses the central idea of Chapter 2 in its most direct and practical application, the synchronization of data frames in digital links. The considered architecture locates the exact position of the data frames by detecting sync words placed in front them and captures the payload. The architecture has as input the received de-modulated bitstream on which it applies digital correlation and based on the value of this correlation it locates the data frame. Due to the long-correlated sync words used the architecture achieves synchronization with an error rate 10~100 times better than existing methods. Due to the simple method and the minimalist circuit structure inherited by Chapter 2, the full implementation of this architecture is achieved on a very cheap commercial platform (NEXYS 4 DDR) with 40% of the resources and a supported bit rate close to 15Gbps.

Keywords: Digital design, High-performance architectures, Field Programmable Logic Arrays (FPGAs), Benes networks, Routing algorithms, Telecommunications networks, Digital communications, Bit sequence detection, Data frame synchronization

Ευχαριστίες-Αντί προλόγου

Εντάχθηκα στο Εργαστήριο Φωτονικών Επικοινωνιών στις αρχές του 2020 εν μέσω πανδημίας Covid. Οι συνθήκες εκείνης της περιόδου έπαιξαν καθοριστικό ρόλο στο πως τελικά εξελίχθηκε η πορεία της διατριβής. Ερχόμενος από τον τομέα των ψηφιακών κυκλωμάτων η αλληλεπίδραση με τις φωτονικές τεχνολογίες και γενικά τις επικοινωνίες υψηλών ταχυτήτων με οδήγησε να προσεγγίσω τα προβλήματα με καινοτόμες μεθόδους και ως εκ τούτου, ο τρόπος σκέψης που αναπτύχθηκε κατά την διάρκεια αυτής της διαδικασίας αποτυπώθηκε στα τελικά συμπεράσματα που παρουσιάζονται σε αυτήν την εργασία.

Θέλω να ευχαριστήσω τον επιβλέποντα Καθηγητή κ.Ηρακλή Αβραμόπουλο για την εμπιστοσύνη που μου έδειξε σε όλη την διάρκεια της παραμονής μου στο Εργαστήριο έτσι ώστε απερίσπαστος και προσηλωμένος να ολοκληρώσω το έργο μου. Επίσης, θα ήθελα να ευχαριστήσω τον Δρ.

Παναγιώτη Γκρούμα, με το οποίο συνεργάστηκα στο πρώτο κομμάτι της διατριβής και η εποικοδομητική συνεισφορά του με βοήθησε πάρα πολύ στο να κάνω τα πρώτα μου βήματα καθώς επίσης τον συνεργάτη Δρ. Νίκο Λύρα για τις πολύτιμες συμβουλές του και την κ.Πόλυ για την βοήθειά της. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συνεχή συμπαράσταση και υποστήριξη που μου προσέφερε σε όλη αυτή τη διαδρομή.

Περιεχόμενα

Περίληψη.....	5
Abstract.....	8
Ευχαριστίες-Αντί προλόγου.....	11
Περιεχόμενα.....	13
Πίνακας συντημήσεων.....	14
Κεφάλαιο 1: Αλγόριθμος δρομολόγησης δικτύου Benes.....	15
1.1 Εισαγωγή.....	15
1.2 Βασικές αρχές.....	21
1.3 Προτεινόμενος αλγόριθμος.....	27
1.3.1 Αλγόριθμος.....	27
1.3.2 Απόδειξη λειτουργίας.....	28
1.4 Περιγραφή αρχιτεκτονικής.....	32
1.4.1 Αναστροφή πίνακα δρομολόγησης.....	34
1.4.2 Μηχανισμός ταξινόμησης.....	35
1.4.3 Δείκτης μη δρομολογημένου διακόπτη.....	37
1.4.4 Δρομολόγηση διακοπών και πίνακες δρομολόγησης επόμενων επιπέδων.....	38
1.5 Υλοποίηση.....	43
1.6 Δρομολόγηση δικτύων Benes γενικής μορφής.....	44
1.7 Βιβλιογραφία κεφαλαίου.....	50
Κεφάλαιο 2: Κύκλωμα αναγνώρισης ακολουθιών bit.....	54
2.1 Εισαγωγή.....	54
2.2 Βασικές αρχές.....	56
2.2.1 Δέντρο αθροιστής.....	58
2.2.2 Δέντρο συγκριτής.....	59
2.3 Περιγραφή αρχιτεκτονικής.....	59
2.3.1 Βασική μονάδα συσχέτισης.....	59
2.3.2 Ολικός ανιχνευτής αλληλουχιών.....	60
2.4 Βιβλιογραφία.....	62
Κεφάλαιο 3: Συγχρονισμός πλαισίων δεδομένων με αναγνώριση μοτίβων.....	63
3.1 Εισαγωγή.....	63
3.2 Βασικές αρχές.....	65
3.3 Περιγραφή αρχιτεκτονικής.....	68
3.3.1 Μονάδα παραθύρου (ΜΠ).....	68
3.3.2 Μονάδα συσχέτισης (ΜΣ).....	69
3.3.3 Μονάδα σύλληψης ωφέλιμου φορτίου (ΜΩΦ).....	71
3.4 Κυκλωματική Υλοποίηση.....	73
3.5 Σημασία των αποτελεσμάτων.....	75
3.6 Βιβλιογραφία κεφαλαίου.....	78
Παράρτημα : Πρωτόκολλο SPI.....	81
Λίστα σχημάτων.....	83
Λίστα πινάκων.....	85
Δημοσιεύσεις.....	86
Extended Abstract.....	87

Πίνακας συντμήσεων

FPGA: Field-programmable gate array

ASIC: Application-specific integrated circuit

VHDL: Hardware Description Language

SPI: Serial Peripheral Interface

AWGN: Additive White Gaussian Noise

SNR: Signal-to-Noise Ratio

Κεφάλαιο 1: Αλγόριθμος δρομολόγησης δικτύου Benes

1.1 Εισαγωγή

Η ενσωμάτωση δυνατοτήτων επικοινωνίας μέσω δικτύου (ethernet services) σε πληθώρα συσκευών και εφαρμογών (ηλεκτρικά είδη σπιτιού [1], αυτοκίνητα [2],) έχει οδηγήσει στην εκρηκτική αύξηση της διαδικτυακής κίνησης [3]. Θέματα όπως η κατανάλωση ενέργειας [4] και η επεξεργαστική ισχύς [5] αποτελούν βασικούς παράγοντες για τον σχεδιασμό και κατασκευή των υποδομών των σύγχρονων δικτύων. Μέσα σε αυτές τις υποδομές σημαντική θέση κατέχουν τα κέντρα δεδομένων (Data Centers) τα οποία ακολουθώντας τις ανάγκες των δικτύων χρειάζεται να κλιμακώνονται συνεχώς. Ένα από τα σημαντικότερα δομικά χαρακτηριστικά των κέντρων δεδομένων είναι τα δίκτυα διασύνδεσης τόσο σε επίπεδο κόμβων (ιεραρχική δομή δικτύων) όσο και σε επίπεδο επεξεργαστικής δυνατότητας (πολυεπεξεργαστικά συστήματα).

Τα δίκτυα διασύνδεσης αποτελούν το συνηθέστερο εργαλείο για την αποτελεσματική επικοινωνία και τον συντονισμό μεταξύ στοιχείων μέσα σε ένα σύστημα/δίκτυο. Δίκτυα τέτοιας φύσης χρησιμοποιούνται από επεξεργαστές, μονάδες μνήμης και περιφερειακές συσκευές έτσι ώστε να οδηγήσουν στην δημιουργία συνόλων με δυνατότητες κατά πολύ μεγαλύτερες από τις μεμονωμένες μονάδες. Χαρακτηριστικό παράδειγμα τέτοιας διαδικασίας είναι οι πολυπύρρηνοι επεξεργαστές, τα κέντρα δεδομένων και τα διαφόρων ειδών δίκτυα επικοινωνίας.

Ένα δίκτυο διασύνδεσης αποτελείται από τρία βασικά μέρη [6]. Τους κόμβους, τις διασυνδέσεις και τους διακόπτες. Ο συνδυασμός τους καθορίζει την τοπολογία του δικτύου. Κόμβοι είναι οι μονάδες που εκπέμπουν ή λαμβάνουν πληροφορία όπως π.χ. ένας επεξεργαστής μέσα σε ένα πολύ-επεξεργαστικό σύστημα ή ένας server μέσα σε ένα κέντρο δεδομένων [7]. Διασυνδέσεις είναι τα κανάλια μέσα στα οποία εκπέμπονται και λαμβάνονται οι πληροφορίες από τους κόμβους. Οι διασυνδέσεις ειδικά αποτελούν πολύ σημαντικό παράγοντα απόδοσης του δικτύου καθώς σε μεγάλο βαθμό καθορίζουν τη ταχύτητα και τη καθυστέρηση της μετάδοσης πληροφορίας [8]. Οι διακόπτες έχουν ως ρόλο τη σωστή κατεύθυνση της πληροφορίας μέσα από διάφορες διασυνδέσεις από τον πομπό στον δέκτη ώστε σε συνδυασμό με όλα τα παραπάνω στοιχεία να αποτελεί την τοπολογία του δικτύου[9].

Τα δίκτυα διασύνδεσης μπορούν να κατηγοριοποιηθούν ανάλογα με το μέγεθος, τις μεθόδους δρομολόγησης τους και τη τοπολογία τους. Οι κύριοι τύποι δικτύων είναι:

- Δίκτυα κοινού διαύλου: Σε αυτά, όλα τα στοιχεία του δικτύου έχουν κοινούς διαύλους τους οποίους χρησιμοποιούν για επικοινωνία. Αυτό τα καθιστά πολύ πιο απλά στη κατασκευή, γεγονός όμως που πληρώνεται με τη δυσκολία κλιμάκωσης τους σε μεγάλα μεγέθη [10].

- Δίκτυα από σημείο προς σημείο (point to point): Σε αυτά τα δίκτυα οι κόμβοι συνδέονται απευθείας εξασφαλίζοντας άμεση επικοινωνία χαμηλής καθυστέρησης και μεγάλης χωρητικότητάς. Προσφέρουν κλιμάκωση στο μέγεθος αλλά δεν είναι τόσο απλά στη κατασκευή όπως η κατηγορία δικτύων κοινού διαύλου [11].
- Δίκτυο τύπου crossbar: Αυτά τα δίκτυα έχουν τη μορφή δισδιάστατου πλέγματος όπου οι κάθετες και οριζόντιες γραμμές είναι οι διασυνδέσεις με τους κόμβους. Στο σημείο τομής αυτών των διασυνδέσεων βρίσκονται οι διακόπτες του δικτύου. Η ιδιαιτερότητα αυτής της δομής δικτύου είναι να επιτρέπει τη χρησιμοποίηση πολλών εξόδων από μία είσοδο ταυτόχρονα. Λόγω αυτού έχουν πολύ μεγάλη αποδοτικότητα αλλά το κόστος και η πολυπλοκότητα κατασκευής τους αυξάνονται ραγδαία με την αύξηση του μεγέθους τους [12].
- Πολυεπίπεδα δίκτυα: Όπως μαρτυρά και το όνομα πρόκειται για δίκτυα με πολλά επίπεδα διακοπών και συνδέσεων που στηρίζονται στην κατάλληλη ρύθμιση των εν λόγω στοιχείων έτσι ώστε να κατευθύνουν μια είσοδο στην κατάλληλη έξοδο. Είναι μία μέση λύση μεταξύ πολυπλοκότητας κατασκευής, κλιμάκωσης και απόδοσης, γι' αυτό τελικά προτιμώνται για εφαρμογές μεσαίων και μεγάλων μεγεθών. Παράδειγμα τέτοιων δικτύων είναι τα δίκτυα Clos και πιο συγκεκριμένα μια υποκατηγορία τους, τα δίκτυα Benes που χρησιμοποιούνται σε πληθώρα εφαρμογών [13].

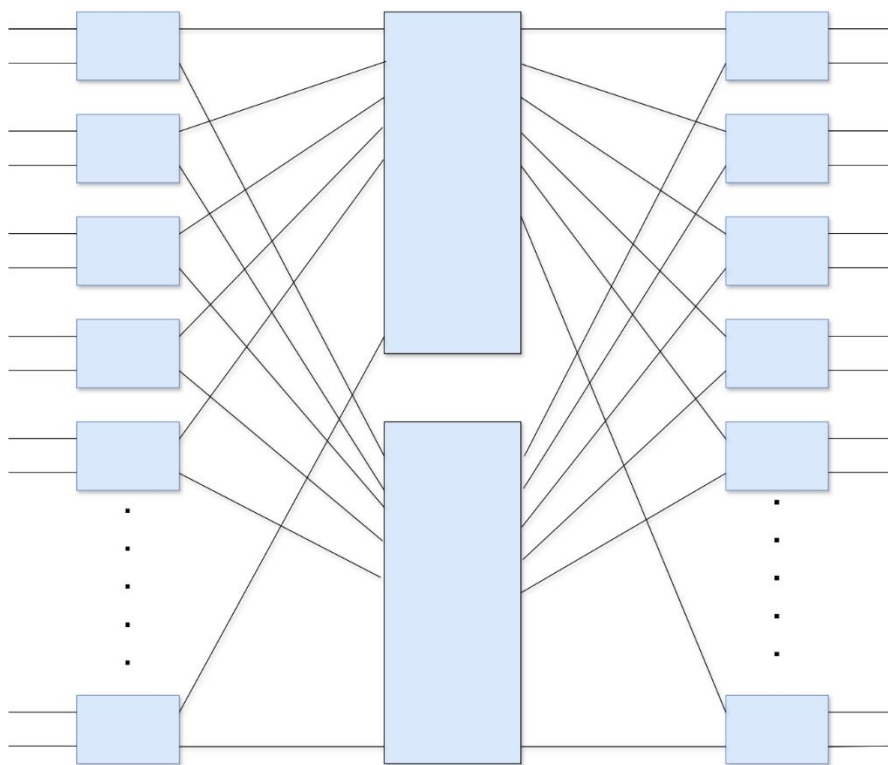
Κάποιες πολύ σημαντικές εφαρμογές στις οποίες τα δίκτυα διασύνδεσης παίζουν κομβικό ρόλο μπορούν να συνοψιστούν ως εξής :

- Υπολογιστικά συστήματα υψηλής απόδοσης.: Υπολογιστές με πολύ μεγάλο αριθμό επεξεργαστών βασίζονται στην αποδοτικότητα των δικτύων διασυνδέσεών τους ώστε να επιτελούν υπολογιστικά βαριές λειτουργίες όπως λειτουργίες cloud (cloud computing) [14], προσομοίωσης [15], ανάλυση μεγάλου όγκου δεδομένων (big data analytics) [16] και νευρωνικά δίκτυα [17]. Οι αυξημένες απαιτήσεις αυτών των εργασιών καθιστούν απαραίτητη τη γρήγορη, χωρίς λάθη και καθυστερήσεις επικοινωνία μεταξύ των επεξεργαστικών μονάδων μέσα από κατάλληλα δίκτυα διασύνδεσης.
- Πολυπύρηντοι επεξεργαστές: Κατεβαίνοντας ένα επίπεδο παρακάτω, οι ίδιοι οι επεξεργαστές αποτελούνται σήμερα από πολλούς πυρήνες, μονάδες μνήμης και άλλες περιφερειακές μονάδες οι οποίες είναι απαραίτητο να συγχρονιστούν με τη καλύτερη δυνατή ακρίβεια. Τα δίκτυα διασύνδεσης εξασφαλίζουν την αξιόπιστη λειτουργία του επεξεργαστή σαν μονάδα και κατ' επέκταση βοηθούν στη σωστή λειτουργία των συστημάτων υψηλής απόδοσης [18].
- Κέντρα δεδομένων(data centers): Τα κέντρα δεδομένων είναι κομβικά για τη τέλεση μεγάλης κλίμακας λειτουργιών όπως οι λειτουργίες cloud, υπηρεσίες δικτύου (web services) κ.α. Τα δίκτυα διασύνδεσης επιστρατεύονται σε αυτή τη περίπτωση για να εξασφαλίσουν την ομαλή, αξιόπιστη και χωρίς καθυστερήσεις επικοινωνία μεταξύ server, συσκευών δικτύου (modem, router) και χώρων αποθήκευσης [19].

Η παρούσα εργασία επικεντρώνεται σε ένα συγκεκριμένο τύπο δικτύου που εντάσσεται στην υποκατηγορία των πολυεπίπεδων δικτύων [13]. Το δίκτυα Benes [20] ανήκουν στην ευρύτερη κατηγορία πολυεπίπεδων δικτύων Clos [21] και πήραν το όνομα τους από τον Τσέχο μαθηματικό Václav Edvard "Vic" Beneš ο οποίος ερεύννησε τις ιδιότητες τους εκτεταμένα την δεκαετία του 60. Η πιο χρήσιμη ιδιότητα των δικτύων Benes, η οποία βοήθησε στην εξάπλωσή τους, είναι ότι μπορούν να εξυπηρετήσουν κάθε πιθανό συνδυασμό εισόδων-εξόδων χωρίς να υπάρχει παρεμβολή μεταξύ των δρομολογήσεων. Είναι δηλαδή non-blocking δίκτυα.

Η βασική δομική μονάδα του δικτύου Benes είναι οι διακόπτες 2×2 οι οποίοι διατάσσονται με τέτοιο τρόπο έτσι ώστε να αποδίδουν τις εξής ιδιότητες στο συνολικό δίκτυο:

1. Το Benes είναι ένα δίκτυο αναδρομικής μορφής που σημαίνει ότι ένα $N \times N$ δίκτυο (N εισόδοι, N εξόδοι) χρειάζεται δύο $N/2 \times N/2$ δίκτυα Benes για να κατασκευαστεί. Τα δίκτυα $N/2 \times N/2$ αναλύονται περισσότερο μέχρι να εκφυλιστούν σε 2×2 διακόπτες. Η γενική μορφή ενός $N \times N$ δικτύου Benes φαίνεται στο Σχήμα 1-1.



Σχήμα 1-1: Γενική μορφή δικτύου Benes.

2. Το Benes είναι αναδιατάξιμο χωρίς μπλοκάρισμα (Rearrangeable Non-blocking) και μπορεί να εξυπηρετήσει κάθε πιθανό πίνακα δρομολόγησης (αντιστοιχία εισόδων εξόδων) χωρίς να μπλοκάρονται διαδρομές, με την μόνη προϋπόθεση ότι κάθε φορά που έρχεται νέος πίνακας δρομολόγησης η κατάσταση των διακοπών υπολογίζεται από την αρχή.

3. Λόγω της αναδρομικής μορφής του ο αριθμός εισόδων του πρέπει να είναι πάντα δύναμη του 2, $2^r = N$. Αυτό έχει ως αποτέλεσμα το δίκτυο να έχει λογαριθμικό βάθος σε σχέση με τον αριθμό εισόδων. Συγκεκριμένα ένα δίκτυο Benes N εισόδων διαθέτει $2 \cdot \log_2 N - 1$ στάδια (stages).

Στα επόμενα μετά την εισήγησή τους χρόνια, η έρευνα πάνω στα δίκτυα Benes συνεχίστηκε τόσο σε επίπεδο αλγορίθμων δρομολόγησης όσο και σε επίπεδο πρακτικής υλοποίησης και τα δίκτυα Benes έγιναν ακόμα περισσότερο ελκυστικά με αποτέλεσμα να υιοθετηθούν σε πολλούς εμπορικούς τομείς όπου η λογαριθμική σχέση μεταξύ αριθμού εισόδων και σταδίων, αλλά και η ικανότητα να ικανοποιείται κάθε πιθανό σχέδιο δρομολόγησης είναι απαραίτητες. Ένας από τους πιο σημαντικούς τομείς εφαρμογής των δικτύων Benes είναι οι οπτικοί μεταγωγείς μεγάλης κλίμακας οι οποίοι χρησιμοποιούνται στα κέντρα δεδομένων (data centers) [22].

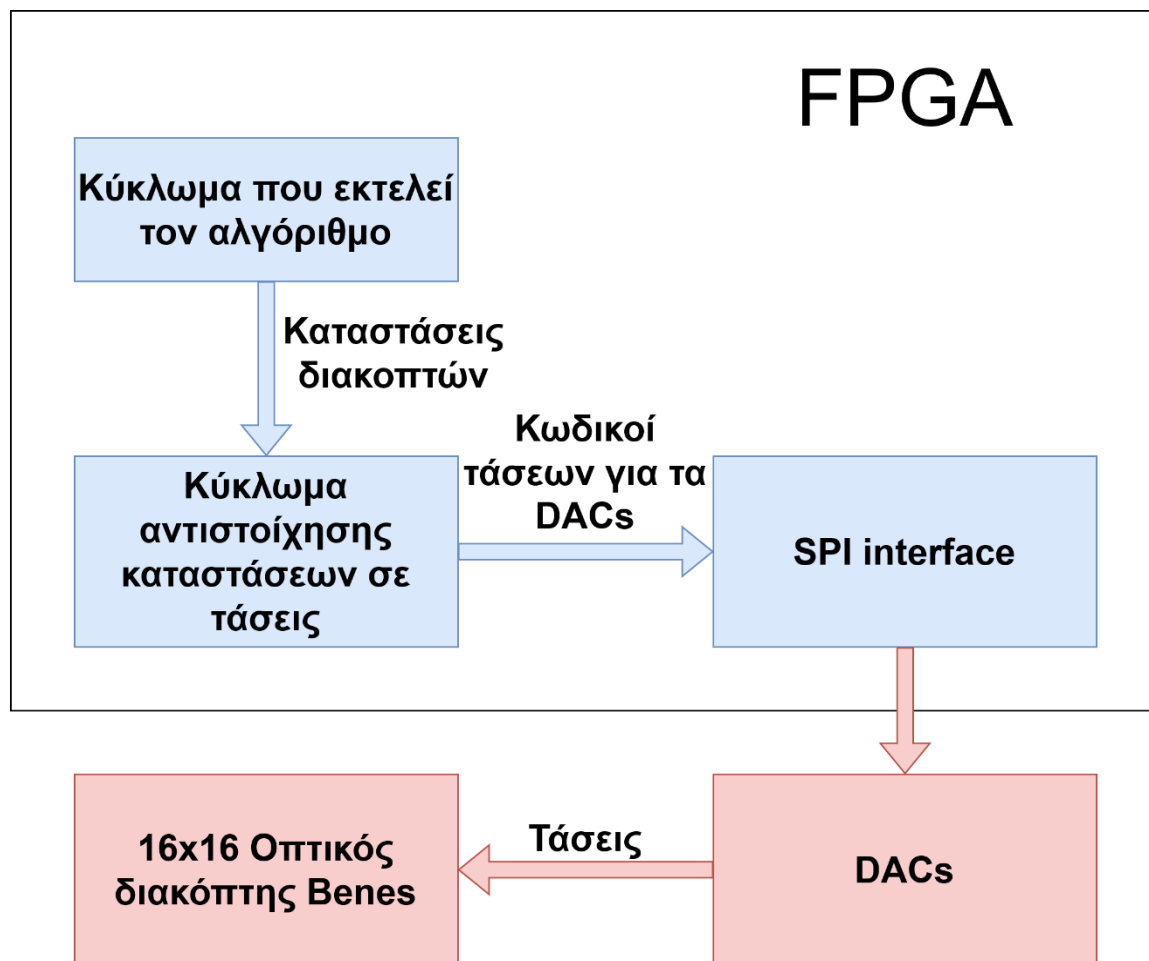
Η χρησιμοποίηση οπτικών μεταγωγέων στα κέντρα δεδομένων μελετάται εδώ και καιρό [23] όμως πρόσφατα λόγω του κορεσμού των αμιγώς ηλεκτρικών μεταγωγέων έχει αναδειχθεί η χρήση ισοδύναμων με ηλεκτρική οδήγηση [24]. Ο λόγος για αυτήν την τάση είναι τα πλεονεκτήματα που παρέχουν οι ηλεκτροοπτικοί διακόπτες [25] κάποια εκ' των οποίων είναι:

1. η δυνατότητα κλιμάκωσης της χωρητικότητας και του αριθμού των κόμβων,
2. η ενεργειακή αποδοτικότητα,
3. η δυνατότητα αναβάθμισης (modular upgradability) και
4. ο περιορισμός του κόστους από την κατάργηση των μεγαλύτερων και ενεργειακά κοστοβόρων ηλεκτρικών μεταγωγέων ενώ διατηρούνται οι μικρότεροι πιο ευέλικτοι στα άκρα του δικτύου.

Το δίκτυο Benes συγκεκριμένα συναντάται συχνά σε μορφή οπτικού διακόπτη σε διάφορα μεγέθη (16×16 [26], 32×32 [27]) λόγω των χρήσιμων ιδιοτήτων του (rearrangably non-blocking, λογαριθμικό βάθος, απλή διαδικασία δρομολόγησης).

Ο αλγόριθμος δρομολόγησης δικτύου Benes ο οποίος θα παρουσιαστεί σε αυτήν την ενότητα χρησιμοποιήθηκε στα πλαίσια του Ευρωπαϊκού έργου POETCIS [28-29] για την δρομολόγηση ενός οπτικού μεταγωγέα 16×16 που ακολουθεί την αρχιτεκτονική Benes και αποτελεί την πιο αποδοτική υλοποίηση σε υλικό από άποψη κατανάλωσης πόρων και ταχύτητας από οποιαδήποτε άλλη στην βιβλιογραφία. Παρ' ότι τα οπτικά χαρακτηριστικά του συγκεκριμένου οπτικού μεταγωγέα είναι πολύ σημαντικά, στο παρών κεφάλαιο θα επικεντρωθούμε μόνο στην δρομολόγηση του ως δίκτυο Benes. Οι 2×2 διακόπτες [30] μιμούνται τη μαθηματική λειτουργία των switches και έχουν δύο πιθανές καταστάσεις, cross και bar. Cross είναι η κατάσταση στην οποία η είσοδος 1 πάει στην είσοδο 2 και η είσοδος 2 στην είσοδο 1 και bar όπου συμβαίνει το αντίθετο ($1 \rightarrow 1$, $2 \rightarrow 2$). Η κατάσταση αυτή ελέγχεται από υποδοχές τάσεων στις οποίες πρέπει να εφαρμοστούν συγκεκριμένες τάσεις έτσι ώστε ένας διακόπτης να λάβει την μία ή την άλλη κατάσταση. Καθώς είναι απαραίτητο η

δρομολόγηση του συνολικού δικτύου να γίνεται όσο τον δυνατόν γρηγορότερα, ο μόνος τρόπος για να επιτευχθεί ο στόχος ήταν η υλοποίηση του αλγορίθμου ως κύκλωμα. Ο αλγόριθμος, που παρουσιάζεται αναλυτικά στο Κεφάλαιο 1, έχει ως είσοδο έναν πίνακα δρομολόγησης σύμφωνα με τον οποίο αντιστοιχίζονται είσοδοι σε εξόδους. Ο πίνακας αυτός χρησιμοποιείται έτσι ώστε να υπολογιστούν οι σωστές καταστάσεις για κάθε διακόπτη (bar ή cross). Αφού γίνει αυτό, οι καταστάσεις αντιστοιχίζονται στα κατάλληλα ζευγάρια τάσεων τα οποία στέλνονται μέσω πρωτοκόλλου SPI[31] σε DAC που οδηγούν τους υποδοχείς τάσεων του μεταγωγέα για κάθε διακόπτη. Το Σχήμα 1-2 παρουσιάζει την λογική αλληλουχία της διαδικασίας.



Σχήμα 1-2: Λογικό διάγραμμα του κυκλώματος δρομολόγησης μέσα στην διάταξη.

Το δίκτυο Benes είναι μια αποδεδειγμένα αποτελεσματική λύση για διασύνδεση θυρών σε δίκτυα. Για τον λόγο αυτό έχουν προταθεί πληθώρα διαφορετικών μεθόδων δρομολόγησης, παράλληλες και γραμμικές. Ένας από τους γραμμικούς αλγόριθμους είναι ο αλγόριθμος βρόχου Orferman[32], που ξεκινά από τους διακόπτες στα εξωτερικά επίπεδα και καταλήγει στο κεντρικό με την πιθανότητα όμως να χρειαστεί οπισθοδρόμηση. Χωρίζει το δίκτυο σε μικρότερα δίκτυα και τα δρομολογεί αναδρομικά όλα, δημιουργώντας μία πλήρης διαδρομή. Άλλος γραμμικός αλγόριθμος είναι αυτός του Waksman [33] που πρότείνει μια αναδρομική διαδικασία για εκτέλεση σε συστήματα με μία

επεξεργαστική μονάδα. Και τα δύο παραδείγματα έχουν πολυπλοκότητα $n \cdot \log n$, που είναι η χαμηλότερη δυνατή για μονοεπεξεργαστικό σύστημα [32]. Στον παράλληλο τομέα, οι Nassimi και Sahni [34][35] παρουσίασαν έναν παράλληλο αλγόριθμο που είναι σημαντικά ταχύτερος από τις γραμμικές μεθόδους, αλλά η πολυπλοκότητά του ποικίλλει από $(\log n)^2$ έως $(\log n)^4$ ανάλογα με τον τύπο του συστήματος στο οποίο εφαρμόζεται. Ο παράλληλος αλγόριθμος του Lee [36] είναι από τα λίγα παραδείγματα που έχουν υλοποιηθεί σε πλατφόρμα υλικού και προσφέρει τη χαμηλότερη πολυπλοκότητα $(\log n)^2$ καταναλώνοντας μόνο $n/2$ μονάδες επεξεργασίας. Υπάρχουν και άλλοι αλγόριθμοι που δρομολογούν Benes δίκτυα [37][38] όμως είναι ουσιαστικά όλοι παραλλαγές των προαναφερθέντων. Η λύση που παρουσιάζεται στο [39] είναι η πλησιέστερη, από άποψη λειτουργίας, στη λύση που καλύπτεται σε αυτό το κεφάλαιο. Παρόλο που είναι μία από τις λύσεις που χρησιμοποιούν δισδιάστατους (2D) πίνακες, σε ορισμένες περιπτώσεις, παράγει την ίδια ρύθμιση για το δίκτυο. Η συγκεκριμένη δημοσίευση όμως είναι καθαρά θεωρητική και δεν περιλαμβάνει κάποια πρακτική υλοποίηση σε υλικό ώστε να υπάρχει σύγκριση.

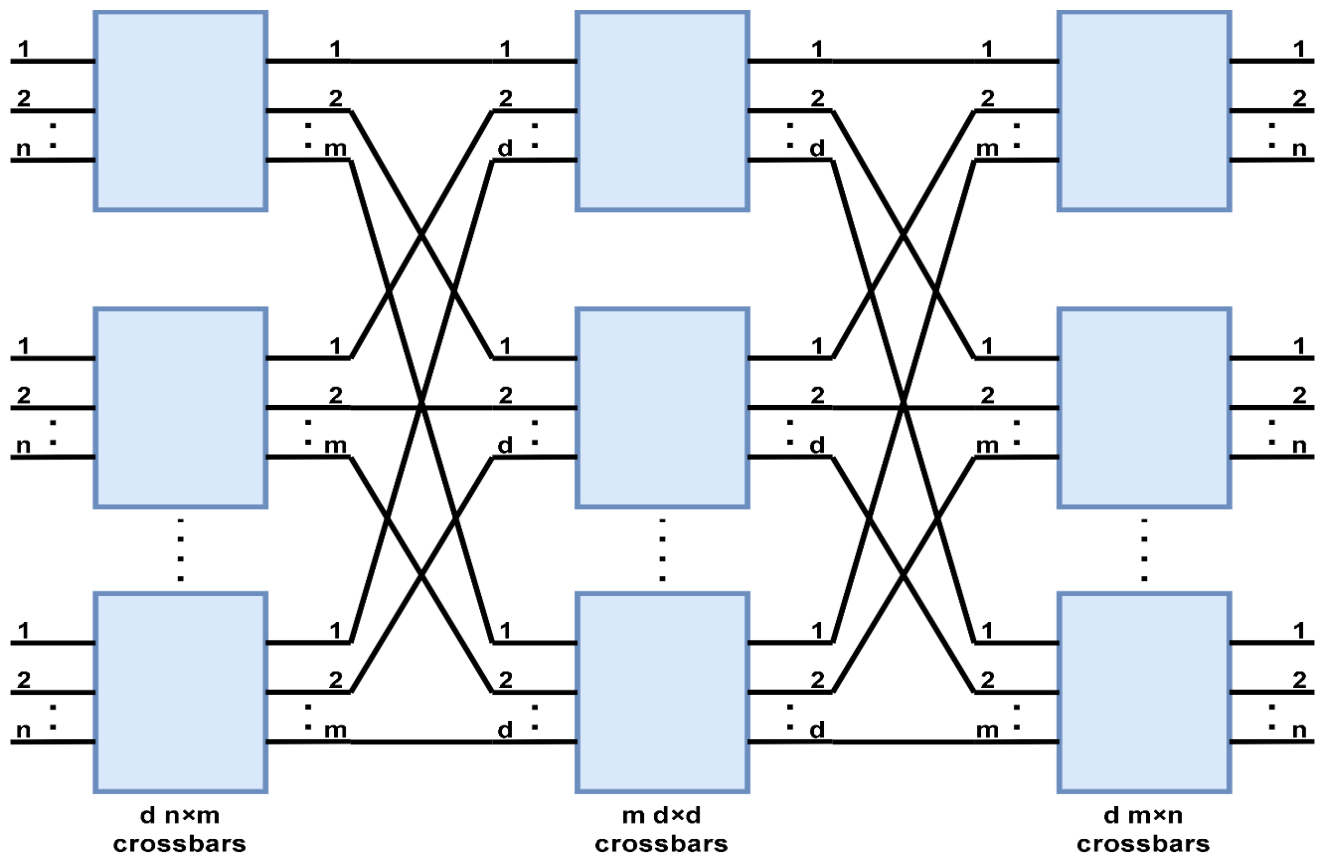
Έχουν γίνει προσπάθειες να μεταφραστούν ορισμένοι αλγόριθμοι σε εφαρμογές υλικού [40]. Ωστόσο όλες αυτές οι προσπάθειες επικεντρώνονται κυρίως σε παράλληλους αλγόριθμους δρομολόγησης. Επιπλέον, όλες οι ήδη καθιερωμένες μέθοδοι, γραμμικές και παράλληλες, περιλαμβάνουν σύνθετους υπολογισμούς. Οι γραμμικές μέθοδοι εστιάζουν γενικά στο χρήση μεταθέσεων πάνω σε δυσδιάστατους πίνακες (2D) [38], με ορισμένες από αυτές να απαιτούν οπισθοδρόμηση(backtracking) σε διακόπτες μέσα στο δίκτυο που ήδη έχουν τεθεί σε κατάσταση [32], μειώνοντας έτσι την απόδοσή τους. Οι παράλληλες υλοποιήσεις επικεντρώνονται γύρω από τον αλγόριθμο του Lee [36], ο οποίος αν και πιο γρήγορος, καταναλώνει πολλούς πόρους όταν μεταφράζεται σε υλικό. Σε αυτό το κεφάλαιο παρουσιάζεται μια πρωτότυπη, απλή αλγοριθμική λύση μαζί με την κυκλωματική υλοποίησή της που είναι πιο αποδοτική σε σχέση με τις γραμμικές μεθόδους από άποψη χρόνου ολοκλήρωσης και πολύ πιο εύκολο να υλοποιηθεί σε σχέση με τις παράλληλες μεθόδους.

Η υλοποίηση είναι γενικού χαρακτήρα και πειραματικά επαληθεύτηκε για διάφορους αριθμούς εισόδων. Εκμεταλλευόμενοι την αναδρομική φύση του δικτύου, το πρόβλημα δρομολόγησης ανάγεται σε μια στοιχειώδη ταξινόμηση των ζευγών εισόδου-εξόδου. Η θέση που καταλαμβάνει κάθε ζεύγος στην ταξινομημένη λίστα καθορίζει την κατάσταση των εσωτερικών διακοπών. Όλοι οι διακόπτες δρομολογούνται γραμμικά και δεν υπάρχει οπισθοδρόμηση. Το κύριο εμπόδιο σε αυτό το εγχείρημα ήταν η συμπίκνωση των πολύπλοκων πράξεων και μεταθέσεων πάνω σε δυσδιάστατους πίνακες που χρησιμοποιούνται από του άλλους αλγορίθμους. Η υπέρβαση αυτού του εμποδίου ήταν καθοριστική ώστε να είναι ο προτεινόμενος αλγόριθμος πιο αποδοτικός από την βιβλιογραφία.

Η οργάνωση του κεφαλαίου έχει ως εξής. Η ενότητα 1.2 περιέχει τις βασικές προτάσεις που αποτελούν τα θεμέλια της αρχιτεκτονικής. Η ενότητα 1.3 περιλαμβάνει την θεωρητική απόδειξη λειτουργίας του αλγορίθμου. Η ενότητα 1.4 καλύπτει την υλοποίηση του ως κυκλωματική αρχιτεκτονική και στην ενότητα 1.5 βρίσκονται τα αποτελέσματα της υλοποίησης πάνω στις πλατφόρμες υλικού. Τέλος, στην ενότητα 1.6 η χρήση του αλγόριθμου επεκτείνεται θεωρητικά σε Benes δίκτυα γενικής μορφής χωρίς όμως υλοποίηση.

1.2 Βασικές αρχές

Το δίκτυο Benes είναι μια υποκατηγορία δικτύων διασύνδεσης που προέκυψε από την οικογένεια Clos. Τα Clos δίκτυα έχουν τρία στάδια: το στάδιο εισόδου, το μεσαίο στάδιο και το στάδιο εξόδου, κάθε στάδιο αποτελείται από έναν συγκεκριμένο αριθμό διακοπών crossbar. Τρεις αριθμοί χρησιμοποιούνται για την περιγραφή ενός δικτύου Clos (n, m, d). Το στάδιο εισόδου αποτελείται από d $n \times m$ διακόπτες crossbar (συνολικός αριθμός εισόδων $d \times n = N$), το μεσαίο στάδιο αποτελείται από m $d \times d$ διακόπτες και το στάδιο εξόδου αποτελείται από d $m \times n$ διακόπτες (συνολικός αριθμός εισόδων/εξόδων $d \times n = N$). Ένας διακόπτης μπορεί να δρομολογήσει οποιαδήποτε είσοδο σε οποιαδήποτε έξοδο, υπό την προϋπόθεση ότι η έξοδος δεν είναι ήδη κατειλημμένη. Το Σχήμα 1-3 παρουσιάζει ένα δίκτυο Clos τριών σταδίων.

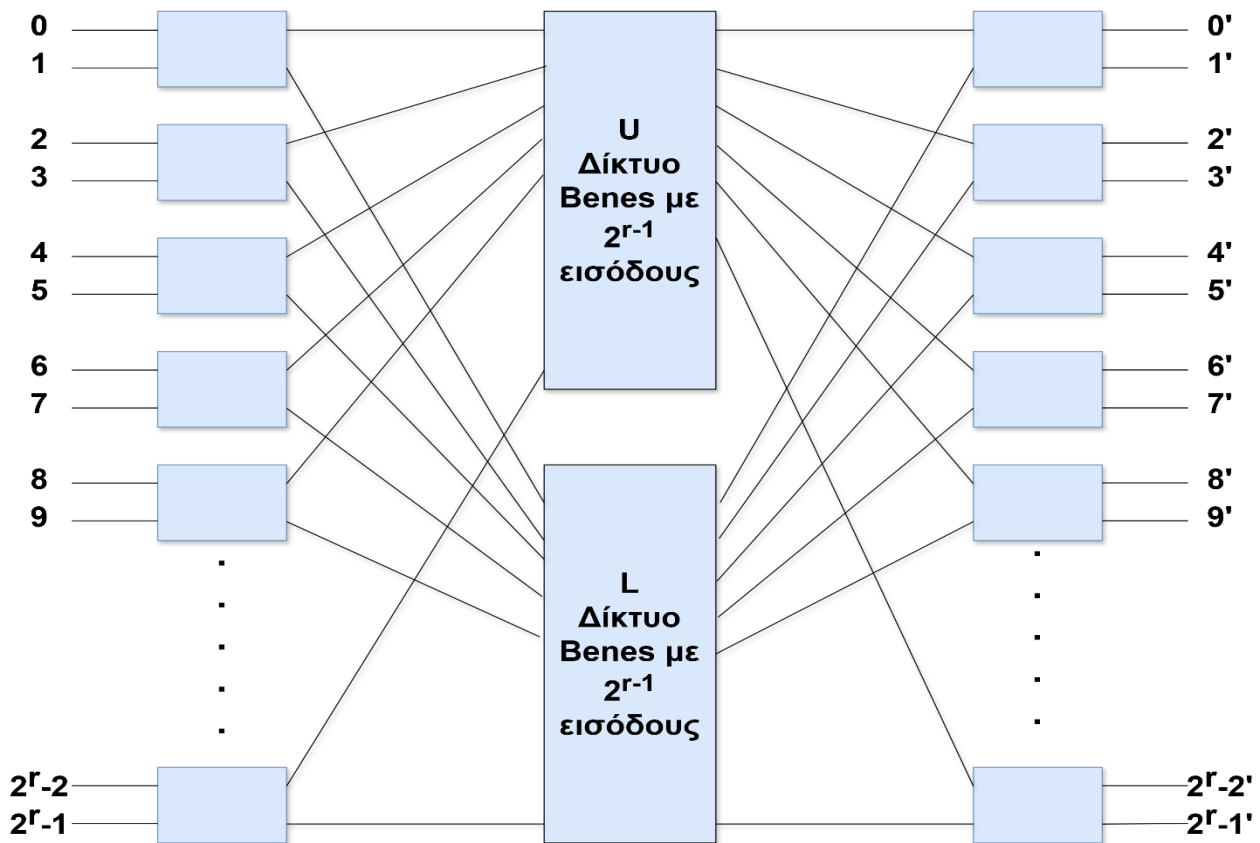


Σχήμα 1-3: Γενική μορφή Clos δικτύου.

