2011

Algebraic Models of Constant Node Degree Interconnection Networks

Khadidja Bendjilali
Lehigh University

ALGEBRAIC MODELS OF CONSTANT NODE DEGREE INTERCONNECTION NETWORKS

by
Khadidja Bendjilali

A Dissertation
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy
in
Computer Engineering

Lehigh University
January 2012

This dissertation is accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

(Date)

(Accepted Date)

Professor Meghanad Wagh

Professor Bruce Dodson

Professor Tiffany Jing Li

Professor Zhiyuan Yan

Acknowledgements

" In The Name of Allah Most Gracious Most Merciful"

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Meghanand Wagh for all the hope he has put on me, for his continuous guidance and endless patience and support. Professor Wagh has been a great advisor, I will forever be thankful to him. I would like also to thank the members of my dissertation committee, Professor Tiffany Jing Li, Professor Bruce Dodson, and Professor Zhiyuan Yan for their helpful comments and advices during the revision of my dissertation draft. I am gifted to have been raised in a family where education is highly valued and praised. I would like to thank my great parents, Dr. Boualem Bendjilali and Lila Belkacem, to the most special person in my life Dr. Fethi Belkhouche and every member of my family for their love, endless support and encouragement. Without their continuous support and love I would not have been able to accomplish this work.

Contents

List of Tables

List of Figures

Abstract

Binary representation has been widely used to model many common interconnection networks such as the Butterflies (BF), Cube Connected Cycles (CCC), Shuffle Exchange (SE), Hypercubes and deBruijn (DB) networks. However, binary models are difficult to analyze and complex to use, except for a few select ones such as the Hypercubes. In this research we exploit new algebraic representations for BF, CCC, SE and DB networks. While algebraic models for BF and DB are available in the literature, this dissertation provides algebraic models for CCC and SE for the first time. The simplicity of the models and access to powerful algebraic techniques allows us to explore the structural properties of these networks. In particular, we have found all the automorphisms of BF and CCC networks and the effect of these automorphisms on graph edges. This has allowed us to provide strategies to map algorithms on networks with faulty edges, which is an important problem in parallel processing. We illustrate our methods by mapping Hamilton cycle on the butterfly under various edge fault scenarios. This dissertation also exploits the algebraic machinery to find paths in SE and optimal paths in the CCC networks.

Chapter 1

Introduction

1.1 Introduction

Over the last few decades, the semiconductor technology has delivered increasingly faster and complex and yet smaller integrated circuits. Unfortunately, this ability to create chips of shrinking sizes and higher complexities has now hit the technological barriers. On the other hand, applications are becoming increasingly complex and need faster solutions. The only way to solve these highly dynamic and complex problems is by using parallel computing paradigm. It is therefore an accepted premise that parallel processing using a larger number of processors will be the future of computing.

While somewhat specialized parallel machines based on SIMD (single-instruction-multiple-data) and shared memory have also been designed, the most common parallel architecture is the distributed memory parallel machine. In such a machine, multiple processors work independently on different parts of the application using their own program and data memories. They exchange data and partial results with each other using communication links between them. These links, together, form an interconnection network of the machine. Unfortunately, the communication speeds have not kept up with the computational speeds. As a result, the performance of message passing parallel architectures and multi-core chips depends, to a large extent, on the underlying interconnection network.

The choice of the interconnection network also affects other key characteristics of the system such as the ease of algorithm development, overall speed, reliability, scalability and complexity of physical layout. Therefore communication network of a parallel processor dominates its performance.

Interconnection networks can be modeled as graphs whose nodes represent processors, and edges, the communication paths between them. Hypercubes, butterflies, cube connected cycles and meshes are some of the popular graphs on which many of the existing parallel machines are based [1]. To run a computation on a parallel machine, one partitions the task into a set of sub-tasks and develops a task graph. One then maps the task graph on the interconnection graph of the architecture such that the communicating sub-tasks are mapped (as far as possible) on processors that have a direct link between them. Graph theory is one of the most powerful mathematical tools for designing and analyzing interconnection networks. Selecting an appropriate topological structure of an interconnection network is a major problem in designing distributed memory parallel machine.

Interconnection networks are characterized by the following parameters:

- *Network size:* the number of nodes in the network. Large network size is desirable since it implies more processors and hence higher throughput.

- *Node degree:* the number of communication links connected to a node. The degree of a node in interconnection network directly determines the complexity of communication hardware within that processor node. Clearly a small node degree is desirable. In addition, if the node degree is constant with respect to the size of the network then the same node hardware may be employed to build networks of different sizes. Thus constant node degree is important for scalability and economical of parallel computing.

- *Bisection width:* The minimum number of links which need to be cut in order to divide the network into two equal halves. A large bisection width is desirable since the bisection width limits the rate of data transfer between the two halves of the network, thus affecting the performance of communication.

- *Network diameter:* The maximum shortest path between any two nodes in the network. The shortest path is the minimum number of links which must be traversed in order to connect two nodes in the network. The network diameter is the longest of such shortest paths between *any pair of nodes* and therefor represents the maximum number of links that a message may have to traverse before it reaches its destination. A smaller diameter implies a shorter time for message communication and results into a higher speed of algorithm execution.

- *Existence of mappings of parallel algorithms:* Mapping algorithms on architectures is an important requirement of an interconnection network. To reduce the communication overhead, a good match is necessary between the structure of parallel algorithm represented by the guest graph (in which the nodes represent subtasks and edges, communication between subtasks), and the network topology represented by the host graph. There are several desirable properties such as dilation, congestion, loading, etc. to estimate the quality of a mapping.

- *Symmetry:* An interconnection network is symmetric if it looks the same from any node. Symmetry allows simpler algorithm mappings, easier communication strategies, task remapping and fault avoidance.

We now review some common interconnection networks in the light of the desired properties stated in this section.

1.2 Some promising networks

This section reviews some of the common interconnection networks in the light of the desired properties.

The simplest interconnection network is a *Ring* of N nodes [1]. Each node in the ring is connected to only two other nodes in a cycle with diameter of $N/2$.

It has constant node degree of 2 and a constant bisection of width of 2. Ring network contains several attractive properties such as simplicity, extensibility and symmetry. Ring networks are suitable for implementing simple algorithms with low communication costs. However, mapping most guest graphs onto Ring is difficult because of the simple architecture and the large diameter of this architecture.

A two-dimensional *Torus network* is defined to have $n \times n = N$ nodes, formed by an $n \times n$ array of nodes with each connected to its immediate neighbor in the row and column, including the wrap around edges. Torus is one of the attractive interconnection networks due to its symmetry and application to solving finite element problems. It has a constant node degree of 4 and a diameter of $O(\sqrt{N})$ [1].

Many networks use labels that are binary strings. We therefore explain the notation first. Let Z_n denote the group of integers $\{0, 1, \ldots, n-1\}$ under the operation of addition modulo n and Z_2^n, the group of binary vectors of length n under the operation of modulo 2 addition.

A topology that has been used in several parallel machines is the Hypercube. The n-dimensional hypercube H_n is defined to have 2^n nodes labeled with elements of Z_2^n [2,1]. Two nodes are connected with an edge if and only if their labels differ in exactly one bit. Hypercube H_n can be constructed from two H_{n-1} hypercubes by connecting the corresponding nodes. This hierarchical property of a hypercube simplifies the development of communication strategies as well as the mappings of a large number of parallel algorithms on the hypercube. Hypercube H_n also has many other nice properties such as a small diameter n and a large bisection width 2^{n-1}. The non-constant node degree, however gives the hypercube poor scalability. A network with a large number of nodes becomes very complex, which is reflected in the network cost.