Οι εισόδοι που εισέρχονται σε crossbar εισόδου μπορούν να δρομολογηθούν στην επιθυμητή έξοδο ενός crossbar εξόδου μέσω οποιουδήποτε διακόπτη μεσαίας βαθμίδας. Διαφορετικές τιμές για n, m, d συνεπάγονται διαφορετικές ιδιότητες για το δίκτυο Clos. Η μορφή του δικτύου Clos μπορεί να γενικευτεί σε οποιονδήποτε περιττό αριθμό σταδίων. Αντικαθιστώντας κάθε διακόπτη crossbar στη μεσαία βαθμίδα με ένα νέο δίκτυο Clos 3 σταδίων μπορεί να κατασκευαστεί ένα δίκτυο Clos 5 σταδίων. Με την επανάληψη της διαδικασίας μπορούν να κατασκευαστούν δίκτυα Clos 7, 9, 11 σταδίων. Έχει αποδειχθεί ότι όταν $m \geq n$, το δίκτυο Clos είναι αναδιατάξιμο μη-μπλοκαρισμένο (rearrangably non-blocking) που σημαίνει ότι όλες οι πιθανές συνθήκες δρομολόγησης ($N!$) μπορούν να εκπληρωθούν εάν όλα τα στοιχεία μεταγωγής αναδιατάσσονται κάθε φορά που υπάρχει αλλαγή στον πίνακα δρομολόγησης. Το δίκτυο Benes είναι ένα δίκτυο Clos όπου $n=m=2$. Σε ένα δίκτυο Benes μεγέθους $N \times N$ (εισόδους $\times$ εξόδοι) ο αριθμός των εισόδων/εξόδων είναι δύναμη του 2 ($N=2^r$). Όλοι οι διακόπτες στο δίκτυο Benes είναι 2×2 .

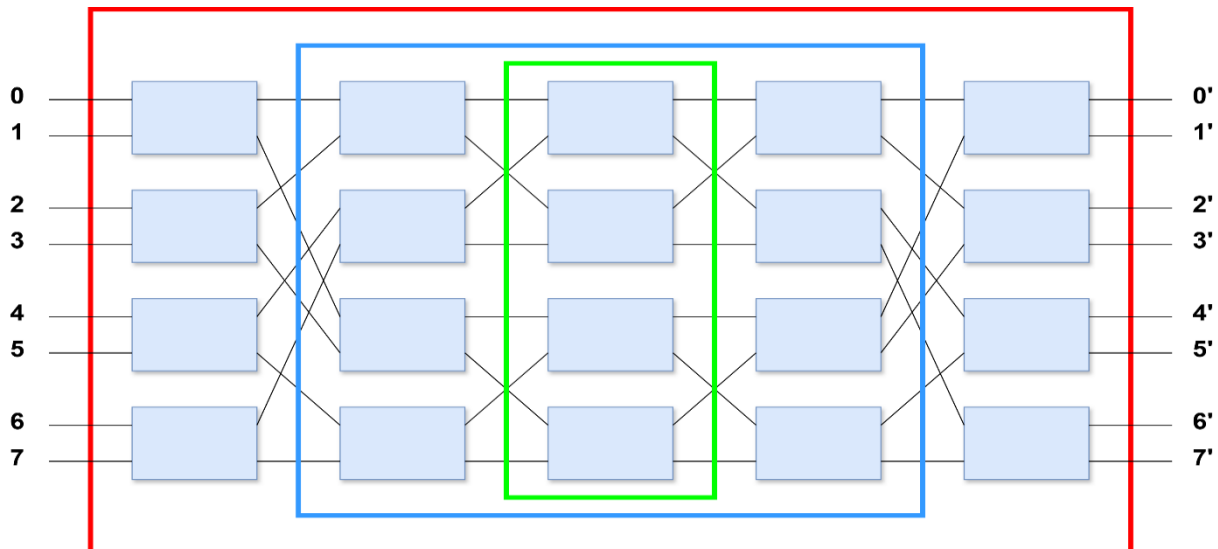
Η μορφή του δικτύου Benes οδηγεί σε δύο πολύ σημαντικά χαρακτηριστικά. Πρώτον, ως μέρος του δικτύου Clos με διακόπτες 2×2 , το δίκτυο Benes είναι ένα δίκτυο διασύνδεσης με δυνατότητα αναδιάταξης, χωρίς μπλοκάρισμα. Δεύτερον, ένα δίκτυο Benes μεγέθους N μπορεί να κατασκευαστεί από δύο δίκτυα Benes στοιχείων $N/2$ τα οποία με τη σειρά τους μπορούν να κατασκευαστούν από δύο $N/4$ Benes δίκτυα το καθένα. Ο αλγόριθμος που παρουσιάζεται σε αυτό το κεφάλαιο εκμεταλλεύεται αυτήν την ιδιότητα για τη δρομολόγηση του δικτύου. Η πράξη της δρομολόγησης συνεπάγεται την καλωδίωση όλων των εσωτερικών διακοπών 2×2 crossbar με τρόπο που να ικανοποιεί έναν δεδομένο πίνακα δρομολόγησης. Ο αλγόριθμος είναι επεκτάσιμος και πρακτικά εφαρμόσιμος για πολλές εισόδους (16×16 , 32×32 , 64×64) με ελάχιστους πόρους σε πλατφόρμες χαμηλού κόστους. Παρόλο που δημιουργήθηκε για έναν γενικό αριθμό εισόδων, οι αριθμοί και οι θεωρητικές εξηγήσεις χρησιμοποιούν μόνο ένα δίκτυο 8×8 για να επιδείξουν τη λειτουργικότητα του αλγορίθμου. Η δρομολόγηση μεγαλύτερων δικτύων θα ήταν αδύνατο να ακολουθηθεί στο χαρτί.

Πρόταση 1: Η γενική αναδρομική μορφή ενός δικτύου Benes φαίνεται στο *Σχήμα 1-4*. Έχει $N = 2^r$, $r > 1$, εισόδους και $N = 2^r$ εξόδους. Οι θύρες στην αριστερή πλευρά ($0, 1, 2, \dots, 2^r-1$) μπορούν να θεωρηθούν οι θύρες εισόδου και οι θύρες στη δεξιά πλευρά ($0', 1', 2', \dots, 2^r-1'$) οι θύρες εξόδου. Επειδή τα δίκτυα Benes είναι συμμετρικά, οι θύρες εισόδου και εξόδου μπορούν να αντιστραφούν, εφόσον η δρομολόγηση είναι ίδια. Υπάρχουν 2^{r-1} διακόπτες εισόδου 2^{r-1} διακόπτες εξόδου και 2 υποδίκτυα στο κέντρο, πάνω (U) και κάτω (L). Αυτά τα 2 υποδίκτυα είναι δίκτυα Benes με 2^{r-1} εισόδους, το ήμισυ του αρχικού.



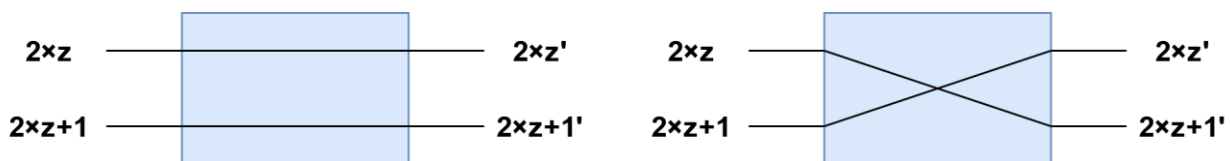
Σχήμα 1-4: Εξώτατο επίπεδο του Δικτύου Benes. Τα δύο υποδίκτυα έχουν μισό αριθμό εισόδων.

Πρόταση 2: Η φόρμα που φαίνεται στο Σχήμα 1-4 αποτελεί το εξώτατο επίπεδο του Δικτύου Benes που έχει τους ίδιους διακόπτες εισόδου και εξόδου με το ίδιο το δίκτυο. Συνολικά, σε ένα δίκτυο με $N=2^r$ εισόδους υπάρχουν r επίπεδα ($\log_2 N$). $r-1$ επίπεδα με διακόπτες 2^r (είσοδος και έξοδος) και 1 επίπεδο με διακόπτες 2^{r-1} . Το επίπεδο $k \in [0, r-1]$ έχει 2^k υποδίκτυα με $2^{(r-k)}$ εισόδους, εξόδους το καθένα. Το Σχήμα 1-5 παρουσιάζει ένα δίκτυο Benes 8 εισόδων-εξόδων. Όπως μπορούμε να δούμε, είναι ένα δίκτυο Benes με $2^3 = 8$ ($r = 3$) θύρες εισόδου-εξόδου. Το κόκκινο είναι το επίπεδο 0 ($k = 0$, $2^{3-0} = 8$ θύρες εισόδου-εξόδου ανά υποδίκτυο, $2^0 = 1$ υποδίκτυο που είναι το αρχικό), το μπλε είναι το επίπεδο 1 ($k = 1$, $2^{3-1} = 4$ είσοδοι-θύρες εξόδου ανά υποδίκτυο, $2^1 = 2$ υποδίκτυα) και το πράσινο είναι το επίπεδο 2 ($k = 2$, $2^{3-2} = 2$ θύρες εισόδου-εξόδου ανά υποδίκτυο, $2^2 = 4$ διακόπτες/υποδίκτυα). Με κάθε επίπεδο, ο αριθμός των υποδικτύων διπλασιάζεται και ο αριθμός των εισόδων ανά υποδίκτυο μειώνεται στο μισό. Για το λόγο αυτό, ο αριθμός των διακοπών εισόδου και εξόδου κάθε επιπέδου παραμένει ίδιος, εκτός από το τελευταίο. Το επίπεδο $k = r-1$ είναι το κέντρο του δικτύου Benes και οι διακόπτες και τα υποδίκτυα είναι ισοδύναμα (ένας διακόπτης είναι ένα υποδίκτυο).



Σχήμα 1-5: Εσωτερική δομή δικτύου Benes 8×8. Το επίπεδο 0 σημειώνεται με κόκκινο, το 1 με μπλε και το 2 με πράσινο.

Πρόταση 3: Ο διακόπτης (είσοδος ή έξοδος) z έχει 2 θύρες εισόδου, $2 \times z$, και $2 \times z + 1$ και 2 θύρες εξόδου, $2 \times z'$ και $2 \times z + 1'$. Για κάθε διακόπτη, είναι δυνατές 2 ρυθμίσεις (Σχήμα 1-6). Όταν το $2 \times z$ είναι συνδεδεμένο στο $2 \times z + 1'$ και το $2 \times z + 1$ συνδέεται στο $2 \times z'$, η ρύθμιση είναι σταυρωτή (cross). Όταν το $2 \times z$ είναι συνδεδεμένο στο $2 \times z'$ και το $2 \times z + 1$ συνδέεται στο $2 \times z + 1'$, η ρύθμιση είναι μπάρα (bar). Ο σκοπός του αλγορίθμου δρομολόγησης είναι η ικανοποίηση των απαιτούμενων διαδρομών δρομολόγησης με τη σωστή ρύθμιση όλων των διακοπών του δικτύου.



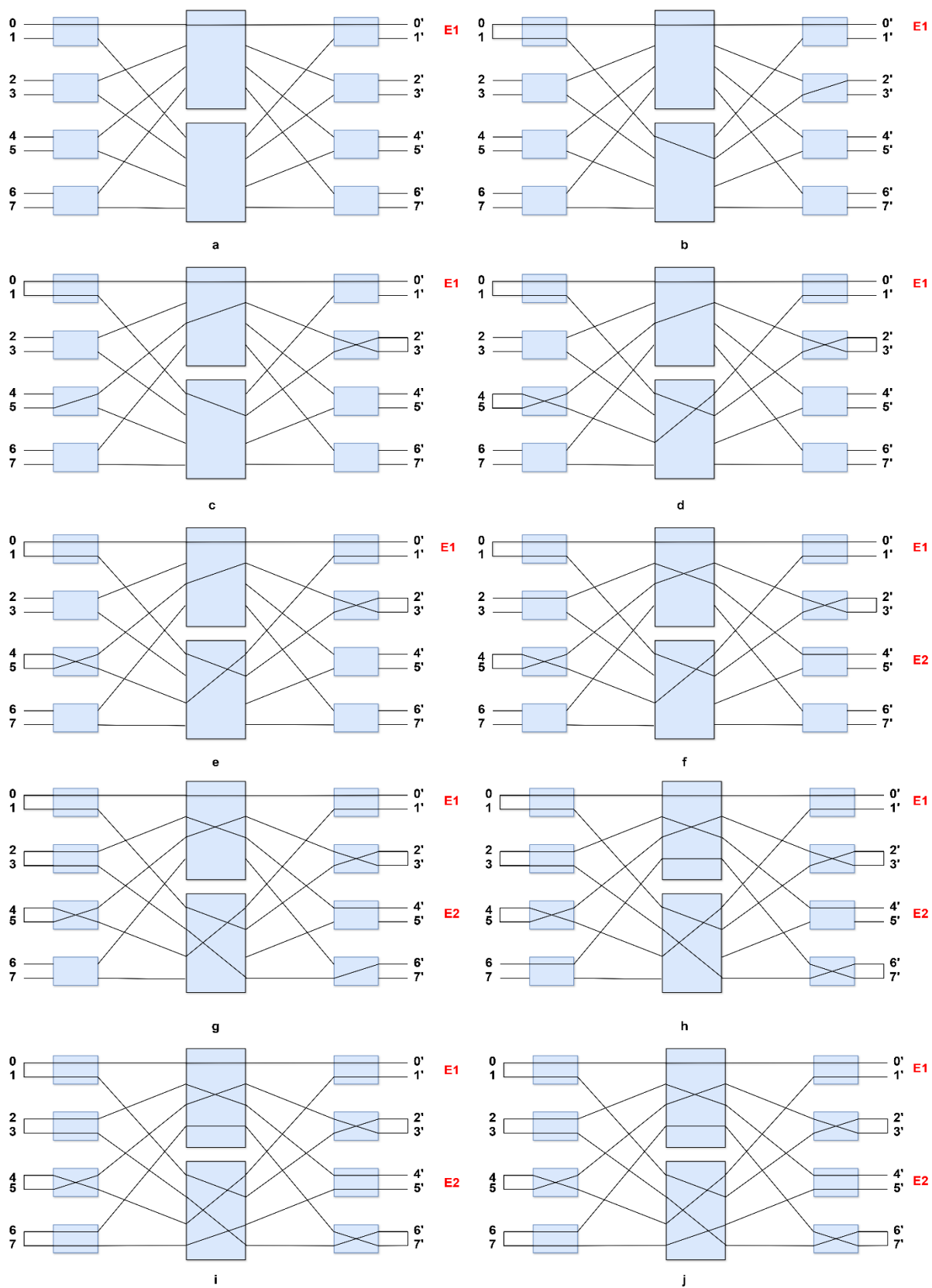
Σχήμα 1-6: Οι δύο καταστάσεις ενός διακόπτη. Bar (αριστερά) και Cross (δεξιά).

Πρόταση 4: Λόγω της γενικής μορφής του δικτύου (Πρόταση 1) και επειδή οι $2 \times z$ και $2 \times z + 1$ είναι θύρες του ίδιου διακόπτη, συνδέονται σε διαφορετικά υποδίκτυα (πάνω και κάτω) ανεξάρτητα από τη ρύθμιση (σταυρός ή μπάρα).

Πρόταση 5: Ως συνέπεια των προτάσεων 4 και 3 Εάν ο διακόπτης είναι ράβδος (bar), η είσοδος $2 \times z$ συνδέεται στο ανώτερο υποδίκτυο και το $2 \times z + 1$ συνδέεται στο κάτω υποδίκτυο. Αν ο διακόπτης είναι σταυρός (cross), ισχύει το αντίθετο. Το ίδιο ισχύει και για τις εξόδους.

Πρόταση 6: Εάν το υποδίκτυο, με το οποίο συνδέεται κάθε είσοδος και έξοδος, είναι γνωστό, τότε βάσει της Πρότασης 5, όλοι οι διακόπτες μπορούν να ρυθμιστούν σωστά. Επιπλέον, η γνώση του πίνακα εισόδου-εξόδου των υποδικτύων του επόμενου επιπέδου μπορεί να χρησιμοποιηθεί για τη διαμόρφωσή τους. Επομένως, η ακολουθία που δρομολογεί με επιτυχία το επίπεδο 0 (το πιο εξωτερικό επίπεδο) μπορεί να χρησιμοποιηθεί αναδρομικά για τη δρομολόγηση των εσωτερικών

επιπέδων και τελικά ολόκληρου του δικτύου Benes. Η ίδια η διαδικασία δρομολόγησης του εξωτερικού επιπέδου παράγει τους πίνακες δρομολόγησης εισόδου-εξόδου των εσωτερικών δικτύων. Στο Σχήμα 1-7 φαίνονται καθαρά οι γραμμές που συνδέουν τις εισόδους-εξόδους των εσωτερικών δικτύων. Καθένα από αυτά δίκτυα παίζει τον ίδιο ρόλο για τα δικά του υποδίκτυα και διαδίδει τους νέους πίνακες δρομολόγησης. Αυτή η διαδικασία συνεχίζεται μέχρι να δρομολογηθούν πλήρως όλα τα επίπεδα.



Σχήμα 1-7: Τα βήματα εκτέλεσης του αλγορίθμου.

1.3 Προτεινόμενος αλγόριθμος

1.3.1 Αλγόριθμος

Ο αλγόριθμος εκμεταλλεύεται την αναδρομική φύση του Δικτύου Benes όπως φαίνεται στην πρόταση 6 για να δρομολογήσει το επίπεδο 0 (εξώτατο επίπεδο) και στη συνέχεια χρησιμοποιεί απλώς τους πίνακες δρομολόγησης που παράγονται για τα υποδίκτυα για να συνεχίσει τη διαδικασία δρομολόγησης στο επόμενο επίπεδο μέχρι το τέλος. Μια τυπική εκτέλεση του αλγορίθμου έχει ως εξής.

- 1) Επίλεξε μια θύρα εξόδου ενός μη ρυθμισμένου διακόπτη και ένα υποδίκτυο (πάνω ή κάτω). Εάν δεν υπάρχουν μη ρυθμισμένοι διακόπτες, τερμάτισε τη διαδικασία. Γενικά, όταν ο αλγόριθμος ξεκινά τη δρομολόγηση για πρώτη φορά, επιλέγουμε την έξοδο 0'. Εάν επιστρέψουμε σε αυτό το βήμα κατά τη δρομολόγηση, μπορούμε να επιλέξουμε οποιαδήποτε θύρα που δεν έχουμε επισκεφτεί. Ανεξάρτητα από το ποια είναι η αρχική θύρα εξόδου ή το υποδίκτυο, ο αλγόριθμος παράγει σωστή ρύθμιση.
- 2) Δρομολόγησε τη θύρα εξόδου μέσω του επιλεγμένου υποδικτύου μέχρι την αντίστοιχη είσοδο (Εικόνα 1-7a).
- 3) Επίλεξε την άλλη είσοδο του ίδιου διακόπτη και δρομολόγησέ την πίσω στην αντίστοιχη έξοδο της χρησιμοποιώντας το αντίθετο υποδίκτυο από αυτό που χρησιμοποιήθηκε στο Βήμα 1 (Εικόνα 1-7b). Αυτό είναι σημαντικό. Είτε το άνω είτε το κάτω υποδίκτυο είναι επιλεγμένο στο Βήμα 1, δεν θα αλλάξει το αποτέλεσμα. Ωστόσο, είναι απαραίτητο να εναλλάσσουμε υποδίκτυα. Εάν το Βήμα 1 χρησιμοποιούσε το άνω υποδίκτυο, το Βήμα 3 πρέπει να χρησιμοποιήσει το κάτω υποδίκτυο. Εάν το Βήμα 1 χρησιμοποιούσε το κάτω υποδίκτυο, το Βήμα 3 πρέπει να χρησιμοποιήσει το άνω υποδίκτυο. Αυτό διασφαλίζει την ισορροπία των κατειλημμένων διαδρομών μεταξύ των δύο πιθανών διαδρομών (άνω και κάτω) και ουσιαστικά εγγυάται ότι για κάθε δρομολόγηση προς έναν διακόπτη, υπάρχει πάντα μια δρομολόγηση από αυτόν.
- 4) Εάν ο διακόπτης εξόδου στον οποίο οδηγεί η διαδρομή έχει ήδη τεθεί, μεταβείτε στο Βήμα 1 (Εικόνα 1-7d). Εάν ο διακόπτης εξόδου δεν είναι κατειλημμένος, επιλέξτε την έξοδο του ίδιου διακόπτη και του αρχικού επιλεγμένου υποδικτύου και προχωρήστε στο Βήμα 2 (Εικόνα 1-7c). Στην πρώτη περίπτωση, ολοκληρώνεται ένα μονοπάτι. Οι διακόπτες που ανήκουν στο ίδιο μονοπάτι δεν επηρεάζουν την δρομολόγηση του υπόλοιπου δικτύου. Στη δεύτερη περίπτωση, επιλέγουμε την έξοδο του ίδιου διακόπτη για να διατηρήσουμε την πορεία διαδρομής. Το υποδίκτυο που επιλέχθηκε στο Βήμα 1 για κάθε σύμπλεγμα είναι πάντα το αντίθετο από το υποδίκτυο του βήματος 3. Με άλλα λόγια, τα μονοπάτια εναλλάσσουν υποδίκτυα συνεχώς.

Ο Πίνακας 1-1 είναι ο πίνακας δρομολόγησης που αντιστοιχεί στην ακολουθία δρομολόγησης που φαίνεται στο Σχήμα 1-7.

Πίνακας 1-1: Πίνακας δρομολόγησης ενός δικτύου Benes 8×8 .

Θύρες εισόδου	Θύρες εξόδου
0	0'
1	2'
2	4'
3	6'
4	1'
5	3'
6	7'
7	5'

Στο Σχήμα 1-7a γίνεται η έναρξη της διαδικασίας με την θύρα εξόδου 0' και το άνω υποδίκτυο (Βήμα 1). Στη συνέχεια δρομολογείται η πρώτη διαδρομή (Βήμα 2). Στο Σχήμα 1-7b, η είσοδος του ίδιου διακόπτη δρομολογείται πίσω στις εξόδους (Βήμα 3). Στο Σχήμα 1-7c ο αλγόριθμος βρίσκεται στο Βήμα 4. Δεν έχει γίνει επίσκεψη στον διακόπτη εξόδου, επομένως προχωράμε δρομολογώντας την έξοδο του διακόπτη πίσω στις εισόδους μέσω του άνω υποδικτύου (Βήμα 2). Το Σχήμα 1-7d δείχνει την ίδια διαδικασία που επαναλαμβάνεται δρομολογώντας την είσοδο στην έξοδο μέσω του κάτω υποδικτύου αυτή τη φορά (Βήμα 3). Στο Σχήμα 1-7e ο διακόπτης εξόδου (διακόπτης 0' με θύρες 0', 1') έχει ήδη δρομολογηθεί, οπότε ο αλγόριθμος προχωρά στο Βήμα 1 και επιλέγει την έξοδο 4' ως αρχή της διαδρομής 2 και το ανώτερο υποδίκτυο ως υποδίκτυο έναρξης (Βήμα 1). Το Σχήμα 1-7f δείχνει τη διαδικασία που ξεκινά ξανά με τη δρομολόγηση από το 4' στο 2 (Βήμα 2). Στο Σχήμα 1-7g η διαδρομή επιστρέφει στις εξόδους (Βήμα 3). Στο Σχήμα 1-7h, η Έξοδος 7' επιλέγεται (βήμα 4) και δρομολογείται (Βήμα 2) αφού βρίσκεται στον ίδιο διακόπτη με την 6'. Τέλος, στο Σχήμα 1-7i, δρομολογούμε την είσοδο του ίδιου διακόπτη πίσω στις εξόδους (Βήμα 3). Το Σχήμα 1-7j παρουσιάζει την ολοκληρωμένη δρομολόγηση.

Η ολοκλήρωση της διαδικασίας συνεπάγεται τη ρύθμιση όλων των διακοπών. Κατά συνέπεια, ο πίνακας δρομολόγησης εισόδου-εξόδου για τα επόμενα δίκτυα μπορεί να φανεί σχεδιασμένος στο Σχήμα 1-7j. Στη συνέχεια, χρησιμοποιείται για τη δρομολόγηση των υποδικτύων με τον ίδιο ακριβώς τρόπο. Αν $2^r = N$ είναι οι εισοδοί τότε η συνολική πολυπλοκότητα είναι $N \times \log_2 N$. Αυτή είναι η γραμμική πολυπλοκότητα ενός αλγορίθμου δρομολόγησης Benes. Δεν υπάρχει γραμμική λύση που να μπορεί να ολοκληρώσει τη δρομολόγηση σε λιγότερα βήματα. Ένα πράγμα που πρέπει να αναφερθεί είναι ότι σε αντίθεση με τον αλγόριθμο Orferman, δεν υπάρχει οπισθοδρόμηση (backtracking) για την αλλαγή των ρυθμίσεων των διακοπών [32].

1.3.2 Απόδειξη λειτουργίας

Αυτή η ενότητα περιέχει την απόδειξη της εγκυρότητας του αλγορίθμου. Η διαδικασία που περιγράφεται στην προηγούμενη ενότητα θα οδηγεί πάντα σε μια έγκυρη λύση ανεξάρτητα από τις παραμέτρους αρχικοποίησης (διακόπτες εκκίνησης και υποδίκτυο). Όπως υπαγορεύει η πρόταση 6, η δρομολόγηση του εξώτατου επιπέδου συνεπάγεται τη δρομολόγηση ολόκληρου του δικτύου. Το παράδειγμα στο *Σχήμα 1-7* δείχνει πώς η δρομολόγηση του εξώτατου επιπέδου παράγει τους πίνακες δρομολόγησης για τα επόμενα υποδίκτυα οπότε εάν η διαδικασία δρομολόγησης αποδειχθεί σωστή για το πρώτο επίπεδο, είναι σωστή για όλα τα επίπεδα.

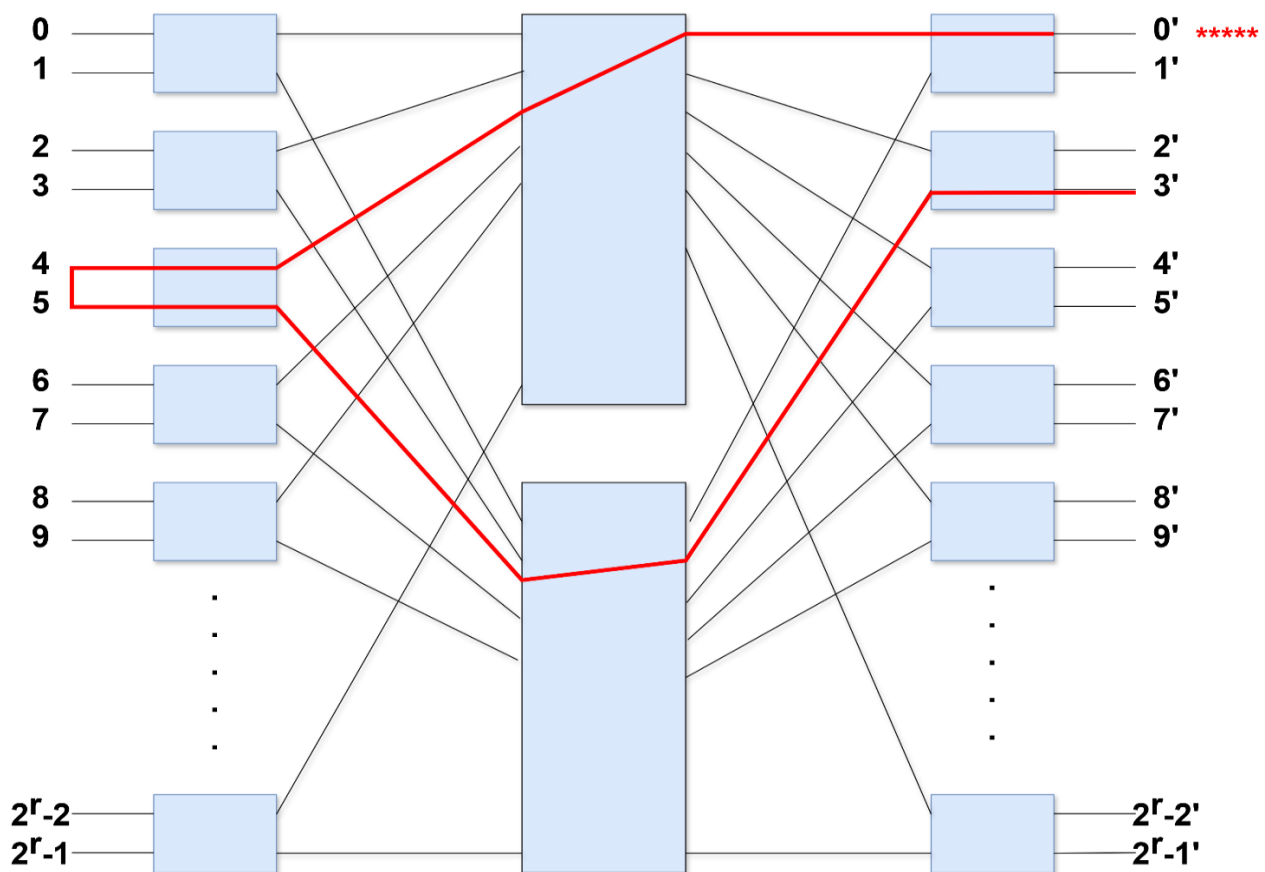
Για λόγους παρουσίασης, τα βήματα μεταξύ της δρομολόγησης μιας εξόδου και της δρομολόγησης της επόμενης εξόδου (Βήματα 1 έως 4) αναφέρονται ως επανάληψη. Σε κάθε επανάληψη, ρυθμίζονται δύο διακόπτες. Επομένως, στο επίπεδο 0 με 2^r εισόδους-εξόδους χρειάζονται 2^{r-1} επαναλήψεις. Αποδεικνύουμε ότι η επανάληψη a_0 οδηγεί στην a_1 και η επανάληψη a_i οδηγεί στην a_{i+1} , (εν συντομία $a_0 \rightarrow a_1$, $a_i \rightarrow a_{i+1}$) χωρίς αποτυχία. Αυτό σημαίνει ότι το αρχικό βήμα είναι πάντα έγκυρο και όλα τα βήματα μετά από αυτό είναι πάντα έγκυρα, ώστε το δίκτυο να δρομολογείται σωστά όταν τελειώσει η διαδικασία.

Περίπτωση $a_0 \rightarrow a_1$. Είναι προφανές ότι όλα τα μονοπάτια είναι διαθέσιμα και η πρώτη επανάληψη μπορεί να συμπληρώνεται πάντα χωρίς αποτυχία (*Σχήμα 1-8*).

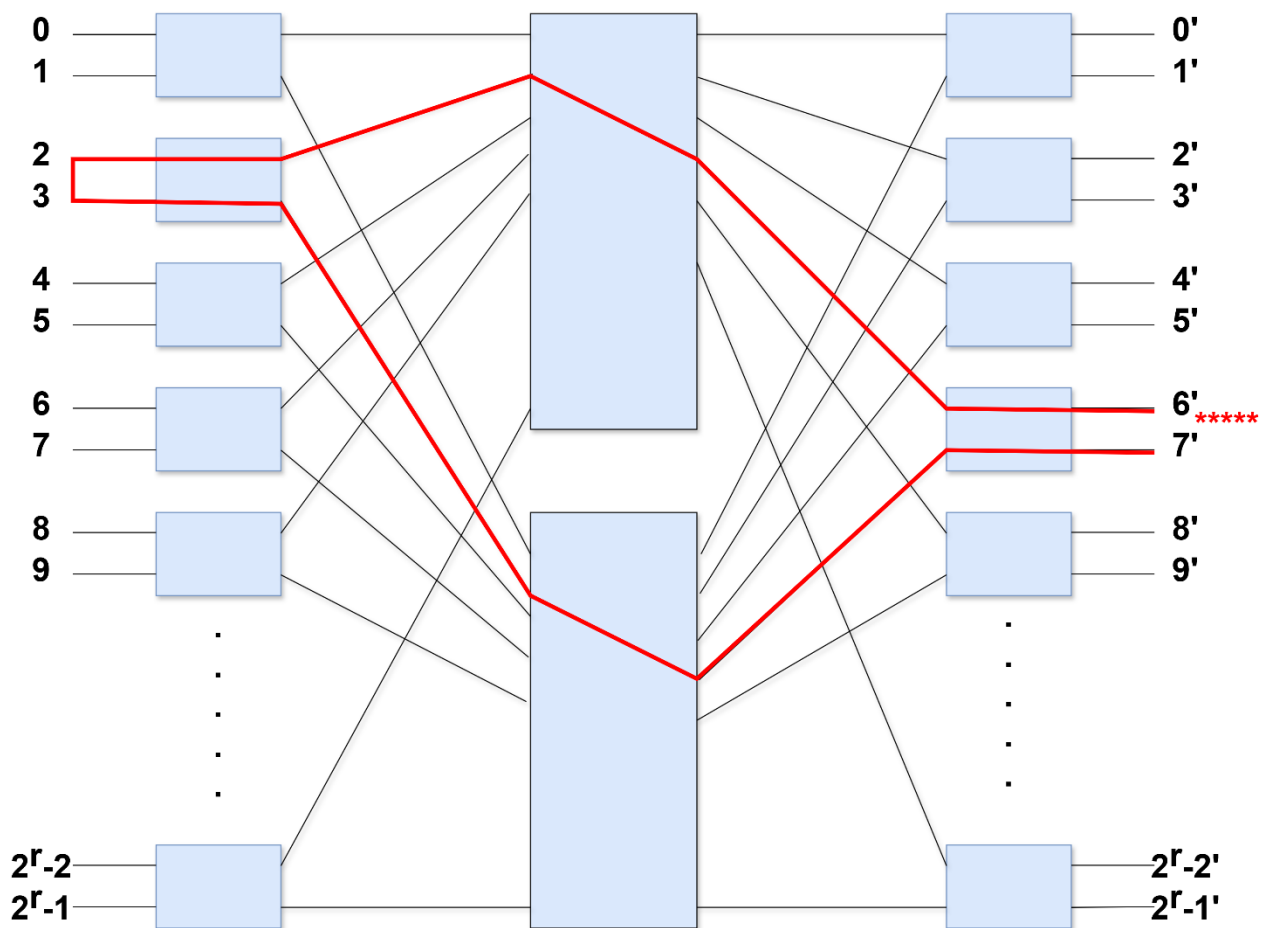
Περίπτωση $a_k \rightarrow a_{k+1}$. Πρώτον, είναι σαφές ότι αφού και οι δύο εισόδοι ενός διακόπτη εισόδου δρομολογούνται σε μία επανάληψη, η δρομολόγηση της μίας θύρας σε έναν διακόπτη εισόδου (Βήμα 2) ακολουθείται αμέσως από την δρομολόγηση της άλλης θύρας εισόδου στον αντίστοιχο διακόπτη (Βήμα 3). Είναι αδύνατο ένας διακόπτης εισόδου να είναι μερικώς δρομολογημένος (*Σχήμα 1-9*). Αυτό σημαίνει ότι στην επανάληψη a_k , ο διακόπτης εισόδου που πρέπει να δρομολογηθεί έχει και τις δύο διαδρομές ανοιχτές. Για την διαδρομή πίσω στην επόμενη έξοδο, υπάρχουν δύο δυνατότητες

- 1) Ο διακόπτης εξόδου στο τέλος της επανάληψης δεν είναι ρυθμισμένος και μπορεί απλά να ρυθμιστεί με τον ίδιο τρόπο όπως όλοι οι άλλοι: σε αυτήν την περίπτωση, ο αλγόριθμος προχωρά ως συνήθως επιλέγοντας την έξοδο του ίδιου διακόπτη ως σημείο εκκίνησης της επανάληψης a_{k+1} .
- 2) Εάν ο διακόπτης εξόδου είναι δρομολογημένος, τότε έχουμε φτάσει στο τέλος της διαδρομής βρόχου και πρέπει να επιλεγεί μια νέα διαδρομή. Πιθανότητα σύγκρουσης δεν υπάρχει αφού η διαδρομή που κλείνει τον βρόχο επιστρέφει πάντα στην πλευρά εξόδου μέσω διαφορετικού υποδικτύου από αυτό που ξεκίνησε τον βρόχο. Παρατηρήστε ότι στο *Σχήμα 1-10*, διαδρομή 7-1', που τελειώνει τον βρόχο, δρομολογείται μέσω του κάτω υποδικτύου και η διαδρομή 4-0', η οποία ξεκινά το βρόχο, δρομολογείται μέσω του άνω υποδικτύου. Μεταξύ των μονοπατιών 4-0' και 7-1', υπάρχουν δύο μονοπάτια, 5-3' και 6-2'. Η σειρά δρομολόγησης είναι 4-0', 5-3', 6-2', 7-1'. Αν προστεθούν μερικές

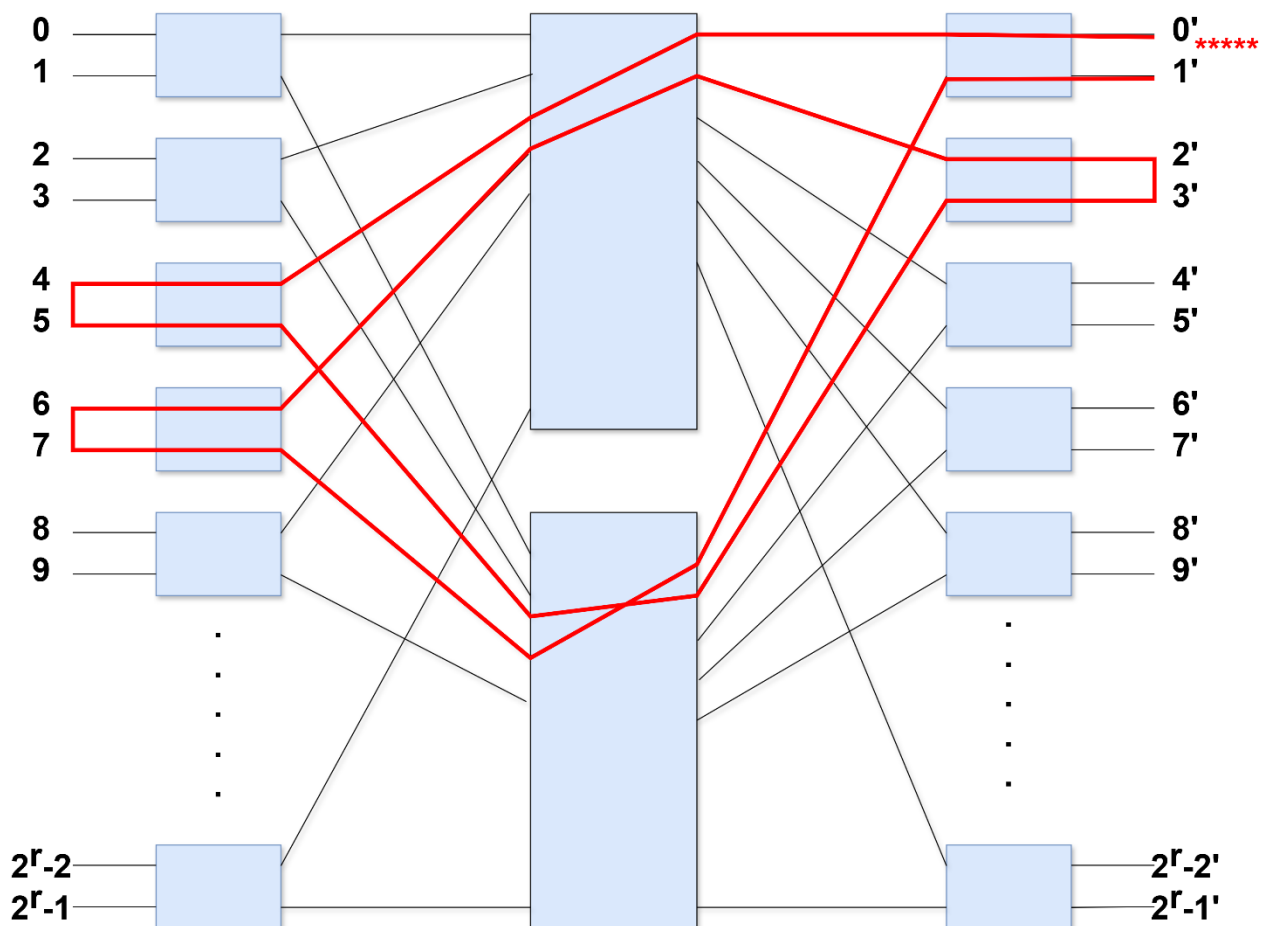
ακόμη διαδρομές στον βρόχο, αυτές πρέπει να είναι πολλαπλάσια του δύο. Για παράδειγμα, στο Σχήμα 1-10, αν αλλάξουμε διαδρομή 7-1' σε 7-5', τότε πρέπει να προσθέσουμε 3-4' στην πλευρά εισόδου και μετά πάλι τέλος στην πλευρά εξόδου με 2-1'. Ολόκληρη η διαδρομή τώρα είναι 4-0', 5-3', 6-2', 7-5', 3-4', και 2-1'. Το μονοπάτι πρέπει να επισκεφθεί πρώτα τις εισόδους για να επιστρέψει στις εξόδους. Αυτό κοστίζει τουλάχιστον δύο μονοπάτια που σημαίνει ότι ο αριθμός των διαδρομών σε ένα βρόχο μεταξύ τέλους και αρχής είναι πάντα πολλαπλάσιο του δύο. Επομένως όσες κι αν είναι οι διαδρομές που βρίσκονται μεταξύ της διαδρομής x-0' και y-1', αυτές βρίσκονται πάντα σε διαφορετικά υποδίκτυα.



Σχήμα 1-8: Η αρχή της διαδικασίας δρομολόγησης.



Σχήμα 1-9: Τυχαίο στιγμιότυπο στην διαδικασία δρομολόγησης.



Σχήμα 1-10: Ο διακόπτης (0',1') ολοκληρώνει το μονοπάτι (κόκκινο χρώμα). Το μονοπάτι είναι κλειστό και ανεξάρτητο από τις καταστάσεις των όλων των διακοπών που δεν ανήκουν σε αυτό.

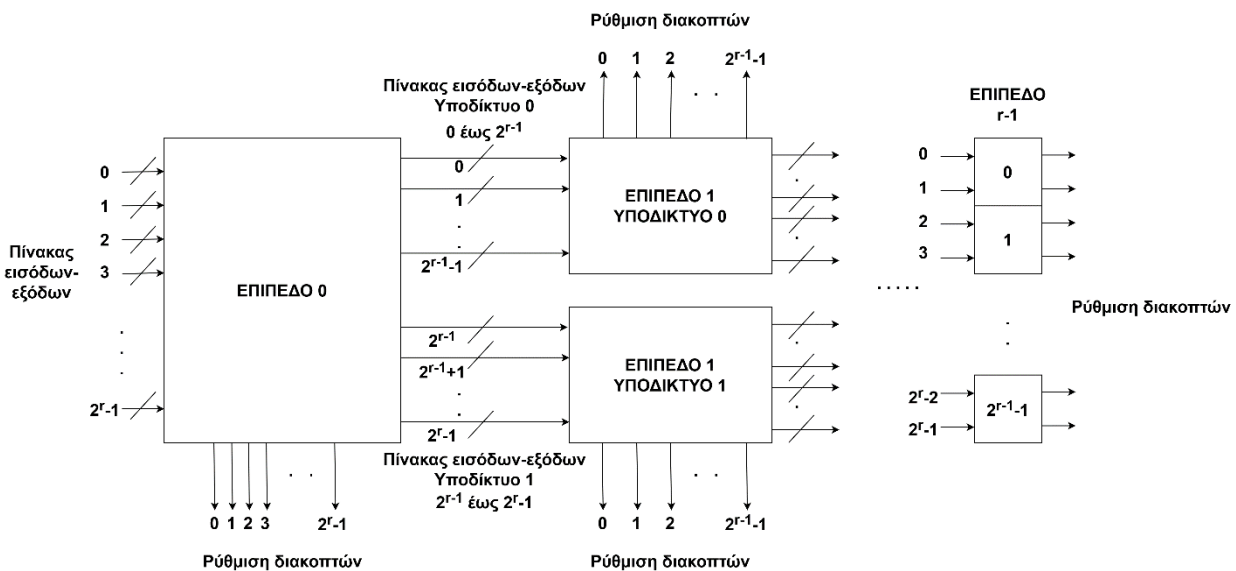
Για να συνοψίσουμε, τώρα γνωρίζουμε ότι $a_0 \rightarrow a_1$ και $a_k \rightarrow a_{k+1}$. Η διαδικασία δρομολογεί με επιτυχία ολόκληρο το εξωτερικό επίπεδο. Εφαρμόζοντας τον ίδιο αλγόριθμο στα επόμενα επίπεδα, δρομολογείται ολόκληρο το δίκτυο.

1.4 Περιγραφή αρχιτεκτονικής

Αυτή η ενότητα καλύπτει την υλοποίηση σε επίπεδο αρχιτεκτονικής του αλγορίθμου που περιγράφεται στις προηγούμενες ενότητες. Η είσοδος είναι ο πίνακας δρομολόγησης εισόδου-εξόδου και η έξοδος είναι η ρύθμιση των διακοπών. Μετά την ολοκλήρωση της δρομολόγησης, αναμένεται ότι κάθε έξοδος μπορεί να χρησιμοποιηθεί μέσω της αντίστοιχης εισόδου της. Η αναδρομική φύση του Benes επιτρέπει τη χρήση της ίδιας κυκλωματικής αρχιτεκτονικής για την δρομολόγηση κάθε επιπέδου και υποδικτύου ανεξαρτήτως αριθμού εισόδων-εξόδων. Η συνολική αρχιτεκτονική για γενικό αριθμό εισόδων φαίνεται στο Σχήμα 1-11. Κάθε μπλοκ υποδικτύου είναι το κύκλωμα που δρομολογεί το συγκεκριμένο υποδίκτυο. Όλα τα μπλοκ λειτουργούν πανομοιότυπα. Σε κάθε επίπεδο, ο αριθμός των υποδικτύων διπλασιάζεται και ο αριθμός των εισόδων ανά υποδίκτυο

υποδιπλασιάζεται. Η κύρια έξοδος κάθε επιπέδου είναι η ρύθμιση των διακοπών. Ο αριθμός των διακοπών είναι ο αριθμός εισόδων-εξόδων (μισοί διακόπτες εισόδου και μισοί διακόπτες εξόδου) και είναι ίδιος (2^r διακόπτες) για κάθε επίπεδο, εκτός από το κέντρο (διακόπτες 2^{r-1}). Εκτός από την ρύθμιση των διακοπών, κάθε μπλόκ υποδικτύου δημιουργεί επίσης τους πίνακες εισόδου-εξόδου που δίνονται στα επόμενα μπλοκ υποδίκτυα.

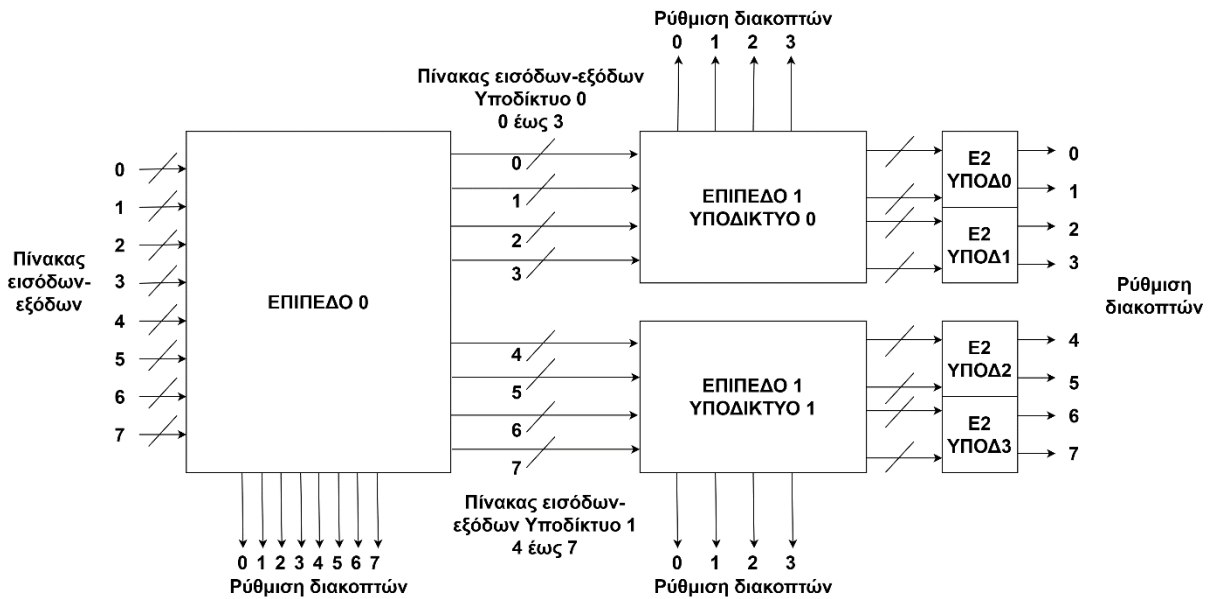
Η πράξη της «δρομολόγησης» εισόδων και εξόδων μεταφράζεται σε μια απλή ταξινόμηση των αντιστοιχίσεων εισόδου-εξόδου. Αυτή η ταξινόμηση διασφαλίζει ότι στη νέα λίστα, τα ζεύγη σε θέσεις $2 \times z$ δρομολογούνται μέσω του άνω υποδικτύου και τα ζεύγη σε θέσεις $2 \times z + 1$ δρομολογούνται μέσω του κάτω υποδικτύου. Κάθε κυκλωματικό μπλοκ αποτελείται από 4 μονάδες: την μονάδα αναστροφής πίνακα δρομολόγησης που παράγει τον αντίστροφο πίνακα δρομολόγησης με βάση τις εξόδους έτσι ώστε να μην χρειάζεται να γίνει γραμμική αναζήτηση για να βρεθεί ποια είσοδος συνδέεται σε συγκεκριμένη έξοδο, την μονάδα ταξινόμησης που ταξινομεί τα ζευγάρια με συγκεκριμένο τρόπο ώστε να γίνεται η δρομολόγηση στους διακόπτες κάθε επιπέδου, την μονάδα εύρεσης μη δρομολογημένου διακόπτη που λειτουργεί επικουρικά βρίσκοντας αδέσμευτους διακόπτες σε περίπτωση νέου μονοπατιού και τέλος την μονάδα δρομολόγησης διακοπών που παράγει τις καταστάσεις των διακοπών σε μορφή δυαδικού διανύσματος αλλά και τους πίνακες δρομολόγησης για τα επόμενα υποδίκτυα.



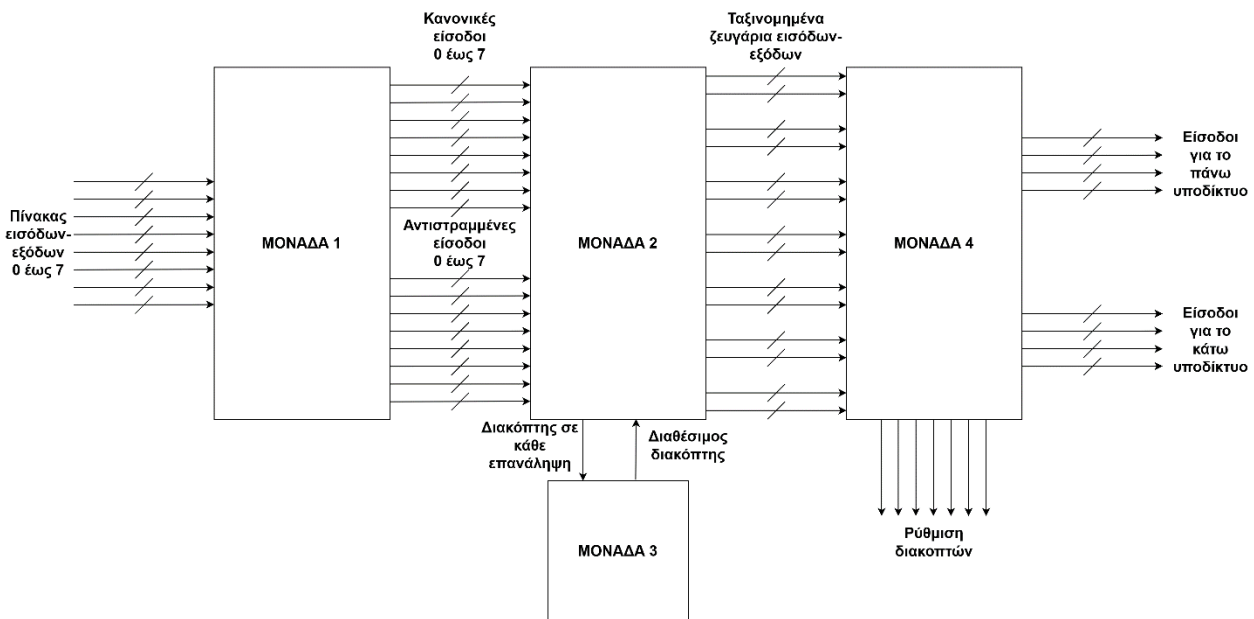
Σχήμα 1-11: Γενικευμένο κύκλωμα δρομολόγησης δικτύου Benes $2^r=N$ εισόδων. Η έξοδος του είναι όλα τα σήματα με σήμανση 'Ρύθμιση διακοπών' τα οποία περιέχουν την κατάσταση κάθε διακόπτη (cross ή bar).

Για την πιο λεπτομερή εξήγηση του συστήματος, χρησιμοποιήθηκε ένα παράδειγμα 8 εισόδων $\times$ εξόδων (Σχήμα 1-12). Από όλα τα υποδίκτυα (Πρόταση 2) στο δίκτυο, εστιάζουμε στη λειτουργία του πρώτου επιπέδου (Σχήμα 1-13) (8 θύρες εισόδου-εξόδου, 8 διακόπτες). Η πρόταση 6 διασφαλίζει

ότι μόλις δρομολογηθεί το πρώτο στρώμα, όλα τα άλλα ακολουθούν με τον ίδιο τρόπο. Η δρομολόγηση μεγαλύτερων δικτύων είναι σχεδόν αδύνατο να ακολουθηθεί στο χαρτί.



Σχήμα 1-12: Το κύκλωμα για δίκτυο Benes 8×8.



Σχήμα 1-13: Το κύκλωμα για πρώτο επίπεδο δικτύου Benes 8×8.

1.4.1 Αναστροφή πίνακα δρομολόγησης

Αυτή η ενότητα δημιουργεί το διάνυσμα αντίστροφης δρομολόγησης εισόδου-εξόδου όπου οι έξοδοι είναι ταξινομημένες κατά αυξητική σειρά. Έχοντας ως παράδειγμα τον πίνακα δρομολόγησης που δόθηκε παράγουμε το αντίστοιχο αντίστροφο πίνακα.

Το αντίστροφο διάνυσμα περιέχει τις θύρες εξόδου στην αριστερή πλευρά σε αύξουσα σειρά και την αντίστοιχη θύρα εισόδου τους στη δεξιά πλευρά. Στο παράδειγμά μας, η θέση 2' στο νέο διάνυσμα

συμπληρώνεται με τον αριθμό 1, η θέση 0' με τον αριθμό 0 και το 4' με το 2. Ολόκληρο το αντίστροφο διάνυσμα που φαίνεται στον Πίνακα 1-2.

Πίνακας 1-2: Αντίστροφος πίνακας δρομολόγησης.

Θύρες εξόδου	Θύρες εισόδου
0'	0
1'	4
2'	1
3'	5
4'	2
5'	7
6'	3
7'	6

Η αριστερή πλευρά των δύο διανυσμάτων είναι πάντα σε αύξουσα σειρά οπότε δεν την χρειαζόμαστε. Η δεξιά πλευρά είναι η είσοδος που τροφοδοτείται στο πρώτο μπλοκ που ξεκινά ουσιαστικά την διαδικασία δρομολόγησης. Στην περίπτωση ενός γενικού αριθμού θυρών εισόδου, ο πίνακας αντιστροφής παράγεται με τον ίδιο τρόπο. Η μόνη προϋπόθεση για την δημιουργία αντίστροφου να μην υπάρχουν δύο ή περισσότερες θύρες από την μία πλευρά που θα πρέπει να συνδεθούν με την ίδια θύρα της άλλης πλευράς. Η δημιουργία του αντίστροφου χρειάζεται 1 κύκλο ρολογιού ανεξαρτήτως μεγέθους.

1.4.2 Μηχανισμός ταξινόμησης

Αυτό το μπλοκ υλοποιεί ουσιαστικά τον αλγόριθμο ταξινόμησης. Σε κάθε κύκλο, δρομολογεί δύο διαδρομές: μία από τις εξόδους στις εισόδους και μία από τις εισόδους πίσω στις εξόδους. Στο περιβάλλον του κυκλώματος, «δρομολόγηση» σημαίνει ταξινόμηση ζευγών εισόδου-εξόδου με τρόπο που ζεύγη σε θέσεις $2 \times m$ δρομολογούνται μέσω του άνω δικτύου και τα ζεύγη σε $2 \times m + 1$ θέσεις δρομολογούνται μέσω του κάτω υποδικτύου. Είτε είναι πάνω κάτω, είτε κάτω και πάνω, δεν έχει καμία διαφορά στη λύση. Χρησιμοποιώντας τον πίνακα δρομολόγησης και τον αντίστροφό του μια λεπτομερής εκτέλεση του αλγορίθμου φαίνεται παρακάτω. Όπως αναφέραμε στην Ενότητα 3, μία επανάληψη αποτελείται από όλα τα βήματα, που δρομολογούν το μονοπάτι από ένα διακόπτη εξόδου σε ένα διακόπτη εισόδου και μετά ξανά σε ένα διακόπτη εξόδου.

1η επανάληψη. Οποιαδήποτε έξοδος μπορεί να επιλεγεί για να ξεκινήσει ο αλγόριθμος, αλλά στην περίπτωσή μας, επιλέχθηκε η έξοδος 0' ως το σημείο εκκίνησης της πρώτης διαδρομής (Πίνακας 1-3).

Πίνακας 1-3: Πρώτη επανάληψη.

Θύρες εισόδου**Θύρες εξόδου**

0

0

1

2'

2η επανάληψη. Η έξοδος για την επόμενη επανάληψη είναι 3' αφού βρίσκεται στον ίδιο διακόπτη με το 2' (Πίνακας 1-4).

Πίνακας 1-4: Δεύτερη επανάληψη.

Θύρες εισόδου**Θύρες εξόδου**

0

0'

1

2'

5

3'

4

1'

3η επανάληψη. Το 1' βρίσκεται στον ίδιο διακόπτη με το 0', επομένως πρέπει να δημιουργηθεί μια νέα διαδρομή. Το 4' είναι το νέο σημείο εκκίνησης (Πίνακας 1-5).

Πίνακας 1-5: Τρίτη επανάληψη.

Θύρες εισόδου**Θύρες εξόδου**

0

0'

1

2'

5

3'

4

1'

2

4'

3

6'

4η επανάληψη. Το 7' είναι στον ίδιο διακόπτη με το 6'. Τελευταία επανάληψη αφού δρομολογούνται όλοι οι διακόπτες (Πίνακας 1-6). Αυτό είναι το ίδιο παράδειγμα που παρουσιάζεται στο Σχήμα 1-7.

Πίνακας 1-6: Τέταρτη επανάληψη.

Θύρες εισόδου**Θύρες εξόδου**

0

0'

1

2'

5

3'

4

1'

2

4'

3

6'

6

7'

Στην περίπτωση ενός γενικού αριθμού εισόδων, ο αλγόριθμος είναι ο ίδιος. Επαναλαμβάνουμε συνέχεια τα ίδια βήματα μέχρι να δρομολογηθούν όλοι οι διακόπτες. Η διαδικασία είναι γραμμική. Για είσοδο $2^r = N$, χρειάζονται $2^{r-1} = N/2$ κύκλοι ρολογιού καθώς σε ένα κύκλο δρομολογούνται δύο διακόπτες.

1.4.3 Δείκτης μη δρομολογημένου διακόπτη

Το μπλοκ εύρεσης μη δρομολογημένου διακόπτη είναι αναγκαίο για την εύρεση νέων σημείων εκκίνησης σε περίπτωση κλειστού μονοπατιού. Για παράδειγμα, στην επανάληψη 2 (*Πίνακας 1-5*), η διαδρομή καταλήγει σε βρόχο στο διακόπτη εξόδου $0'$ (θύρες εξόδου $0', 1'$). Για να συνεχιστεί η διαδικασία, είναι απαραίτητο να βρεθεί ένα νέο σημείο εκκίνησης. Όταν δεν υπάρχουν πλέον διαθέσιμοι διακόπτες, όλοι οι βρόχοι είναι κλειστοί και ο αλγόριθμος τελειώνει. Η λογική περιγραφή της μονάδας έχει ως εξής.

Κάθε στοιχείο διακόπτη γίνεται μέρος μιας συνδεδεμένης λίστας με την βοήθεια δύο πινάκων. Ο πίνακας «αριστερά» περιέχει τις καταχωρήσεις λίστας στα αριστερά κάθε στοιχείου και ο πίνακας «δεξιά» περιέχει τις καταχωρήσεις λίστας στα δεξιά κάθε στοιχείου. Ο επικεφαλής της λίστας είναι το πρώτο διαθέσιμο ζευγάρι. Όταν ένα στοιχείο χρησιμοποιείται (δηλαδή δρομολογείται), εξάγεται από τη λίστα με σύνδεση των δύο διπλανών. Εάν επιλεγεί η κεφαλή, ο δείκτης κεφαλής μετακινείται αριστερά. Η λίστα είναι πάντα οργανωμένη από αριστερά προς τα δεξιά. Θεωρείται ότι ο διακόπτης εξόδου z' έχει τις θύρες $2 \times z'$ και $2 \times z' + 1'$. Ένα ισοδύναμο τρέξιμο σύμφωνα με τη πρώτη υλοποίηση του 8×8 Benes είναι ως εξής.

1η επανάληψη. Οι δύο πίνακες (αριστερά) και (δεξιά) έχουν τέσσερις καταχωρήσεις, μία για κάθε διακόπτη εξόδου ($0', 1', 2', 3'$). Ο διακόπτης $0'$ (θύρες $0', 1'$) δεν συνεισφέρει αφού πάντα δρομολογείται πρώτος. Τα άλλα κελιά αρχικοποιούνται με την ίδια λογική (*Πίνακας 1-7*). Η κεφαλή στη λίστα είναι ο διακόπτης $1'$ (θύρες $2', 3'$), κεφαλή = $1'$.

Πίνακας 1-7: Πρώτη επανάληψη (λίστα διακοπτών).

Αριθμός διακόπτη εξόδου	Αριστερά	Δεξιά
3'	1'	2'
2'	3'	1'
1'	2'	3'
0'	X	X

2η επανάληψη. Ο διακόπτης εξόδου 1' εξάγεται από τη λίστα. Ο διακόπτης 2' (4',5') γίνεται κεφαλή της λίστας, κεφαλή = 2' (Πίνακας 1-8).

Πίνακας 1-8: Δεύτερη επανάληψη (λίστα διακοπτών).

Αριθμός διακόπτη εξόδου	Αριστερά	Δεξιά
3'	2'	2'
2'	3'	3'
1'	X	X
0'	X	X

3η επανάληψη. Συναντάται βρόχος διαδρομής. Ο διακόπτης εξόδου 2' (κεφαλή λίστας) χρησιμοποιείται για να συνεχιστεί η δρομολόγηση. Κατά συνέπεια, ο διακόπτης 2' εξάγεται από τη λίστα (Πίνακας 1-9).

Πίνακας 1-9: Τρίτη επανάληψη (λίστα διακοπτών).

Αριθμός διακόπτη εξόδου	Αριστερά	Δεξιά
3'	3'	3'
2'	X	X
1'	X	X
0'	X	X

4η επανάληψη. Ο διακόπτης εξόδου 3' εξάγεται από τη λίστα. Η δρομολόγηση ολοκληρώθηκε (Πίνακας 1-10).

Πίνακας 1-10: Τέταρτη επανάληψη (λίστα διακοπτών).

Αριθμός διακόπτη εξόδου	Αριστερά	Δεξιά
3'	X	X
2'	X	X
1'	X	X
0'	X	X

Μια τέτοια υλοποίηση βελτιώνει δραματικά την ταχύτητα του κυκλώματος καθώς μας επιτρέπει να αποφύγουμε τη χρήση λογικών κατασκευών με μεγάλες κρίσιμες διαδρομές (critical paths), όπως έναν κωδικοποιητή προτεραιότητας για να αναζητήσουμε ξεκίνημα νέου μονοπατιού.

1.4.4 Δρομολόγηση διακόπτων και πίνακες δρομολόγησης επόμενων επιπέδων

Η τελευταία μονάδα διαβάσει την ταξινομημένη λίστα των ζευγών και δρομολογεί τους διακόπτες εισόδου και εξόδου ανάλογα. Διαβάσει κάθε δύο καταχωρήσεις γιατί οι παρακείμενες εγγραφές ανήκουν στον ίδιο διακόπτη (0 και 1, 4 και 5 κ.ο.κ). Σε κάθε διακόπτη z εάν η θύρα $2 \times z$ είναι στο

πάνω υποδίκτυο, τότε ο διακόπτης είναι bar. Αν είναι στο κάτω υποδίκτυο, τότε ο διακόπτης είναι cross. Το αντίθετο ισχύει για την θύρα $2 \times z + 1$. Η μονάδα επίσης προετοιμάζει τους πίνακες εισόδου-εξόδου για τα επόμενα δύο υποδίκτυα. Η διαδικασία ολοκληρώνεται σε $n/2$ κύκλους.

Αφού ταξινομήσουμε τις καταχωρήσεις με τις προηγούμενες μονάδες, έχουμε τα υποδίκτυα διέλευσης τους (Πίνακας 1-11). Όλα τα ζεύγη σε θέσεις $2 \times m$ δρομολογούνται μέσω του άνω Benes υποδικτύου και όλα τα ζεύγη σε $2 \times m + 1$ δρομολογούνται μέσω του κάτω υποδικτύου. Για την δρομολόγηση ελέγχουμε τις θέσεις $2 \times m$ στις δύο στήλες θυρών εισόδου και εξόδου και σταδιακά δρομολογούμε το επίπεδο.

Στη θέση 0, υπάρχει η θύρα εισόδου 0 και η θύρα εξόδου 0'. Και οι δύο θύρες 0 και 0' είναι οι πρώτες θύρα του αντίστοιχου διακόπτη τους, έτσι και οι δύο διακόπτες δρομολογούνται ως bar (Πίνακας 1-12).

Πίνακας 1-11: Πάνω (U) και Κάτω (L) υποδίκτυα.

Θύρες εισόδου	Θύρες εξόδου	Υποδίκτυο
0	0'	U
1	2'	L
5	3'	U
4	1'	L
2	4'	U
3	6'	L
6	7'	U
7	5'	L

Πίνακας 1-12: Πρώτη επανάληψη (δρομολόγηση διακοπών).

Νούμερο διακόπτη (εις/εξ)	Κατάσταση δ.εισ.	Κατάσταση δ.εξ.
0	bar	bar
1		
2		
3		

Η επόμενη θέση $2 \times m$ είναι η θέση 2 (3η σειρά). Οι αντίστοιχες θύρες είναι 5 και 3'. Και τα δυο είναι η δεύτερη θύρα του αντίστοιχου διακόπτη τους και συνδέονται στο πάνω υποδίκτυο, επομένως και οι δύο διακόπτες είναι cross (διακόπτης 2 για 5 και διακόπτης 1' για 3') (Πίνακας 1-13).

Πίνακας 1-13: Δεύτερη επανάληψη (δρομολόγηση διακοπών).

Νούμερο διακόπτη (εις/εξ)	Κατάσταση δ.εισ.	Κατάσταση δ.εξ.
0	bar	bar
1		cross

2

cross

3

Η θέση 4 (5η σειρά) περιέχει τις θύρες 2 και 4'. Και οι δύο είναι η πρώτη θύρα τους αντίστοιχου διακόπτη στο πάνω υποδίκτυο οπότε είναι bar (*Πίνακας 1-14*).

Πίνακας 1-14: Τρίτη επανάληψη (δρομολόγηση διακοπών).

Νούμερο διακόπτη (εις/εξ) **Κατάσταση δ.εισ.** **Κατάσταση δ.εξ.**

0

bar

bar

1

bar

cross

2

cross

bar

3

Η θέση 6 (7η σειρά) περιέχει τις θύρες 6 και 7'. Εδώ, η θύρα 6 είναι η πρώτη θύρα εισόδου του αντίστοιχου διακόπτη και η 7' η δεύτερη που συνδέονται στο πάνω υποδίκτυο συνεπώς ο διακόπτης εισόδου είναι bar και εξόδου cross (*Πίνακας 1-15*).

Πίνακας 1-15: Τέταρτη επανάληψη (δρομολόγηση διακοπών).

Νούμερο διακόπτη (εις/εξ) **Κατάσταση δ.εξ.** **Κατάσταση δ.εις.**

0

bar

bar

1

bar

cross

2

cross

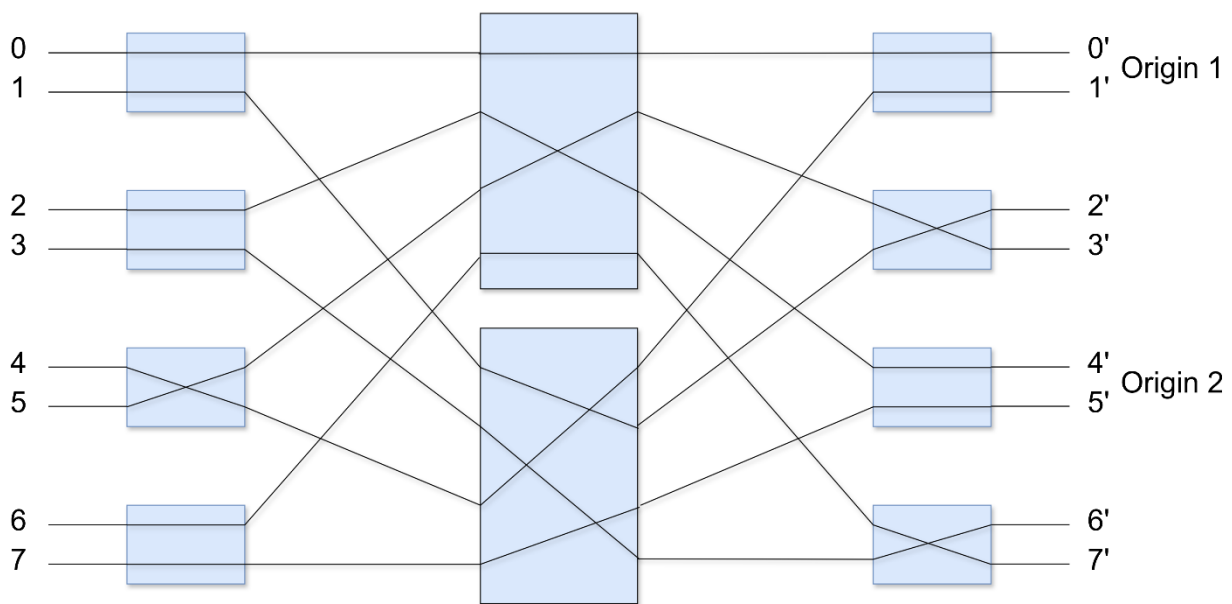
bar

3

bar

cross

Ολόκληρο το επίπεδο παρουσιάζεται δρομολογημένο στο *Σχήμα 1-14*. Όσον αφορά τους πίνακες δρομολόγησης εισόδου στα επόμενα επίπεδα (άνω και κάτω υποδίκτυα) οι συνδέσεις φαίνονται καθαρά πάνω στο σχήμα. Οι καταστάσεις διακοπών κάθε επιπέδου (cross και bar) δηλώνονται με ένα δυαδικό διάνυσμα του οποίου το κάθε bit αντιστοιχεί σε ένα διακόπτη και η τιμή του δηλώνει την κατάστασή του αντίστοιχου διακόπτη, 1 για cross και 0 για bar. Για το συγκεκριμένο παράδειγμα το διάνυσμα που παράγεται είναι 00100101. Η διαδικασία για την δρομολόγηση των εσωτερικών υποδικτύων συνεχίζεται στους *Πίνακες 1-16* και *1-17*.



Σχήμα 1-14: Η εικόνα της δρομολόγησης όλων των διακοπών μετά την ταξινόμηση.

Πίνακας 1-16: Ζευγάρια δρομολόγησης για το πάνω υποδίκτυο.

Διακόπτες εισόδου	Διακόπτες εξόδου	Υποδίκτυο
0	0'	U
5	3'	U
2	4'	U
6	7'	U

Πίνακας 1-17: Ζευγάρια δρομολόγησης για το κάτω υποδίκτυο.

Διακόπτες εισόδου	Διακόπτες εξόδου	Υποδίκτυο
1	2'	L
4	3'	L
3	4'	L
7	7'	L

Αφού χωριστούν τις διαδρομές που περνούν από το πάνω και το κάτω υποδίκτυο, διαιρούνται με το 2. Αυτή η διαίρεση είναι απαραίτητη για να παραχθεί μια έγκυρη δρομολόγηση πίνακα δεδομένου ότι τα υποδίκτυα έχουν το μισό αριθμό θυρών εισόδου και εξόδου (πίνακες 1-18 και 1-19).

Πίνακας 1-18: Ζευγάρια δρομολόγησης για το πάνω υποδίκτυο (διαιρούμε με το 2).

Διακόπτες εισόδου	Διακόπτες εξόδου	Υποδίκτυο
0	0'	U
2	1'	U

1	2'	U
3	3'	U

Πίνακας 1-19: Ζευγάρια δρομολόγησης για το κάτω υποδίκτυο (διαιρούμε με το 2).

Διακόπτες εισόδου	Διακόπτες εξόδου	Υποδίκτυο
0	1'	L
2	0'	L
1	3'	L
3	2'	L

Τέλος, τα ζεύγη οργανώνονται με τον ίδιο τρόπο (αύξουσα σειρά εισόδου) όπως το αρχικός πίνακας δρομολόγησης εισόδου-εξόδου (πίνακες 1-20 και 1-21).

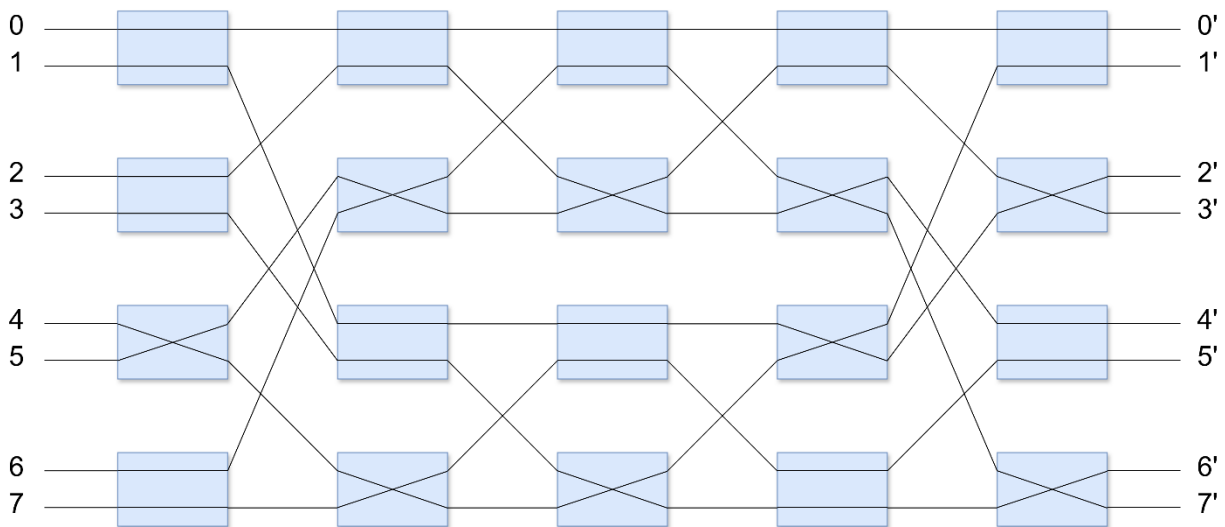
Πίνακας 1-20: Πίνακας δρομολόγησης για το πάνω υποδίκτυο.

Διακόπτες εισόδου	Διακόπτες εξόδου	Υποδίκτυο
0	0'	U
1	2'	U
2	1'	U
3	3'	U

Πίνακας 1-21: Πίνακας δρομολόγησης για το κάτω υποδίκτυο.

Διακόπτες εισόδου	Διακόπτες εξόδου	Υποδίκτυο
0	1'	L
1	3'	L
2	0'	L
3	3'	L

Σε αυτό το σημείο, οι πίνακες μεταβιβάζονται ως είσοδοι στα υποδίκτυα του επόμενου επιπέδου και χρησιμοποιούνται για τη δρομολόγησης του (Πρόταση 6). Η διαδικασία επαναλαμβάνεται μέχρι να τεθούν όλοι οι διακόπτες σε όλα τα επίπεδα σειρά. Ολόκληρο το πλήρως δρομολογημένο δίκτυο φαίνεται στο Σχήμα 1-15.



Σχήμα 1-15: Το δίκτυο Benes συνολικά δρομολογημένο.

Η διαδικασία είναι γραμμική και είναι ακριβώς ίδια για έναν γενικό αριθμό εισόδων. Απλώς συνεχίζουμε να ρυθμίζουμε τους διακόπτες μέχρι το τέλος. Αφού χρησιμοποιούνται μόνο θέσεις $2 \times m$ για εισόδους 2^r , χρειάζονται $2^{r-1} = n/2$ κύκλοι ρολογιού. Οι κύκλοι των υπόλοιπων μονάδων είναι, μονάδα 1: 1, μονάδα 2: $n/2$, μονάδα 3: 1, μονάδα 4: $n/2$, και έτσι συνολικά έχουμε, $1 + n/2 + 1 + n/2 = 2 + n \approx n$ κύκλους. Λόγω της αναδρομικότητας του δικτύου Benes (Πρόταση 1) υπάρχουν $\log_2 n = r$ επίπεδα υποδικτύων. Η πολυπλοκότητα ολόκληρης της αρχιτεκτονικής είναι $\log_2 n \times n$, η ελάχιστη πολυπλοκότητα για τη δρομολόγηση ενός Benes δίκτυο με μία μονάδα επεξεργασίας [32].

1.5 Υλοποίηση

Το κύκλωμα υλοποιήθηκε δύο φορές. Αρχικά η αρχιτεκτονική υλοποιήθηκε για να διαπιστωθεί η απόδοση της σε σχέση με άλλες υλοποιήσεις αλγορίθμων δρομολόγησης. Η δεύτερη υλοποίηση αφορούσε τις ανάγκες του έργου. Η υλοποίηση για το έργο ήταν πιο περίπλοκη καθώς μαζί με το κύκλωμα του αλγορίθμου χρειάστηκε να υλοποιηθεί και το κύκλωμα διεπαφής με το υπόλοιπο μέρος της διάταξης.

Για την πρώτη υλοποίηση χρησιμοποιήθηκε η πλακέτα ZCU-104. Η υλοποίηση αφορούσε μόνο το κύκλωμα δρομολόγησης. Η κατανάλωση υλικού, ισχύος, η συχνότητα λειτουργίας και ο συνολικός χρόνος δρομολόγησης φαίνεται στον Πίνακα 1-22 για κυκλώματα δρομολόγησης δικτύων 8x8, 16x16, 32x32, 64x64.

Πίνακας 1-22: Αποτελέσματα απομονωμένης υλοποίησης.

Μέγεθος Benes	8	16	32	64
LUTs	1416	3656	13549	35925
FlipFlops	561	1558	4002	9993
P (W)	0.036	0.105	0.241	0.437

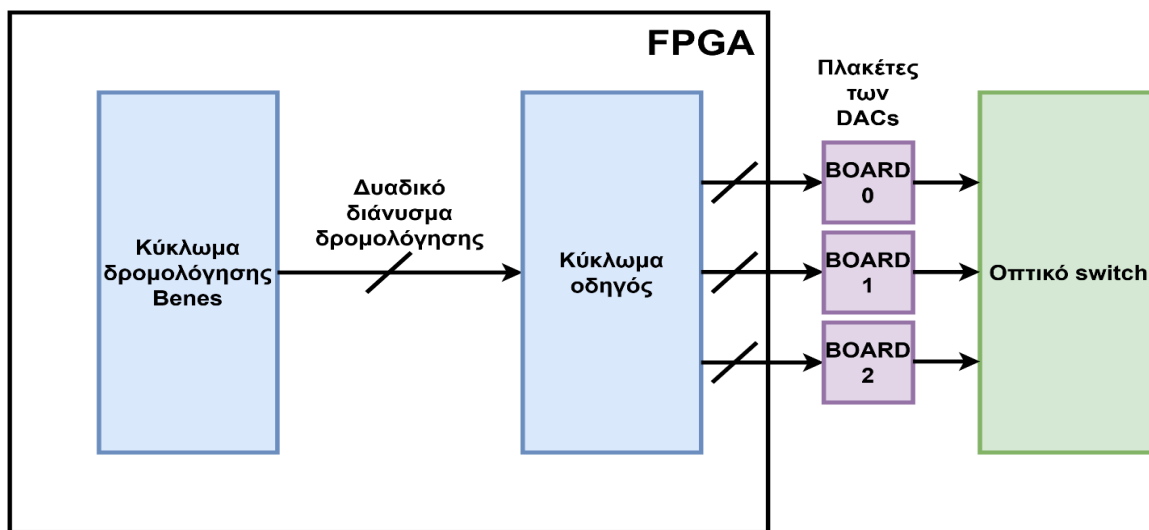
Συχνότητα (MHz)	250	250	166	125
Χρόνος δρομολόγησης(ns)	96	256	960	3072

Από όλες τις υπόλοιπες υλοποιήσεις στην Βιβλιογραφία η πιο κοντινή υλοποίηση από άποψη απόδοσης στην προτεινόμενη είναι η [40]. Πρόκειται για μια υλοποίηση ASIC ενός διαφορετικού αλγορίθμου δρομολόγησης με μικρότερη αλγοριθμική πολυπλοκότητα $(\log_2 n)^2$. Έχοντας ως μετρική σύγκρισης τον πολλαπλασιασμό υλικού με συνολικού χρόνου δρομολόγησης και λαμβάνοντας υπ' όψιν ότι στις πλατφόρμες ASIC τα κυκλώματα γενικά καταναλώνουν λιγότερους πόρους υλικού ενώ είναι πιο γρήγορα [41], υπολογίζεται ότι η μετρική για την προτεινόμενη υλοποίηση (χρόνος δρομολόγησης × υλικό) είναι 6.12 φορές μεγαλύτερη.