The n-dimensional Shuffle Exchange graph, SE_n, has 2^n nodes, each labeled with an element of Z_2^n. A node $(v_1 v_2 \ldots v_n)$ is connected to three distinct nodes: $(v_1 v_2 \ldots v_n \oplus 2^n)$ (an *exchange edge*), $(v_2 v_3 \ldots v_n v_1)$ and $(v_n v_1 \ldots v_{n-1})$ (*shuffle edges*) [3]. SE_n has $3 \times 2^{n-1}$ edges and a diameter of $2n-1$. Its node degree is ≤ 3.

The n-dimensional deBruijn graph, DB_n was introduced to overcome the Hypercube disadvantage that the degree grows as the size of the network increases. DB_n

has 2^n nodes, each labeled with an element of Z_2^n. A node $(v_1 v_2 \dots v_n) \in DB_n$ is connected to $(v_2 v_3 \dots v_n\ 0)$, $(v_2 v_3 \dots v_n\ 1)$, $(0\ v_1 v_2 \dots v_{n-1})$ and $(1\ v_1 v_2 \dots v_{n-1})$ [4]. DB_n has 2^{n-1} edges and a diameter of n. Its node degree is ≤ 4. Shuffle Exchange, and deBruijn, though non-symmetric, are still considered important because of their small node degree and small diameter.

The Cube Connected Cycles network of degree n, CCC$_n$ was developed as a hypercube derivative by replacing each node of a degree n hypercube by a cycle of n nodes [5]. CCC$_n$ has $n2^n$ nodes, each labeled by a pair (m, V) where $m \in Z_n$, and $V \in Z_2^n$. A node (m, V) of CCC is connected to only three other nodes: $(m+1, V)$, $(m-1, V)$ and $(m, V \oplus 2^m)$ as shown in Fig. 5.1, where $V \oplus 2^m$ is the string V with mth bit complemented. The diameter of CCC is 6 when $n = 3$ and $2n + \lfloor n/2 \rfloor - 2$ when $n > 3$ [6]. This low diameter and the low constant node degree imply that CCC may be very useful for parallel architectures.

The Wrapped Butterfly graph BF$_n$, $n \geq 3$, is defined to have $n2^n$ nodes, each labeled with a pair (m, V) where $m \in Z_n$ and $V \in Z_2^n$ [1]. A node (m, V) is connected to four distinct nodes: $(m+1, V)$, $(m+1, V \oplus 2^m)$, $(m-1, V)$ and $(m-1, V \oplus 2^{m-1})$ as shown in Fig. 2.3. BF$_n$ represents a good trade-off between the cost and the performance of a parallel machine. It has a large number of nodes ($n2^n$), fixed node degree (4), low diameter ($\lfloor 3n/2 \rfloor$), symmetry, and ability to support a variety of parallel algorithms [1, 7–10].

Table 1.1 summarizes the topological properties of the various interconnection networks mentioned in this section.

This dissertation focuses on constant node degree networks since these are the only *trully* scalable networks from hardware perspective. However, as described above, other properties such as a small diameter and the availabilty of mappings of common parallel algorithms are crutial to the performance of such networks. The prominent networks that have a constant node degree include the Ring, the Torus, the Tree, the Shuffle Exchange network, the deBruijn network, the Cube Connected Cycles and the Wrapped Around Butterfly. Of these, the Ring and the Torus have a substantially large diameter, $O(N)$ and $O(\sqrt{N})$ respectively, where N is the number

Table 1.1: Comparison of Hypercube (H_n), Ring (R_n), Torus ($T_{n,n}$), Shuffle Exchange (SE_n), deBruijn (DB_n), Cube Connected Cycles (CCC_n), and Butterfly (B_n) interconnection networks

Network name	No. of proc	Dia	Node degree	Bisection width	Symmetry
H_n	2^n	n	n	2^{n-1}	Yes
R_n	n	$n/2$	2	2	Yes
$T_{n,n}$	n^2	n	4	$2n$	Yes
SE_n	2^n	$2n-1$	≤ 3	$n/log\ n$	No
DB_n	2^n	n	≤ 4	$2^n/n$	No
CCC_n	$n2^n$	$2n$	3	2^{n-1}	Yes
B_n	$n2^n$	$\lfloor 3n/2 \rfloor$	4	2^n	Yes

of nodes. Thus they do not really scale well in performance. The Tree, Shuffle Exchange and the deBruijn networks do not possess symmetry, an important property that is required for fault tolerance as well as to simplify algorithm mappings. Thus neither of these three networks are scalable in this context.

The two scalable networks that have symmetry, constant node degree and acceptable diameter, $O(\log N)$, are thus the Cube Connected Cycles and the Wrapped Butterflies. Our research focuses on these two architectures.

1.3 Existing algebraic models for interconnection networks

This section reviews the existing algebraic models for constant node degree interconnection networks. The objective is to provide the reader with a concise introduction to these models.

The first general framework for constant node degree interconnection networks was introduced by Akers and Krishnamurthy [2, 11]. They showed that by using Cayley graphs, one can obtain constant node-degree interconnection networks, all of which are *symmetric*. Given a group G and a subset S called the generator set, an interconnection network can be defined as a Cayley Graph, in which each vertex is labeled by an element of G and two vertices $u, v \in G$ are connected if and only if there exists an element $s \in S$, such that $v = us$. To ensure bidirectionality of edges and to avoid self loops, S should be closed under inverses and should not contain identity element. The node degree of this graph is the number of elements in the subset S.

All Cayley graphs are node symmetric. All the common symmetric networks such as hypercubes, rings, toruses, butterflies can be constructed as Cayley graphs using appropriately defined groups and generator sets. Further, this concept has also been used to develop new interconnection networks with attractive properties, e. g., star graphs and pancake graphs [2,12,13]. As an illustration, Hypercube H_n of dimension n can be constructed as a Cayley graph of the group Z_2^n under the binary operation of bit-wise XOR, and the generator set S to be the set of n binary vectors of weight 1.

Cayley Graphs, for the first time, offered a unified view of symmetric interconnection networks. Unfortunately, even though they provide new models for some older networks and a procedure to generate new networks, they do not simplify the investigation of either topological (other than the symmetry and node degree) or mapping properties of these networks.

Cayley graphs cannot model non-symmetric networks such as Shuffle Exchange and deBruijn graphs which have important topological properties (refer to Table 1.1). An extension of the Cayley graph concept known as the Cayley Coset graphs was therefore developed by Annexstein et al. [11]. Given a group G and its subgroup H, and a generator subset S as in the case of a Cayley graph, a Cayley coset graph with $|G|/|H|$ nodes may be defined as follows. Every node of the graph is labeled by a (say) left coset of H in G. Node Hx is connected to node Hy if and only if there exists an $s \in S$ such that $x_1 s = y_1$, where $x_1 \in Hx$ and $y_1 \in Hy$. If the subset H is trivial, i.e., if $H = \{e\}$ where e is the identity element of G, then the Cayley Coset Graph is the same as the Cayley graph obtained from G and the set of generator S. Thus, Cayley coset graphs can be used to represent both symmetric and non-symmetric graphs. Unfortunately this model also suffers from the same drawback as the Cayley graphs, namely that it can be used effectively to design new interconnection networks but it does not simplify the investigation of the existing networks. In order to benefit from the model one needs to make the model powerful enough to explore all the structural properties.

The third kind of algebraic model used to describe interconnection networks uses finite fields. Since a field admits both addition and multiplication, this model often is more powerful. Currently models that use finite fields are available for the deBruijn networks [14, 15] and for the Wrapped Butterfly networks [10]. In these models, the graph nodes are expressed using elements of finite fields (for the deBruijn network) or elements of a direct product of groups and finite fields (for Wrapped Butterflies). The connectivity between nodes can then be expressed as a simple algebraic relationship between the node labels. This allows one to explore the structural properties of these networks in much more direct fashion using powerful algebraic techniques. Further, such a model allows one to map algorithms to parallel machines rather easily [10].