Για τις ανάγκες του έργου το κύκλωμα υλοποιήθηκε στην πλακέτα NEXYS 4 DDR, με στόχο την δρομολόγηση του οπτικού μεταγωγέα 16×16 , ο οποίος αποτελείται από μικρότερους 2×2 οπτικούς διακόπτες και η δομή του είναι ταυτόσημη με αυτή ενός Benes 16×16 . Έχει 4 επίπεδα και $2^4 - 1 = 7$ στάδια. Κάθε στάδιο έχει 8 2×2 οπτικούς διακόπτες άρα συνολικά $8 \times 7 = 56$ 2×2 οπτικούς διακόπτες. Η οπτικοί αυτοί διακόπτες είναι δύο εισόδων × εξόδων και μιμούνται την μαθηματική λειτουργία των διακοπών που περιεγράφηκε στις προηγούμενες ενότητες. Η κατάσταση τους ελέγχεται από την τιμή δύο υποδοχών τάσης. Αυτό σημαίνει πως για να ελεγχθούν όλοι οι διακόπτες του δικτύου χρειάζονται να δοθούν $56 \times 2 = 112$ τάσεις στο μεταγωγέα. Καθώς η έξοδος του κυκλώματος είναι μόνο οι καταστάσεις των διακοπών (bar ή cross) οι οποίες δίνονται σαν ένα δυαδικό διάνυσμα (με τιμές 0 ή 1 σε κάθε bit) δεν γίνεται να δοθούν όπως είναι στο μεταγωγέα. Θα πρέπει να μεσολαβήσει ένα δεύτερο κύκλωμα υποδοχής (interface) το οποίο θα λαμβάνει το δυαδικό διάνυσμα, θα το μετατρέπει με κάποιο μηχανισμό στις σωστές τάσεις και στην συνέχεια με πρωτόκολλο επικοινωνίας θα το στέλνει σε κάποια συσκευή η οποία θα μπορεί να ασκήσει την σωστή τάση στους υποδοχείς του δικτύου. Οι συσκευές που επιλέχθηκαν για αυτό το ρόλο είναι τα evaluation board ad5370[42] της analog devices.

Κάθε evaluation board περιέχει 40 dacs. Τα dacs (digital to analog converters) είναι ειδικά κυκλώματα που λαμβάνουν μία ψηφιακή τιμή συγκεκριμένης ακρίβειας (8,16,32 bit) και σύμφωνα με μια συνάρτηση μεταφοράς εφαρμόζουν την κατάλληλη τάση στις εξόδους τους. Η ψηφιακές αυτές τιμές στέλνονται από το FPGA στα evaluation boards μέσω του πρωτοκόλλου SPI (Παράρτημα) το οποίο μεταδίδει σειριακά την τιμή για κάθε dac σε καθένα evaluation board ξεχωριστά. Το πρωτόκολλο αυτό είναι πολύ διαδεδομένο και χρησιμοποιείται ευρέως από πολλές εταιρίες σαν interface για τις συσκευές τους. Συνολικά χρειάζονται 3 evaluation boards (με 40 dacs το καθένα) έτσι ώστε να ικανοποιηθούν οι 112 υποδοχείς όλων των διακοπών (2×56).

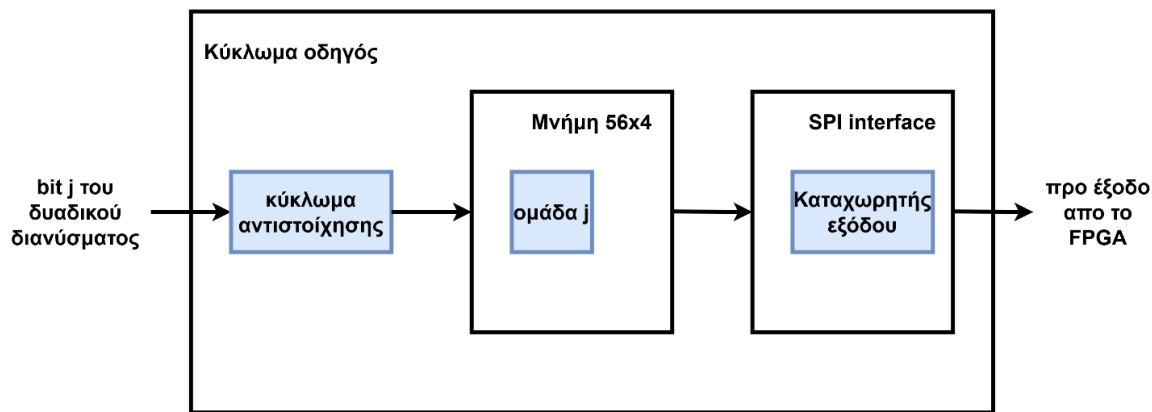
Η αντιστοίχιση του δυαδικού διανύσματος το οποίο παράγεται από το κύκλωμα του αλγορίθμου, με το σωστό δίδυμο τιμών τάσης για κάθε 2×2 γίνεται από ένα ενδιάμεσο κύκλωμα που μεσολαβεί μεταξύ του κυκλώματος του αλγορίθμου και των εξόδων του FPGA. Οι εξόδοι αυτοί είναι ουσιαστικά τα σήματα του πρωτοκόλλου SPI που αντιστοιχούν σε κάθε evaluation board. Το ενδιάμεσο κύκλωμα λαμβάνει τις τιμές του διανύσματος, τις μεταφράζει σε δίδυμα τάσεων τα οποία αποθηκεύει σε μια προσωρινή μνήμη και μόλις ολοκληρωθεί η αποθήκευση ξεκινάει να στέλνει τις τιμές μέσω του SPI στα boards. Τα boards με την σειρά τους ρυθμίζουν τα dacs και η δρομολόγηση του γενικού 16×16 οπτικού μεταγωγέα ολοκληρώνεται. Στο Σχήμα 1-16 βλέπουμε το πως το κύκλωμα συνδέεται με τα υπόλοιπα μέρη του συστήματος οδήγησης.



Σχήμα 1-16: Η υλοποίηση του κυκλώματος στο πλαίσιο του έργου.

Ο μηχανισμός που χρησιμοποιεί το ενδιάμεσο κύκλωμα είναι ένα απλός πίνακας αντιστοίχισης. Αφού οι διακόπτες χρειάζονται δύο τάσεις για να τεθούν σε μία κατάσταση (bar ή cross) συνολικά χρειάζονται για κάθε διακόπτη τέσσερις τιμές. Αυτό σημαίνει πως ο πίνακας αντιστοίχισης έχει συνολικά καταχωρημένες 4×56 τιμές ή αλλιώς 56 ομάδες των 4 τιμών.

Η διαδικασία της αντιστοίχισης μπορεί να συνοψιστεί ως εξής. Για ένα συγκεκριμένο 2×2 διακόπτη που αντιστοιχεί στην θέση i , του δυαδικού διανύσματος, το κύκλωμα οδηγός ελέγχει εάν η τιμή είναι 0 ή 1. Εάν η τιμή είναι 0 τότε επιλέγει τις πρώτες 2 τιμές ομάδας μνήμης i (των 4 τιμών), εάν είναι 1 τότε επιλέγει τις άλλες δυο. Οι τιμές αυτές στέλνονται στου καταχωρητές εξόδων οι οποίοι ελέγχονται από τα SPI interface που επικοινωνούν με κάθε πλακέτα. Μόλις η διαδικασία ολοκληρωθεί για όλα τα ψηφία του διανύσματος του κύκλωμα ενεργοποιεί τα SPI interfaces τα οποία μεταδίδουν τους κωδικούς τάσεις στις πλακέτες και ενεργοποιούν τις εξόδους dacs τους. Η διαδικασία της αντιστοίχισης παρουσιάζεται στο Σχήμα 1-17.



Σχήμα 1-17: Η μετάφραση του δυαδικού διανύσματος σε τιμές τάσεις για τα DACs.

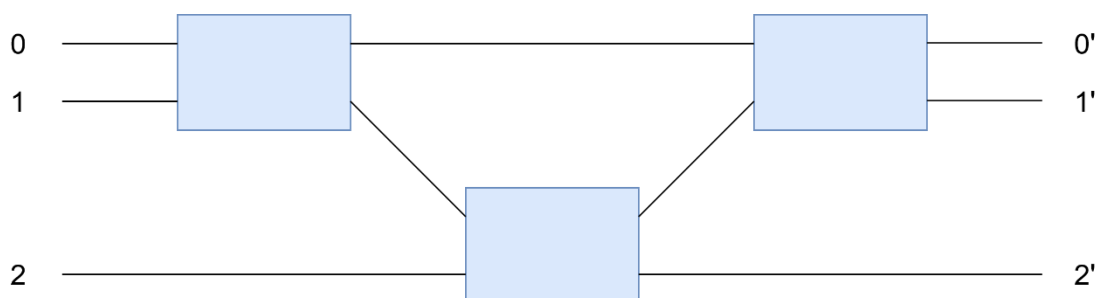
Η κατανάλωση του υλικού, ισχύος καθώς και η μέγιστη δυνατή συχνότητα λειτουργίας σύμφωνα με το πρόγραμμα νινάδο φαίνονται στον Πίνακα 1-23.

Πίνακας 1-23: Αποτελέσματα υλοποίησης για τις ανάγκες του έργου.

Μέγεθος Benes	16
LUTs	3961
FlipFlops	1695
P (W)	0.15
Συχνότητα (MHz)	100
Χρόνος δρομολόγησης(ns)	640

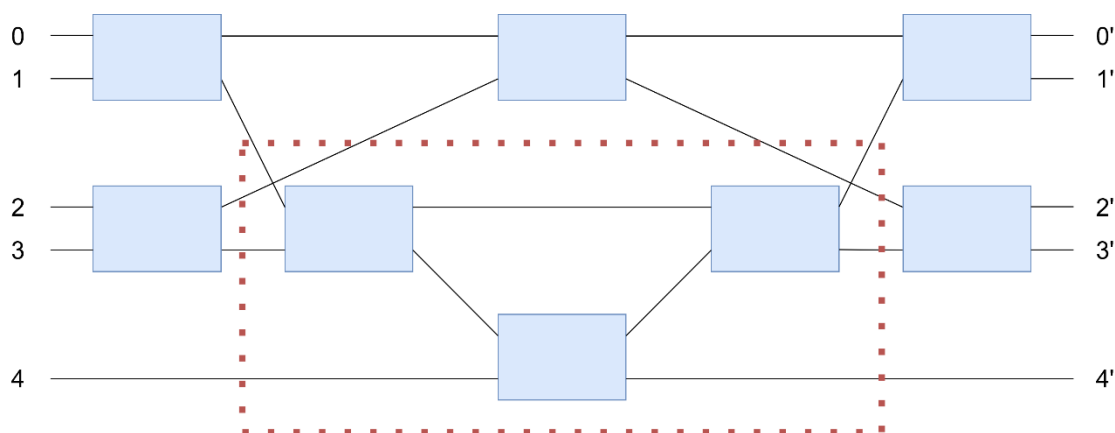
1.6 Δρομολόγηση δικτύων Benes γενικής μορφής (arbitrary size Benes Network) με τον ίδιο αλγόριθμο.

Καθ' όλη την διάρκεια του κεφαλαίου το Δίκτυο Benes αντιμετωπίζεται ως ένα δίκτυο του οποίου ο αριθμός των εισόδων(και εξόδων) είναι αυστηρά δυνάμεις του 2. Είναι δυνατόν όμως να κατασκευαστούν δίκτυα τα οποία έχουν παρόμοιες ιδιότητες με το κλασσικό δίκτυο Benes (λογαριθμική σχέση εισόδων και καθυστέρησης, rearrangeably non-blocking) με εισόδους οποιοδήποτε αριθμό. Τα δίκτυα αυτά ονομάζονται δίκτυα Benes γενικής μορφής (arbitrary size Benes Networks)[43]. Η κατασκευή ενός τέτοιου δικτύου ξεκινάει από το δίκτυο 3×3 Benes που φαίνεται στο Σχήμα 1-18.

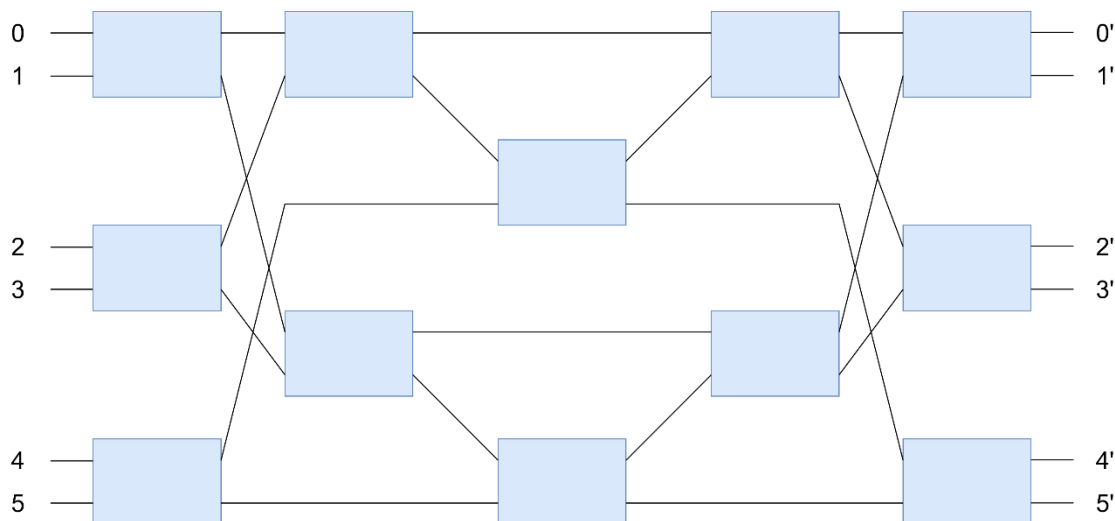


Σχήμα 1-18: Benes γενικής μορφής 3×3.

Χρησιμοποιώντας το AS-Benes 3×3 συνθέτονται τα δίκτυα 5×5 και 6×6 των Σχημάτων 1-19, 1-20 αντίστοιχα.

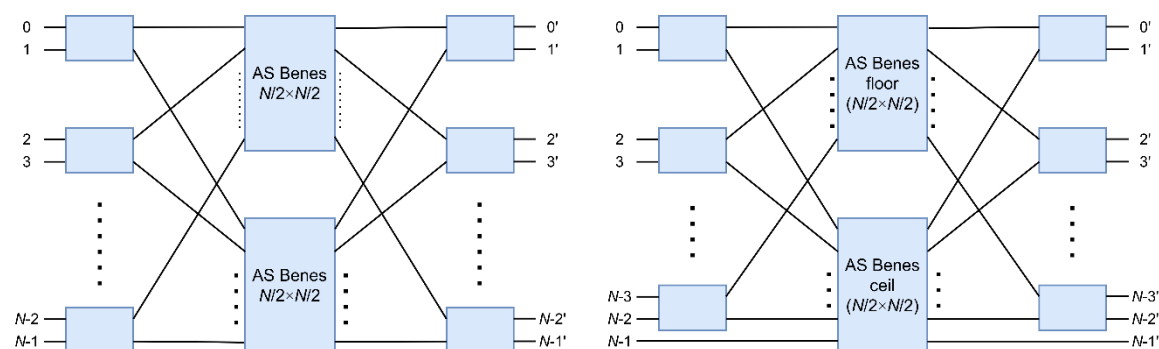


Σχήμα 1-19: AS-Benes 5×5 .



Σχήμα 1-20: AS-Benes 6×6 .

Γενικεύοντας περαιτέρω καταλήγουμε στην γενική μορφή των AS-Benes δικτύων που φαίνεται στο Σχήμα 1-21.



Σχήμα 1-21: Γενική μορφή AS-Benes.

Τα υποδίκτυα είναι και αυτά AS-Benes μικρότερου μεγέθους που σημαίνει ότι και το AS-Benes είναι ένα δίκτυο αναδρομικής μορφής όπως το κανονικό Benes. Αν ο αριθμός των εισόδων διαιρείται με το 2 τότε το πάνω και το κάτω υποδίκτυο είναι μεγέθους $N/2 \times N/2$. Αν ο αριθμός των

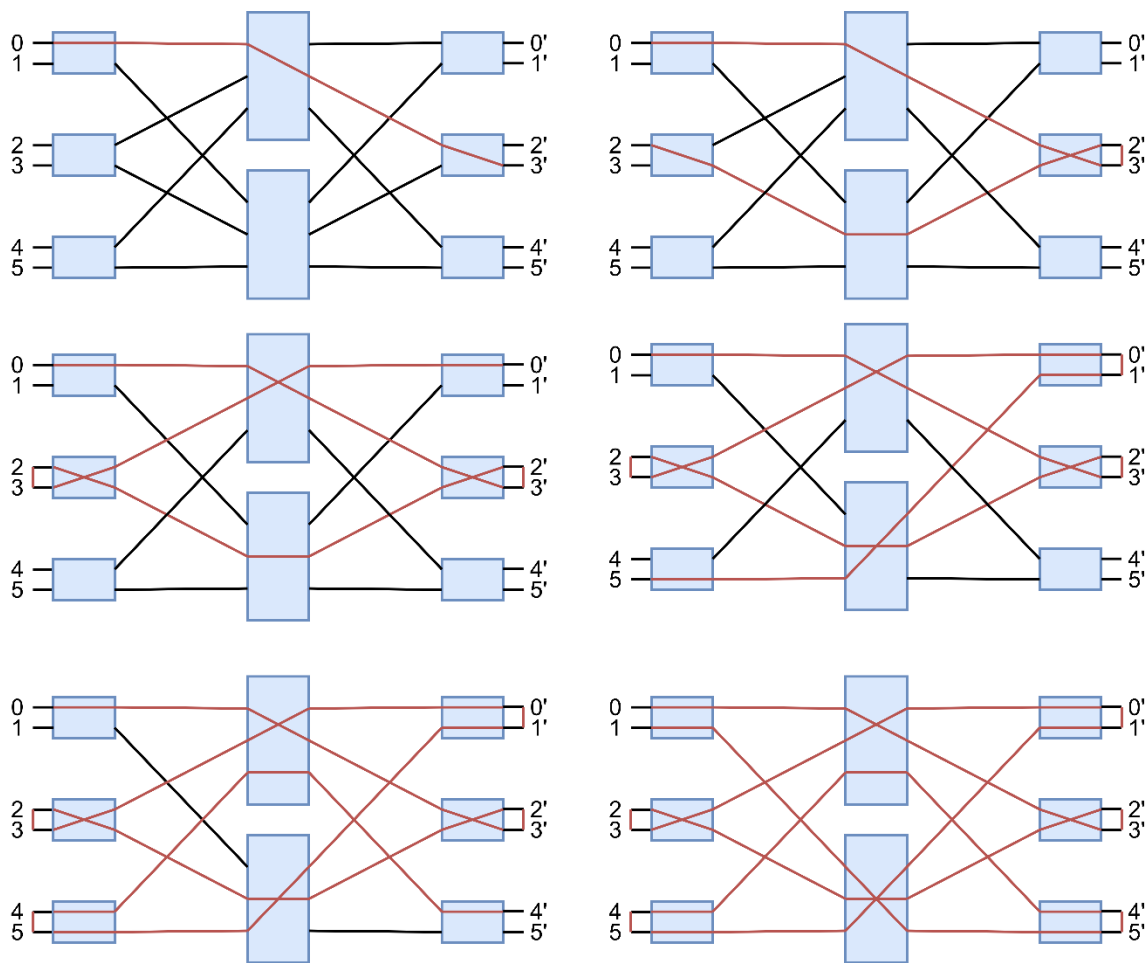
εισόδων δεν διαιρείται με το 2 τότε το πάνω υποδίκτυο έχει εισόδους $\text{floor}(N/2)$ και το κάτω $\text{ceil}(N/2)$.

Η διαδικασία δρομολόγησης των δικτύων AS-Benes μπορεί να γίνει με τον αλγόριθμο που παρουσιάστηκε στις προηγούμενες ενότητες. Αρχικά παρατηρούμε ότι αφού και το AS-Benes είναι αναδρομικής μορφής ισχύει ότι εάν δρομολογηθεί σωστά το πρώτο επίπεδο τότε μπορούν να δρομολογηθούν όλα. Ακόμα, τα βασικά στοιχεία του AS-Benes είναι ίδια με αυτά του Benes (δηλαδή 2×2 διακόπτες cross ή bar). Αυτό σημαίνει ότι η δρομολόγηση του δικτύου συνεπάγεται τον σωστό χαρακτηρισμό της κατάστασης κάθε διακόπτη ως bar ή cross ώστε να ικανοποιείται ο εκάστοτε πίνακας δρομολόγησης. Λαμβάνοντας υπόψιν τα παραπάνω μπορεί να αποδειχθεί ότι ο αρχικός αλγόριθμος μπορεί να δρομολογήσει σωστά AS-Benes δίκτυα χωρίς αλλαγές.

Για τα δίκτυα AS-Benes με άρτιο αριθμό εισόδων είναι προφανές ότι η δρομολόγηση θα γίνει σωστά με τον προτεινόμενο αλγόριθμο καθώς τα πάνω και κάτω υποδίκτυα έχουν τον ίδιο αριθμό εισόδων, εξόδων. Αυτό σημαίνει ότι γίνεται να δρομολογούμε το μονοπάτι εναλλάξ όπως και στην περίπτωση του συμβατικού Benes. Στο Σχήμα 1-22 φαίνεται η διαδικασία δρομολόγησης ενός δικτύου 6×6 με τον ακόλουθο πίνακα δρομολόγησης.

Πίνακας 1-24: Πίνακας δρομολόγησης 6×6 AS-Benes.

Είσοδοι	Έξοδοι
0	3'
1	5'
2	2'
3	0'
4	4'
5	1'



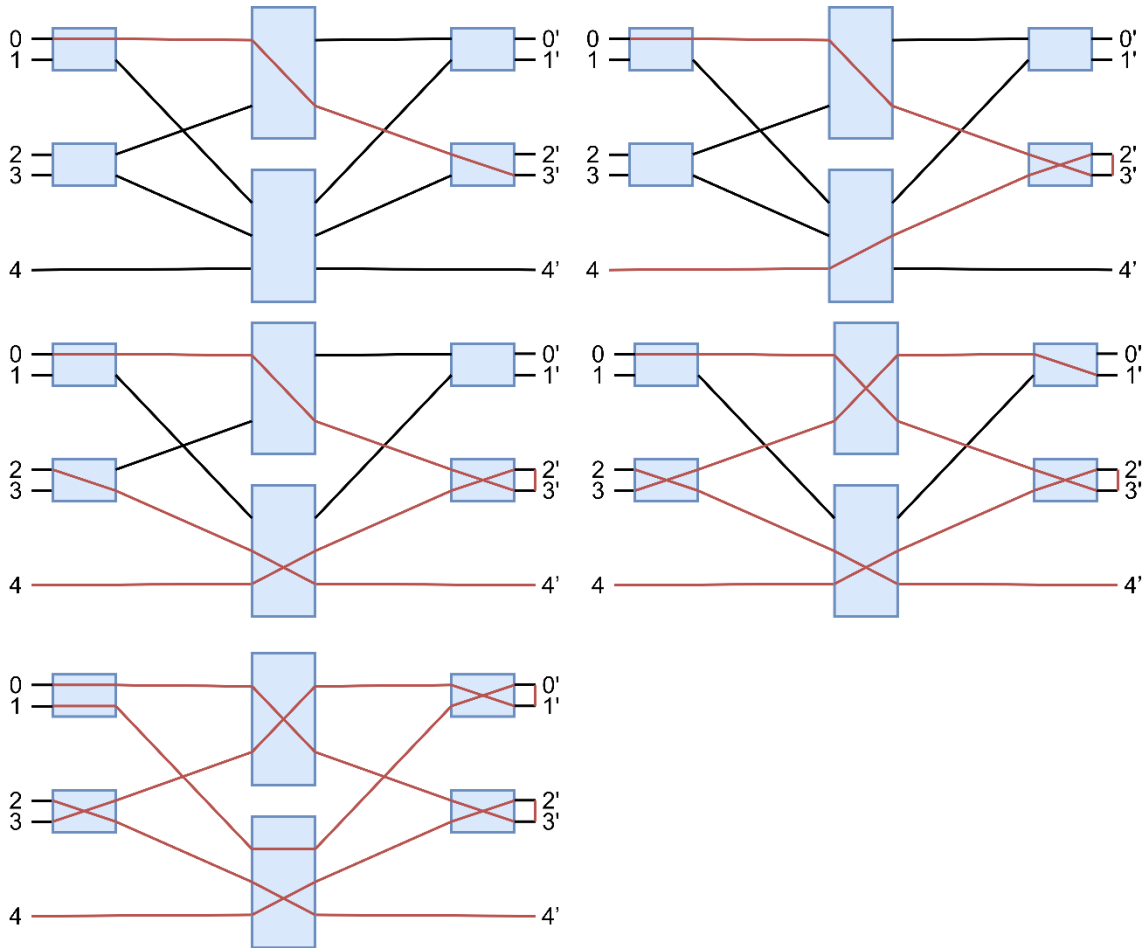
Σχήμα 1-22: Δρομολόγηση AS-Benes 6×6.

Και για τα δίκτυα AS-Benes με περιττό αριθμό εισόδων ο ίδιος αλγόριθμος οδηγεί πάντα σε σωστό αποτέλεσμα. Παρότι το πάνω και το κάτω υποδίκτυο δεν έχουν τον ίδιο αριθμό εισόδων, εξόδων η επιπλέον είσοδος και έξοδος του κάτω υποδικτύου συνδέεται αποκλειστικά με την εναπομείναντα είσοδο, έξοδο του δικτύου. Ως κόμβοι του μονοπατιού, η είσοδος (και έξοδος) $N-1$ ($N-1'$) είναι από μόνες τους κλειστά μονοπάτια (δεν υπάρχει τρόπος έτσι ώστε να συνεχίσει το μονοπάτι από το πάνω υποδίκτυο) και μόλις το μονοπάτι φτάσει σε αυτά πρέπει να ξαναρχίσει από καινούργιο αδέσμευτο διακόπτη. Αυτό σημαίνει πως δεν γίνεται να κολλήσει η δρομολόγηση σε κάποιο σημείο λόγω έλλειψης μονοπατιού προς κάποιο υποδίκτυο και άρα στο τέλος της διαδικασίας το δίκτυο είναι δρομολογημένο σωστά. Στο Σχήμα 1-23 παρουσιάζεται η δρομολόγηση ενός δικτύου 5×5 βήμα-βήμα.

Πίνακας 1-25: Πίνακας δρομολόγησης 5×5 AS-Benes.

Είσοδοι	Έξοδοι
0	3'
1	0'

2	4'
3	1'
4	2'



Σχήμα 1-23: Δρομολόγηση AS-Benes 5×5.

Από όλα τα παραπάνω εξάγεται το συμπέρασμα πως ο προτεινόμενος αλγόριθμος μπορεί να δρομολογήσει δίκτυο Benes οποιαδήποτε μορφής σε θεωρητικό επίπεδο. Σε πρακτικό επίπεδο θα πρέπει να υπάρξουν ορισμένες αλλαγές στην αρχιτεκτονική έτσι ώστε να λαμβάνονται υπόψιν όλες οι περιπτώσεις, πρόβλημα το οποίο αφήνεται ως αντικείμενο μελλοντικής έρευνας.

1.7 Βιβλιογραφία κεφαλαίου

- [1] R. Hema, S. Subbulakshmi, M. Sundararajan and M. Nagarajan, "Automation of Smart Home Appliances Using IoT," 2024 10th International Conference on Electrical Energy Systems (ICEES), Chennai, India, 2024, pp. 1-5, doi: 10.1109/ICEES61253.2024.10776820.
- [2] Olumuyiwa, O.; Chen, Y. Virtual CANBUS and Ethernet Switching in Future Smart Cars Using Hybrid Architecture. Electronics 2022, 11, 3428. <https://doi.org/10.3390/electronics11213428>

- [3] Ethernet Alliance, *2024 Ethernet Roadmap: Digital Version*, Mar. 2024. [Online]. Available: <https://ethernetalliance.org/wp-content/uploads/2024/03/2024-Ethernet-Roadmap-Digital-Version-March-2024.pdf>. [Accessed: Nov. 13, 2024].
- [4] Koronen, Carolina, Max Åhman, and Lars J. Nilsson. "Data centres in future European energy systems—energy efficiency, integration and policy." *Energy Efficiency* 13.1 (2020): 129-144, <https://doi.org/10.1007/s12053-019-09833-8>
- [5] de Assuncao, Marcos Dias, Alexandre da Silva Veith, and Rajkumar Buyya. "Distributed data stream processing and edge computing: A survey on resource elasticity and future directions." *Journal of Network and Computer Applications* 103 (2018): 1-17, <https://doi.org/10.1016/j.jnca.2017.12.001>.
- [6] Kruskal, C.P. and Snir, M., 1986. A unified theory of interconnection network structure. *Theoretical Computer Science*, 48, pp.75-94, [https://doi.org/10.1016/0304-3975\(86\)90084-8](https://doi.org/10.1016/0304-3975(86)90084-8).
- [7] B. Qiao, F. Shi and W. Ji, "A New Hierarchical Interconnection Network for Multi-core Processor," *2007 2nd IEEE Conference on Industrial Electronics and Applications*, Harbin, China, 2007, pp. 246-250, doi: 10.1109/ICIEA.2007.4318408.
- [8] Chuan-Lin Wu and Tse-Yun Feng, "The Reverse-Exchange Interconnection Network," in *IEEE Transactions on Computers*, vol. C-29, no. 9, pp. 801-811, Sept. 1980, doi: 10.1109/TC.1980.1675679.
- [9] Zhang, Xiaodong, Elise Miller-Hooks, and Kevin Denny. "Assessing the role of network topology in transportation network resilience." *Journal of transport geography* 46 (2015): 35-45, doi: <https://doi.org/10.1016/j.jtrangeo.2015.05.006>
- [10] Bermond, J.C. and Ergincan, F.Ö., 1996. Bus interconnection networks. *Discrete Applied Mathematics*, 68(1-2), pp.1-15, [https://doi.org/10.1016/0166-218X\(95\)00046-T](https://doi.org/10.1016/0166-218X(95)00046-T).
- [11] Duato, J., Yalamanchili, S. and Ni, L., 2002. *Interconnection networks*. Elsevier.
- [12] B. Webb and A. Louri, "A class of highly scalable optical crossbar-connected interconnection networks (SOCNs) for parallel computing systems," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 5, pp. 444-458, May 2000, doi: 10.1109/71.852398.
- [13] Amodu, O.A.; Othman, M.; Yunus, N.A.M.; Hanapi, Z.M. A Primer on Design Aspects and Recent Advances in Shuffle Exchange Multistage Interconnection Networks. *Symmetry* 2021, 13, 378. <https://doi.org/10.3390/sym13030378>
- [14] A. M. Maliszewski, E. Roloff, E. D. Carreño, D. Griebler, L. P. Gasparý and P. O. A. Navaux, "Performance and Cost-aware HPC in Clouds: A Network Interconnection Assessment," *2020 IEEE Symposium on Computers and Communications (ISCC)*, Rennes, France, 2020, pp. 1-6, doi: 10.1109/ISCC50000.2020.9219554.

- [15] K. Ahmed, J. Liu, S. Eidenbenz and J. Zerr, "Scalable Interconnection Network Models for Rapid Performance Prediction of HPC Applications," *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Sydney, NSW, Australia, 2016, pp. 1069-1078, doi: 10.1109/HPCC-SmartCity-DSS.2016.0151.
- [16] A. Ruhela, S. Xu, K. V. Manian, H. Subramoni and D. K. Panda, "Analyzing and Understanding the Impact of Interconnect Performance on HPC, Big Data, and Deep Learning Applications: A Case Study with InfiniBand EDR and HDR," *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, New Orleans, LA, USA, 2020, pp. 869-878, doi: 10.1109/IPDPSW50202.2020.00147.
- [17] S. M. Nabavinejad, M. Baharloo, K. -C. Chen, M. Palesi, T. Kogel and M. Ebrahimi, "An Overview of Efficient Interconnection Networks for Deep Neural Network Accelerators," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 3, pp. 268-282, Sept. 2020, doi: 10.1109/JETCAS.2020.3022920.
- [18] Ying Ping Zhang, Taikyeong Jeong, Fei Chen, Haiping Wu, R. Nitzsche and G. R. Gao, "A study of the on-chip interconnection network for the IBM Cyclops64 multi-core architecture," *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, Rhodes, Greece, 2006, pp. 10 pp.-, doi: 10.1109/IPDPS.2006.1639301.
- [19] C. Kachris and I. Tomkos, "A Survey on Optical Interconnects for Data Centers," in *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1021-1036, Fourth Quarter 2012, doi: 10.1109/SURV.2011.122111.00069.
- [20] Lenfant, "Parallel Permutations of Data: A Benes Network Control Algorithm for Frequently Used Permutations," in *IEEE Transactions on Computers*, vol. C-27, no. 7, pp. 637-647, July 1978, doi: 10.1109/TC.1978.1675164.
- [21] H. J. Chao, Zhigang Jing and S. Y. Liew, "Matching algorithms for three-stage bufferless Clos network switches," in *IEEE Communications Magazine*, vol. 41, no. 10, pp. 46-54, Oct. 2003, doi: 10.1109/MCOM.2003.1235594.
- [22] Cheng, Qixiang, et al. "Recent advances in optical technologies for data centers: a review." *Optica* 5.11 (2018): 1354-1370, doi: <https://doi.org/10.1364/OPTICA.5.001354>
- [23] C. Kachris, K. Kanonakis and I. Tomkos, "Optical interconnection networks in data centers: recent trends and future challenges," in *IEEE Communications Magazine*, vol. 51, no. 9, pp. 39-45, September 2013, doi: 10.1109/MCOM.2013.6588648.

- [24] B. G. Lee and N. Dupuis, "Silicon Photonic Switch Fabrics: Technology and Architecture," in *Journal of Lightwave Technology*, vol. 37, no. 1, pp. 6-20, 1 Jan.1, 2019, doi: 10.1109/JLT.2018.2876828.
- [25] S. J. Ben Yoo, "Prospects and Challenges of Photonic Switching in Data Centers and Computing Systems," in *Journal of Lightwave Technology*, vol. 40, no. 8, pp. 2214-2243, 15 April15, 2022, doi: 10.1109/JLT.2021.3136570.
- [26] Shen, Lin, et al. "Silicon optical filters reconfigured from a 16×16 Benes switch matrix." *Optics Express* 27.12 (2019): 16945-16957, <https://doi.org/10.1364/OE.27.016945>
- [27] Qiao, Lei, Weijie Tang, and Tao Chu. " 32×32 silicon electro-optic switch with built-in monitors and balanced-status units." *Scientific Reports* 7.1 (2017): 42306, doi: <https://doi.org/10.1038/srep42306>
- [28] Nikolaidis, D.; Groumas, P.; Kouloumentas, C.; Avramopoulos, H. Novel Benes Network Routing Algorithm and Hardware Implementation. *Technologies* **2022**, *10*, 16. <https://doi.org/10.3390/technologies10010016>
- [29] K. European Commission. CoPackaging of Terabit Direct-Detection and Coherent Optical Engines and Switching Circuits in multiChip Modules for Datacenter Networks and the 5G Optical Fronthaul. Available online: <https://cordis.europa.eu/project/id/871769> (accessed on 23 December 2024).
- [30] Song, Lijia, et al. "Toward calibration-free Mach–Zehnder switches for next-generation silicon photonics." *Photonics Research* 10.3 (2022): 793-801, <https://doi.org/10.1364/PRJ.447478>
- [31] F. Leens, "An introduction to I2C and SPI protocols," in *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8-13, February 2009, doi: 10.1109/MIM.2009.4762946.
- [32] D. C. Opferman and N. T. Tsao-wu, "On a class of rearrangeable switching networks part I: Control algorithm," in *The Bell System Technical Journal*, vol. 50, no. 5, pp. 1579-1600, May-June 1971, doi: 10.1002/j.1538-7305.1971.tb02569.x.
- [33] Abraham Waksman. 1968. A Permutation Network. *J. ACM* 15, 1 (Jan. 1968), 159–163. <https://doi.org/10.1145/321439.321449>.
- [34] Nassimi and Sahni, "A Self-Routing Benes Network and Parallel Permutation Algorithms," in *IEEE Transactions on Computers*, vol. C-30, no. 5, pp. 332-340, May 1981, doi: 10.1109/TC.1981.1675791.
- [35] David Nassimi and Sartaj Sahni. 1980. An optimal routing algorithm for mesh-connected Parallel computers. *J. ACM* 27, 1 (Jan. 1980), 6–29. <https://doi.org/10.1145/322169.322172>

- [36] T. T. Lee and S. Y. Liew, "Parallel routing algorithms in Benes-Clos networks," in *IEEE Transactions on Communications*, vol. 50, no. 11, pp. 1841-1847, Nov. 2002, doi: 10.1109/TCOMM.2002.805258.
- [37] A. Chakrabarty, M. Collier and S. Mukhopadhyay, "Matrix-Based Nonblocking Routing Algorithm for Beneš Networks," *2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, Athens, Greece, 2009, pp. 551-556, doi: 10.1109/ComputationWorld.2009.72.
- [38] Kyungsook Yoon Lee, "A New Benes Network Control Algorithm," in *IEEE Transactions on Computers*, vol. C-36, no. 6, pp. 768-772, June 1987, doi: 10.1109/TC.1987.1676970.
- [39] Abbas Karimi, Kiarash Aghakhani, Seyed Ehsan Manavi, Faraneh Zarafshan, S.A.R. Al-Haddad, Introduction and Analysis of Optimal Routing Algorithm in Benes Networks, *Procedia Computer Science*, Volume 42, 2014, Pages 313-319, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2014.11.068>.
- [40] Jiang, Y. Hardware Implementation of Parallel Algorithm for Setting Up Benes Networks. *Int. J. High Perform. Syst. Archit.* 2017,7, 26–40, <https://doi.org/10.1504/IJHPSA.2017.083645>.
- [41] A. Boutros and V. Betz, "FPGA Architecture: Principles and Progression," in *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 4-29, Secondquarter 2021, doi: 10.1109/MCAS.2021.3071607.
- [42] Analog Devices, "AD5370: 16-Channel, 16-Bit Digital-to-Analog Converter," Analog Devices, [Online]. Available: <https://www.analog.com/en/products/ad5370.html>. [Accessed: Dec. 18, 2024].
- [43] Chang, Chihming, and Rami Melhem. "Arbitrary size Benes networks." *Parallel Processing Letters* 7.03 (1997): 279-284. doi: <https://doi.org/10.1142/S0129626497000292>

Κεφάλαιο 2: Κύκλωμα αναγνώρισης ακολουθιών bit.

2.1 Εισαγωγή

Το Κεφάλαιο 2 είναι αφιερωμένο σε ένα διαφορετικό πρόβλημα από τη δρομολόγηση δικτύων. Διαπραγματεύεται την ανάπτυξη μιας πρωτότυπης δομής με στόχο την αναγνώριση μοτίβων-ακολουθιών bit. Η αναγνώριση ακολουθιών bit είναι μια σημαντική λειτουργία των σύγχρονων συστημάτων υλικού και έχει έντονη παρουσία σε πολλούς τομείς. Αξιοσημείωτα παραδείγματα περιλαμβάνουν την ταξινόμηση επικεφαλίδας πακέτων [1], την ανίχνευση εισβολών δικτύου [2-3], και πολλά άλλα, συμπεριλαμβανομένης της επεξεργασίας εικόνας και της αναζήτησης κειμένου [4]. Ουσιαστικά, η προτεινόμενη αρχιτεκτονική αφορά την αναγνώριση μιας ακολουθίας συγκεκριμένου μεγέθους που δίνεται ως είσοδος από ένα σύνολο ακολουθιών που βρίσκεται στη μνήμη του συστήματος που εκτελεί τη λειτουργία [5]. Δεδομένου ότι το μέγεθος των ακολουθιών είναι

συνήθως μεγαλύτερο από 1000 bit, η κύρια μέθοδος που χρησιμοποιείται από τις προτάσεις της βιβλιογραφίας είναι η ανάλυση της ακολουθίας εισόδου σε μικρότερες υποακολουθίες και στη συνέχεια η χρησιμοποίηση μικρών και αποδοτικών στοιχείων επεξεργασίας για την εκτέλεση αντιστοίχισης με λογικές πράξεις (logical operations) στη μέγιστη απόδοση [6].

Οι σύγχρονες μέθοδοι είναι πολύ αποτελεσματικές στην αναγνώριση ακολουθιών, ωστόσο έχουν ένα σοβαρό μειονέκτημα. Δεν μπορούν να ανιχνεύσουν την ακολουθία όταν το μοτίβο έχει αλλοιωθεί σημαντικά. Ενώ υπάρχουν πολλές αποτελεσματικές λύσεις για την ανίχνευση ακολουθιών [5, 6] που χρησιμοποιούν προηγμένες μεθόδους για τη μεγιστοποίηση της συχνότητας λειτουργίας και τη βέλτιστη χρήση των μπλοκ μνήμης, υπάρχει ελάχιστη έως καθόλου πρόβλεψη για την ανίχνευση ακολουθιών υπό αλλοιώσεις σε υψηλές ταχύτητες. Δεδομένου ότι τα ψηφιακά δεδομένα μπορούν να αναπαριστούν οτιδήποτε, η δυνατότητα να ανιχνεύονται μεγάλου μήκους ακολουθίες bit που έχουν υποστεί σημαντική αλλοίωση μπορεί να αποδειχθεί πολύ σημαντική σε πεδία όπως η αναγνώριση εικόνας [7].

Η προτεινόμενη αρχιτεκτονική λύνει το πρόβλημα της ανίχνευσης χρησιμοποιώντας τη μέθοδο της ψηφιακής συσχέτισης σε συνδυασμό με δέντρα αθροιστών. Δέντρο αθροιστών είναι μια βασική δομή κυκλώματος που χρησιμοποιείται για την άθροιση M αριθμών με το βέλτιστο τρόπο σε $\log_2 M$ βήματα. Το δέντρο αθροιστών έχει $\log_2 M$ επίπεδα (όσα και βήματα). Σε κάθε επίπεδο οι αριθμοί προσθέτονται ανά δύο (όποιος περισσεύει προχωράει στο επόμενο επίπεδο μόνος του) και το αποτέλεσμα της πρόσθεσης είναι είσοδος για το επόμενο επίπεδο. Σε κάθε επίπεδο οι αθροίσεις υποδιπλασιάζονται και στο τέλος δίνεται το συνολικό άθροισμα των αριθμών. Γενικά οι αριθμοί που προσθέτονται είναι πολλών bit (multi-bit) όμως για την ανίχνευση ακολουθιών θεωρούμε ότι οι αριθμοί είναι δυαδικά ψηφία (0 ή 1) και το τελικό άθροισμα δείχνει το πλήθος των λογικών 1. Έχοντας αυτό ως δεδομένο μπορούμε να αποφανθούμε για το αν ένα δυαδικό διάνυσμα εισόδου μεγέθους M είναι το διάνυσμα που ψάχνουμε εφόσον χρησιμοποιήσουμε πύλες XNOR (πύλη ισοδυναμίας) στα αντίστοιχα ψηφία (του διανύσματος εισόδου και του γνωστού διανύσματος) και στη συνέχεια προσθέσουμε τις εξόδους των πυλών με ένα δέντρο αθροιστών. Η τιμή του αθροίσματος είναι μια ένδειξη για το πόσο κοντά, από θέμα ομοιότητας, είναι τα δύο διανύσματα. Μεγάλες τιμές αθροίσματος δείχνουν πως τα διανύσματα παρουσιάζουν μεγάλη ομοιότητα ενώ χαμηλές τιμές δείχνουν πως είναι ανόμοια. Μεγάλο ποσοστό ομοιότητας μπορεί να είναι ένδειξη πως το διάνυσμα εισόδου είναι το διάνυσμα που αναζητείται και έχει υποστεί αλλοιώσεις (bit errors). Η μέθοδος αυτή μοιράζεται την πολυπλοκότητα του δέντρου αθροιστή ($\log_2 M$).

Παρότι το δέντρο αθροιστών είναι μια πολύ ευέλικτη δομή που μπορεί να χρησιμοποιηθεί όχι μόνο στην απλή άθροιση αριθμών αλλά και στην ανίχνευση ακολουθιών, έχει μια σημαντική αδυναμία. Για μεγάλες τιμές του M οι αθροιστές που απαιτούνται για να φτιαχτεί το δέντρο, σε μια απευθείας

υλοποίηση, προκύπτει να είναι πολλών bit. Αυτό σημαίνει πως σταδιακά η κρίσιμη διαδρομή (critical path) του κυκλώματος αυξάνεται και καταλήγει η χρονική καθυστέρηση των υπολογισμών να είναι μη αποδεκτή. Για να ξεπεράσουμε την αδυναμία αυτή και να επιτρέψουμε την κλιμάκωση της ανίχνευσης ακολουθιών σε οποιοδήποτε μέγεθος, αναπτύσσουμε μια πρωτότυπη κυκλωματική δομή βασισμένη σε επιμέρους υπο-δέντρα αθροιστών και κυκλώματα συγκριτών.

Η μέθοδος που ακολουθείται για να κατασκευαστεί η προτεινόμενη δομή είναι η εξής: Έστω ότι έχουμε μια ψηφιακή ακολουθία εισόδου μεγέθους M και θέλουμε να εξετάσουμε πόσο όμοια είναι με μία γνωστή από πριν ακολουθία. Καθώς το M είναι πολύ μεγάλο, η απευθείας χρήση δέντρου αθροιστών οδηγεί σε αθροιστές μεγάλου μεγέθους και χαμηλής συχνότητας οπότε δεν είναι βέλτιστη λύση. Μπορούμε όμως να σπάσουμε την ακολουθία σε Q τμήματα μεγέθους M/Q το καθένα τα οποία υποθέτουμε ότι μπορούν να ελεγχθούν με δέντρα αθροιστών αποδεκτού μεγέθους. Στο άθροισμα εξόδου του κάθε υπο-δέντρου τοποθετείται ένας συγκριτής ο οποίος ελέγχει το άθροισμα και αν είναι πάνω από μία συγκεκριμένη τιμή δίνει την λογική τιμή 1 αλλιώς δίνει την λογική τιμή 0. Με αυτήν την διαδικασία καταλήγουμε σε μια ακολουθία μεγέθους $Q(<M)$ χρησιμοποιώντας αθροιστές μεγέθους $\text{ceil}(\log_2 M/Q)$ (όχι $\text{ceil}(\log_2 M)$). Είναι ξεκάθαρο ότι αν επαναλάβουμε τη διαδικασία αυτή πολλές φορές, η ακολουθία θα μικρύνει ξανά έως ότου εκφυλιστεί σε ένα bit. Είναι επίσης ξεκάθαρο ότι ανάλογα με τις τιμές με τις οποίες συγκρίνουν οι συγκριτές τα αποτελέσματα των επιμέρους υπο-δέντρων, αλλάζει το ποσοστό και η κατανομή ομοιότητας που απαιτείται ώστε το κύκλωμα να ανιχνεύσει το διάνυσμα εισόδου σωστά. Η ευελιξία αυτή που επιδεικνύει η συγκεκριμένη δομή είναι ο λόγος για τον οποίο μπορεί να ανιχνεύσει μεγάλες ακολουθίες ψηφίων σε πολύ μεγάλες ταχύτητες (η κρίσιμη διαδρομή εξαρτάται μόνο από το Q) ακόμα και αν έχουν υποστεί αλλοιώσεις από θόρυβο. Η υλοποίηση της παραπάνω μεθόδου σε λειτουργικό ψηφιακό κύκλωμα είναι το αντικείμενο του Κεφαλαίου 2.

2.2 Βασικές αρχές

Ο κύριος σκοπός της ανίχνευσης ακολουθιών/διανυσμάτων bit είναι η αντιστοίχιση ακολουθίας εισόδου σε μια γνωστή ακολουθία που βρίσκεται στη μνήμη του συστήματος. Συνήθως, οι γνωστές ακολουθίες αναφέρονται ως κανόνες και το μέγεθος του σύνολο κανόνων είναι γνωστό ως σύνολο κανόνων. Τα σύνολα κανόνων ποικίλλουν σε μέγεθος αλλά γενικά είναι χαμηλότερα ή ίσα με 1000 [6]. Σε κάποιες υλοποιήσεις, οι ακολουθίες αναλύονται περαιτέρω σε υποακολουθίες, αλλά αυτό δεν θα συμβαίνει στην προτεινόμενη αρχιτεκτονική. Ο Πίνακας 2-1 παρουσιάζει μία ακολουθία εισόδου, μαζί με το σύνολο κανόνων που χρησιμοποιείται. Το μέγεθος του κανόνα είναι 5 και το μήκος της ακολουθίας είναι 10.

Πίνακας 2-1: Σύνολο κανόνων.

Ακολουθία εισόδου	0011011101
Κανόνας 1	1101001101
Κανόνας 2	0001011101
Κανόνας 3	1001011111
Κανόνας 4	111110101
Κανόνας 5	1011110100

Στη συγκεκριμένη περίπτωση, η ακολουθία εισόδου δεν ταιριάζει ακριβώς με καμία καταχώρηση, αλλά μπορεί να αντιστοιχιστεί με τον Κανόνα 2, καθώς είναι ο πλησιέστερος ως προς την απόσταση Hamming κανόνας. Καθορίζοντας το πλησιέστερο διάνυσμα bit ως προς την απόσταση μπορεί απλά να επιτευχθεί με xnor'ing (τα bit σε αντίστοιχες θέσεις περνούν από πύλες xnor) της ακολουθίας εισόδου με κάθε κανόνα και στη συνέχεια άθροιση των bits του προϊόντος. Ο Πίνακας 2-2 απεικονίζει αυτή τη λειτουργία.

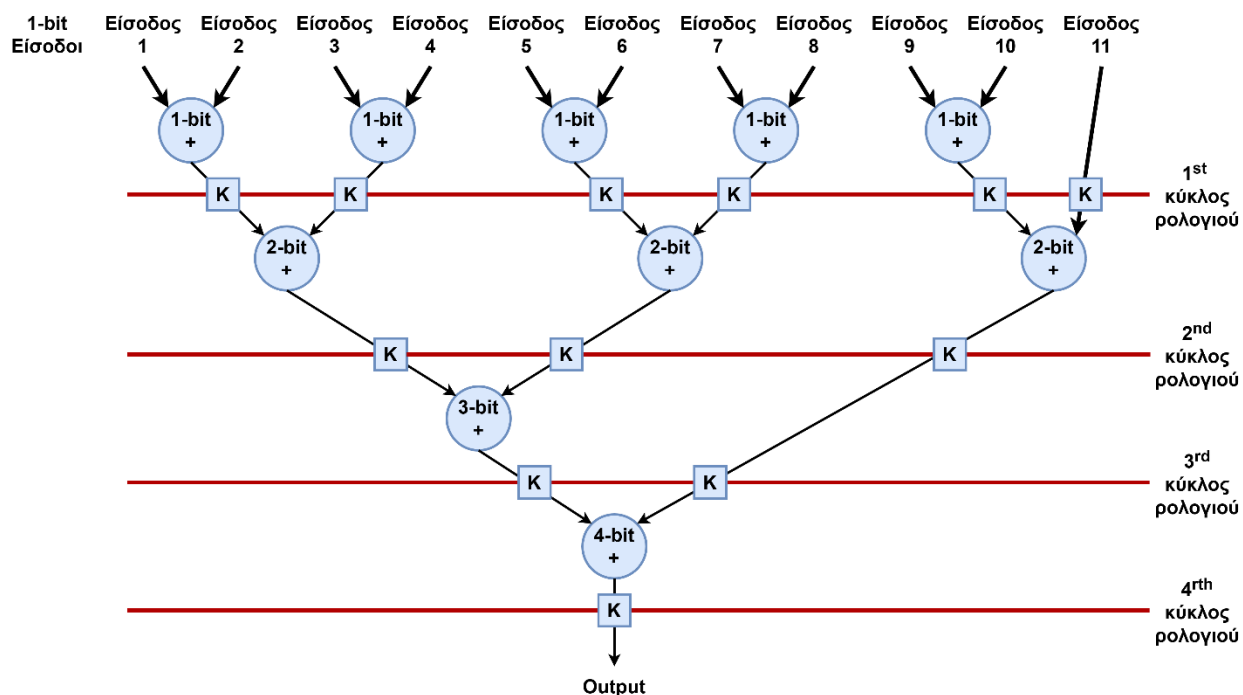
Πίνακας 2-2: Η ομοιότητα των ακολουθιών.

Ακολουθία εισόδου	Κανόνας 2	XNOR	Άθροισμα XNOR
0	0	1	9(1001)
0	0	1	
1	0	0	
1	1	1	
0	0	1	
1	1	1	
1	1	1	
1	1	1	
0	0	1	
1	1	1	

Η υψηλότερη τιμή για το άθροισμα είναι ο αριθμός των bit των διανυσμάτων, όταν κανόνας και είσοδος είναι ίδιες, και η χαμηλότερη 0, όταν κανόνας και είσοδος είναι συμπληρωματικές. Η απλότητα αυτής της μεθόδου ταιριάζει στον μινιμαλιστικό τόνο του κυκλώματος. Η επίτευξη της υλοποίησής του με τους λιγότερους δυνατούς πόρους και την υψηλότερη δυνατή συχνότητα λειτουργίας, παρέχοντας παράλληλα τη μέγιστη ακρίβεια, είναι ο στόχος. Πριν προχωρήσουμε στην περιγραφή της αρχιτεκτονικής, θα παρουσιάσουμε δύο υποκυκλώματα που θα χρησιμοποιηθούν εκτενώς.

2.2.1 Δέντρο αθροιστής

Ένα δέντρο αθροιστής είναι ένα πολύ γνωστό κύκλωμα που υπολογίζει το άθροισμα ενός συνόλου αριθμών. Η γενική μορφή φαίνεται στο Σχήμα 2-1 για ένα μοντέλο με 11 bit (κάθε είσοδος είναι ένα bit).



Σχήμα 2-1: Δέντρο αθροιστής για 11 bit.

Οι κύκλοι είναι αθροιστές κατάλληλου μεγέθους για κάθε επίπεδο, ενώ τα τετράγωνα αντιπροσωπεύουν καταχωρητές. Η προσθήκη καταχωρητών μεταξύ των επιπέδων των αθροιστών επιτρέπει τη διοχέτευση (pipelining). Σε κάθε επίπεδο, οι αριθμοί προστίθενται σε ζεύγη. Εάν έχει απομείνει αριθμός, προχωρά μόνος του. Αυτό σημαίνει ότι στο επίπεδο i υπάρχουν $\frac{n}{2^i}$ προσθέσεις ($\lceil \frac{n}{2^i} \rceil$ ή $\lfloor \frac{n}{2^i} \rfloor$) με $\lceil \log_2 n \rceil$ επίπεδα συνολικά. Το τελικό άθροισμα έχει επίσης μήκος bit $\lceil \log_2 n \rceil$ οπότε ο μεγαλύτερος αθροιστής που χρειάζεται είναι $\lceil \log_2 n \rceil$ bit. Η κρίσιμη διαδρομή αυτού του αθροιστή καθορίζει τη μέγιστη συχνότητα λειτουργίας ολόκληρου του δέντρου. Αυτός είναι ο κύριος λόγος που το δέντρο αθροιστή όπως είναι, δεν μπορεί να υποστηρίξει πολύ μεγάλες ακολουθίες. Οι αθροιστές 32 bit είναι σημαντικά πιο αργοί από αθροιστές 4 bit όταν αναγκάζονται να εκτελέσουν πρόσθεση σε έναν κύκλο ρολογιού. Ακόμα, οι μεγαλύτεροι αθροιστές θα οδηγήσουν σε τεράστια κατανάλωση υλικού, η οποία τελικά θα αντισταθμίσει την αποτελεσματικότητα των δέντρων. Για να μπορεί το κύκλωμα να χειρίζεται ακολουθίες bit των 1000 bit ενώ λειτουργεί στη συχνότητα αθροιστών 4-bit, χωρίς σημαντική επιβάρυνση υλικού, πρέπει να γίνουν σημαντικές τροποποιήσεις.

2.2.2 Δέντρο συγκριτής

Ένα δέντρο συγκριτής είναι μια παραλλαγή του δέντρου αθροιστή που αντί για αθροιστές αποτελείται από συγκριτές. Ακριβώς όπως υποδηλώνει το όνομα, δεδομένου ενός συνόλου n αριθμών, μπορεί να καθορίσει το μέγιστο ή το ελάχιστο αυτών των αριθμών. Διατηρώντας τη θέση κάθε αριθμού σε παράλληλους καταχωρητές, μπορεί να εξάγει όχι μόνο τη μέγιστη τιμή αλλά και τη θέση του μέσα στο σύνολο. Όπως το δέντρο αθροιστών, έχει $\text{ceil}(\log_2 n)$ επίπεδα και χρειάζεται τον ίδιο αριθμό κύκλων για να δώσει αποτέλεσμα (δεδομένου ότι έχει το ίδιο pipelining). Στο πλαίσιο του κυκλώματος, αυτό θα χρησιμοποιηθεί για να καθοριστεί ποιος κανόνας είναι πιο κοντά στην ακολουθία εισόδου.

2.3 Περιγραφή αρχιτεκτονικής

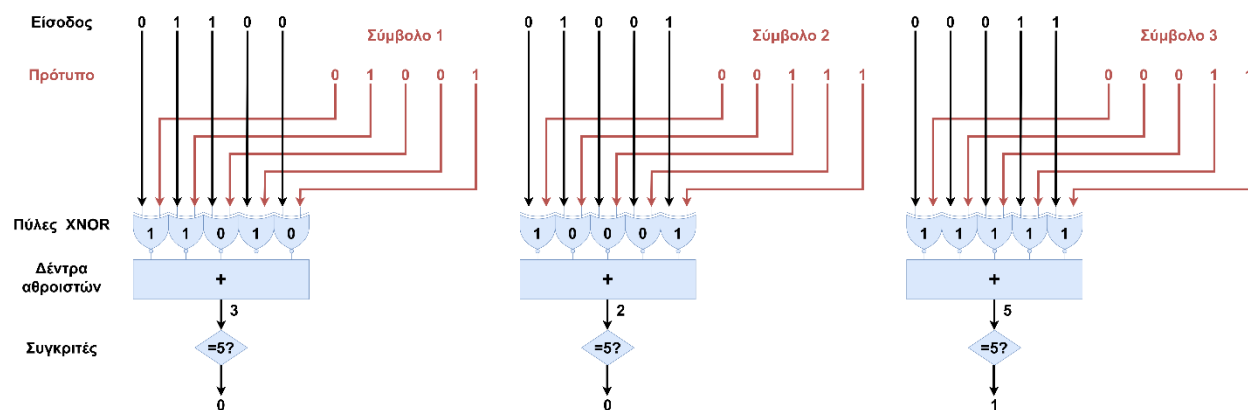
Η αρχιτεκτονική που εκτελεί την αντιστοίχιση μεταξύ της ακολουθίας εισόδου και μιας ακολουθίας στην μνήμη θα παρουσιαστεί σε δύο φάσεις. Σε πρώτη φάση θα παρουσιάσουμε μία μικρότερη απλουστευμένη αρχιτεκτονική την οποία θα χρησιμοποιήσουμε στην συνέχεια για να κατασκευάσουμε την μεγαλύτερη και πιο πολύπλοκη τελική έκδοσής της. Η έκδοση αυτή μπορεί να διατηρεί την ίδια συχνότητα λειτουργίας ανεξάρτητα από το πόσο μεγάλη είναι η ακολουθία.

2.3.1 Βασική μονάδα συσχέτισης

Υποθέτουμε ότι ολόκληρο το μοτίβο εισόδου έχει μήκος n bit. Συσχετίζουμε (correlation με πύλες XNOR) το μοτίβο εισαγωγής με μία καταχώρηση μνήμης (οι αντίστοιχες θέσεις bit τροφοδοτούνται στις πύλες xnor). Μετά κατανέμουμε αυτά τα bit σε q σύμβολα των m bits το καθένα, άρα έχουμε $q \times m = n$. Τα m bits που προκύπτουν από την συσχέτιση (έξοδοι XNOR πυλών) του κάθε συμβόλου q οδηγούνται στη συνέχεια σε δέντρο αθροιστή. Το άθροισμα των bit συσχετισμού μας δίνει μια εκτίμηση για το πόσο κοντά είναι η ακολουθία εισόδου στην ακολουθία μνήμης. Μετά το δέντρο αθροιστή, όλα τα αθροίσματα διέρχονται από συγκριτές. Αν κάποιο άθροισμα είναι ίσο με τον αριθμό bit ανά σύμβολο (m), τότε η έξοδος του συγκριτή γίνεται 1, υποδεικνύοντας ότι το εντοπίστηκε το σωστό σύμβολο. Στη συνέχεια, ελέγχοντας την έξοδο των συγκριτών, μπορούν να καθοριστούν πόσα σύμβολα εντοπίστηκαν. Εάν εντοπίστηκαν όλα τα σύμβολα q , τότε το μοτίβο εντοπίστηκε. Ένα παράδειγμα φαίνεται στο *Σχήμα 2-2* για $q = 3$ και $m = 5$. Όλες οι τιμές σημειώνονται στα καλώδια του κυκλώματος για ευκολία και σαφήνεια. Ο αριθμός που οι συγκριτές χρησιμοποιούν για να προσδιορίσουν εάν το σύμβολο εντοπίστηκε ή όχι ορίζεται ως το βάρος της βασικής μονάδας συσχέτισης. Στο *Σχήμα 2-2*, το βάρος είναι 5.

Είναι σαφές ότι η βασική μονάδα συσχέτισης δεν μπορεί να χρησιμοποιηθεί για τον προσδιορισμό μεγάλων μοτίβων ($n > 100$). Αν υπάρχουν πάρα πολλά σύμβολα ($q > > 1$) τότε οι έξοδοι των

συγκριτών είναι πάρα πολλές για έλεγχο σε έναν κύκλο ρολογιού. Αν είναι και τα σύμβολα μεγάλα, τα δέντρα αθροιστών απαιτούν αθροιστές που περιλαμβάνουν πολλά bit, γεγονός που αυξάνει περισσότερο την συχνότητα λειτουργίας. Για αυτούς τους λόγους, η πραγματική ανίχνευση ακολουθιών θα εκτελεστεί χρησιμοποιώντας μια πιο εμπλουτισμένη μορφή της βασικής μονάδας συσχέτισης.



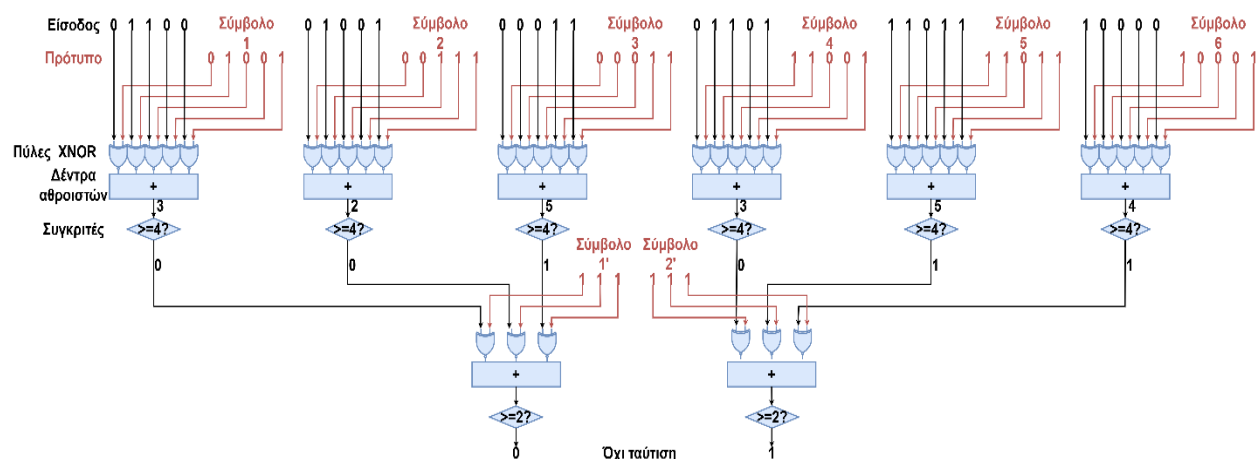
Σχήμα 2-2: Μικρότερος συγκριτής μοτίβου.

2.3.2 Ολικός ανιχνευτής αλληλουχιών

Ο ολικός ανιχνευτής αλληλουχιών είναι ένα κύκλωμα με πολλαπλά στάδια. Κάθε στάδιο περιλαμβάνει μία βασική μονάδα συσχέτισης (βλ. Ενότητα 2.3.1) όπου οι έξοδοι του συγκριτή τροφοδοτούνται στην μονάδα συσχέτισης του επόμενου σταδίου. Το Σχήμα 2-3 παρουσιάζει το κύκλωμα ενός ολικού ανιχνευτή αλληλουχιών. Ο αριθμός των δυαδικών ψηφίων εισόδου διατηρείται χαμηλός στο παράδειγμα, προκειμένου να απεικονιστεί καθαρά ολόκληρο το κύκλωμα. Αυτή η έκδοση έχει δύο στάδια. Το πρώτο στάδιο (στάδιο 1) περιλαμβάνει 30 bit, οργανωμένα σε 5 bit ανά σύμβολο για 6 σύμβολα. Στο δεύτερο στάδιο, οι εισοδοί πέφτουν από 30 σε 6 που είναι οργανωμένα σε 2 σύμβολα των 3 bit το καθένα. Τα τελευταία 2 bit δείχνουν εάν ολόκληρο το μοτίβο των 30 bit εντοπίστηκε. Μπορούμε να δούμε ότι τα βάρη των συγκριτών είναι τώρα υψηλότερα ή ίσα με 4 για το πρώτο στάδιο και υψηλότερα ή ίσα με 2 για το δεύτερο στάδιο. Αυτό σημαίνει ότι υπάρχει περιθώριο λάθους. Επιπλέον, το μοτίβο που πρέπει να εντοπιστεί εμφανίζεται μόνο στα σύμβολα του πρώτου σταδίου. Για το στάδιο 2, όλα τα σύμβολα αντικαθίστανται με συνδυασμό 111 δεδομένου ότι τα επόμενα στάδια χρησιμοποιούνται για να προσδιοριστεί πόσο καλά το πρώτο στάδιο ανίχνευσε την ακολουθία. Όπως μπορούμε να δούμε, μόνο ένα μέρος της ακολουθίας εντοπίστηκε με βάση τα βάρη που χρησιμοποιήθηκαν από τους συγκριτές και δεν υπάρχει αντιστοιχία. Η καθυστέρηση του κυκλώματος υπολογίζεται ως $\text{ceil}(\log_2 5) + 1 + \text{ceil}(\log_2 3) + 1 = 7$ κύκλοι ρολογιού. Ο όρος $1 + 1$ αντιπροσωπεύει τους κύκλους του ρολογιού που χρειάζονται οι συγκριτές. Για εξοικονόμηση υλικού, οι συγκριτές μπορούν να υλοποιηθούν ως συνδυαστικά

κυκλώματα ώστε να μην καταναλώνουν κύκλους ρολογιού. Οι όροι $\log_2$ είναι προϊόν των αθροιστών. Γενικά, για ένα κύκλωμα t σταδίων, οι κύκλοι ρολογιού που χρειάζεται το κύκλωμα είναι $\text{κύκλοι} = t + \sum_{i=1}^t [\text{ceil}(\log_2 b_i)]$ κύκλοι όπου το b_i αντιπροσωπεύει τα bit ανά σύμβολο για κάθε στάδιο.

Κατασκευάζοντας τον ανιχνευτή με αυτόν τον τρόπο, πετύχαμε δύο σημαντικά πράγματα. Πρώτον, για μεγάλες ακολουθίες, οι αθροιστές δεν χρειάζεται να κλιμακώνονται επ' αόριστον καθώς μπορούμε να χρησιμοποιήσουμε περισσότερα στάδια για να διατηρούνται χαμηλά τα bit ανά σύμβολο. Αυτό σημαίνει ότι το μέγεθος της ακολουθίας, σε bit, δεν έχει καμία επίδραση στο critical path (και ως αποτέλεσμα της συχνότητας λειτουργίας και της απόδοσης). Δεύτερον, χειραγωγώντας τα βάρη στους συγκριτές, μπορούμε να προσαρμόσουμε την ικανότητα ανίχνευσης του κυκλώματος. Ως παράδειγμα, προσφέρουμε την υλοποιημένη έκδοση του κυκλώματος με 1200 bit εισόδου και 3 στάδια [8]. Η μέθοδος αυτή μπορεί να κλιμακωθεί σε οποιοδήποτε μήκος ακολουθίας. Διαφορετικοί συνδυασμοί αριθμού επιπέδων, μεγέθους συμβόλων και βαρών μπορούν να χρησιμοποιηθούν για να μεγιστοποιηθεί η ικανότητα ανίχνευσης.



Σχήμα 2-3: Συγκριτής μοτίβου 2 επιπέδων.

Στο πρώτο στάδιο (πρώτη βασική μονάδα συσχέτισης) τα 1200 bit αναλύονται σε 120 σύμβολα από 10 bit. Ακολουθώντας τη λογική του Σχήματος 2-3, χρησιμοποιούμε 120 δέντρα αθροιστών μεγέθους 10 bit. Τα αποτελέσματα αυτών των δέντρων τροφοδοτούνται σε 120 συγκριτές, οι οποίοι έχουν ένα προκαθορισμένο βάρος. Τα αποτελέσματα των συγκριτών τροφοδοτούνται τώρα στην δεύτερη βασική μονάδα συσχέτισης. Αυτή έχει είσοδο 120 bit, που αναλύονται σε 12 σύμβολα των 10 bit. Σε αυτά εφαρμόζεται η ίδια διαδικασία και στη συνέχεια οι έξοδοι τους (12 bit) τροφοδοτούνται στο τελευταία μονάδα συσχέτισης, όπου αναλύονται σε 1 σύμβολο των 12 bit. Η έξοδος της τελευταίας μονάδας είναι 1 bit, αποτέλεσμα του μοναδικού δέντρου αθροιστών της μονάδας. Αν το bit είναι 1, η ακολουθία εντοπίστηκε. Για 1200 bit το κύκλωμα χρειάζεται $3 + 2 \times$

$\text{ceil}(\log_2 10) + \text{ceil}(\log_2 12) = 15$ κύκλους ρολογιού. Ο μεγαλύτερος αθροιστής που χρησιμοποιείται είναι 4 bit.

Ενδεικτικά, μία μονάδα που μπορεί να ανιχνεύσει μία ακολουθία μήκους 1200 bit, με ακρίβεια λάθους 10^{-4} , υπό αλλοιώσεις της τάξης των 5×10^{-2} , χρειάζεται μόνο 3051 LUTs και 5531 FFs (flipflops) και μπορεί να λειτουργεί με συχνότητα 667 MHz (για υλοποίηση στο FPGA ZCU106 Ultrascale+). Η μινιμαλιστική φύση της αρχιτεκτονικής επιτρέπει την κλιμάκωση της σε μεγάλα μεγέθη χωρίς την αντίστοιχη κατανάλωση του υλικού. Το βασικότερο συμπέρασμα από την υλοποίηση της αρχιτεκτονικής του κεφαλαίου αυτού είναι πως μπορεί να πραγματοποιηθεί ανίχνευση πολύ μεγάλου μήκους ακολουθιών bit με σημαντική αλλοίωση «αναίμακτα».

2.4 Βιβλιογραφία κεφαλαίου

- [1] F. Pong and N. -F. Tzeng, "HaRP: Rapid Packet Classification via Hashing Round-Down Prefixes," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1105-1119, July 2011, doi: 10.1109/TPDS.2010.195.
- [2] T. N. Thinh, T. T. Hieu, Van Quoc Dung and S. Kittitornkun, "A FPGA-based deep packet inspection engine for Network Intrusion Detection System," *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Phetchaburi, Thailand, 2012, pp. 1-4, doi: 10.1109/ECTIcon.2012.6254301.
- [3] H. Le and V. K. Prasanna, "A Memory-Efficient and Modular Approach for Large-Scale String Pattern Matching," in *IEEE Transactions on Computers*, vol. 62, no. 5, pp. 844-857, May 2013, doi: 10.1109/TC.2012.38.
- [4] P. Dlugosch, D. Brown, P. Glendenning, M. Leventhal and H. Noyes, "An Efficient and Scalable Semiconductor Architecture for Parallel Automata Processing," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3088-3098, Dec. 2014, doi: 10.1109/TPDS.2014.8.
- [5] A. V. Alyushin, S. A. Alyushin and V. G. Arkhangelsky, "Scalable processor core for high-speed pattern matching architecture on FPGA," *2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, Moscow, Russia, 2016, pp. 148-153, doi: 10.1109/DIPDMWC.2016.7529380.
- [6] Y. R. Qu, S. Zhou and V. K. Prasanna, "High-performance architecture for dynamically updatable packet classification on FPGA," *Architectures for Networking and Communications Systems*, San Jose, CA, USA, 2013, pp. 125-136, doi: 10.1109/ANCS.2013.6665195.
- [7] M. V. Thomas, C. Kanagasabapathi and S. S. Yellampalli, "VHDL implementation of pattern based template matching in satellite images," *2017 International Conference On Smart Technologies*

For Smart Nation (SmartTechCon), Bengaluru, India, 2017, pp. 820-824, doi: 10.1109/SmartTechCon.2017.8358487.

[8] Nikolaidis D, Groumas P, Kouloumentas C, Avramopoulos H. High-Throughput Bit-Pattern Matching under Heavy Interference on FPGA. *Electronics*. 2023; 12(4):803. <https://doi.org/10.3390/electronics12040803>

Κεφάλαιο 3: Συγχρονισμός πλαισίων δεδομένων με ανίχνευση ακολουθιών

3.1 Εισαγωγή

Το Κεφάλαιο 3 παρουσιάζει την άμεση, πρακτική εφαρμογή των μεθόδων που αναπτύχθηκαν στο Κεφάλαιο 2. Το κεφάλαιο αυτό ασχολείται με το συγχρονισμό πλαισίων δεδομένων κατά τη διάρκεια μιας ψηφιακής επικοινωνίας. Με τον όρο συγχρονισμό πλαισίων εννοούμε την αξιόπιστη ανίχνευση υπό θόρυβο, πλαισίων δεδομένων. Ο συγχρονισμός αυτός επιτυγχάνεται με την τοποθέτηση μεγάλου μήκους ακολουθίας bit μπροστά από κάθε πλαίσιο και την ανίχνευσή της (ψηφιακή συσχέτιση) από κύκλωμα παρόμοιας αρχιτεκτονικής με αυτήν που αναπτύχθηκε στο Κεφάλαιο 2. Αυτό έχει ως αποτέλεσμα να μπορούν να χρησιμοποιηθούν μεγάλου μήκους ακολουθίες bit ώστε η ανίχνευση να επιτυγχάνεται με πάρα πολύ μεγάλη ακρίβεια και με πολύ μικρή καθυστέρηση.

Ο συγχρονισμός πλαισίων μέσω της ανίχνευσης ακολουθιών (ψηφιακή συσχέτιση: digital correlation) δεν είναι κάτι καινούργιο. Επινοήθηκε από τον Barker στο [1], και είναι η πιο βασική μορφή συγχρονισμού πλαισίων. Χρησιμοποιείται σε εφαρμογές όπως αισθητήρες [2] ακόμα και σήμερα. Ο Barker χρησιμοποίησε συγκεκριμένες δυαδικές ακολουθίες/λέξεις που ονομάζονται ακολουθίες Barker οι οποίες διαθέτουν την ιδανική ιδιότητα αυτοσυσχέτισης. Αυτό σημαίνει ότι οι πλευρικοί λοβοί αυτοσυσχέτισης είναι όσο το δυνατόν μικρότεροι. Για πολύ καιρό ήταν ευρέως αποδεκτό ότι αυτή ήταν η βέλτιστη μέθοδος συγχρονισμού πλαισίων, μέχρι που ο Massey εισήγαγε τον κανόνα μέγιστης πιθανότητας προσθέτοντας έναν μη γραμμικό διορθωτικό όρο στον συσχετιστή Barker για να λαμβάνονται υπόψη τα δεδομένα που περιβάλλουν τις λέξεις συγχρονισμού [3]. Ο Massey ανέπτυξε επίσης μία εύκολα υλοποιήσιμη λύση για άμεση εφαρμογή με παρόμοια απόδοση όπως απέδειξε ο Nielsen [4]. Το βασικό πρόβλημα με την μέθοδο του Barker είναι ότι το μέγεθος των ακολουθιών είναι πολύ μικρό.

Από τότε, έχουν προταθεί διάφορες παραλλαγές της ψηφιακής συσχέτισης όπως η παρατήρηση πολλών πλαισίων (όχι μόνο ενός) για να εξαχθούν συμπεράσματα για την θέση τους [5] αλλά και τροποποιημένοι αλγόριθμοι μέγιστης πιθανότητας [6]. Ακόμα, παρόλο που η ψηφιακή συσχέτιση

είναι σχετικά παλιά μέθοδος εξακολουθεί να ανταγωνίζεται σε όρους απόδοσης πιο εξελιγμένες σύγχρονες μεθόδους συγχρονισμού πλαισίων όπως ο κοινός συγχρονισμός πλαισίου και διόρθωσης λαθών (joint frame synchronization and error correction) LDPC [7]. Η εξεταζόμενη αρχιτεκτονική επιτυγχάνει τον συγχρονισμό με την προσθήκη ακολουθιών μπροστά από τα πλαίσια δεδομένων και την ανίχνευση τους από δέντρα αθροιστών και συγκριτών με ψηφιακή συσχέτιση. Καθώς μέχρι τώρα δεν υπήρχε τρόπος να αυξηθεί πρακτικά σε τέτοιο μέγεθος το μήκος των ακολουθιών που χρησιμοποιούνταν με συσχέτιση, η ακρίβεια είναι πολύ μεγαλύτερη από τις άλλες μεθόδους. Βασίζεται στην μεθοδολογία του Κεφαλαίου 2 με την έννοια ότι χρησιμοποιεί συγκριτές για να ανιχνεύσει την ύπαρξη του πλαισίου υπό θόρυβο, όμως το μέγεθος των ακολουθιών που απαιτούνται ακόμα και για μεγάλη ακρίβεια συγχρονισμού δεν είναι τόσο μεγάλο που να χρειάζεται ο διαχωρισμός σε Q τμήματα. Αυτό σημαίνει πως τα δέντρα αθροιστών δεν χρειάζεται να τροποποιηθούν (όπως στο Κεφάλαιο 2).

Η προτεινόμενη αρχιτεκτονική υλοποιήθηκε σε περιβάλλον Vivado με γλώσσα VHDL πάνω στο χαμηλού κόστους FPGA NEXYS 4 DDR. Πέρα από το συγχρονισμό (ανίχνευση πλαισίων) αποδίδει και το ωφέλιμο φορτίο κάθε πλαισίου σαν επιπλέον δυνατότητα. Ο έλεγχος της ορθής λειτουργίας έγινε με emulation (συνθετικά δεδομένα και υλοποιημένο κύκλωμα). Τα αποτελέσματα έδειξαν ότι η ακρίβεια συγχρονισμού είναι κατά πολύ μεγαλύτερη από αντίστοιχες εργασίες στη βιβλιογραφία. Ενδεικτικά, για τις παραλλαγές της μεθόδου βέλτιστης πιθανότητας [5] και [6] η ακρίβεια συγχρονισμού της προτεινόμενης αρχιτεκτονικής (πιθανότητα λάθους συγχρονισμού $\sim 4 \times 10^{-4}$) στον AWGN για $\text{SNR} > 0\text{dB}$ είναι $10 \sim 10^2$ φορές καλύτερη από όλα τα αποτελέσματα που αναφέρονται. Το ίδιο ισχύει και για τις μεθόδους που χρησιμοποιούν κώδικες διόρθωσης για συγχρονισμό. Στις δημοσιεύσεις [7] και [8] παρουσιάζονται αποτελέσματα συγχρονισμού με την βοήθεια LDPC για πάρα πολλές διαφορετικές παραμέτρους. Η αρχιτεκτονική (πιθανότητα λάθους συγχρονισμού $\sim 4 \times 10^{-5}$) είναι και εδώ $10 \sim 10^2$ φορές καλύτερη από τα αποτελέσματα της βιβλιογραφίας στον AWGN δίαυλο για $\text{SNR} > 1\text{dB}$. Και για τις δύο κατηγορίες μεθόδων η ακρίβεια για μικρότερες τιμές του SNR είναι πιθανότατα χειρότερη από $10 \sim 10^2$ όμως δεν μπορεί να εκτιμηθεί με βεβαιότητα από τα διαγράμματα που δίνονται. Σε ότι αφορά την υλοποίηση, για την πρώτη κατηγορία ([5],[6]) δεν αναφέρονται υλοποιήσεις. Για την δεύτερη ([7],[8]) χρησιμοποιούνται αποκωδικοποιητές LDPC [9] που εξ' ορισμού είναι λιγότερο αποδοτικοί από την προτεινόμενη αρχιτεκτονική. Συγκεκριμένα προσφέρουν μετά βίας το 1/5 της ροής δεδομένων ενώ καταναλώνουν περισσότερους πόρους παρότι υλοποιούνται σε πιο προηγμένες πλατφόρμες υλικού (ASIC αντί για FPGA) [10]. Αυτό έχει ως αποτέλεσμα η προτεινόμενη αρχιτεκτονική να είναι ένα χρήσιμο εργαλείο για εφαρμογές που δίνουν προτεραιότητα στην εξοικονόμηση πόρων όπως το IoT (Internet Of Things). Επίσης, επειδή ο συγχρονισμός μπορεί να επιτευχθεί με πολλών ειδών ακολουθίες, η αρχιτεκτονική μπορεί να

χρησιμοποιηθεί και ως μηχανισμός ασφάλειας που καθιστά την μετάδοση των πλαισίων μη ανιχνεύσιμη από εξωτερικούς παρατηρητές.