Binary representation has been used to model common interconnection networks such as Wrapped Butterfly, Hypercube, Cube Connected Cycles, deBruijn, and Shuffle Exchange. Unfortunately, the binary model is quite difficult to analyze except for a few selected ones such as Hypercube. For example, in a Wrapped Butterfly network, the destination of (m, V) is $(m+1, V \oplus 2^m)$ the complexity of this representation should be apparent from the fact that the second coordinate of the destination is a function of both V and m, the two coordinates of the source. Because of this, one cannot treat the two coordinates independently making it very difficult to explore the topological properties of the network.

With the advances in the VLSI technology, it is now possible to build parallel machines with a large number of processors. However, larger the machine, higher is the probability that one or more of its processors and links will develop a fault. Thus, for the underlying networks of these large machines, mapping of algorithms on faulty graphs becomes an important issue [16–20]. Unfortunately, earlier work with these models did not explore mappings on networks with faults. Finite field based models are powerful enough to yield results in this domain as well. In particular, these models may yield easy expression of such powerful concepts as the automorphisms of the graphs. Since automorphism is a natural way to remap an algorithm on a faulty network, it represents an important direction for exploration.

This research develops a new approach to mappings on faulty butterflies using an algebraic model first given in [10]. We show that with this model, it is rather simple to obtain all the automorphisms of the butterfly. Automorphisms can be used to translate an algorithm mapping to one that avoids node faults. For example, an algorithm mapping can avoid a faulty node N_{faulty} by using a free node N_{free} (assuming one exists) and an automorphism $\phi(\cdot)$ of the interconnection graph such that $\phi(N_{free}) = N_{faulty}$. By remapping tasks on each node N to node $\phi(N)$, one can run the algorithm entirely on fault free nodes. Automorphisms have also been used to obtain better VLSI layouts of butterfly networks [21, 22].

This research obtains all the automorphisms of the degree n wrapped butterfly BF_n (Theorems 1, 4, 6). We explore the edge transformations in butterfly networks due to automorphisms. In particular, we show that exactly $n2^n$ automorphisms of BF_n affect all the edges in a column similarly (Theorem 7). As an example, we show that a butterfly BF_n supports a Hamilton cycle even when it has up to 2^n faulty edges of the same type (to be defined later) in each column except one (Theorem 11). Hamiltonian cycle provide an optimal all to all broadcast in architectures that have processor constrained communication (a processor can receive only one message at any time). As a corollary, one can show that BF_n is Hamiltonian with up to $n - 1$ random edge faults distributed one per column. The remaining $n2^n$ automorphisms change the type of exactly half the edges in a column while the other edges retain their type (Theorems 9, 10). Further, one can design automorphisms to achieve the desired edge transformation. This allows one to map algorithms onto butterfly machines with edge faults. As examples, we show that a Butterfly BF_n supports a Hamilton cycle even when it has faulty edges in all but two of its rows as long as the faults in a given set of rows are constrained to one type and those outside to one type as well (Theorems 13, 14). Further, the requirement of two fault-free rows can be lifted when n is odd (Theorems 15, 16). Our procedure allows one to map the Hamilton cycle on to the faulty butterfly easily and directly. The simplicity of the automorphism and the resultant edge mappings show promise of wide applicability of this technique to a variety of applications.

Binary models of Cube Connected Cycles (CCC$_n$) suffer from similar defects. The connectivity of CCC$_n$ using the binary model is much too complex to obtain many of the useful properties of the network. In this research, we propose a new model for CCC$_n$ based upon the direct product of a cyclic group and a finite field. With this new model, one can avail of powerful algebraic techniques to investigate the structure and mappings of CCC$_n$. Similar algebraic models developed previously for the deBruijn network [15] and the wrapped butterflies [10] have allowed efficient mappings of cycles and trees on the butterflies and provided insights into intricate structural properties such as the automorphisms [23,24]. This new model helps solve similar problems in CCC$_n$. Besides proving this model, this reserach demonstrates its use to obtain paths in CCC$_n$. We also provide a similar new model of SE$_n$ and explore the relationship between the Shuffle Exchange and the deBruijn networks.

Finally, this research also obtains all the automorphisms of the Cube Connected Cycles and explores the edge transformations in CCC$_n$ networks due to automorphisms.

1.4 Organization of the dissertation

The rest of this dissertation is organized as follows. The necessary mathematical background required for the rest of the dissertation is presented in the next Chapter, where we briefly review some basic results in finite fields. Chapter 2 also reviews the algebraic model of the Wrapped Butterfly network using direct product of finite fields and cyclic groups. Chapter 3 obtains all the automorphisms of a Wrapped Butterfly network. It also investigates the translation of Butterfly edges by automorphisms, and proposes a new strategy for algorithm mappings on an architecture with faulty edges. The new algebraic model of the Shuffle Exchange is defined and proved in Chapter 4. We prove that the Shuffle Exchange network is a subgraph of the deBruijn network of the same size. In Chapter 5 the new algebraic model of the Cube Connected Cycles network is introduced and proved. We then provide all the automorphisms of the Cube Connected Cycles network. We also investigate the effect of automorphisms on the Cube Connected Cycles edges. Finally, Chapter 6 summarizes the most important results of this work.

Chapter 2

Mathematical Preliminaries and Algebraic Models

This chapter presents the mathematical framework on finite fields and their properties to be used in the subsequent chapters. The definitions and background provided in this chapter will be used to treat the interconnection networks; in particular the *Wrapped Butterfly* (B_n) and the *Cube Connected Cycles* (CCC_n). This chapter is organized in these sections. Section 2.1 presents the main definitions concerning finite fields. Section 2.2 discusses the dual of polynomial basis of finite fields. Section 2.3 discusses the automorphism definition. An overview of the new representation of B_n and its isomorphism to the binary node labels is provided in Section 2.4. Section 2.5 is devoted to mapping of cycles of possible lengths to B_n, then provide simple procedures to merge cycles under this model. Section 2.6 provides an overview of the algebraic model of the deBruijn network.

2.1 Finite Fields

In this section, we discuss the main definitions and properties of finite fields and groups that will be used in this research. For more extensive coverage of the topic, reader is referred to [25, 10]. Group is one of the important structures we employ.

It is defined as follows.

Definition 1 *[26] A set G with a binary operation $\otimes$ is called a group iff*

1. *G is closed under $\otimes$.*
2. *$\otimes$ is associative, i.e., $a \otimes(b \otimes c) = (a \otimes b) \otimes c$ for any $a, b, c \in G$.*
3. *There is an identity element $e \in G$ such that $a \otimes e = e \otimes a = a$ for all $a \in G$.*
4. *For each $a \in G$, there exists $a, b \in G$ such that $a \otimes b = b \otimes a = e$.*

A group in which $a \otimes b = b \otimes a$ for all $a, b \in G$ is called commutative group.

Now we define a field structure.

Definition 2 *A set F with binary operations $\oplus$ (generally called addition) and $\otimes$ (generally called multiplication) is called a field iff*

1. *F is a commutative group under $\oplus$. Let e denote its identity.*
2. *Set $F - \{e\}$ is a commutative group under $\otimes$.*
3. *$\otimes$ is distributes over $\oplus$, i.e., $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$ for any $a, b, c \in F$.*

Definition 3 *A finite field also called (Galois field) is a field that contains finite number of elements.*

Finite fields are algebraic structures that support the four basic operations: addition, subtraction, multiplication, and division. They do share many of the properties of the other fields such as the fields of real numbers or complex numbers. The only difference is that finite fields have a finite number of elements unlike the fields of real or complex numbers. Finite fields are important in algebraic geometry, cryptography, Galois theory and coding theory among others. It is known that a finite field has exactly p^n elements where p is a prime and n is a positive integer. Further, there is only one finite field (within isomorphism) of any size. A Galois Field with p^n elements is denoted by $GF(p^n)$.