Τα αποτελέσματα της ακρίβειας συγχρονισμού είναι αναμενόμενα όταν πρόκειται για τις μεθόδους ψηφιακής συσχέτισης. Είναι γενικά αποδεκτό και στην θεωρία ότι η συσχέτιση είναι αρκετά ακριβής όταν γίνεται με ακολουθίες μεγάλου μήκους [11]. Ο λόγος για τον οποίο αυτή η μέθοδος δεν προτιμάται μέχρι τώρα είναι γιατί δεν υπήρχε τρόπος να κλιμακωθεί πρακτικά το μέγεθος των ακολουθιών που χρησιμοποιούνται. Αυτό ανάγκαζε τους χρήστες να χρησιμοποιούν ακολουθίες μικρού μήκους και να βασίζονται σε πρωτόγονες μεθόδους υλοποίησης περιορίζοντας την ακρίβεια και την ταχύτητα των εφαρμογών. Ακόμα, καθώς η μέθοδος ανίχνευσης της ακολουθίας του Κεφαλαίου 3 είναι βέλτιστη ως προς την πολυπλοκότητα ($\log_2 n$), είναι εξ' ορισμού πιο αποδοτική (σε ότι αφορά τον συγχρονισμό) από οποιαδήποτε υλοποίηση αποκωδικοποιητή LDPC (βέλτιστη πολυπλοκότητα $n \times \log n$).

Η μεγάλη ακρίβεια που προσφέρει η αρχιτεκτονική μας επιτρέπει να προσεγγίσουμε τη μετάδοση με διαφορετικό τρόπο απ' ότι παλιότερα. Επειδή οι μέθοδοι ανίχνευσης των πλαισίων μέχρι τώρα δεν ήταν πολύ αξιόπιστες υπήρχε η ανάγκη να γίνεται η καλύτερη δυνατή επεξεργασία στα πλαίσια δεδομένων που λαμβάνονται επιτυχώς, ώστε να εξάγεται όσο το δυνατόν περισσότερη ωφέλιμη πληροφορία. Αυτό είχε ως αποτέλεσμα οι μοντέρνοι κώδικες διόρθωσης (LDPC, Polar) να λειτουργούν κοντά στο όριο του Shannon (το θεωρητικό όριο για την καλύτερη δυνατή μετάδοση πληροφορίας). Η ικανότητα αυτή όμως πληρώνεται με μεγάλους χρόνους κωδικοποίησης και αποκωδικοποίησης. Με την αύξηση της ακρίβειας συγχρονισμού που πετυχαίνουμε, έχουμε την επιλογή να μειώσουμε τις απαιτήσεις και να στέλνουμε δεδομένα πολλές φορές, όταν απαιτείται, χρησιμοποιώντας πιο απλές μεθόδους με χειρότερη αλλά ταχύτερη διόρθωση λαθών.

3.2 Βασικές αρχές

Η βασική μονάδα μεταφοράς δεδομένων είναι το πλαίσιο (data frame). Συγχρονισμός πλαισίων ονομάζεται η αξιόπιστη ανίχνευση της θέσης κάθε πλαισίου με τέτοιο τρόπο έτσι ώστε το φορτίο δεδομένων του (data payload) να παραδίδεται με επιτυχία στο σύστημα λήψης. Η αρχιτεκτονική επιτυγχάνει συγχρονισμό πλαισίων ανιχνεύοντας λέξεις με ψηφιακή συσχέτιση. Διαφορετικές μορφές και μήκη λέξεων παράγουν διαφορετικά αποτελέσματα. Εκτός από την λέξη συγχρονισμού είναι θεμιτό να διαμορφώνονται και τα ψηφία γύρω της έτσι ώστε να ενισχύουν την πιθανότητα σωστού εντοπισμού της θέσης της. Η ακολουθία που απαρτίζεται από την λέξη συγχρονισμού και τα γειτονικά της ψηφία καλείται σηματοδότης του πλαισίου (frame marker).

Οι σηματοδότες είναι της μορφής $a_k b_l c_k$ (μετάδοση από δεξιά προς τα αριστερά, c_k είναι πρώτη και a_k τελευταία), όπου a_k και c_k είναι ακολουθίες μήκους k bit, και b_l είναι μια ακολουθία μήκους l bits

με $l \gg 2 \times k$; Το b_l είναι τυχαίο (δημιουργείται τυχαία, που σημαίνει ότι κάθε bit έχει ίσες πιθανότητες να είναι 0 ή 1), αλλά το a_k και το c_k πρέπει να είναι το τελευταία και τα πρώτα k bit (αντίστοιχα) του b_l αντίστροφα ($a_k = \text{NOT}$ (τελευταία k bits του b_l), $c_k = \text{NOT}$ (πρώτα k bits του b_l)); Η ακολουθία b_l είναι η λέξη συγχρονισμού που χρησιμοποιεί η αρχιτεκτονική για να ανιχνεύσει το πλαίσιο, και οι ακολουθίες a_k και c_k είναι τα παραποιημένα γειτονικά ψηφία. Το a_k και το c_k δεν χρησιμοποιούνται από την αρχιτεκτονική για ανίχνευση. Ο ρόλος τους είναι να κρατούν τις τιμές συσχέτισης σε όλες τις άλλες θέσεις εκτός από το πρώτο bit του b_l χαμηλά, ενισχύοντας την ακρίβεια ανίχνευσης της αρχιτεκτονικής. Αυτό το φαινόμενο παρατηρείται ευκολότερα και είναι πολύ ισχυρότερο όταν το b_l είναι μεγάλο (>70 bit). Έτσι το a_k και το c_k επηρεάζουν την έκβαση της ανίχνευσης έμμεσα (χωρίς να ανιχνεύονται οι ίδιες).

Μόλις δημιουργηθεί η ακολουθία b_l , θεωρείται προκαθορισμένη (δηλαδή, γνωστή σε πομπό και δέκτη). Για παράδειγμα, ας υποθέσουμε ότι $l = 32$ bit. Μια τυχαία ακολουθία 32-bit όπως δημιουργείται από μια γεννήτρια τυχαίας δυαδικής ακολουθίας είναι το 11001010101110101110001100000111. Μόλις δημιουργηθεί η ακολουθία, παραμένει τυχαία με μαθηματική έννοια, ωστόσο είναι πλέον προκαθορισμένη και γνωστή. Προκαθορισμένη στο πλαίσιο του κεφαλαίου σημαίνει ότι η b_l δημιουργείται πριν η επικοινωνία γίνει και χρησιμοποιείται κατά τη διάρκειά της.

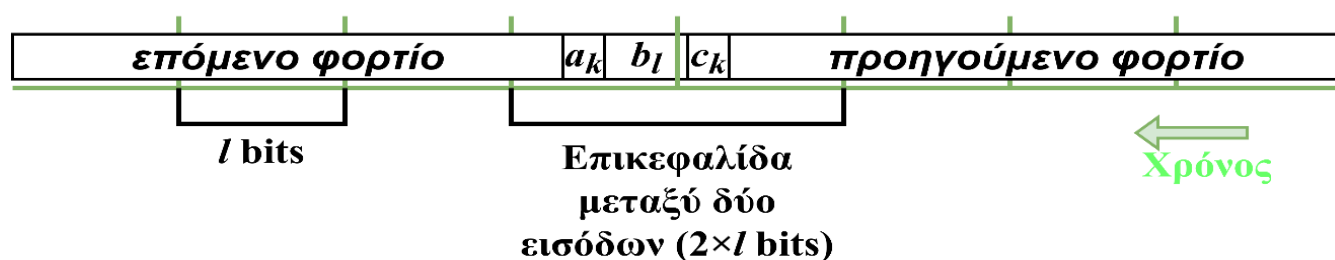
Αυτή η μορφή σηματοδότη (λέξη συγχρονισμού και γειτονικά ψηφία) δεν είναι βέλτιστη. Χρησιμοποιήθηκε επειδή είναι απλό να κατασκευαστεί και έδειξε καλά αποτελέσματα. Άλλες ακολουθίες που έχουν πολύ καλύτερες ιδιότητες συσχέτισης, όπως αυτές που κατασκευάζονται στο [11], μπορούν επίσης να χρησιμοποιηθούν. Η μορφή ολόκληρου του πλαισίου είναι $payload_p a_k b_l c_k$ και μεταδίδεται από δεξιά προς τα αριστερά. Το $payload_p$ είναι το φορτίο δεδομένων του πλαισίου (μήκος p bits) και το $a_k b_l c_k$ ο σηματοδότης. Η ανίχνευση της ακολουθίας b_l βασίζεται στις ακόλουθες τρεις προτάσεις:

1. Για να υπολογιστεί ο αριθμός των όμοιων στοιχείων σε αντίστοιχες θέσεις μεταξύ δύο διανυσμάτων v_1 και v_2 ίδιου μεγέθους s , εφαρμόζουμε την πράξη XNOR σε στοιχεία στις αντίστοιχες θέσεις και προσθέτουμε τα αποτελέσματα (ψηφιακή συσχέτιση). Το άθροισμα αντιπροσωπεύει πόσα παρόμοια ψηφία έχουν. Όταν η τιμή του αθροίσματος είναι υψηλή, τα δύο διανύσματα είναι παρόμοια. Όταν η τιμή του αθροίσματος είναι χαμηλή, τα δύο διανύσματα δεν είναι παρόμοια. Ουσιαστικά είναι το αντίθετο της απόστασης Hamming. Το άθροισμα αναφέρεται ως άθροισμα ομοιότητας. Η πρόταση αυτή είναι γνωστή από το Κεφάλαιο 3.
2. Εάν έχουμε ένα πολύ μεγαλύτερο διάνυσμα v_3 μεγέθους z όπου $z \gg s$, μπορούμε να χρησιμοποιήσουμε τη διαδικασία στην πρόταση 1 για να προσδιορίσουμε εάν το v_1 υπάρχει μέσα στο v_3 . Αυτό επιτυγχάνεται με την εφαρμογή της 1 σε όλες τις πιθανές θέσεις στις οποίες το v_1 μπορεί να

είναι μέσα στο v_3 . Υπάρχουν $z - s + 1$ πιθανές θέσεις για το v_1 μέσα στο v_3 , οι οποίες είναι $s-1$ έως 0 , s έως 1 , $s + 1$ έως $2 \dots z - 2$ έως $z - s - 1$, και $z - 1$ έως $z - s$. Αυτό σημαίνει ότι η πρόταση 1 εφαρμόζεται $z - s + 1$ φορές, και για κάθε επανάληψη, το v_1 είναι το v_1 και το v_2 αντιστοιχεί στο διάνυσμα που κατασκευάζεται από τα στοιχεία στο εσωτερικό του v_3 των αντίστοιχων θέσεων (για παράδειγμα, στοιχεία στις θέσεις $s-1$ έως 0 για την πρώτη επανάληψη). Η διαδικασία παράγει $z - s + 1$ αθροίσματα ομοιότητας sum_i . Εάν οποιαδήποτε από αυτά ισούται με s ή είναι κοντά στο s , τότε αυτό σημαίνει ότι το v_1 ή ένα διάνυσμα που μοιάζει με αυτό εντοπίστηκε μέσα στο v_3 . Αυτή η διαδικασία λειτουργεί ανεξάρτητα από το πόσο μεγάλο είναι το μέγεθος v_3 (δηλαδή το z).

3. Για την αρχιτεκτονική, το v_3 είναι η εισερχόμενη ψηφιακή ροή (input bitstream) και το v_1 είναι η ακολουθία b_l . Το b_l εντοπίζεται μέσα στο v_3 χρησιμοποιώντας την πρόταση 2 και η αρχιτεκτονική απομονώνει το ωφέλιμο φορτίο του πλαισίου που συνοδεύει.

Αυτές οι τρεις αρχές αποτελούν τη βάση για τη λειτουργία της αρχιτεκτονικής και, εφεξής, θα αναφέρονται με παρένθεση ((1) που αναφέρεται στην πρόταση 1). Η είσοδος της αρχιτεκτονικής είναι η ψηφιακή ροή, η οποία προκύπτει από την αποδιαμόρφωση της εισερχόμενης μετάδοσης. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι η είσοδος δίνεται ως παράλληλο διάνυσμα bit μεγέθους l (ίδιο μέγεθος με την ακολουθία b_l). Με άλλα λόγια, αντί να λαμβάνει ένα bit στη συχνότητα f , η αρχιτεκτονική λαμβάνει l bit στη συχνότητα $f_{op} = \frac{f}{l}$ και ο συνολικός ρυθμός μετάδοσης bit είναι $f_{op} \times l$. Υποθέτουμε επίσης ότι αν τα παράλληλα διανυσματικά στοιχεία bit εισόδου είναι $l-1$ έως 0 , το bit στη θέση 0 μεταδόθηκε νωρίτερα στο χρονοδιάγραμμα εισερχόμενης ροής bit από το bit στη θέση $l-1$. Το Σχήμα 3-1 είναι η οπτική αναπαράσταση του τρόπου με τον οποίο η αρχιτεκτονική «βλέπει» την εισερχόμενη ροή δεδομένων. Ο χρόνος ρέει προς τα αριστερά, επομένως τα δείγματα προς τα δεξιά μεταδόθηκαν νωρίτερα.



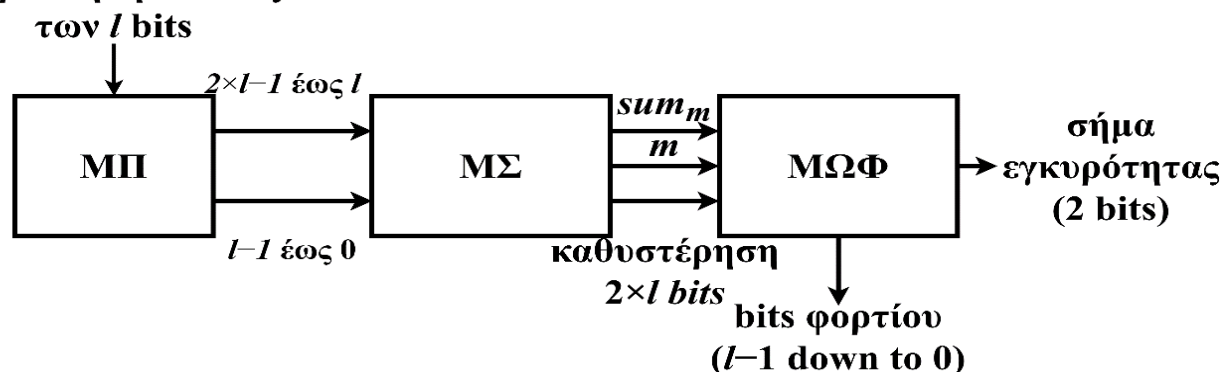
Σχήμα 3-1: Λήψη ροής δεδομένων σε πραγματικό χρόνο. Η ροή λαμβάνεται σε κομμάτια των l bits. Η επικεφαλίδα $a_k b_l c_k$ βρίσκεται μεταξύ δύο πλαισίων.

Η ροή δυαδικών ψηφίων διαχωρίζεται σε μπλοκ l bit ($l - 1$ έως 0), που είναι το μέγεθος της εισόδου. Μπορούμε να δούμε ξεκάθαρα ότι η b_l βρίσκεται πάντα μεταξύ δύο εισόδων ($2 \times l > l$). Η αρχιτεκτονική απομονώνει αυτά τα $2 \times l$ bits κάθε φορά, τα συσχετίζει με το προκαθορισμένο b_l , βρίσκει την αρχή του ωφέλιμου φορτίου και το αιχμαλωτίζει. Λαμβάνεται επίσης υπόψη η σπάνια περίπτωση που η κεφαλίδα b_l είναι τέλεια ευθυγραμμισμένη με ένα δείγμα εισόδου.

3.3 Περιγραφή Αρχιτεκτονική

Η αρχιτεκτονική αποτελείται από τρεις μονάδες: Μονάδα Παραθύρου (ΜΠ), Μονάδα Συσχέτισης (ΜΣ) και Μονάδα Ωφέλιμου Φορτίου (ΜΩΦ). Η ΜΠ αποτελείται από έναν καταχωρητή $2 \times l$ bit που χρησιμοποιείται για να απομονώσει την περιοχή της εισερχόμενης ροής στην οποία βρίσκεται η ακολουθία b_l . Στέλνει τα περιεχόμενα του καταχωρητή στη ΜΣ έτσι ώστε να μπορεί να υπολογιστεί η θέση του ωφέλιμου φορτίου. Η ΜΣ συσχετίζει την προκαθορισμένη αποθηκευμένη ακολουθία b_l με την περιοχή που απομονώνεται από τη ΜΠ (σύμφωνα με (2)) και υπολογίζει δύο τιμές, sum_m και m . sum_m είναι το μέγιστο άθροισμα ομοιότητας στο τρέχον παράθυρο ($2 \times l$ bit) και m είναι η θέση στην οποία εντοπίστηκε. Εκτός από αυτές τις δύο τιμές, η ΜΣ παρέχει επίσης τη συγχρονισμένη είσοδο, δηλαδή τα περιεχόμενα του καταχωρητή της ΜΠ για τα οποία υπολογίζονται οι τιμές, στη ΜΩΦ για σωστό συγχρονισμό. Η ΜΩΦ χρησιμοποιεί sum_m και m για τον προσδιορισμό της θέσης του ωφέλιμου φορτίου και τη συγχρονισμένη είσοδο για την αιχμαλώτισή του. Ένα σήμα εξόδου εγκυρότητας ($valid_out$) χρησιμοποιείται για να ειδοποιήσει το υπόλοιπο σύστημα για την επιτυχή σύλληψη. Το σύνολο της αρχιτεκτονικής φαίνεται στο Σχήμα 3-2.

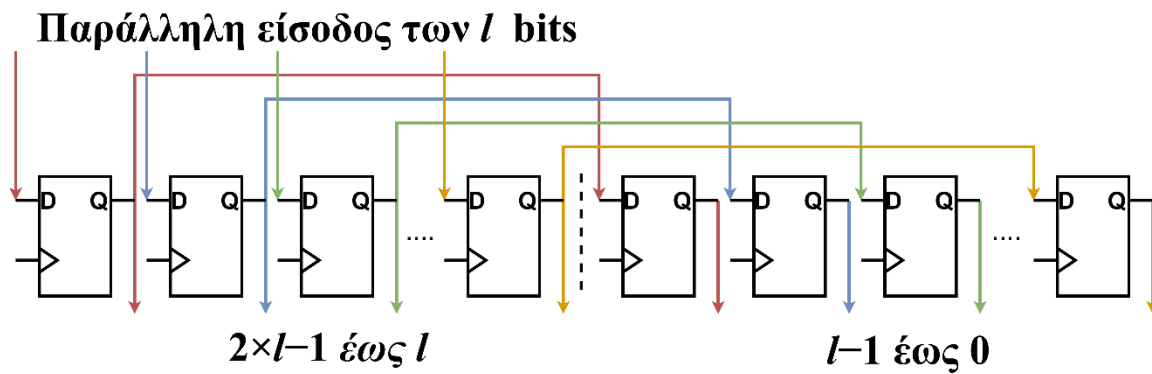
Παράλληλη είσοδος



Σχήμα 3-2: Σύνολο αρχιτεκτονικής.

3.3.1 Μονάδα παραθύρου (ΜΠ)

Η ΜΠ απομονώνει μια συγκεκριμένη περιοχή της εισερχόμενης ροής δυαδικών ψηφίων, έτσι ώστε η ΜΣ να μπορεί να προσδιορίσει εάν το b_l (και, ως αποτέλεσμα, το πλαίσιο) ανιχνεύθηκε στη συγκεκριμένη περιοχή. Η μονάδα αποτελείται από έναν καταχωρητή $2 \times l$ bit (στοιχεία μνήμης $2 \times l - 1$ έως 0). Η είσοδος l -bit εισέρχεται στο αριστερό μισό του καταχωρητή (στοιχεία $2 \times l - 1$ έως 1) από τις θύρες εισόδου στον κύκλο ρολογιού t . Στον κύκλο ρολογιού $t + 1$, τα περιεχόμενα του αριστερού μισού μεταφέρονται στο δεξί μισό (στοιχεία $l - 1$ έως 0) διατηρώντας τη σειρά τους και τα νέα bits εισέρχονται στο αριστερό μισό. Ουσιαστικά, τα στοιχεία μνήμης (D flip flops) που απέχουν l θέσεις μεταξύ τους συνδέονται σειριακά. Η μονάδα ΜΠ φαίνεται στο Σχήμα 3-3.



Σχήμα 3-3: Αρχιτεκτονική ΜΠ.

$2 \times l$ είναι ο ελάχιστος αριθμός στοιχείων μνήμης που απαιτούνται για την καταγραφή της ακολουθίας b_l όταν το μέγεθος της παράλληλης εισόδου είναι l bit. Μπορούμε εύκολα να το δείξουμε αυτό λαμβάνοντας υπόψη τα ακόλουθα. Εάν το διάνυσμα εισόδου έχει μήκος l bit, η αρχική θέση της b_l μπορεί να είναι οποιοδήποτε από αυτά τα l bit, που σημαίνει ότι όλες οι πιθανές θέσεις για τη b_l είναι $2 \times l - 2$ έως $l - 1$ (b_l ξεκινά από το bit $l - 1$), $2 \times l - 3$ έως $l - 2$ (b_l ξεκινά από το bit $l - 2$) ... $l - 1$ έως 0 (b_l ξεκινά από το 0). Επειδή τα στοιχεία σε κάθε κύκλο ρολογιού μετατοπίζονται l bit, οι θέσεις $2 \times l - 1$ έως l είναι ίδιες με τις θέσεις $l - 1$ έως 0 (μετατοπίζονται προς τα δεξιά κατά l bit) και δεν χρειάζεται να ελεγχθούν. Το να βρεθεί ποια από αυτές τις θέσεις είναι η πιο πιθανή είναι ο ρόλος της Μονάδας Συσχέτισης (ΜΣ).

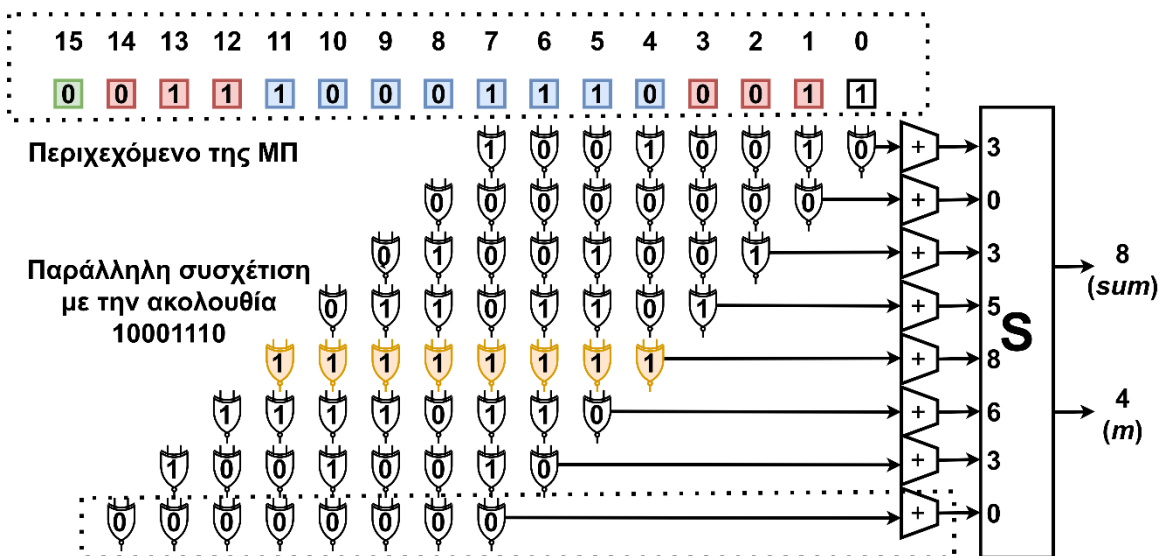
3.3.2 Μονάδα συσχέτισης (ΜΣ)

Η ΜΣ είναι ο πυρήνας της αρχιτεκτονικής και αποτελείται από τρία κύρια υποκυκλώματα: Τα παράλληλα δέντρα αθροιστή, τον επιλογέα και την προσωρινή μνήμη καθυστέρησης εισόδου. Τα παράλληλα δέντρα αθροιστή εκτελούν την ενέργεια που περιγράφεται στο (2). Συσχετίζουν, παράλληλα (ψηφιακά όπως περιγράφεται στο (1)), την αποθηκευμένη ακολουθία b_l με τα διανύσματα σε όλα οι πιθανές θέσεις b_l . Όπως αναφέρθηκε, αυτές οι θέσεις είναι $2 \times l - 2$ έως $l - 1$, $2 \times l - 3$ έως $l - 2$... $l - 1$ έως 0 . Το αποτέλεσμα αυτής της διαδικασίας είναι l τιμές sum_i , η υψηλότερη από τις οποίες είναι η πιο πιθανή θέση για το b_l . Για κάθε θέση, η τιμή συσχέτισης sum_i υπολογίζεται χρησιμοποιώντας πύλες XNOR (για τη συσχέτιση) και ένα δέντρο αθροιστή (για τον υπολογισμό του αθροίσματος ομοιότητας). Τα δέντρα αθροιστή είναι αυτά που περιεγράφηκαν στο κεφάλαιο 2.

Κάθε δέντρο αθροιστή αντιστοιχεί σε ένα άθροισμα ομοιότητας sum_i , επομένως συνολικά χρησιμοποιούνται l δέντρα αθροιστή. Η καθυστέρηση του δέντρου αθροιστή είναι $\text{ceil}(\log_2 l)$ κύκλοι ρολογιού. Το υποκύκλωμα «Επιλογέας» είναι ένα δέντρο συγκριτών όπως αυτό περιεγράφηκε στο κεφάλαιο 2. Επιλέγει το μεγαλύτερο άθροισμα ομοιότητας sum_m και δίνει σαν έξοδο αυτό μαζί με

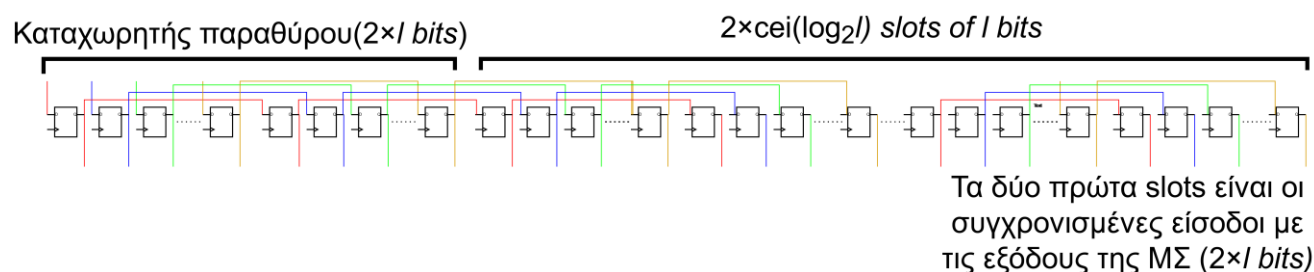
την θέση του m . Η τιμή της θέσης m ουσιαστικά αντιστοιχεί στο πρώτο bit της b_l . Το δέντρο συγκριτών έχει την ίδια καθυστέρηση σε κύκλους με το δέντρο αθροιστών δηλαδή $\text{ceil}(\log_2 l)$.

Ένα παράδειγμα του κυκλώματος της ΜΠ μαζί με τα παράλληλα δέντρα και τον Επιλογέα όταν $l = 8$ (μήκος b_l) και $k = 3$ (μήκος a_k, c_k) φαίνονται στο Σχήμα 3-4. Με αυτά τα μήκη, αν και πολύ μικρά για πρακτικές εφαρμογές, φαίνεται ξεκάθαρα η διαδικασία συγχρονισμού. Στο συγκεκριμένο παράδειγμα, η ακολουθία που αναζητείται (b_l) είναι 10001110. Για μεγαλύτερο μέγεθος l , το κύκλωμα που απεικονίζεται είναι πανομοιότυπο. Η μόνη διαφορά είναι ένας μεγαλύτερος καταχωρητής ΜΠ και περισσότεροι συσχετιστές.



τιμές, 6 και 5, εμφανίζονται μόνο μία φορά η καθεμία. Αυτό μπορεί να αποδοθεί εν μέρει στην ύπαρξη των a_k και c_k , τα οποία επηρεάζουν έμμεσα την ακρίβεια της ανίχνευσης.

Εφόσον η ΜΣ έχει καθυστέρηση $2 \times \text{ceil}(\log_2 l)$, η ΜΩΦ (επόμενη ενότητα) δεν μπορεί να χρησιμοποιήσει τον καταχωρητή της ΜΠ για τη σύλληψη του ωφέλιμου φορτίου, καθώς οι είσοδοι δεν είναι συγχρονισμένες με τις εξόδους. Για να συλλάβει αποτελεσματικά τα δεδομένα ωφέλιμου φορτίου, η ΜΩΦ πρέπει να μπορεί να «βλέπει» τον καταχωρητή παραθύρου με καθυστέρηση συγχρονισμένη με τις αντίστοιχες εξόδους της ΜΣ. Για να γίνει αυτό, η ΜΣ είναι εξοπλισμένη με προσωρινή μνήμη καθυστέρησης. Αυτή προκαλεί σειριακή καθυστέρηση $2 \times \text{ceil}(\log_2 l)$ κύκλων ρολογιού στην είσοδο και τη συγχρονίζει με τις αντίστοιχες τιμές sum_m και m . Έχει δηλαδή $2 \times \text{ceil}(\log_2 l)$ slots των l bits και τα bit στο εσωτερικό του μετακινούνται ανά l θέσεις όπως στον καταχωρητή παραθύρου. Ο καταχωρητής καθυστέρησης απεικονίζεται στο Σχήμα 3-5.



Σχήμα 3-5: Γραμμή καθυστέρησης αποτελούμενη από D flip-flops όπως ο καταχωρητής παραθύρου. Τα bit μετακινούνται προς τα δεξιά με shifts ανά l bits.

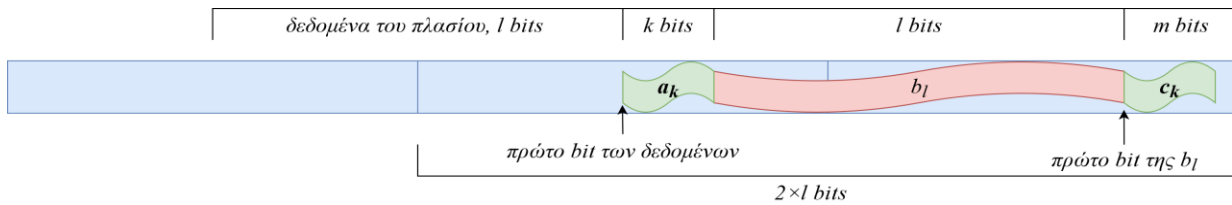
Η καθυστερημένη είσοδος, μαζί με τις δύο τιμές, δίνεται στη ΜΩΦ για να γίνει η σύλληψη ωφέλιμου φορτίου.

3.3.3 Μονάδα σύλληψης ωφέλιμου φορτίου (ΜΩΦ)

Η ΜΩΦ χρησιμοποιεί τις δύο εξόδους της ΜΣ μαζί με την συγχρονισμένη είσοδο για να υπολογίσει σωστά τη θέση του ωφέλιμου φορτίου και να συλλάβει τα δεδομένα. Όπως αναφέρθηκε στο (1), όταν η τιμή της ψηφιακής συσχέτισης είναι υψηλή, τα δύο δυαδικά διανύσματα είναι κοντά ως προς την ομοιότητα. Συνεπώς, εάν η τιμή (η οποία είναι η υψηλότερη από όλες τις θέσεις) είναι πάνω από ένα ορισμένο όριο, τότε αυτό σημαίνει ότι ο συσχετιστής έχει εντοπίσει ένα διάνυσμα στη θέση m που είναι πολύ κοντά στη b_l . Εάν το κατώφλι έχει ρυθμιστεί κατάλληλα, η ΜΩΦ μπορεί όχι μόνο να εντοπίσει το ωφέλιμο φορτίο αλλά και να το αιχμαλωτίσει ανεξάρτητα από το μέγεθος του πλαισίου. Για το λόγο αυτό, η επιλογή της κατάλληλης τιμής για το κατώφλι είναι καθοριστικής σημασίας για τη σωστή λειτουργία της αρχιτεκτονικής. Μέχρι στιγμής, αυτό επιτυγχάνεται με πειραματισμούς. Το κατώφλι πρέπει να είναι αρκετά χαμηλό για να ανιχνεύει τη λέξη συγχρονισμού ακόμη και με την

παρουσία πολλών σφαλμάτων bit, αλλά όχι τόσο χαμηλό ώστε να ενεργοποιείται συχνά από τυχαία δεδομένα.

Για τον προσδιορισμό της αρχικής θέσης της b_l και του ωφέλιμου φορτίου, η μονάδα χρησιμοποιεί μια απλή μονάδα ελέγχου που ελέγχει συνεχώς το άθροισμα ομοιότητας της ΜΣ. Όταν η έξοδος ξεπεράσει το κατώφλι στον κύκλο ρολογιού t , η μονάδα ελέγχου ενεργοποιείται. Υποθέτει ότι το ωφέλιμο φορτίο βρίσκεται στη θέση $l + k + m_t$ και ξεκινά τη σύλληψή του. Στον επόμενο κύκλο $t + 1$, συγκρίνει τη νέα τιμή του $sum_{m_{t+1}}$ με το προηγούμενο sum_{m_t} . Εάν $sum_{m_{t+1}} < sum_{m_t}$ τότε το πρώτο bit της b_l ήταν στη θέση m_t και δεν απαιτούνται αλλαγές. Εάν ισχύει το αντίθετο, το πρώτο bit της b_l βρίσκεται στη θέση m_{t+1} και η διαδικασία λήψης πλαισίου πρέπει να ξεκινήσει ξανά για να καταγράψει τα σωστά bit. Εάν στο κύκλο t το πρώτο bit του πλαισίου είναι εκτός ΜΠ τότε περιμένει ένα κύκλο ώστε να έρθει μέσα στην ΜΠ. Η θέση του ωφέλιμου φορτίου σε σχέση με την επικεφαλίδα πλαισίου φαίνεται στο Σχήμα 3-6.



Σχήμα 3-6: Θέση του φορτίου δεδομένων σε σχέση με την επικεφαλίδα $a_k b_l c_k$ στα πρώτα τρία slots της γραμμής καθυστέρησης

Υπάρχουν πολλοί τρόποι με τους οποίους μπορεί να επιτευχθεί η σύλληψη, αλλά εδώ προτείνεται ο απλούστερος. Υποθέτοντας ότι το μέγεθος του ωφέλιμου φορτίου είναι πολλαπλάσιο του l ($p = n \times l$), το ωφέλιμο φορτίο μπορεί να αποτυπωθεί σε n κύκλους με τον ακόλουθο τρόπο. Χρησιμοποιείται το 3^ο slot των l bits της γραμμής καθυστέρησης και αιχμαλωτίζονται τα bit $l + k + m + l - 1$ έως $l + k + m$ μετρώντας από δεξιά. Καθώς τα bit στην γραμμή καθυστέρησης μετακινούνται l θέσεις, τα παραπάνω bit ανανεώνονται σε κάθε κύκλο με αποτέλεσμα μετά από n κύκλους να έχουν αιχμαλωτιστεί τα δεδομένα όλου του πλαισίου. Το ωφέλιμο φορτίο συλλαμβάνεται με τον ίδιο ρυθμό που εισέρχεται στο κύκλωμα και το σήμα των δύο bit χρησιμοποιείται για να υποδείξει ότι η διαδικασία σύλληψης βρίσκεται σε εξέλιξη. Το μεγαλύτερο πλεονέκτημα της χρήσης αυτής της μεθόδου για τη λήψη των πλαισίων είναι ότι ανεξάρτητα από το πόσο μεγάλο είναι το p (το μέγεθος του ωφέλιμου φορτίου), το μέγεθος του κυκλώματος δεν αλλάζει, παρά μόνο οι κύκλοι που απαιτούνται για τη σύλληψή του. Ενώ η σύλληψη του ωφέλιμου φορτίου βρίσκεται σε εξέλιξη, η μονάδα συσχέτισης απενεργοποιείται για εξοικονόμηση ενέργειας.

3.4 Κυκλωματική υλοποίηση

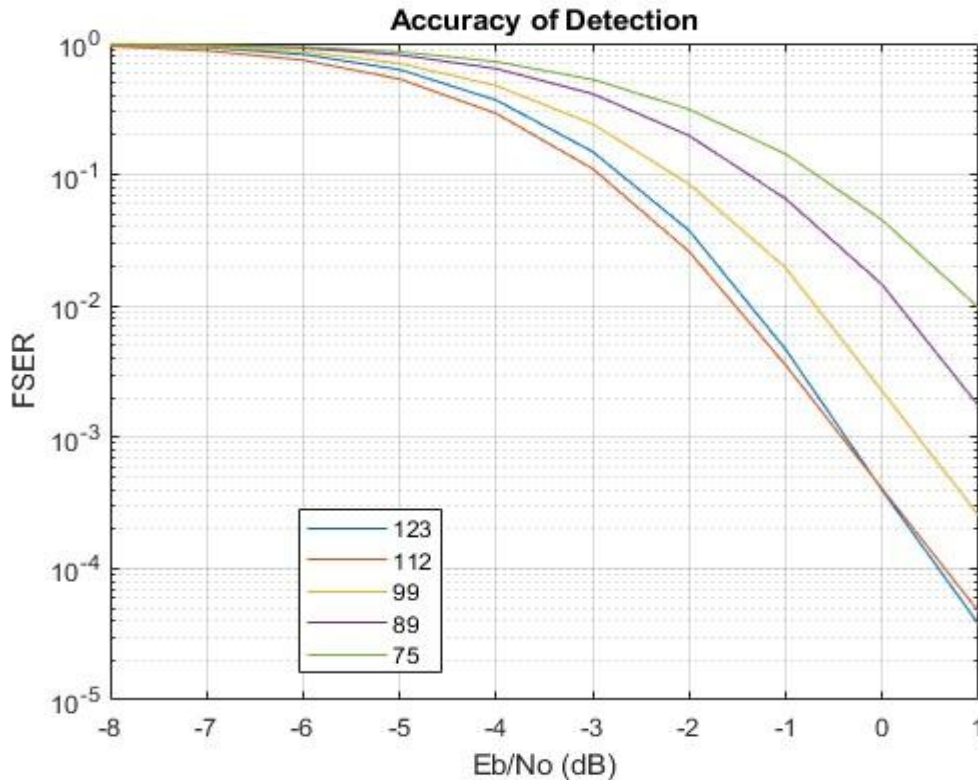
Η αρχιτεκτονική υλοποιήθηκε στη χαμηλού κόστους πλακέτα NEXYS 4 DDR FPGA για πέντε περιπτώσεις. Αυτές ήταν $l = 123, k = 23, th = 93$ (κατώφλι). $l = 112, k = 21, th = 84$; $l = 99, k = 17, th = 76$; $l = 89, k = 19, th = 70$; $l = 75, k = 16, th = 59$. Επιτεύχθηκε μέγιστη συχνότητα λειτουργίας 125 Mhz για όλες τις περιπτώσεις. Οι ρυθμοί bit υπολογίζονται πολλαπλασιάζοντας την παράλληλη είσοδο με τη συχνότητα λειτουργίας, συνεπώς $123 \text{ bit} \times 125 \text{ MHz} = 15,37 \text{ Gbps}$, $112 \text{ bit} \times 125 \text{ MHz} = 14,00 \text{ Gbps}$, $99 \text{ bit} \times 125 \text{ MHz} \times 125 \text{ MHz} = 51 \text{ bits} \times 5. = 11,125 \text{ Gbps}$ και $75 \text{ bit} \times 125 \text{ MHz} = 9,375 \text{ Gbps}$. Οι απαραίτητοι πόροι υλικού, η κατανάλωση ενέργειας και ο ρυθμός bit παρουσιάζονται στον Πίνακα 3-1. Η αρχιτεκτονική δεν καταναλώνει μονάδες DSP επειδή χρησιμοποιεί μόνο αθροιστές μικρού μήκους, συγκριτές, πύλες XNOR και καταχωρητές.