We restrict ourselves to $p = 2$, i.e., finite fields $GF(2^n)$. However, to construct $GF(2^n)$, we first build $GF(2)$, a field that has only two elements 0 and 1. The addition and multiplication operations on $GF(2)$ are defined as addition and multiplication modulo 2. This implies that $X + X = 0$, for any $X \in GF(2)$. Thus $X = -X$ in $GF(2)$.

To enlarge the field of two elements to a field of 2^n elements, one uses a primitive polynomial $p(x)$ of degree n over $GF(2)$. A primitive polynomial over $GF(2)$ has coefficients in $GF(2)$, but has no roots in that field. One can extend $GF(2)$ by including a root of $p(x)$, say α. Since one is constructing a structure in which multiplication is a valid operation, one would expect products of these elements, $\alpha^2, \alpha^3, \alpha^4, \ldots$ to be also valid elements of the new field. However, all these powers cannot be distinct. Recall that α is the root of a polynomial of degree n. Thus, α^n can be expressed in terms of the lower powers of α. Thus any α^i can be expressed as of $\sum_{j=0}^{n-1} a_j \alpha^i$, $a_j \in GF(2)$. One can thus show that the number of distinct powers of α is at most $2^n - 1$. If $p(x)$ is indeed primitive, then all these $2^n - 1$ powers of α are distinct. In fact, $\alpha^{2^n-1} = 1$ and thus, the elements of $GF(2^n)$ are closed under multiplication. Therefore, any of these distinct α powers may be multiplied and the result would still be an element within the finite set of α powers. The finite field thus generated has 2^n elements comprising of 0 and the $2^n - 1$ powers of α enumerated as $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^n-2}\}$.

Fields $GF(2^3)$ and $GF(2^4)$ are illustrated below in Table 2.1 and Table 2.2, respectively. Expressing each element of $GF(2^n)$ in basis $\langle \alpha^{n-1}, \alpha^{n-2}, \ldots, \alpha, 1 \rangle$ is fairly straightforward. For example, in Table 2.2, elements 1, α, α^2 and α^3 are already the basis elements. α^4 can be expressed using lower powers of α using the fact that α is the root of the primitive polynomial $x^4 + x + 1$. Thus $\alpha^4 + \alpha + 1 = 0$, or $\alpha^4 = \alpha + 1$. (Recall that $GF(2^n)$ uses modulo 2 additions.) The expressions for successive higher powers of α are obtained by multiplying the expressions for lower powers by α and replacing any α^4, thus created, by $\alpha + 1$.

Tables 2.1 and 2.2 are important to simplify additions between field elements. For example, using Table 2.2, one may easily add α^{10} and α^{11} in $GF(2^4)$ as $\alpha^{10} + \alpha^{11} = (\alpha^2 + \alpha + 1) + (\alpha^3 + \alpha^2 + \alpha) = \alpha^3 + 1 = \alpha^{14}$.

Table 2.1: Structure of $GF(2^3)$.

Primitive Polynomial: x^3+x+1 Elements and their Relationships:	
0	$\alpha^3=\alpha+1$
1	$\alpha^4=\alpha^2+\alpha$
α	$\alpha^5=\alpha^2+\alpha+1$
α^2	$\alpha^6=\alpha^2+1$
Dual Base $\langle\beta_2,\beta_1,\beta_0\rangle = \langle\alpha,\alpha^2,1\rangle$.	

Table 2.2: Structure of $GF(2^4)$.

Primitive Polynomial: x^4+x+1 Elements and their Relationships:	
0	$\alpha^7=\alpha^3+\alpha+1$
1	$\alpha^8=\alpha^2+1$
α	$\alpha^9=\alpha^3+\alpha$
α^2	$\alpha^{10}=\alpha^2+\alpha+1$
α^3	$\alpha^{11}=\alpha^3+\alpha^2+\alpha$
$\alpha^4=\alpha+1$	$\alpha^{12}=\alpha^3+\alpha^2+\alpha+1$
$\alpha^5=\alpha^2+\alpha$	$\alpha^{13}=\alpha^3+\alpha^2+1$
$\alpha^6=\alpha^3+\alpha^2$	$\alpha^{14}=\alpha^3+1$
Dual Base $\langle\beta_3,\beta_2,\beta_1,\beta_0\rangle = \langle 1,\alpha,\alpha^2,\alpha^{14}\rangle$.	

2.2 Dual of polynomial basis of Finite Fields

One can use an alternate representation for the elements of $GF(2^n)$ over $GF(2)$ using the *dual basis* $\langle\beta_{n-1},\beta_{n-2},\ldots,\beta_0\rangle$. The dual of polynomial basis is unique and its component β_i is defined as that element of $GF(2^n)$ which satisfies

$$Tr(\alpha^j\beta_i) = \begin{cases} 1 & \text{if } j=i, \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where, the *Trace* function $Tr(\cdot) : GF(2^n) \to GF(2)$ is computed as [25]:

$$Tr(x) = \sum_{i=0}^{n-1} x^{2^i}, \qquad x \in GF(2^n).$$

Note that Trace is a linear function. In other words, for any $a, b \in GF(2)$ and $X, Y \in GF(2^n)$, one has

$$Tr(aX + bY) = aTr(X) + bTr(Y).$$

The structure of the primitive polynomial governs the relationships between the dual basis elements. In particular, the dual basis elements β_i, $0 \leq i < n$ of a finite field $GF(2^n)$ are related to each other as given by the following Lemma.

Lemma 1 *Let $\langle \beta_{n-1}, \beta_{n-2}, \ldots, \beta_0 \rangle$ denote the dual base of GF(2^n). Then*

$$\beta_i = \begin{cases} \alpha\beta_0 & \textit{if } i = n-1 \\ \alpha\beta_{i+1} + p_{i+1}\beta_{n-1} & \textit{otherwise,} \end{cases} \tag{2.2}$$

where α is the primitive element of the field and p_i is the coefficient of x^i in the primitive polynomial used to generate the field.

Proof.

To prove that $\beta_{n-1} = \alpha\beta_0$, all we need to show is that the element $\alpha\beta_0$ satisfies the definition of β_{n-1}. Consider any $0 \leq j < n-1$. Because of the properties of β_0,

$$Tr(\alpha^j \alpha\beta_0) = Tr(\alpha^{j+1}\beta_0) = 0.$$

On the other hand, by using the linearity property of the Trace function and the fact that

$$\alpha^n = \sum_{k=0}^{n-1} p_k \alpha^k$$

one gets,

$$Tr(\alpha^{n-1}\alpha\beta_0) = \sum_{k=0}^{n-1} p_k Tr(\alpha^k \beta_0) = p_0 = 1.$$

The relation of β_i and β_{i+1} can be proved similarly. For any $0 \leq i < n-1$, and $0 \leq j < n-1$, $j \neq i$.

$$Tr(\alpha^j(\alpha\beta_{i+1} + p_{i+1}\beta_{n-1})) = Tr(\alpha^{j+1}\beta_{i+1}) + p_{i+1}Tr(\alpha^j\beta_{n-1}) = 0. \qquad (2.3)$$

Further, for $i < n-1$, for $j = n-1$,
by expressing α^n as $\sum_{k=0}^{n-1} p_k\alpha^k$ (since $p(\alpha) = 0$),

$$Tr(\alpha^{n-1}(\alpha\beta_{i+1}+p_{i+1}\beta_{n-1})) = \sum_{k=0}^{n-1} p_k Tr(\alpha^k\beta_{i+1})+p_{i+1}Tr(\alpha^{n-1}\beta_{n-1}) = p_{i+1}+p_{i+1} = 0. \qquad (2.4)$$

Finally, for $j = i < n-1$,

$$Tr(\alpha^i(\alpha\beta_{i+1} + p_{i+1}\beta_{n-1})) = Tr(\alpha^{i+1}\beta_{i+1}) + p_{i+1}Tr(\alpha^i\beta_{n-1})) = 1 + 0 = 1. \qquad (2.5)$$

From (2.3), (2.4) and (2.5),

$$\alpha\beta_{i+1} + p_{i+1}\beta_{n-1} = \beta_i, \qquad for 0 \leq i < n-1$$

. ∎

We use symbol σ to denote the quantity $(\alpha^n + 1)$. Using the fact that $p(\alpha) = 0$, σ can also be expressed as $\sigma = \sum_{i=1}^{n-1} p_i\alpha^i$. The interaction between σ and elements of the dual basis is important to our representation. It is given by the following Lemma.

Lemma 2 *Let $\langle\beta_{n-1}, \ldots, \beta_1, \beta_0\rangle$ denote the dual basis of $GF(2^n)$. Then*

$$Tr(\sigma\beta_i) = \begin{cases} 0 & \textit{if } i = 0, \\ p_i & \textit{otherwise}, \end{cases}$$

$$Tr(\alpha^{-1}\sigma\beta_i) = \begin{cases} 0 & \textit{if } i = n-1 \\ p_{i+1} & \textit{otherwise.} \end{cases}$$

Proof. One has

$$\begin{aligned} Tr(\sigma\beta_i) = Tr((\alpha^n+1)\beta_i) &= Tr(\sum_{j=1}^{n-1} p_j\alpha^j\beta_i) \\ &= \sum_{j=1}^{n-1} p_j Tr(\alpha^j\beta_i) \end{aligned} \tag{2.6}$$

The trace function in (2.6) is 0 except when $j = i$, when it is 1. This gives the value of $Tr(\sigma\beta_i)$ stated in the lemma. The value of $Tr(\alpha^{-1}\sigma\beta_i)$ can be computed similarly. ∎

2.3 Graph Automorphism

Graph isomorphism represents the problem of testing whether two graphs are identical. An isomorphism ϕ from a graph $G = (V_G, E_G)$ to a graph $H = (V_H, E_H)$ is one to one mapping of V_G onto V_H that preserves connectivity, that is $\phi : V_G \to V_H$ such that for any two vertices u and $v \in V_G$, we have $(\phi(u), \phi(v)) \in E_H$ if and only if $(u, v) \in E_G$.

An automorphism of a graph is an isomorphism from a graph to itself. Automorphisms are useful to *remap* an algorithm without affecting its performance on a graph in the event of failure of a node or an edge. It is also important in multi user machines when algorithm mappings need to be moved every time a new user demands some part of the architecture. Recently, automorphisms of architecture graphs have also been used to design better VLSI layout of multicore chips [21,22].

To illustrate the concept of graph automorphism, consider a graph shown in Fig. 2.1, and a mapping function defined in Table 2.3. To prove that the resulting graph after applying ϕ is an automorphic copy of graph in Fig. 2.1, one needs to show that this mapping satisfies the automorphism properties. It is easy to check that the specified ϕ is a one to one mapping from the graph nodes to themselves which means every node is an image of one and only one node. Secondly, for every pair of connected nodes in the graph, their images are also connected, and if the two nodes are not connected in the graph then their images remain unconnected.

For example nodes B and E are connected in the graph. One can see that their images under ϕ, A and C respectively, are also connected. Therefore, one can often call an automorphism as relabeling the node graph as shown in Fig. 2.2. In this graph, nodes are relabeled with their images. In Chapter 3, and Chapter 5 we will determine all possible automorphisms of the Wrapped Butterfly and the Cube Connected Cycles graphs.

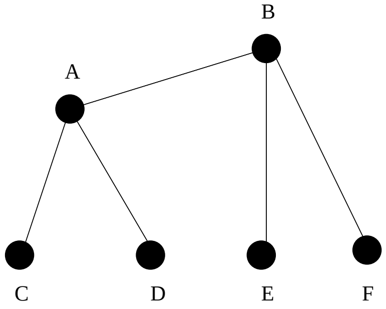

Figure 2.1: Original Graph.

Table 2.3: Automorphism Function

N	$\phi(N)$
A	B
B	A
C	F
D	E
E	C
F	D

2.4 An algebraic model of the butterfly graph

Before we provide the algebraic model for the butterfly, we first discuss the conventional binary model.

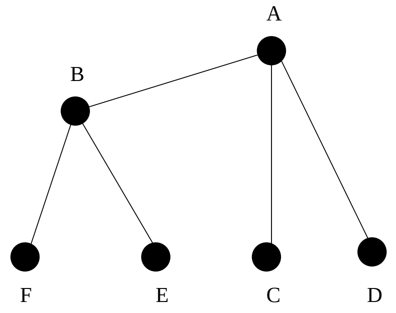

Figure 2.2: Relabeled Graph.

Let Z_n denote the group of integers 0 through $n-1$ under the operation of addition modulo n and Z_2^n, the group of binary vectors of length n under the operation of modulo 2 addition. Then the wrapped butterfly graph BF$_n$, $n \geq 3$, is defined to have $n2^n$ nodes each labeled by the pair (m, V), where $m \in Z_n$, $V \in Z_2^n$. The nodes of BF$_n$ may be arranged in a $2^n \times n$ array such that (m, V) is in the located in m-th column and V-th row. Each node is connected only to nodes in the neighboring columns (except for the *wrap-around* links between the nodes of the 0-th and the $n-1$-th columns). BF$_n$ graph is shown in Fig. 2.5. BF$_n$ is a symmetric, undirected regular graph of degree 4. BF$_n$ has a logarithmic diameter of $\lfloor 3n/2 \rfloor$ and a vertex connectivity 4, i.e., for any pair of nodes there exist 4 node disjoint paths between them. BF$_n$ supports many parallel algorithms well [1,7–9,27,28]. It is shown that one can map cycles and trees on BF$_n$ with relatively low dilation [29–31]. A node (m, V) is connected to four distinct nodes: $(m+1, V)$, $(m+1, V \oplus 2^m)$, $(m-1, V)$ and $(m-1, V \oplus 2^{m-1})$ as shown in Fig. 2.3. Note that the third and the fourth edges are inverses of the first and the second edges respectively. Thus the edges of a wrapped butterfly are bidirectional, i.e., corresponding to an edge from $(m_1,\ V_1)$ to $(m_2,\ V_2)$, there is also an edge from $(m_2,\ V_2)$ to $(m_1,\ V_1)$.

In this classical definition of the wrapped butterfly, the rows of destinations $(m+1, V \oplus 2^m)$ and $(m-1, V \oplus 2^{m-1})$ are dependent on both the column and the row of the source (m, V). This complicates mappings of algorithm task graphs on the butterfly.