Πίνακας 3-1: Αποτελέσματα υλοποίησης.

b_l	LUTs (%)	FlipFlops (%)	P(W)	Ρυθμός bit(Gbps)
123	20027 (32)	23608 (18)	0.842	15.375
112	16393 (26)	19530 (15)	0.728	14
99	13657 (22)	15817 (12)	0.617	12.375
89	11136 (18)	13779 (11)	0.529	11.125
75	8112 (13)	10253 (8)	0.413	9.375

Για τη έλεγχο της ορθής λειτουργίας της αρχιτεκτονικής χρησιμοποιήθηκαν VIVADO και MATLAB. Οι ροές εισόδου δυαδικών ψηφίων προετοιμάστηκαν στο MATLAB και εισήχθησαν στον πάγκο δοκιμών (testbench) VHDL ως αρχεία txt. Το επιλεγμένο σχήμα διαμόρφωσης είναι 16QAM και το εύρος του SNR είναι $[-8, 1]$, ο δίαυλος επικοινωνίας είναι AWGN.

Οι είσοδοι προετοιμάστηκαν για κάθε ακέραια τιμή SNR ξεχωριστά (για όλα τα l). Αποτελούνται από 21.368 πλαίσια της μορφής $payload_{pack}$. Τα bitstreams (ροές) δημιουργήθηκαν, διαμορφώθηκαν με 16QAM και οδηγήθηκαν μέσω του καναλιού AWGN. Για κάθε τιμή SNR χρησιμοποιήθηκε διαφορετικός σηματοδότης (ack_{pack}). Μετά από αυτή τη διαδικασία, οι ροές δεδομένων αποδιαμορφώθηκαν, καταγράφηκαν σε αρχεία txt και δόθηκαν ως είσοδος στα κυκλώματα που υλοποιήθηκαν. Τα ωφέλιμα φορτία ήταν μεγέθους 12.300, 11.200, 99.000, 89.000, 75.000 (πολλαπλάσια των 123, 112, 99, 89, 75), καταγράφηκαν σε αρχεία txt και ελέγχθηκαν από το MATLAB. Το ποσοστό σφάλματος συγχρονισμού πλαισίου (FSER) φαίνεται στο Σχήμα 3-6.



Σχήμα 3-7: Αποτελέσματα ακρίβειας εντοπισμού.

Εφόσον δεν εφαρμόστηκε καμία πρόσθετη επεξεργασία στο σήμα (μόνο διαμόρφωση/αποδιαμόρφωση), ο ρυθμός σφάλματος bit πάνω στο bitstream για κάθε τιμή SNR είναι γνωστός θεωρητικά και μπορεί να υπολογιστεί με συναρτήσεις MATLAB όπως `berawgn` ή `qammod/qamdemod`. Για τις τιμές ενδιαφέροντος SNR, -1 , 0 και 1 , οι αντίστοιχες τιμές BER είναι $0,1637$, $0,1410$ και $0,1190$. Αυτό ισχύει για όλες τις διαφορετικές εκδόσεις της αρχιτεκτονικής επειδή χρησιμοποιήθηκε το ίδιο σχήμα διαμόρφωσης (16QAM). Η διαφορά στην ακρίβεια προκύπτει λόγω των διαφορετικών μηκών b_l . Η εξαγωγή μιας έκφρασης για το FSER είναι επίσης δυνατή, καθώς το ποσοστό σφάλματος bit (16QAM στο κανάλι AWGN) μπορεί να υπολογιστεί απευθείας. Ωστόσο, ακόμα κι αν συμβαίνει αυτό, η τυχαία φύση της b_l ως δυαδική ακολουθία καθιστά τον μαθηματικό υπολογισμό του FSER εξαιρετικά δύσκολο. Προς το παρόν, εάν η b_l δημιουργηθεί τυχαία, τα μόνα διαθέσιμα δεδομένα θα προέλθουν από δοκιμές. Βλέπουμε ότι το υλοποιημένο κύκλωμα επιτυγχάνει ποσοστό σφάλματος συγχρονισμού πλαισίου $3,9311 \times 10^{-4}$, $4,0247 \times 10^{-4}$, $0,0022$, $0,0145$, $0,045$ ($l = 123, 112, 99, 89, 75$ για 0 dB, αντίστοιχα). Το ποσοστό των χαμένων πλαισίων γίνεται $3,7439 \times 10^{-5}$, $4,6799 \times 10^{-5}$, $2,5271 \times 10^{-4}$, $0,0017$, $0,0097$ σε 1 dB SNR. Είναι κατανοητό ότι, γενικά, όσο μειώνεται το μέγεθος της λέξης συγχρονισμού, τόσο μειώνεται και η ακρίβεια. Η υλοποίηση $l = 75$ έχει σημαντικά χειρότερη ικανότητα ανίχνευσης από την έκδοση $l = 123$. Το $l = 112$ φαίνεται να είναι καλύτερο σε χαμηλότερα SNR, αλλά τελικά, το $l = 123$ το ξεπερνά. Αυτό μπορεί να αποτελεί ένδειξη ότι το k πρέπει να είναι μεγαλύτερο από 23 για $l = 123$.

Εάν χρησιμοποιήσουμε $SNR = 0$ και 1 dB (για το κανάλι AWGN) ως βάση σύγκρισης, μπορούμε να συγκρίνουμε τα αποτελέσματα ακρίβειας με άλλες μεθόδους και να δούμε πόσο καλά λειτουργεί η αρχιτεκτονική.

Στο μοντέρνο συγχρονισμό πλαισίων οι δύο κύριες μέθοδοι που έχουν επικρατήσει είναι οι παραλλαγές του κανόνα πιθανότητας του Massey[5][6] και ο συγχρονισμός πλαισίων με την βοήθεια κωδικοποίησης διόρθωσης λαθών όπως LDPC[7][8]. Το [5] είναι μια παραλλαγή του Massey όπου γίνεται παρατήρηση πολλών πλαισίων (όχι μόνο ενός) για την επίτευξη σωστού συγχρονισμού. Η μεγαλύτερη ακρίβεια που πετυχαίνει είναι 0.98 για $SNR=0$ dB που είναι κατά πολύ μικρότερη της προτεινόμενης μεθόδου, παρότι χρησιμοποιούνται διαμορφώσεις BPSK και QPSK (χαμηλότερης τάξης από την 16QAM). Στο [6] παρουσιάζεται ένας σειριακός αλγόριθμος (σειριακός με την έννοια ότι εφαρμόζεται σειριακά στο datastream) που υπολογίζει κάποιες μετρικές πιθανότητας παρόμοιες με αυτές του Massey για συγκεκριμένα παράθυρα χρόνου. Η μικρότερη πιθανότητα αποτυχίας συγχρονισμού είναι κοντά στο 4×10^{-3} για $SNR = 0$ dB, ποσοστό χαμηλότερο από της αρχιτεκτονικής. Και σε αυτήν την μέθοδο χρησιμοποιήθηκε BPSK διαμόρφωση. Ο συγχρονισμός πλαισίων με την βοήθεια κωδικοποίησης LDPC είναι η νέα μέθοδος της οποίας η χρήση συνεχώς αυξάνεται λόγω της δημοφιλίας των LDPC στις νέες τεχνολογίες (5G [12]). [7] και [8] περιέχουν αποτελέσματα συγχρονισμού αυτής της μεθόδου τα οποία συγκρίνουν με την μέθοδο του Massey αλλά και με την κλασσική μέθοδο ψηφιακής συσχέτισης που χρησιμοποιεί η αρχιτεκτονική. Στο [7] η μικρότερη πιθανότητα λάθους είναι 10^{-4} με λέξεις μεγέθους 78 και στο [8] 10^{-2} με λέξεις μεγέθους 35 bits. Παρατηρούμε ότι η αρχιτεκτονική χρειάζεται μεγαλύτερο μέγεθος λέξεων για να φτάσει την ίδια ακρίβεια που είναι λογικό καθώς οι ακολουθίες b_l δεν είναι βέλτιστες.

Για τις παραλλαγές του Massey δεν υπάρχουν πρακτικές υλοποιήσεις. Για τον συγχρονισμό LDPC οι αποκωδικοποιητές LDPC[13] που χρησιμοποιούνται στην διαδικασία είναι κυκλώματα σαφώς πιο αργά. Σύμφωνα με πρόσφατη έρευνα για LDPC αποκωδικοποιητές [10] ακόμα και μεταξύ πρόσφατων υλοποιήσεων [14][15][16] ο ρυθμός bit που προσφέρεται είναι κατά πολύ χαμηλότερος (2.264 έναντι 15 Gbps) για παρόμοιες ισχύς (0.212 W). Τα αποτελέσματα αυτά γίνονται ακόμα πιο σημαντικά αν προσέξουμε ότι όλες αυτές οι υλοποιήσεις έγιναν σε ASIC που σημαίνει ότι θα έπρεπε κανονικά να είναι πιο γρήγορες και να καταναλώνουν λιγότερη ισχύ[17]. Επίσης η αποκωδικοποίηση LDPC έχει μεγαλύτερη πολυπλοκότητα από την ψηφιακή συσχέτιση που σημαίνει ότι η διαδικασία του συγχρονισμού θα καθυστερήσει περισσότερο [18]. Λόγω των παραπάνω, θα ήταν ίσως προτιμότερο να χρησιμοποιείται η προτεινόμενη αρχιτεκτονική για συγχρονισμό πλαισίων και οι LDPC κώδικες για διόρθωση λαθών.

Πέρα από τα παραπάνω, η αρχιτεκτονική έχει κάποια επιπλέον πλεονεκτήματα που συνδέονται με τον μινιμαλιστικό σχεδιασμό της. Μπορεί να συλλάβει οποιοδήποτε μέγεθος ωφέλιμου φορτίου

χωρίς να αλλάξει το μέγεθος της. Μπορεί να χρησιμοποιηθεί ως μηχανισμός ασφαλείας για την κάλυψη της μετάδοσης στο φυσικό επίπεδο [19] επειδή η ακολουθία λέξεων συγχρονισμού *b₁* είναι μη προβλέψιμη. Μπορεί επίσης να εφαρμοστεί σε πλατφόρμες πολύ χαμηλού κόστους, γεγονός που την καθιστά κατάλληλη για εφαρμογές με αυστηρούς περιορισμούς πόρων, όπως το IoT [20]. Τέλος, η ίδια η αρχιτεκτονική μπορεί να χρησιμοποιηθεί με πολλούς τρόπους. Για παράδειγμα, μπορεί κανείς να στείλει πρώτα πλαίσια ευθυγράμμισης και μετά πλαίσια δεδομένων ή να ξεκινήσει αμέσως την αποστολή πλαισίων δεδομένων. Η αρχιτεκτονική μπορεί να παρέχει τη θέση του πλαισίου μαζί με το ωφέλιμο φορτίο, ώστε να μπορεί να χρησιμοποιηθεί με κάθε τρόπο. Το μόνο που εγγυάται είναι η πιθανότητα να χαθεί ένα πλαίσιο κάτω από ένα συγκεκριμένο SNR. Το ερώτημα ποια είναι η βέλτιστη στρατηγική που πρέπει να ακολουθηθεί κατά τη χρήση της (πρώτα πλαίσια ευθυγράμμισης και στη συνέχεια πλαίσια δεδομένων ή απλώς πλαίσια δεδομένων από την αρχή), όσον αφορά τη συνολική αποτελεσματική απόδοση, χρειάζεται περαιτέρω διερεύνηση και θα αποτελέσει αντικείμενο μελλοντικής εργασίας.

Στο επόμενο και τελευταίο κεφάλαιο της εργασίας θα εξετάσουμε τις επιπτώσεις που έχει η αρχιτεκτονική στην παρούσα τεχνολογία συγχρονισμού πλαισίων καθώς και τις ενδείξεις ότι η συγκεκριμένη μέθοδος είναι η βέλτιστη σε επίπεδο πρακτικής υλοποίησης.

3.5 Σημασία των αποτελεσμάτων

Στο τελευταίο κεφάλαιο της εργασίας θα εξετάσουμε γιατί η ακρίβεια συγχρονισμού που προσφέρει η αρχιτεκτονική είναι σχεδόν «τερματική». Είναι, δηλαδή, κοντά στη μέγιστη δυνατή ακρίβεια που μπορούν να εκμεταλλευτούν τα σύγχρονα τηλεπικοινωνιακά συστήματα. Βασικός στόχος όλων των τηλεπικοινωνιακών συστημάτων είναι η μετάδοση πληροφορίας χωρίς λάθη (errorless transmission). Η σημαντικότητα της μετάδοσης χωρίς λάθη φαίνεται από το γεγονός πως διαδεδομένα πρωτόκολλα όπως το IP απορρίπτουν ολόκληρα πακέτα δεδομένων αν υπάρχουν σφάλματα στην επικεφαλίδα (header checksum) [21]. Αυτό έχει ως αποτέλεσμα την δραματική πτώση της ωφέλιμης ροής πληροφορίας όταν υπάρχουν πολλά σφάλματα bit [22].

Επειδή στον πραγματικό κόσμο τα σφάλματα bit είναι αναπόφευκτα, για να είναι δυνατή η μετάδοση χωρίς λάθη έχουν επινοηθεί μέθοδοι που διορθώνουν τα λάθη σε κάποιο βαθμό. Οι μέθοδοι αυτοί ονομάζονται κωδικοί διόρθωσης λαθών (error correction codes) και η λειτουργία τους στηρίζεται σε μία βασική ιδέα. Για ένα μήνυμα με μήκος z bits, παράγεται μέσω μια διαδικασίας κωδικοποίησης ένα άλλο μήνυμα $z+v$ bits. Το μήνυμα που παράγεται μέσω της κωδικοποίησης είναι μοναδικό και έχει μεγαλύτερο αριθμό bit από το αρχικό. Ο λόγος μεταξύ του μεγέθους του αρχικού μηνύματος και του τελικού είναι ο λόγος κωδικοποίησης $\frac{z}{z+v} = R$ (code rate) και δείχνει πόσα παραπάνω bit στέλνουμε για να μπορεί να διορθωθεί το μήνυμα μετά την

μετάδοση. Υπάρχουν πολλοί κώδικες διόρθωσης λαθών όπως Turbo[23], LDPC[24], polar[25]) και ο καθένας χρησιμοποιεί εντελώς διαφορετική μέθοδο κωδικοποίησης. Περισσότερες λεπτομέρειες για τον τρόπο λειτουργίας των κωδίκων είναι εκτός θέματος και δεν θα εξεταστούν. Αυτό που αφορά το συγχρονισμό πλαισίων είναι τα θεωρητικά όρια της ικανότητας διόρθωσης, δηλαδή το μέγιστο ποσοστό λαθών που μπορεί να διορθώσει ένας κώδικας ώστε να έχουμε μετάδοση χωρίς λάθη.

Σύμφωνα με τον Shannon η χωρητικότητα του καναλιού AWGN για διακριτή μετάδοση (discrete AWGN channel) είναι 1 bits/per symbol όταν SNR=0dB [26] σε 16QAM. Αυτό σημαίνει ότι δεν μπορεί να σταλεί περισσότερη ωφέλιμη πληροφορία από 1 bit σε κάθε σύμβολο της 16QAM, με αποτέλεσμα να χρειαζόμαστε λόγο κωδικοποίησης $R=\frac{1}{4}$ ($z=1$ και $v=3$). Για τους LDPC και Polar ο λόγος υπάρχει θεωρητικά όμως η συντριπτική πλειοψηφία των υλοποιήσεων τους δεν τον χρησιμοποιεί[27][28]. Στις τεχνολογίες 5G, LDPC και Polar χρησιμοποιούνται με $R=\frac{1}{2}$ [29]. Για τους Turbo codes ο λόγος είναι πιο συχνός[30] όμως γενικά αυτοί οι κώδικες είναι παλιότερης τεχνολογίας και έχουν παραγκωνισθεί από τους LDPC και Polar στα μοντέρνα δίκτυα[31]. Άλλου τύπου κώδικες όπως Reed Solomon, BCH κτλ θεωρούνται επίσης ξεπερασμένοι με πολύ περιορισμένες εφαρμογές. Γενικά χρησιμοποιούν $R>\frac{1}{4}$ [32]. Η αρχιτεκτονική ξεπερνά το πρόβλημα της χωρητικότητας του καναλιού και έχει τόσο μεγάλη ακρίβεια για τον εξής λόγο. Η πληροφορία για την θέση του πλαισίου μπορεί να θεωρηθεί ως μήνυμα ενός bit. Αυτό το bit κωδικοποιείται με l bits τα οποία τοποθετούνται στην αρχή του πλαισίου. Συνεπώς, η αρχιτεκτονική λειτουργεί σαν ένα κύκλωμα αποκωδικοποίησης με λόγο $R = \frac{1}{l}$ και συνολική ροή για 16QAM $\frac{4}{l}$ που είναι προφανώς πολύ χαμηλότερη από το 1 bit.

Από τα παραπάνω προκύπτει ότι περαιτέρω βελτίωση της ακρίβειας συγχρονισμού έχει νόημα μόνο εάν το σύστημα μπορεί να υποστηρίξει μετάδοση με λάθη. Ουσιαστικές βελτιώσεις στην αρχιτεκτονική μπορούν να γίνουν σε επίπεδο ακρίβειας αυξάνοντας το μέγεθος της ακολουθίας συγχρονισμού και διαχωρίζοντας το μέγεθος της από το μέγεθος της εισόδου.

3.6 Βιβλιογραφία κεφαλαίου

- [1] Barker, R.H. Group Synchronization of Binary Digital Systems. In *Communication Theory*; Academic Press: Cambridge, MA, USA, 1953; pp. 273–287.
- [2] M. Bekar, C. J. Baker, E. G. Hoare and M. Gashinova, "Joint MIMO Radar and Communication System Using a PSK-LFM Waveform With TDM and CDM Approaches," in *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6115-6124, 1 March 1, 2021, doi: 10.1109/JSEN.2020.3043085.
- [3] J. Massey, "Optimum Frame Synchronization," in *IEEE Transactions on Communications*, vol. 20, no. 2, pp. 115-119, April 1972, doi: 10.1109/TCOM.1972.1091127.
- [4] P. Nielsen, "Some Optimum and Suboptimum Frame Synchronizers for Binary Data in Gaussian Noise," in *IEEE Transactions on Communications*, vol. 21, no. 6, pp. 770-772, June 1973, doi: 10.1109/TCOM.1973.1091725.
- [5] E. M. Bastaki, H. H. Tan, Y. Shi and K. B. Letaief, "Frame synchronization based on multiple frame observations," in *IEEE Transactions on Wireless Communications*, vol. 9, no. 3, pp. 1097-1107, March 2010, doi: 10.1109/TWC.2010.03.081474.
- [6] M. Chiani and M. G. Martini, "On sequential frame synchronization in AWGN channels," in *IEEE Transactions on Communications*, vol. 54, no. 2, pp. 339-348, Feb. 2006, doi: 10.1109/TCOMM.2005.863727.
- [7] D. -U. p. a. U. Lee, H. Kim, C. R. Jones and J. D. Villasenor, "Pilotless Frame Synchronization for LDPC-Coded Transmission Systems," in *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2865-2874, July 2008, doi: 10.1109/TSP.2008.917348.
- [8] X. Ding *et al.*, "Customized Joint Blind Frame Synchronization and Decoding Methods for Analog LDPC Decoder," in *IEEE Transactions on Communications*, vol. 72, no. 2, pp. 756-770, Feb. 2024, doi: 10.1109/TCOMM.2023.3327779.
- [9] Z. Feng, M. Xu, L. Xiao, Z. Zhang, J. Zhou and T. Jiang, "LDPC Assisted Blind Frame Synchronization: Efficient Measurement Mechanism and Low-Complexity Algorithm," *2021 7th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2021, pp. 355-359, doi: 10.1109/ICCC54389.2021.9674430.
- [10] O. Ferraz *et al.*, "A Survey on High-Throughput Non-Binary LDPC Decoders: ASIC, FPGA, and GPU Architectures," in *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 524-556, Firstquarter 2022, doi: 10.1109/COMST.2021.3126127.
- [11] S. Matsuyuki and A. Tsuneda, "A Study on Aperiodic Auto-Correlation Properties of Concatenated Codes by Barker Sequences and NFSR Sequences," *2018 International Conference on*

Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2018, pp. 664-666, doi: 10.1109/ICTC.2018.8539367.

[12] Y. Fang, Y. Bu, P. Chen, F. C. M. Lau and S. A. Otaibi, "Irregular-Mapped Protograph LDPC-Coded Modulation: A Bandwidth-Efficient Solution for 6G-Enabled Mobile Networks," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2060-2073, Feb. 2023, doi: 10.1109/TITS.2021.3122994.

[13] J. Nadal and A. Baghdadi, "Parallel and Flexible 5G LDPC Decoder Architecture Targeting FPGA," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1141-1151, June 2021, doi: 10.1109/TVLSI.2021.3072866.

[14] S. Song, H. Cui, J. Tian, J. Lin and Z. Wang, "A Novel Iterative Reliability-Based Majority-Logic Decoder for NB-LDPC Codes," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 8, pp. 1399-1403, Aug. 2020, doi: 10.1109/TCSII.2019.2938562.

[15] Y. -L. Ueng, K. -H. Liao, H. -C. Chou and C. -J. Yang, "A High-Throughput Trellis-Based Layered Decoding Architecture for Non-Binary LDPC Codes Using Max-Log-QSPA," in *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2940-2951, June1, 2013, doi: 10.1109/TSP.2013.2256905.

[16] Toriyama and D. Marković, "A 2.267-Gb/s, 93.7-pJ/bit Non-Binary LDPC Decoder With Logarithmic Quantization and Dual-Decoding Algorithm Scheme for Storage Applications," in *IEEE Journal of Solid-State Circuits*, vol. 53, no. 8, pp. 2378-2388, Aug. 2018, doi: 10.1109/JSSC.2018.2832851.

[17] A. Boutros and V. Betz, "FPGA Architecture: Principles and Progression," in *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 4-29, Secondquarter 2021, doi: 10.1109/MCAS.2021.3071607.

[18] R. Imad, G. Sicot and S. Houcke, "Blind frame synchronization for error correcting codes having a sparse parity check matrix," in *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1574-1577, June 2009, doi: 10.1109/TCOMM.2009.06.070445.

[19] C. Zhang, J. Yue, L. Jiao, J. Shi and S. Wang, "A Novel Physical Layer Encryption Algorithm for LoRa," in *IEEE Communications Letters*, vol. 25, no. 8, pp. 2512-2516, Aug. 2021, doi: 10.1109/LCOMM.2021.3078669.

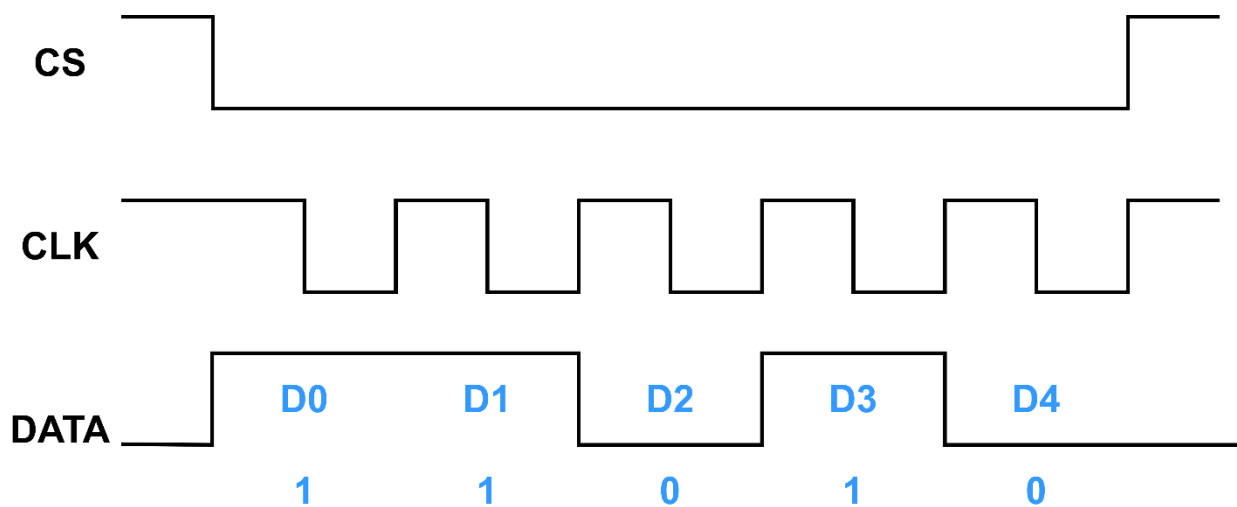
[20] E. M. Migabo, K. D. Djouani and A. M. Kurien, "The Narrowband Internet of Things (NB-IoT) Resources Management Performance State of Art, Challenges, and Opportunities," in *IEEE Access*, vol. 8, pp. 97658-97675, 2020, doi: 10.1109/ACCESS.2020.2995938.

[21] J. Postel, "Internet Protocol," RFC 791, Sept. 1981. [Online]. Available: <https://tools.ietf.org/html/rfc791>.

- [22] Y. Yang and L. Hanzo, "Permutation-Based TCP and UDP Transmissions to Improve Goodput and Latency in the Internet of Things," in *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14276-14286, 15 Sept. 15, 2021, doi: 10.1109/JIOT.2021.3068238.
- [23] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," in *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996, doi: 10.1109/26.539767.
- [24] Hao Zhong and Tong Zhang, "Block-LDPC: a practical LDPC coding system design approach," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 4, pp. 766-775, April 2005, doi: 10.1109/TCSI.2005.844113.
- [25] I. Tal and A. Vardy, "How to Construct Polar Codes," in *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562-6582, Oct. 2013, doi: 10.1109/TIT.2013.2272694.
- [26] LNT, "Exercise 4.10: QPSK Channel Capacity," LNT, [Online]. Available: https://en.lntwww.de/Aufgaben:Exercise_4.Ten:_QPSK_Channel_Capacity. [Accessed: Oct. 8, 2024].
- [27] S. Shao *et al.*, "Survey of Turbo, LDPC, and Polar Decoder ASIC Implementations," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2309-2333, thirdquarter 2019, doi: 10.1109/COMST.2019.2893851.
- [28] V. Bioglio, C. Condo and I. Land, "Design of Polar Codes in 5G New Radio," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 29-40, Firstquarter 2021, doi: 10.1109/COMST.2020.2967127.
- [29] M. Ali, M. Elkhodr, S. Shahrestani, H. N. Qureshi, and S. Khan, "A Survey of Channel Coding Techniques for 5G Wireless Networks," *Telecommunication Systems*, vol. 73, pp. 637-663, Aug. 2019. doi: 10.1007/s11235-019-00630-3.
- [30] M. Qiu, X. Wu, A. G. i. Amat and J. Yuan, "Analysis and Design of Partially Information- and Partially Parity-Coupled Turbo Codes," in *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2107-2122, April 2021, doi: 10.1109/TCOMM.2021.3052227.
- [31] M. Rowshan, M. Qiu, Y. Xie, X. Gu and J. Yuan, "Channel Coding Toward 6G: Technical Overview and Outlook," in *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2585-2685, 2024, doi: 10.1109/OJCOMS.2024.3390000.
- [32] S. Lin, K. Abdel-Ghaffar, J. Li and K. Liu, "A Scheme for Collective Encoding and Iterative Soft-Decision Decoding of Cyclic Codes of Prime Lengths: Applications to Reed-Solomon, BCH, and Quadratic Residue Codes," in *IEEE Transactions on Information Theory*, vol. 66, no. 9, pp. 5358-5378, Sept. 2020, doi: 10.1109/TIT.2020.2978383.

Παράρτημα : Πρωτόκολλο SPI

Το πρωτόκολλο SPI[1] είναι ένα πολύ διαδεδομένο πρωτόκολλο και χρησιμοποιείται από πολλές εταιρίες σαν διεπαφή επικοινωνίας (interface) για τις συσκευές τους. Τα board τα οποία χρησιμοποιήθηκαν για τον οπτικό μεταγωγέα Benes 16×16 είναι τα ad5370 και περιέχουν 40 dacs το καθένα. Για να λάβει το κάθε dac την σωστή τιμή τάσης ο κατάλληλος κωδικός τάσης πρέπει να σταλθεί μαζί με την κατάλληλη εντολή και διεύθυνση του κάθε dac. Αυτό πρέπει να γίνει και για τους 40 dac ξεχωριστά καθώς οι τάσεις για όλους τους οπτικούς διακόπτες 2×2 διαφέρουν από διακόπτη σε διακόπτη ακόμα και για την ίδια κατάσταση (δηλαδή δύο διακόπτες χρειάζονται διαφορετικό ζευγάρι τάσεων για να μουν στην ίδια κατάσταση). Η μορφή μετάδοσης δεδομένων στο SPI φαίνεται στο σχήμα *Σχήμα III*.



Σχήμα III: Σήματα SPI.

Γενικά, το SPI χρειάζεται 2 σήματα δεδομένων (DATA) (όχι ένα όπως απεικονίζεται παραπάνω) για να μπορεί να στέλνει και να δέχεται δεδομένο η συσκευή αλλά καθώς στην δικιά μας περίπτωση το FPGA μόνο στέλνει δεδομένα στα πλακέτες και τους dacs ουσιαστικά κάνουμε χρήση μόνο 3 σημάτων. Το σήμα `chip_select` ενεργοποιεί το interface της συσκευής και συνήθως είναι ενεργό στην τιμή 0 (active low) έτσι ώστε να δέχεται δεδομένα χωρίς να χρειάζεται να αλλάξει τάση. Το σήμα `clk` είναι το ρολόι του οποίου την αρνητική (ή μερικές φορές θετική) ακμή χρησιμοποιεί το interface για να δειγματοληπτήσει το σήμα των δεδομένων. Γι' αυτό η τιμή του σήματος DATA πρέπει να έχει σταθεροποιηθεί πριν την ακμή του CLK.

Για τα board η τιμή κάθε dac δίνεται από συγκεκριμένο τύπο [2]

$$DAC_INPUT = \frac{INPUT_CODE \times (M + 1)}{2^{16}} + C - 2^{15}$$

Η μεταβλητές που μπορούν να επηρεαστούν μέσω του SPI πρωτοκόλλου είναι M (gain), C (offset) και $INPUT_CODE$ που είναι ουσιαστικά η τιμή τάσης που στέλνουμε σε μορφή δυαδικού (κάθε dac

από τα 40 έχει τα δικά του M,C,INPUT_CODE). Με αυτό τον τρόπο τροποποιείται το DAC_CODE το οποίο στην συνέχεια μέσω του τύπου

$$V_{OUT} = 4 \times V_{REF} \frac{DAC_CODE - (4 \times GLOBAL_OFFSET)}{2^{16}} + V_{SIGGRD}$$

μετατρέπεται σε τάση στο DAC. Οι τιμές V_{SIGGND} και V_{REF} είναι σταθερές. Η τιμή $GLOBAL_OFFSET$ χρειάζεται ειδική εντολή για να αλλάξει οπότε μπορεί να θεωρηθεί σταθερή.

Για τις ανάγκες του πειράματος M, C και $GLOBAL_OFFSET$ αφέθηκαν στην κανονική (default) τιμή τους που είναι $M=2^{16}-1$, $C=2^{15}$, $GLOBAL_OFFSET=0x1555$ το οποίο δίνει εύρος [-4V,8V].

Τα board ad454 δέχονται 24 bit σε μορφή SPI. Από αυτά τα πρώτα 2 είναι η εντολή για το board (αν θα αλλάξει τιμή M,C ή INPUT_CODE), τα επόμενα 6 είναι η διεύθυνση του dac σύμφωνα με τον πίνακα διευθύνσεων στο εγχειρίδιο[2] και τα τελευταία 16 είναι η τιμή που στέλνουμε. Για το πείραμα αλλάζουμε μόνο το INPUT_CODE.

Το ενδιαμέσο κύκλωμα μεταφράζει το δυαδικό διάνυσμα δρομολόγησης σε τάσεις τις οποίες αποθηκεύει σε καταχωρητές εξόδου. Οι καταχωρητές εξόδου (40 θέσεις μνήμης) είναι ενσωματωμένη σε κάθε SPI interface και είναι γνωστό σε ποια ομάδα 40 DACs αντιστοιχούν. Μόλις γεμίσουν όλοι οι καταχωρητές εξόδου τα SPI interfaces ξεκινούν να στέλνουν , στο αντίστοιχο board ο καθένας, την σωστή τιμή τάσης για κάθε DAC σε μορφή SPI την μία μετά την άλλη. Στο Σχήμα Π2 παρουσιάζεται ένα στιγμιότυπο από την αποστολή των τιμών σε 2 διαδοχικούς dacs. Σημειώνουμε πως αφήνεται κάποιος χρόνος από την αποστολή το ενός μηνύματος (για τον ένα dac) μέχρι την αποστολή του άλλου όπως ορίζει το εγχειρίδιο



Σχήμα Π2: Μεταφορά δεδομένων στα ad5370

Η διαδικασία αυτή γίνεται και για τους 40 dacs σε όλα τα boards.

Βιβλιογραφία παραρτήματος

[1] F. Leens, "An introduction to I2C and SPI protocols," in *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8-13, February 2009, doi: 10.1109/MIM.2009.4762946.

[2] <https://www.analog.com/en/products/ad5370.html>

Λίστα σχημάτων

Σχήμα 1-1: Γενική μορφή δικτύου Benes.....	17
Σχήμα 1-2: Λογικό διάγραμμα του κυκλώματος δρομολόγησης μέσα στην διάταξη.....	19
Σχήμα 1-3: Γενική μορφή Clos δικτύου.....	21
Σχήμα 1-4: Εξώτατο επίπεδο του Δικτύου Benes. Τα δύο υποδίκτυα έχουν μισό αριθμό εισόδων...23	
Σχήμα 1-5: Εσωτερική δομή δικτύου Benes 8×8 . Το επίπεδο 0 σημειώνεται με κόκκινο, το 1 με μπλε και το 2 με πράσινο.....	24
Σχήμα 1-6: Οι δύο καταστάσεις ενός διακόπτη . Bar (αριστερά) και Cross (δεξιά).....	24
Σχήμα 1-7: Τα βήματα εκτέλεσης του αλγορίθμου.....	26
Σχήμα 1-8: Η αρχή της διαδικασίας δρομολόγησης.....	30
Σχήμα 1-9: Τυχαίο στιγμιότυπο στην διαδικασία δρομολόγησης.....	31
Σχήμα 1-10: Ο διακόπτης $(0',1')$ ολοκληρώνει το μονοπάτι (κόκκινο χρώμα). Το μονοπάτι είναι κλειστό και ανεξάρτητο από τις καταστάσεις των όλων των διακοπών που δεν ανήκουν σε αυτό..	32
Σχήμα 1-11: Γενικευμένο κύκλωμα δρομολόγησης δικτύου Benes $2^r=N$ εισόδων. Η έξοδος του είναι όλα τα σήματα με σήμανση 'Ρύθμιση διακοπών' τα οποία περιέχουν την κατάσταση κάθε διακόπτη (cross ή bar).....	33
Σχήμα 1-12: Το κύκλωμα για δίκτυο Benes 8×8	34
Σχήμα 1-13: Το κύκλωμα για πρώτο επίπεδο δικτύου Benes 8×8	34
Σχήμα 1-14: Η εικόνα της δρομολόγησης όλων των διακοπών μετά την ταξινόμηση.....	41
Σχήμα 1-15: Το δίκτυο Benes συνολικά δρομολογημένο.....	43
Σχήμα 1-16: Η υλοποίηση του κυκλώματος στο πλαίσιο του έργου.....	45
Σχήμα 1-17: Η μετάφραση του δυαδικού διανύσματος σε τιμές τάσεις για τα DACs.....	46
Σχήμα 1-18: Benes γενικής μορφής 3×3	46
Σχήμα 1-19: AS-Benes 5×5	47
Σχήμα 1-20: AS-Benes 6×6	47
Σχήμα 1-21: Γενική μορφή AS-Benes.....	47
Σχήμα 1-22: Δρομολόγηση AS-Benes 6×6	49
Σχήμα 1-23: Δρομολόγηση AS-Benes 5×5	50
Σχήμα 2-1: Δέντρο αθροιστής για 11 bit.....	58
Σχήμα 2-2: Μικρότερος συγκριτής μοτίβου.....	60
Σχήμα 2-3: Συγκριτής μοτίβου 2 επιπέδων.....	61
Σχήμα 3-1: Λήψη ροής δεδομένων σε πραγματικό χρόνο. Η ροή λαμβάνεται σε κομμάτια των l bits. Η επικεφαλίδα $a_k b_l c_k$ βρίσκεται μεταξύ δύο πλαισίων.....	67

Σχήμα 3-2: Σύνολο αρχιτεκτονικής.....	68
Σχήμα 3-3: Αρχιτεκτονική ΜΠ.....	69
Σχήμα 3-4: Παράλληλη συσχέτιση με το εσωτερικό της ΜΠ.....	70
Σχήμα 3-5: Γραμμή καθυστέρησης αποτελούμενη από D flip-flops όπως ο καταχωρητής παραθύρου. Τα bit μετακινούνται προς τα δεξιά με shifts ανά l bits.....	71
Σχήμα 3-6: Θέση του φορτίου δεδομένων σε σχέση με την επικεφαλίδα $a_k b_l c_k$	72
Σχήμα 3-7: Αποτελέσματα ακρίβειας εντοπισμού.....	74

Λίστα πινάκων

<i>Πίνακας 1-1: Πίνακας δρομολόγησης ενός δικτύου Benes 8×8.....</i>	<i>28</i>
<i>Πίνακας 1-2: Αντίστροφος πίνακας δρομολόγησης.....</i>	<i>35</i>
<i>Πίνακας 1-3: Πρώτη επανάληψη.....</i>	<i>36</i>
<i>Πίνακας 1-4: Δεύτερη επανάληψη.....</i>	<i>36</i>
<i>Πίνακας 1-5: Τρίτη επανάληψη.....</i>	<i>36</i>
<i>Πίνακας 1-6: Τέταρτη επανάληψη.....</i>	<i>36</i>
<i>Πίνακας 1-7: Πρώτη επανάληψη (λίστα διακοπών).....</i>	<i>37</i>
<i>Πίνακας 1-8: Δεύτερη επανάληψη (λίστα διακοπών).....</i>	<i>38</i>
<i>Πίνακας 1-9: Τρίτη επανάληψη (λίστα διακοπών).....</i>	<i>38</i>
<i>Πίνακας 1-10: Τέταρτη επανάληψη (λίστα διακοπών).....</i>	<i>38</i>
<i>Πίνακας 1-11: Πάνω (U) και Κάτω (L) υποδίκτυα.....</i>	<i>39</i>
<i>Πίνακας 1-12: Πρώτη επανάληψη (δρομολόγηση διακοπών).....</i>	<i>39</i>
<i>Πίνακας 1-13: Δεύτερη επανάληψη (δρομολόγηση διακοπών).....</i>	<i>39</i>
<i>Πίνακας 1-14: Τρίτη επανάληψη (δρομολόγηση διακοπών).....</i>	<i>40</i>
<i>Πίνακας 1-15: Τέταρτη επανάληψη (δρομολόγηση διακοπών).....</i>	<i>40</i>
<i>Πίνακας 1-16: Ζευγάρια δρομολόγησης για το πάνω υποδίκτυο.....</i>	<i>41</i>
<i>Πίνακας 1-17: Ζευγάρια δρομολόγησης για το κάτω υποδίκτυο.....</i>	<i>41</i>
<i>Πίνακας 1-18: Ζευγάρια δρομολόγησης για το πάνω υποδίκτυο (διαιρούμε με το 2).....</i>	<i>41</i>
<i>Πίνακας 1-19: Ζευγάρια δρομολόγησης για το κάτω υποδίκτυο (διαιρούμε με το 2).....</i>	<i>42</i>
<i>Πίνακας 1-20: Πίνακας δρομολόγησης για το πάνω υποδίκτυο.....</i>	<i>42</i>
<i>Πίνακας 1-21: Πίνακας δρομολόγησης για το κάτω υποδίκτυο.....</i>	<i>42</i>
<i>Πίνακας 1-22: Αποτελέσματα απομονωμένης υλοποίησης.....</i>	<i>43</i>
<i>Πίνακας 1-23: Αποτελέσματα υλοποίησης για τις ανάγκες του έργου.....</i>	<i>46</i>
<i>Πίνακας 1-24: Πίνακας δρομολόγησης 6×6 AS-Benes.....</i>	<i>48</i>
<i>Πίνακας 1-25: Πίνακας δρομολόγησης 5×5 AS-Benes.....</i>	<i>49</i>
<i>Πίνακας 2-1: Σύνολο κανόνων.....</i>	<i>57</i>
<i>Πίνακας 2-2: Η ομοιότητα των ακολουθιών.....</i>	<i>57</i>
<i>Πίνακας 3-1: Αποτελέσματα υλοποίησης.....</i>	<i>73</i>

Λίστα δημοσιεύσεων

- 1) Nikolaidis, D. Novel Minimalist Hardware Architecture for Long Sync Word Frame Synchronization and Payload Capture. Electronics 2024, 13, 3372.
<https://doi.org/10.3390/electronics13173372>
- 2) Nikolaidis, D.; Groumas, P.; Kouloumentas, C.; Avramopoulos, H. Novel Benes Network Routing Algorithm and Hardware Implementation. Technologies 2022, 10, 16.
<https://doi.org/10.3390/technologies10010016>
- 3) Nikolaidis, D.; Groumas, P.; Kouloumentas, C.; Avramopoulos, H. High-Throughput Bit Pattern Matching under Heavy Interference on FPGA. Electronics 2023, 12, 803.
<https://doi.org/10.3390/electronics12040803>
- 4) Raptakis, A.; Gounaridis, L.; Epping, J.P.; Tran, T.L.A.; Aukes, T.; Kleinert, M.; Weigel, M.; Wolfer, M.; Draebenedt, A.; Tsokos, C.; Groumas, P.; Andrianopoulos, E.; Lyras, N.; Nikolaidis, D.; Mylonas, E.; Baxevanakis, N.; Pessina, R.; Schreuder, E.; Dekkers, M.; Seyfried, V.; Keil, N.; Heideman, R.G.; Avramopoulos, H.; Kouloumentas, C. Integrated Heterodyne Laser Doppler Vibrometer Based on Stress-Optic Frequency Shift in Silicon Nitride. Photonix 2023, 4, 30.
<https://doi.org/10.1186/s43074-023-00105-4>

Extended Abstract

1. Introduction

This PhD dissertation concerns the development and implementation of novel hardware architectures on reprogrammable hardware platforms (FPGA) for high-speed networks.