Now we provide the algebraic model of BF$_n$ first presented in [10]. Here, the

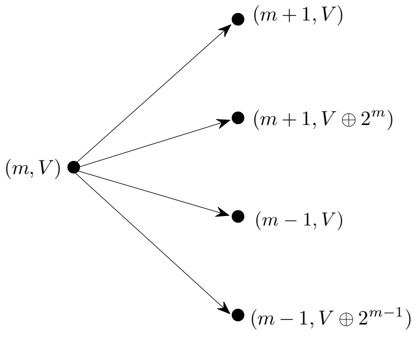

Figure 2.3: Connections from node $(m,\ V)$ in the butterfly network.

nodes of BF_n are labeled with pairs (m, X), $m \in C_n$, $X \in GF(2^n)$, where C_n is the cyclic group of integers 0 through $n-1$ under the operation of addition modulo n and $GF(2^n)$ is the finite field of 2^n elements. We will often refer to m as the column and X, the row, of node (m, X). Let α denote the primitive element of $GF(2^n)$ and $\langle \beta_{n-1}, \beta_{n-2}, \ldots, \beta_0 \rangle$, its dual basis. The node connectivity of graph BF_n can then be described through an algebraic relationship. In particular, a vertex (m, X) of BF_n is connected to the vertices $(m+1, \alpha X)$, $(m+1, \alpha X + \beta_{n-1})$, $(m-1, \alpha^{-1}X)$ and $(m-1, \alpha^{-1}X + \beta_0)$ as shown in Fig. 2.4. For convenience, we refer to these four edges as f, g, f^{-1} and g^{-1} respectively. It is easy to verify that if edge f goes from node N_1 to N_2, then the edge that goes from N_2 to N_1 is f^{-1}. The same observation is also true for g and g^{-1}. Since edges of the butterfly are bidirectional, we often refer to an edge either as f or g without worrying about the direction.

In order to establish the equivalence between the binary labels and the new labels, we use the following mapping $\zeta : Z_n \times Z_2^n \rightarrow C_n \times GF(2^n)$,

$$\zeta(m,\ v_{n-1}v_{n-2}\ldots v_1v_0) = (m,\ \sum_{i=0}^{n-1} v_{(i+m) \bmod n}\, \beta_i). \tag{2.7}$$

Mapping ζ is one-to-one and onto because $\langle \beta_{n-1}, \beta_{n-2}, \ldots, \beta_0 \rangle$ is a basis of $GF(2^n)$. It is proven in [10] that ζ also preserves the connectivity of BF_n. Thus ζ is merely an

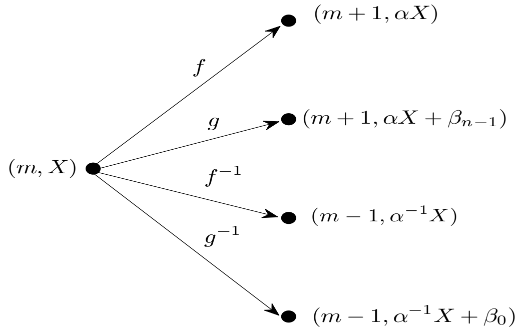

Figure 2.4: Connections from node $(m, X) \in C_n \times GF(2^n)$ in the butterfly network.

isomorphism or relabeling of the butterfly nodes. Table 2.4 provides the mapping ζ between the two representations of B_4. In order to illustrate the entries in this table, consider mapping of a butterfly node $(1,\ 1110) \in Z_n \times Z_2^n$ to its new algebraic setting. The dual basis of $GF(2^4)$ given in Table 2.2 is $\langle \beta_3, \beta_2, \beta_1, \beta_0 \rangle = \langle 1, \alpha, \alpha^2, \alpha^{14} \rangle$. Thus

$$\begin{aligned} \zeta(1,\ 1110) &= (1,\ \alpha^{14} + \alpha^2 + \alpha) \\ &= (1,\ 1 + \alpha + \alpha^2 + \alpha^3) \\ &= (1,\ \alpha^{12}). \end{aligned}$$

Thus the butterfly node with binary label $(1,\ 1110)$ is renamed in the new algebraic notation as $(1,\ \alpha^{12})$. The butterfly graph B_4 relabeled in the algebraic notation is shown in Fig. 2.6.

The simplicity of this model should be apparent from the fact that unlike the binary representation, the two components of the destination of (m, X) are independent. For the proof and examples of the algebraic model of BF_n, reader is referred to [10].

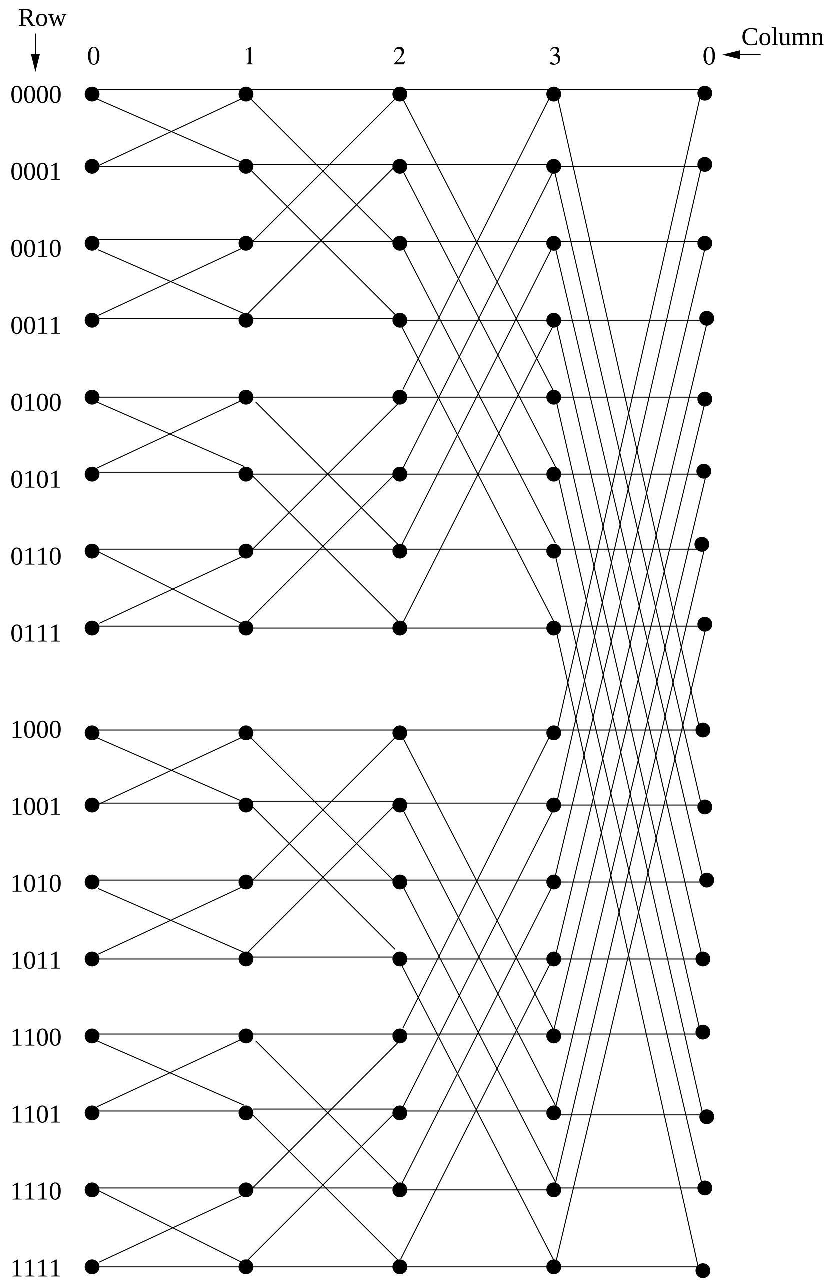

Figure 2.5: Connections of Butterfly B_4 in Binary notation.

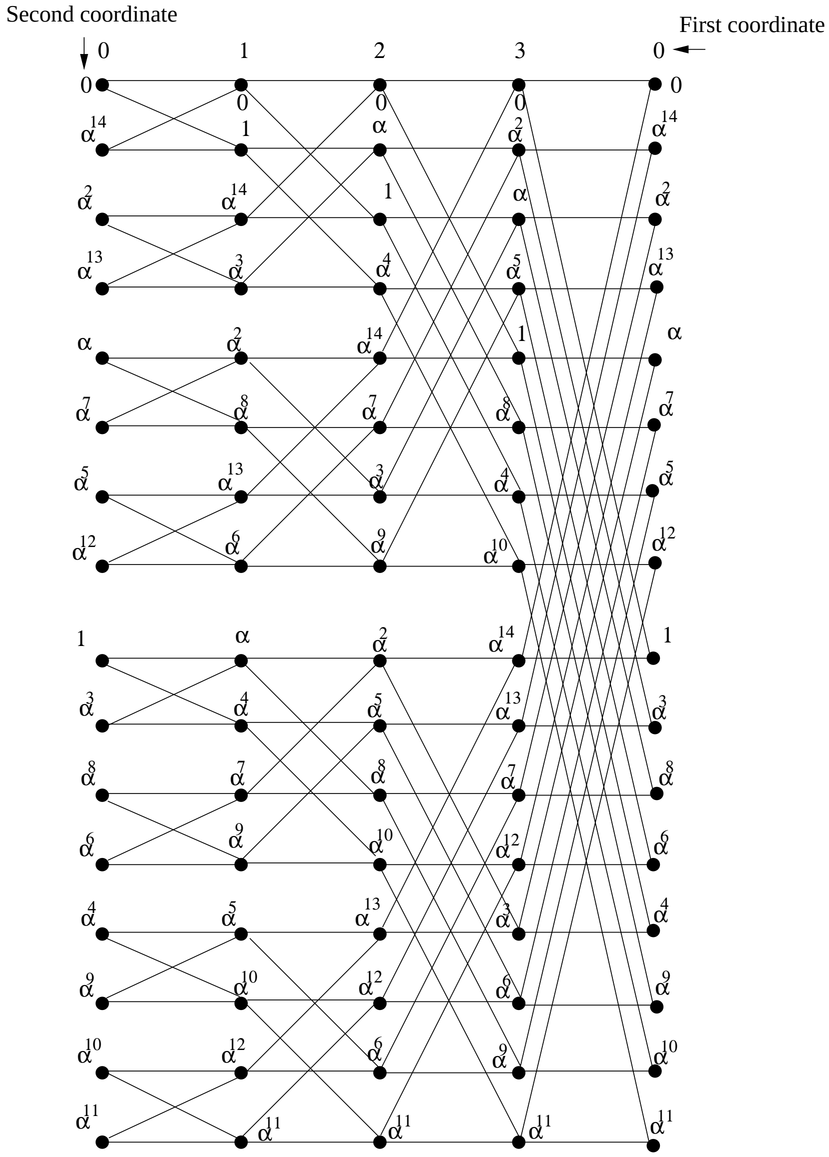

Figure 2.6: Connections of Butterfly B_4 in Algebraic notation.

Table 2.4: Equivalence between the nodes of B_4 and graph $C_4 \times GF(2^4)$.

label	$(m,\ X)$	label	$(m,\ X)$	label	$(m,\ X)$	label	$(m,\ X)$
(0, 0000)	$(0,0)$	(1, 0000)	$(1,0)$	(2, 0000)	$(2,0)$	(3, 0000)	$(3,0)$
(0, 0001)	$(0,\alpha^{14})$	(1, 0001)	$(1,1)$	(2, 0001)	$(2,\alpha)$	(3, 0001)	$(3,\alpha^2)$
(0, 0010)	$(0,\alpha^2)$	(1, 0010)	$(1,\alpha^{14})$	(2, 0010)	$(2,1)$	(3, 0010)	$(3,\alpha)$
(0, 0011)	$(0,\alpha^{13})$	(1, 0011)	$(1,\alpha^3)$	(2, 0011)	$(2,\alpha^4)$	(3, 0011)	$(3,\alpha^5)$
(0, 0100)	$(0,\alpha)$	(1, 0100)	$(1,\alpha^2)$	(2, 0100)	$(2,\alpha^{14})$	(3, 0100)	$(3,1)$
(0, 0101)	$(0,\alpha^7)$	(1, 0101)	$(1,\alpha^8)$	(2, 0101)	$(2,\alpha^7)$	(3, 0101)	$(3,\alpha^8)$
(0, 0110)	$(0,\alpha^5)$	(1, 0110)	$(1,\alpha^{13})$	(2, 0110)	$(2,\alpha^3)$	(3, 0110)	$(3,\alpha^4)$
(0, 0111)	$(0,\alpha^{12})$	(1, 0111)	$(1,\alpha^6)$	(2, 0111)	$(2,\alpha^9)$	(3, 0111)	$(3,\alpha^{10})$
(0, 1000)	$(0,1)$	(1, 1000)	$(1,\alpha)$	(2, 1000)	$(2,\alpha^2)$	(3, 1000)	$(3,\alpha^{14})$
(0, 1001)	$(0,\alpha^3)$	(1, 1001)	$(1,\alpha^4)$	(2, 1001)	$(2,\alpha^5)$	(3, 1001)	$(3,\alpha^{13})$
(0, 1010)	$(0,\alpha^8)$	(1, 1010)	$(1,\alpha^7)$	(2, 1010)	$(2,\alpha^8)$	(3, 1010)	$(3,\alpha^7)$
(0, 1011)	$(0,\alpha^6)$	(1, 1011)	$(1,\alpha^9)$	(2, 1011)	$(2,\alpha^{10})$	(3, 1011)	$(3,\alpha^{12})$
(0, 1100)	$(0,\alpha^4)$	(1, 1100)	$(1,\alpha^5)$	(2, 1100)	$(2,\alpha^{13})$	(3, 1100)	$(3,\alpha^3)$
(0, 1101)	$(0,\alpha^9)$	(1, 1101)	$(1,\alpha^{10})$	(2, 1101)	$(2,\alpha^{12})$	(3, 1101)	$(3,\alpha^6)$
(0, 1110)	$(0,\alpha^{10})$	(1, 1110)	$(1,\alpha^{12})$	(2, 1110)	$(2,\alpha^6)$	(3, 1110)	$(3,\alpha^9)$
(0, 1111)	$(0,\alpha^{11})$	(1, 1111)	$(1,\alpha^{11})$	(2, 1111)	$(2,\alpha^{11})$	(3, 1111)	$(3,\alpha^{11})$

2.5 Cycles in Wrapped Butterflies

This section provides an overview of mapping cycles of all possible lengths on Wrapped Butterfly. For detailed discussion, see [10].

[10] has shown that in BF_n with an even n, cycles of all even lengths L can be mapped except $L = 6$ when $n > 6$ and $L = 10$ when $n > 10$. Similarly for an odd n, cycles of all lengths can be mapped on BF_n except odd $L < n$, $L = 6$ when $n = 5$ or $n > 6$ and $L = 10$ when $n = 7$, $n = 9$ or $n > 10$. Since part of this research deals with mapping cycles on faulty butterflies, we briefly describe here the process of mapping a cycle on BF_n.

To map a cycle of length $L \leq lcm(n, 2^n - 1)$, where L is a multiple of n $L \neq 2^n - 1$, one can start from a vertex (m, X) where m is arbitrary and $X = \beta_{n-1}(1 + \alpha^L)^{-1}$. By continuously using the f edges, the rest of the vertices of the cycle are obtained. The last vertex, $(m - 1, \alpha^{L-1}X)$, is connected to the first vertex (m, X) by a g edge because of the value of X when $L < lcm(n, 2^n - 1)$ and by an f edge when $L = lcm(n, 2^n - 1)$. Note that the second coordinate of each vertex in such cycles is nonzero. On the other hand, by starting from a vertex $(m, 0)$ for an arbitrary m and using f edges between nodes, one can get a cycle of length n that contains all the nodes with their second coordinate 0. We will often refer to this cycle as the 0-cycle.

Two cycles in BF_n may be merged to form a larger cycle as the following Lemma shows.