In recent years there has been a slow but gradual increase in the integration of FPGA platforms in various applications. FPGAs are programmable hardware platforms on which the user can implement various special-purpose digital circuits for specific tasks. They do not have the adaptability of processors, but they perform the tasks assigned to them much faster and with greater accuracy than processors. Similar platforms to FPGAs are ASICs. Their main differences are that ASICs are much faster but less reliable (need testing) and not reprogrammable, while FPGAs are slower than ASICs, consume more power but are more reliable and reprogrammable.

Although FPGAs have clear advantages over processors, they are not used to their full potential. In most cases, they act as simple substitutes for processor units that simply speed up the computation of mathematical formulas and processes. In this paper we introduce methods and digital architectures that focus on high-speed communication and interconnection technologies and are harmonized with the specifics of the hardware. As a result, they outperform previous implementations of theoretical algorithms and can support much higher speeds while consuming fewer resources (hardware and power). In the following paragraphs of this chapter, we provide an introduction to each architecture.

Three architectures are presented in total. The first is a Benes Network switch controller which implements a novel Benes Network routing algorithm developed specifically for hardware. The second architecture is aimed at very long bit pattern detection under heavy interference with high accuracy and the last architecture uses the methods of the second to solve the problem of frame synchronization in digital links. All are implemented on FPGAs with VHDL.

2. Benes network

Benes networks are interconnection networks widely used in many areas. In recent years interest in Benes Networks has been concentrated in their implementation as electro-optical switches in large scale data centres. This trend emerged because of the saturation of purely electrical switches. Compared to them, electrooptical switches enjoy certain advantages such as scalability to large capacity, energy and cost efficiency and modular upgradability. Benes Networks are usually implemented as optical switches (16×16 , 32×32) with electrical(voltage) control signals. The benes network switch controller architecture implements a novel Benes network routing algorithm

designed specifically for hardware implementation. Given a specific routing requirement, the architecture produces a digital output which sets the switch to the correct state to satisfy it. Transfer of the digital signal to the switch is typically done with electrical signals (Digital to Analog converters). Figure 1 presents the recursive form of the Benes Network.

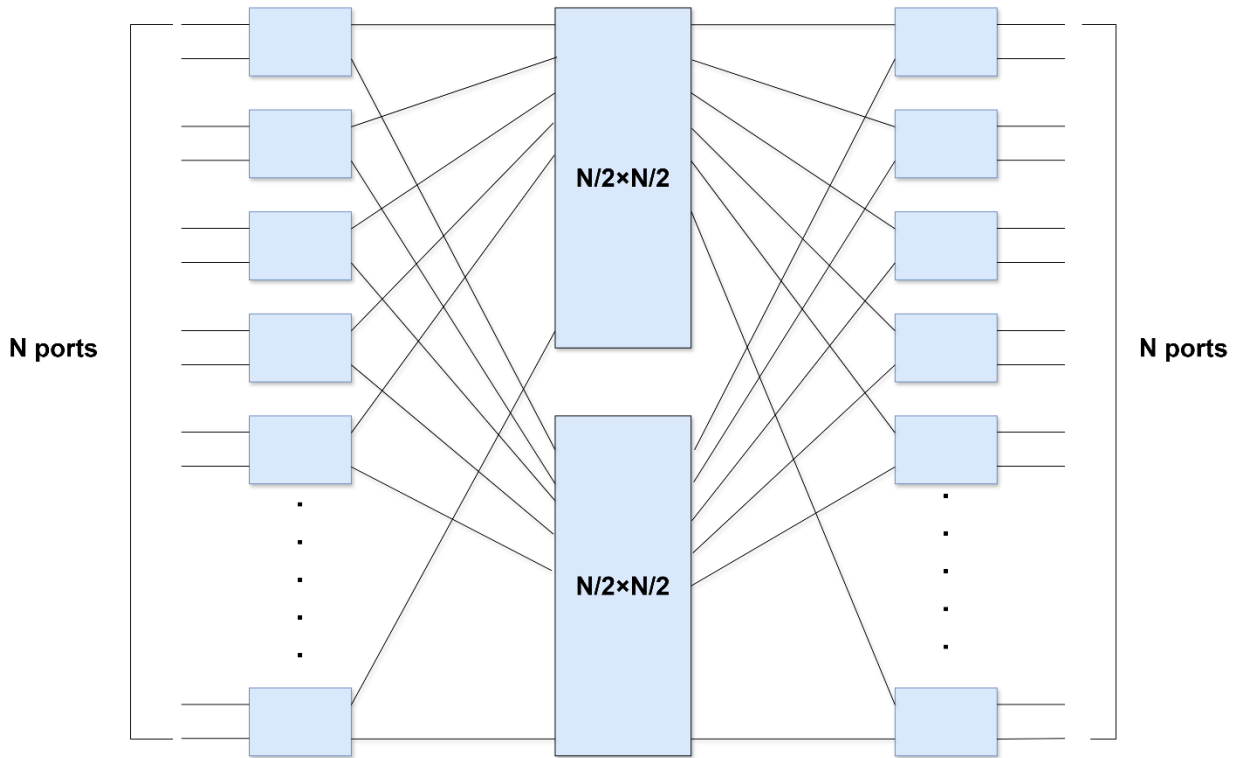


Figure 1: Benes Network of N size.

A benes Network of $N \times N$ ($N=2^n$) ports consists of two smaller Benes Networks of $N/2 \times N/2$ ports and N switches ($N/2$ to each side). All switches are 2×2 and they have two possible settings cross and bar. Both can be seen on figure 2.

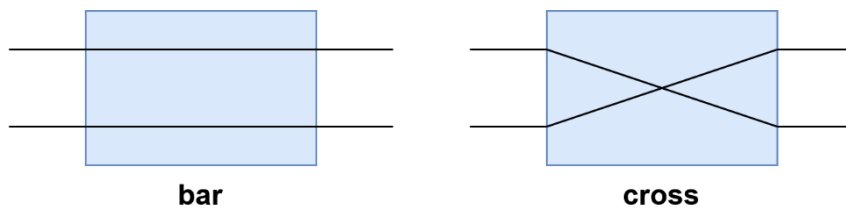


Figure 2: Bar and cross setting of the 2×2 internal switches of the Benes Network.

Routing of the Benes network entails the setting of all internal switches to either cross or bar so that ports from one side are routed to respective ports from the other side based on a given routing table. The routing table is simply a table which matches ports from the two sides. An example routing table can be seen in Table 1.

Table 1: Example of an 8×8 Benes Network routing table.

Left side ports	Right side ports
0	0'
1	2'
2	4'
3	6'
4	1'
5	3'
6	7'
7	5'

Benes Networks are rearrangeable non-blocking which means that they can satisfy any routing table as long as the network switches are set from the beginning every time the routing table changes.

In the context of the thesis, the architecture accepts the routing table as input and produces a binary vector of size $(2 \times r - 1) * 2^{(r-1)}$ (number of switches in the network). This vector contains the setting for all switches. The index of each bit represents a specific switch, and the value of the bit represents its setting. 1 for cross 0 for bar. When the setting dictated by the vector is applied to the respective switch the Benes Network is correctly set. The architecture functions as an accelerator and is implemented generically for any Benes Network size. It is more efficient than any other implementation presented in literature at the time of writing.

2.1 Routing algorithm

The routing algorithm takes advantage of the recursive nature of the Benes Network and routes the switches from the outside layers to the inside layers. Given a routing table, the algorithm routes the outer layer of switches which comprise of the left and right-side switches as seen in Figure 1. This process creates the routing table for the inner $N/2 \times N/2$ Benes Networks which are then routed in the same way. This is repeated until all switches are routed properly. The steps for routing the outer layer of a $N \times N$ Benes Network are the following:

1. Pick a random port to start the routing process. The algorithm will lead to a setting that satisfies the routing table no matter what port is chosen. However, the final setting of the Network switches may vary greatly depending on the origin port. Generally, the first port is the first port of the first switch in the right-side (from the top).

2. Route the right-side port to the respective left-side port through the upper $N/2 \times N/2$ Benes subnetwork.
3. Route the other port of the left-side switch just visited through the lower Benes subnetwork to the respective right-side port.
4. Repeat the process by always routing the other port of the switch last visited while constantly changing the subnetwork used for routing (upper, lower, upper, lower ...). If the other port is already visited a loop has been completed and a new origin port must be picked to start the routing again.

This process can be seen in detail in Figure 3 for the outer layer of an 8×8 Benes Network with the routing table given in table 1. E1 and E2 are origin ports.

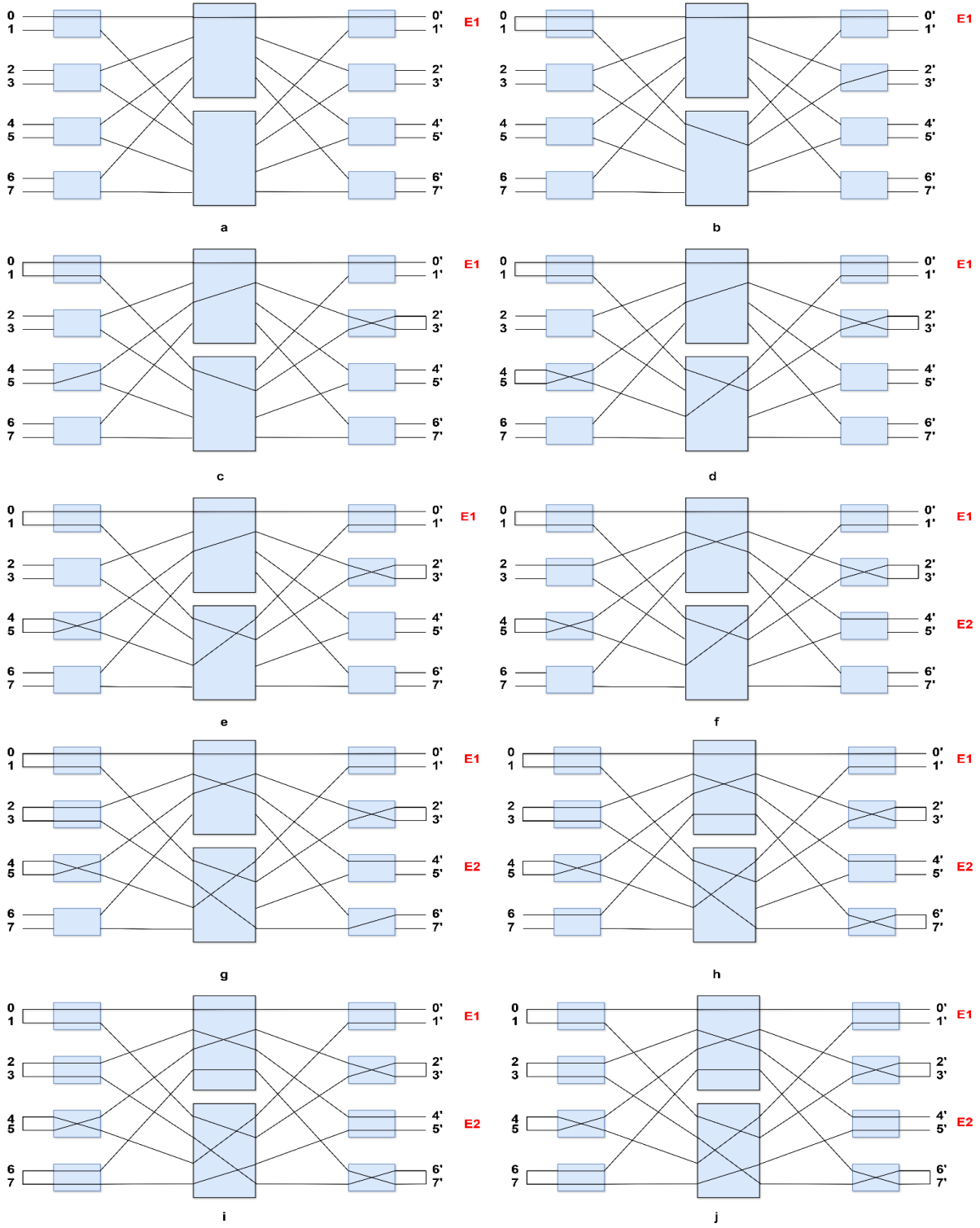


Figure 3: Routing of the outer layer of the 8×8 Benes Network.

It is clear that once the routing of the outer layer is completed the routing tables for the inner subnetworks have also been completed and can be used to continue the routing. The entire process has a complexity of $N \times \log_2 N$. Figure 4 presents a fully rooted 8×8 Benes network with its internal switches set to the correct setting.

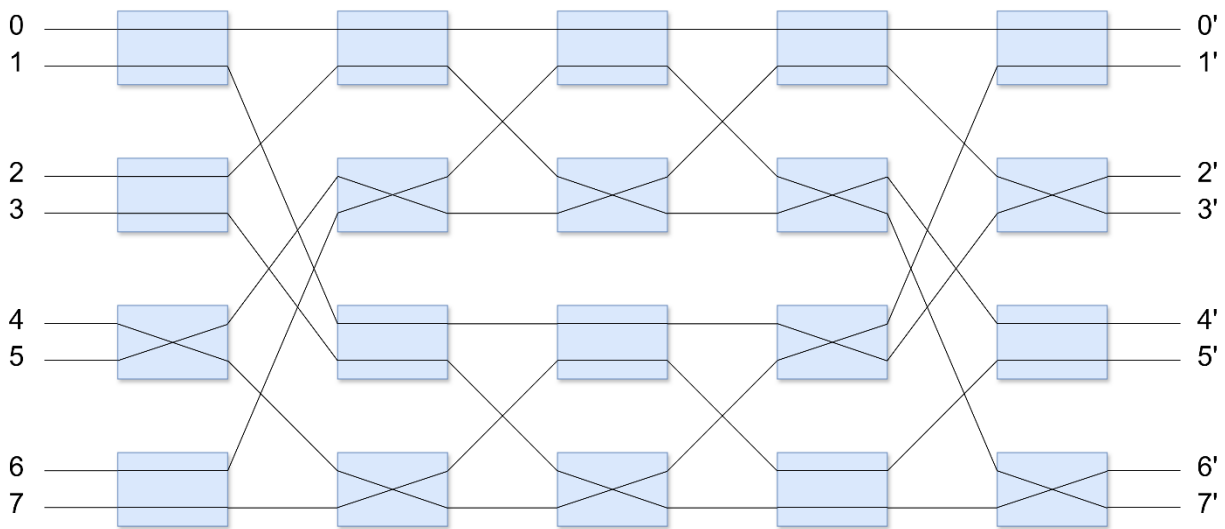


Figure 4: Fully routed 8×8 Benes Network.

2.2 Circuit architecture

The architecture replicates the routing by sorting the input-output port pairs of the routing table in a specific way which reflects the pathing of the algorithm. The position each pair takes on the sorted list shows the correct setting of the input and output port switches the pair belongs to. An example of sorting can be seen in Table 2.

Table 2: Sorted port pairs of the 8×8 Benes Network.

Left side port	Right side port	Subnetwork
0	0'	U
1	2'	L
5	3'	U
4	1'	L
2	4'	U
3	6'	L
6	7'	U
7	5'	L

The routing table (Table 1) is the source of this sorted matrix. The first pair starts from input port 0' (right-side) and steadily moves to different switches as dictated by the algorithm. Pairs in even positions (0,2,4..) are routed through the upper subnetwork and pairs in odd positions (1,3,5..) are routed through the lower subnetwork. If a port (left or right side) is even (0, 2, 4) and is routed through the upper subnetwork then the switch it belongs to must be set to bar, designated by 0 in the respective bit position of the output switch vector, If it is routed through the lower subnetwork the

switch it belongs to must be set to cross. The opposite is true for odd ports (1, 3, 5). Table 3 shows the routing according to this principle.

Table 3: Routing of the switches according to the sorted list.

Number of switch	Left side port state	Right side port state
0	bar	bar
1	bar	cross
2	cross	bar
3	bar	cross

The sorting produced by the architecture matches the correct setting provided by the algorithm in Figure 3. The architecture is comprised of blocks which execute this calculation for each layer of the network. The first block routes the switches on the outer layer and passes the subnetwork routing tables to the next layers. This process is repeated until the final vector containing all switch settings is produced.

The architecture was implemented for multiple sizes on the FPGA ZCU-104 and the results of implementation in terms of LUTs, FFs, Power, operating frequency, time needed for one complete setting can be seen on table 4.

Table 4: Implementation results

Size of Benes	8	16	32	64
LUTs	1416	3656	13549	35925
FlipFlops	561	1558	4002	9993
P (W)	0.036	0.105	0.241	0.437
Frequency (MHz)	250	250	166	125
Routing time(ns)	96	256	960	3072

At the time of writing it is the most efficient implementation of a Benes Network routing algorithm on hardware.

3. Bit Pattern Detection under heavy Interference

The second proposed architecture concerns the detection of very long bit patterns/sequences in the presence of high bit error percentages. It is a preliminary work whose results directly lead to the development of the third and final architecture focused on frame synchronization in digital links.

Long bit sequence/pattern detection is itself a very important function of modern systems and is employed in a plethora of applications such as packet header classification, network intrusion detection, image processing and text search and many others. It is essentially the recognition of

specific patterns in a series of input patterns given to the system. The specific patterns usually reside in the memory of the system and are long (>1000 bits). Due to their size the main method of detection in literature is the division of the main pattern into different sub-patterns which are then given to smaller processing units. These units use simple FSM (finite state machine) structures and execute simple logical operations to detect the smaller sub-patterns which gradually leads to the detection of the larger overall pattern. Even though this method speeds up the process of detection of long bit patterns and yields significant results it presents a significant drawback. It cannot detect patterns which are heavily altered by the presence of bit errors. The second proposed architecture is, at the time of writing, the only practically implementable method by which long big pattern detection under the presence of bit errors can be achieved at very high bit rates with low hardware cost.

Assuming that the specific pattern of length M bits being searched is the key pattern, the architecture compares the key to an input pattern of length M and determines whether the input pattern is the key even with bit errors. This is achieved by combining XNOR gates (equivalency gates) and adder trees. An adder tree is a basic circuit structure that is used for the summation of M elements optimally in $\log_2 M$ steps/clock cycles. It has $\log_2 M$ levels (same as cycles). In every level elements are added by two (leftover numbers proceed as is) and the result proceeds to the next level. With each level, the number of elements is halved and in the end the summation of all M elements is given as the output of the tree. Generally, the elements are multi-bit numbers, however in our case since we only deal with bit pattern detection, we assume that all elements are single bits (1 or 0, both the key and input are M bit patterns). This implies that the summation of all elements of an M -bit pattern is essentially the number of 1s inside the pattern. Using this property, it is very easy to determine how close the key pattern is to the input pattern in terms of bit distances i.e. how many bits are the same by applying the XNOR operation on the bits in respective positions of each pattern. This can be seen in table 5 for $M=10$.

Table 5: Digital correlation through XNOR gates.

key	1	1	0	0	1	0	0	0	0	1
input	1	0	0	1	1	1	0	1	0	0
XNOR	1	0	1	0	1	0	1	0	1	0
Sum	0101 (5)									

This results in M new bits which are the outputs of the XNOR gates. The value of summation of these M bits is the number of similar bits between the two patterns. High summation value indicates that the two patterns are similar, and low summation indicates the opposite. Depending on the circumstance, similarity above a certain threshold may mean that the input pattern is the key pattern

but altered by bit errors. This method of recognition under noise has the same complexity as the adder tree, $\log_2 M$ and the largest adder needed in terms of bits is $\log_2 M + 1$.

By itself, the tree adder method works well for moderate values of M however when M becomes very large the adders required by the tree also become very large. This leads to long critical paths and decreases maximum operating frequency and bit rate and eventually negatively impacts the latency of the circuit. To overcome this scaling problem, a new novel design must be developed which modifies the original adder tree in way that allows it to continuously grow in levels without needing long-bit adders.

The new approach combines adder trees and comparators and is as follow. If the correlation of two very long M bit patterns (key and input) must be calculated without the use of long adders, both patterns are broken down to Q symbols each with M/Q bits. The correlation between two respective sub-patterns/symbols can be determined with a smaller adder tree in $\log_2(M/Q)$ cycles with adders of size $\log_2(M/Q) + 1$. A comparator is placed at the summation output of each tree and compares the value of summation to a predetermined threshold. If the value surpasses the threshold the comparator outputs 1 if it does not the comparator outputs 0. This creates a pattern of Q bits where the number of 1s indicates how many sub-patterns/symbols display enough similarity to surpass the comparator thresholds. The new Q bit pattern can be broken down further into smaller sub-patterns/symbols whose summations are calculated with the same method. Eventually the single bit output of the final comparator indicates if the overall patterns have the desired similarity or not. The largest adders used are $\log_2(M/Q) + 1$ bits long and can be as short as desired by the user. The form of the overall circuit for two patterns of size $M=30$, $Q=6$ can be seen on Figure 5.

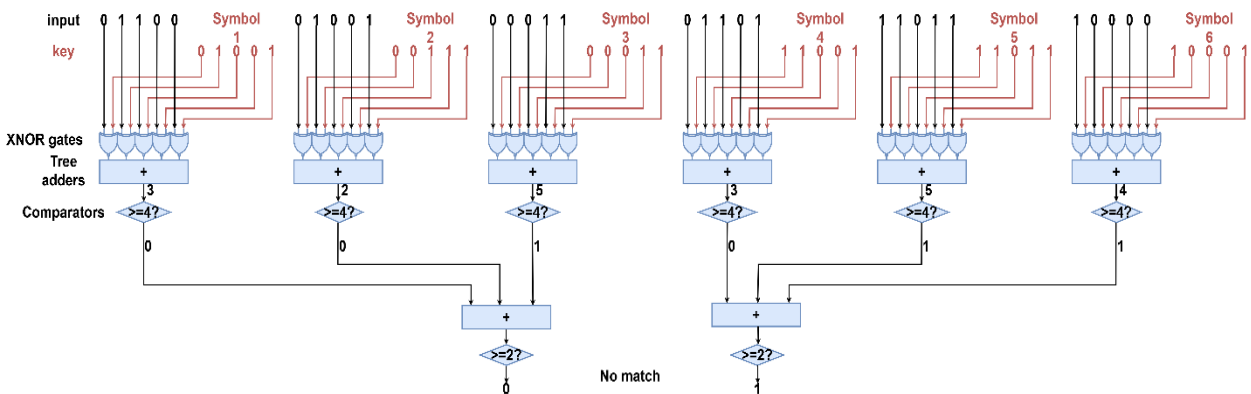


Figure 5: Bit pattern detection architecture for $M=30$, $Q=6$ and two levels of trees.

It is clear by the form of the circuit that the comparator thresholds determine the percentage and density of similar bits required for the circuit to declare the two patterns similar. This flexibility enables the architecture to detect very long sequences at very high bit rates even in the presence of many errors. In the context of the thesis the architecture was implemented in a parallel manner

(multiple units) for $M=1200$ bits broken down to 120 symbols of 10 bits each, then to 12 symbols of 10 bits each and finally to 1 bit (three levels) as part of a system with multiple memory entries and achieved detection accuracy of 10^{-4} with bit error rate close to 5×10^{-2} . One instance of the architecture can be implemented only with 3051 LUTs and 5531 FFs and has an operating frequency of 667MHz. The platform of implementation was the XILINX ZCU106 Ultrascale+.

4. Frame synchronization architecture

The most important implication of the bit pattern detection architecture is that one can perform pattern detection with few resources accurately and detect very long patterns even under significant interference. The immediate application of this result is in the domain of frame synchronization. In digital communications, frame synchronization is the ability of a receiving system to accurately deduce the starting bit of an incoming frame. Many different methods to perform this task exist such as machine learning, joint error correction and frame detection where error codes are used to both correct the error bits and detect the starting bit of the frame. The most basic method of frame synchronization is sync word-based correlation detection where sync words are placed inside the frame and correlation is used to detect their positions. Massey's max likelihood rule further expands this method and proves that for sync word-based synchronization the optimal metric is the correlation metric minus an energy term. The max likelihood rule is staple in frame synchronization and has been expanded upon with different variations focusing on different circumstances.

The third proposed architecture forgoes the optimality of the max likelihood rule to perform digital correlation with long bit sequences and determine the first bit of the frame through the digital correlation metric alone. Even though it is mathematically suboptimal the minimalistic nature of the architecture presented allows the user to use very long sequences without restrictive hardware consumption which leads to much higher detection accuracy than all other methods reported in literature. The architecture not only detects the starting bit of the frame, but it also captures the frame data payload i.e provides it as output.

The architecture consists of three modules. Isolation window, parallel correlation and frame capture. The isolation window module isolates a part of the incoming demodulated received bitstream, the parallel correlation module correlates the isolated part with a predetermined n -bit long sync word placed in front of the frame and the frame capture module accepts the outputs of the correlation module, determines the position of the frame and captures the frame data. It is assumed that the demodulated incoming bitstream is given to the architecture in blocks of n parallel bits (same size as the sync word) and the sync word is encapsulated between two k sized smaller sideguard sequences. These sequences are not part of the sync word and are not detected. They only exist to lower the

correlation values near the sync word so that the architecture can detect the frame more smoothly. Figure 6 presents the form of the bitstream with the sync words, frames and sideguards divided into blocks of n bits. Time flows from right to left.

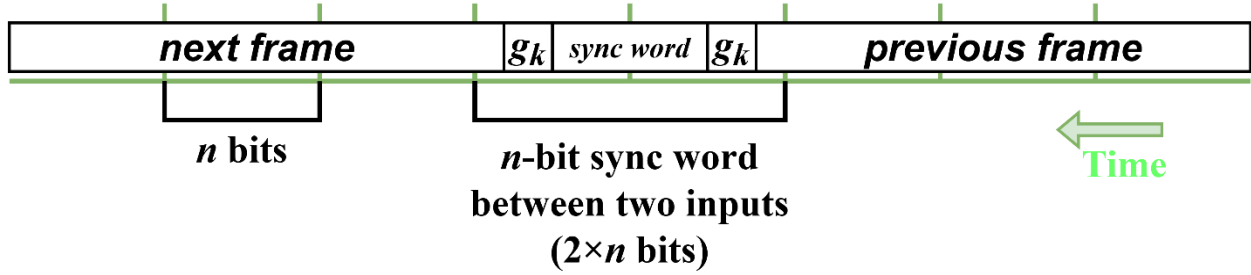


Figure 6: How time flows on the bitstream. The sync word is encapsulated between two sync words.

4.1 Isolation window module

The isolation window module is a register of size $2 \times n$ that consists of two slots of n bits, $2 \times n - 1$ down to n and $n - 1$ down to 0 . Slot $2 \times n - 1$ down to n is the left slot and slot $n - 1$ down to 0 is the right slot, meaning that the indexing goes from right to left following the time-flow of the bitstream. Input of size n enters the left slot of the buffer in clock cycle t and moves to the right slot of the register in the next cycle $t + 1$. Because the indexing of the register and the time-flow of the bitstream are the same the overall bitstream maintains its form inside the isolation window register. Moreover, since the sync word is the same size as the parallel input it will appear in its entirety inside the register with its first bit in positions $n - 1$ down to 0 . Figure 7 presents the circuit of the isolation window module/register.

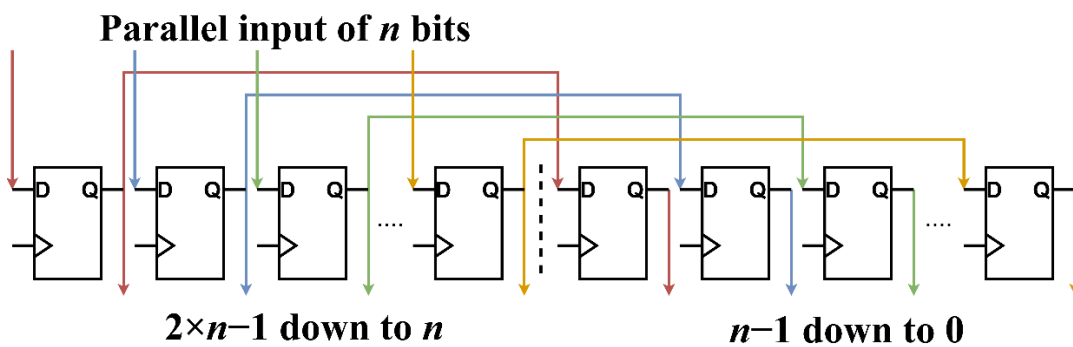


Figure 7: Circuit of the isolation register

The contents of the register are given as input to the parallel correlation module.

4.2 Parallel correlation module

The parallel correlation module correlates the predetermined sync word with the respective bits in every possible position it can occupy inside the isolation window register. Since there are n parallel bits given as input the starting bit of the sync word can be any of these bits. This means that the possible positions for the sync word are $n-1$ down to 0, n down to 1, $n+1$ down to 1 $2 \times n - 2$ down to $n-1$. $2 \times n$ down to n is the same as $n-1$ down to 0 and thus not checked. The correlation module correlates the sync word with the bit patterns in each position using n parallel adder trees and n^2 XNOR gates. All calculations are executed in parallel and the n correlation values produced by the n adder trees enter a comparator tree. The comparator tree determines the highest value out of all n values along with its index which is the index of the starting bit of the sync word inside the left half ($n-1$ down to 0) of isolation register. Comparator trees are circuit elements similar to adder trees but instead of adders they have comparators. They produce the highest or lowest value of the set. With simple modifications they can provide not only the value but also the index. They have the same latency as adder trees. Together the parallel adder trees and comparator tree have a combined latency of $\log_2 n + \log_2 n = 2 \times \log_2 n$. For this reason, the parallel correlation module features a delay register which synchronizes its input with the corresponding output it produces. The delay register consists of $2 \times \log_2 n$ slots of n bits and causes delay equal to the latency of adder and comparator trees. Data enters the last slot ($2 \times \log_2 n - 1$) after the window register and every cycle moves closer to the first slot (0). The first two slots (slot 0 and slot 1) contain the input to the parallel correlation module as it was $2 \times \log_2 n$ clock cycles before and it is the input which produces the correlation output of the same clock cycle. The maximum correlation value, its index and the contents of the first 3 slots of the delay register are given to the frame capture module to enable frame capture. Figure 8 presents the delay register and figure 9 presents the parallel correlation module apart from the delay register (XNOR gates, adder trees and comparator tree) for $n=8$.

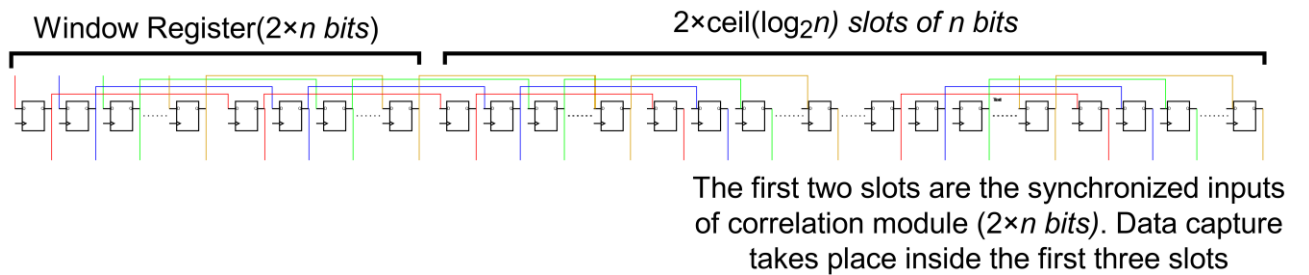


Figure 8: The Delay Register. It consists of $2 \times \text{ceil}(\log_2 n)$ slots of n bits and it comes right after the window register. Data moves to the right as block of n bits. It synchronizes the contents of the Window Register with the outputs of the correlation module.

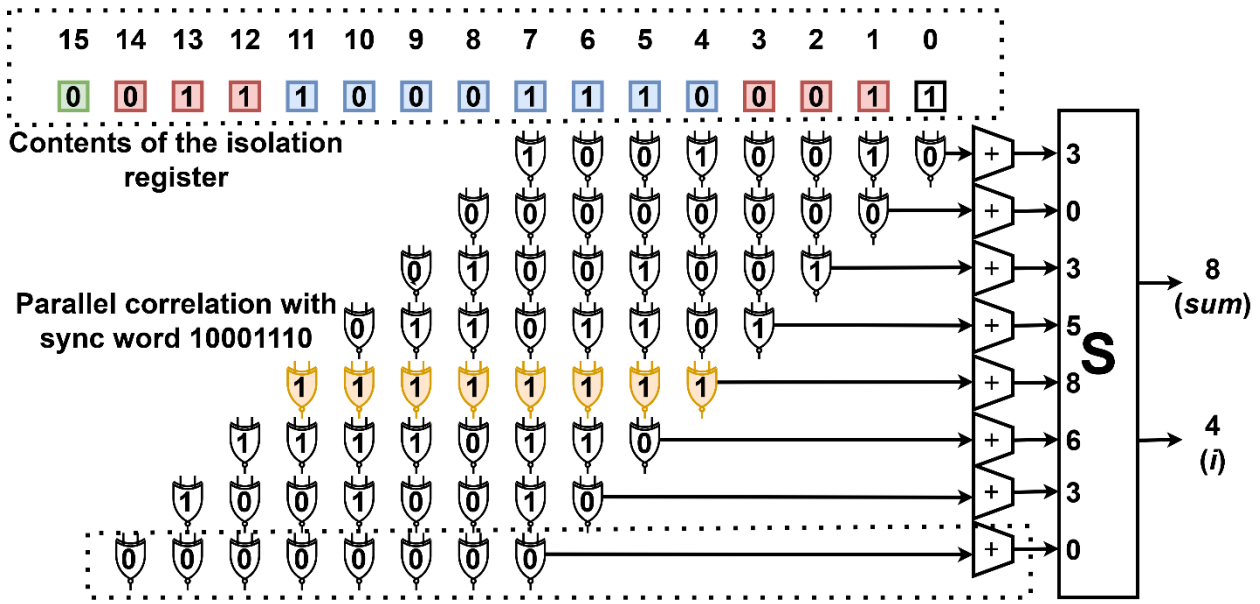


Figure 9: Circuit of parallel correlation minus the delay register when $n=8$. The sync word being searched is 10001110. Circuit S denotes the comparator tree. Positions 15 down to 8 are not checked as their contents are transferred to 7 down to 0 in the next cycle.

4.3 Frame capture module

The frame capture module uses the outputs of parallel correlation (maximum value, position, delayed input) to capture the frame i.e. send the frame data to the output of the architecture. It also provides a valid out signals which signifies validity of the output. To capture the data, it compares the maximum value to a predetermined threshold. When the threshold is surpassed, it assumes that a frame is detected and begins the capture process. The starting bit of the frame data is always at position $k+n+i$ counting from the start of the delay line, where k is the size of the sideguard, n is the size of the sync word and i is the index of the maximum value (first bit of sync word). Data is captured in the first three slots of the buffer (bits $k+n+i+n-1$ down to $k+n+i$) for the next clock cycles depending on the size of the frame. Figure 10 shows the starting bit of the data and frame inside slot 0 of the delay register.

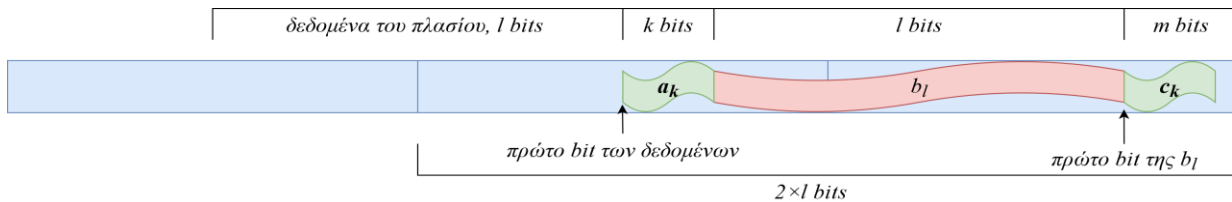


Figure 10: Position of frame data inside the first three slots of the delay register.

4.4 Results

The architecture was implemented for multiple sizes of sync word on the low cost NEXYS 4 DDR FPGA with the Xilinx Vivado tool. The implementation results in terms of LUTs, FlipFlops (FF), Power and Bit Rate can be seen on Table 6.

Sync word size(n)	LUTs (%)	FlipFlops (%)	P(W)	Bit Rate(Gbps)
123	20027 (32)	23608 (18)	0.842	15.375
112	16393 (26)	19530 (15)	0.728	14
99	13657 (22)	15817 (12)	0.617	12.375
89	11136 (18)	13779 (11)	0.529	11.125
75	8112 (13)	10253 (8)	0.413	9.375

For the simulations, the input bitstream was created in MATLAB, modulated with 16QAM, driven through the AWGN channel for different integer SNR values and given to the architecture through a VHDL testbench. The accuracy results are presented in Figure 11 for all sync word size and snr integer values. The side-guard sizes (k) and frame capture thresholds(thr) for every sync word size (n) are the following ($n=123, k=23, thr=93$), ($112, 21, 84$), ($99, 17, 76$), ($89, 19, 70$), ($75, 16, 59$).

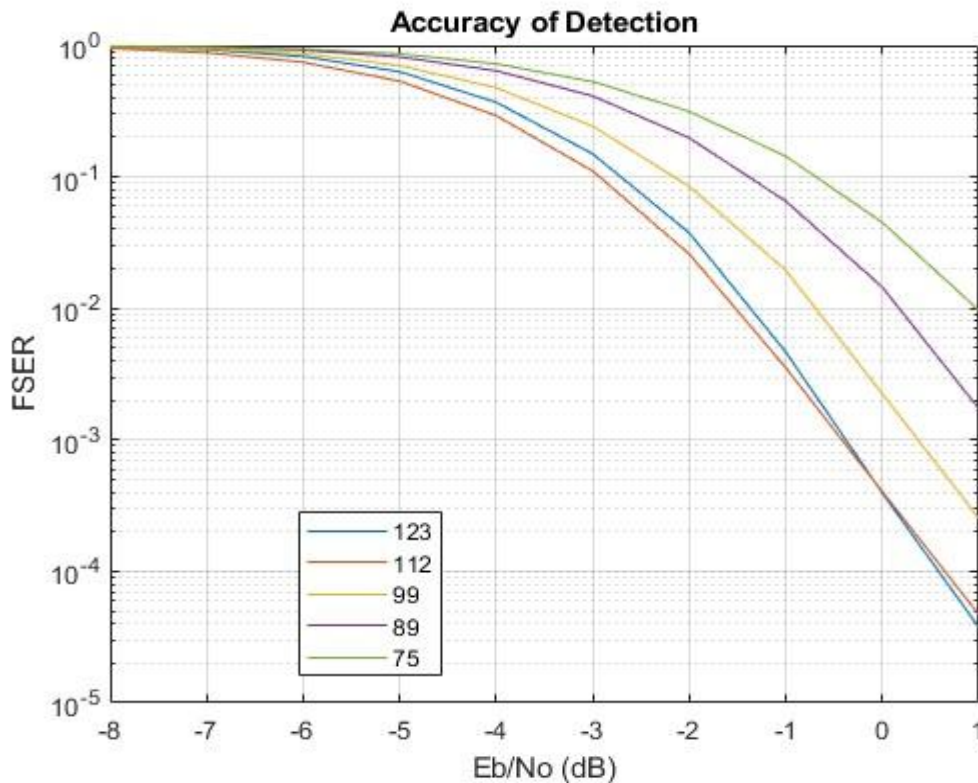


Figure 1: Synchronization accuracy results of the architecture.

Overall, the synchronization accuracy is much higher than other schemes reaching accuracy 10^{-10} times better than in some cases while providing much higher bit rates even when they are implemented on ASIC platform which are superior to FPGA in terms of both hardware consumption and operating frequency. Finally, the architecture can also be used as a pseudo encryption mechanism as the sync word is random, known only to the receiver and transmitter and thus impossible to intercept by third parties.