Lemma 3 *[10] Any two distinct cycles in BF_n may be merged if one cycle contains some vertex $P = (m, X)$ and the other, the vertex $Q = (m, X + \beta_0)$.*

Proof. Let P' and Q' denote the adjacent vertices to P and Q in the two cycles with first index $m+1$. It is easy to see that if $P \to P'$ is an f edge, then so is $Q \to Q'$, or else P' are Q' are not distinct. Similarly, if $P \to P'$ is a g edge, then so is $Q \to Q'$. Cycle merging in both these cases is achieved by dropping the edges $P \to P'$ and $Q \to Q'$ and adding edges $P \to Q'$ and $Q \to P'$. ∎

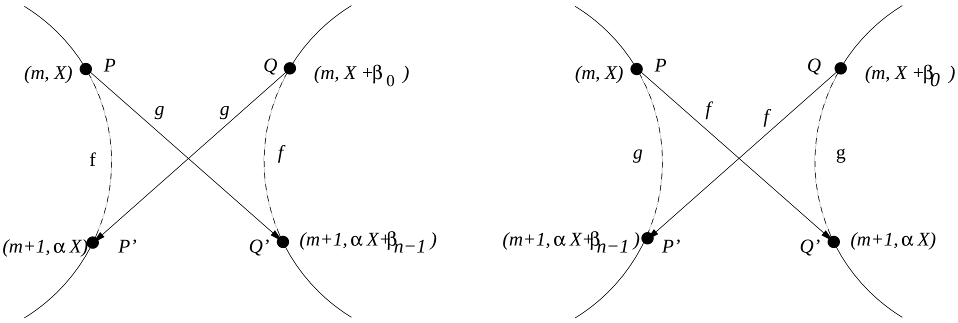

Figure 2.7: Two possible cases of merging two distinct cycles when one cycle contains the vertex (m, X) and the other, the vertex $(m, X + \beta_0)$.

The cycle obtained by merging has a length equal to the sum of the lengths of the two original cycles. Lemma 3 can also be used to obtain a Hamiltonian cycle in BF_n. To achieve this, we first design $\gcd(n, 2^n - 1)$ distinct cycles that include all the vertices with nonzero second coordinate. Each cycle begins from some vertex (m, X), $X \neq 0$, not included in previous cycles and uses only f edges. It is easy to show that each of these cycles will have $n/\gcd(n, 2^n - 1)$ vertices of type (m, β_0).

Since the corresponding vertices $(m, 0)$ belong to the 0-cycle, each of these cycles can be merged with the 0-cycle using Lemma 3. Thus we get the Hamiltonian cycle for BF_n rather easily.

Note that Lemma 3 can also be used to merge a pair of outside adjacent vertices (m, Q) and $(m+1, Q')$ with a cycle provided a vertex (m, P) within the cycle is such that $P = Q + \beta_0$. This observation allows one to create cycles of lengths which are not necessarily multiples of n into BF_n.

2.6 An algebraic model of the deBruijn graph

The deBruijn graph [32] has been one of the interesting and applied graphs. A deBruijn graph of degree n, DB_n, is defined to have 2^n nodes, each with a maximum node degree of four independent of the network size. DB_n is attractive because it has a small constant node degree and a small diameter, n. It however suffers from the fact that its connectivity is difficult to explore. It is neither symmetric nor recursive. Consequently, mapping parallel algorithms on them becomes a very complex task. Since an interconnection network is useful only if real-world application programs can be mapped onto it, there have been intense efforts recently to map standard algorithm skeletons such as the trees and cycles on these networks [4,33–36]. However, apart from [37], which uses graph theoretic relationship between shuffle oriented digraphs and hypercubes, very few results are available that provide a fresh look at the connectivity properties. This lack of proper analytical models to express the connectivity of these networks has proved to be a major roadblock in these analysis.

In this section we show that the nodes of DB_n maybe labeled with elements of the finite fields $GF(2^n)$ such that the node connectivity is expressed through an algebraic relationship between these labels. This allows one to exploit the rich properties of the finite fields to develop good mappings on these networks.

Before we provide the algebraic model for the deBruijn, we first discuss the conventional binary model. DB_n is defined to have 2^n nodes each labeled with an n bit binary string. A node $(v_{n-1}, v_{n-2}, \ldots, v_0) \in \mathrm{DB}_n$ is connected to four nodes

$(0, v_{n-1}, v_{n-2}, \ldots v_1)$, $(1, v_{n-1}, v_{n-2}, \ldots v_1)$, $(v_{n-2}, v_{n-3} \ldots v_0, 0)$ and $(v_{n-2}, v_{n-3} \ldots v_0, 1)$. DB_4 graph is shown in Fig. 2.8.

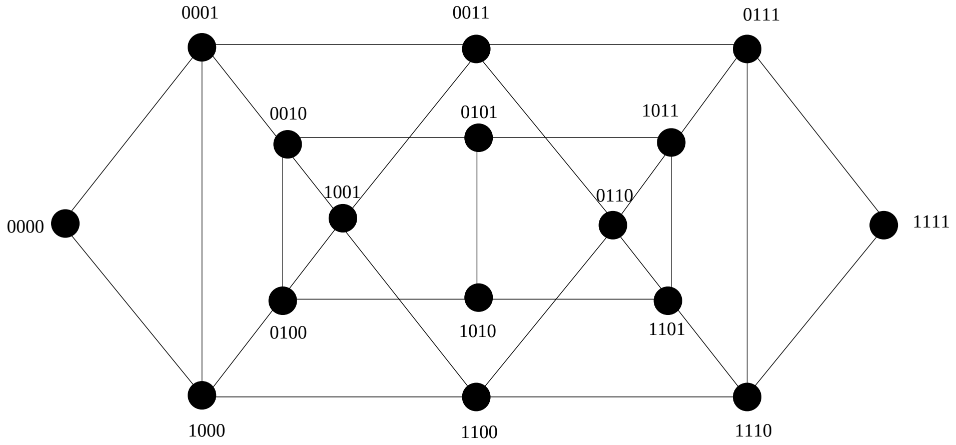

Figure 2.8: Connections of deBruijn DB_4 in Binary notation.

Now we provide the algebraic model of DB_n first presented in [15]. Here, the nodes of DB_n are labeled with elements of the finite field $GF(2^n)$, a finite field of 2^n elements. Let α denote the primitive element of $GF(2^n)$ and $\langle \beta_{n-1}, \beta_{n-2}, \ldots, \beta_0 \rangle$, its dual basis. The node connectivity of graph DB_n can then be described through an algebraic relationship. In particular, A node with label $X \in GF(2^n)$ is connected to nodes αX, $\alpha X + \beta_{n-1}$, $\alpha^{-1} X$ and $\alpha^{-1} X + \beta_0$ as shown in Fig. 2.9.

For convenience, we refer to these four edges as f, g, f^{-1} and g^{-1} respectively. It is easy to verify that if edge f goes from node N_1 to N_2, then the edge that goes from N_2 to N_1 is f^{-1}. The same observation is also true for g and g^{-1}. Since edges of the deBruijn graph are bidirectional, we often refer to an edge either as f or g without worrying about the direction.

In order to establish the equivalence between the binary labels and the algebraic labels, mapping $\zeta : Z_n \times Z_2^n \to C_n \times GF(2^n)$ defined below can be used.

$$\zeta(v_{n-1}, v_{n-2}, \ldots v_1, v_0) = (\sum_{i=0}^{n-1} v_i \; \beta_i). \tag{2.8}$$

Mapping ζ is one-to-one and onto because $\langle \beta_{n-1}, \beta_{n-2}, \ldots, \beta_0 \rangle$ is a basis of $GF(2^n)$. It is proven in [15] that ζ also preserves the connectivity of DB_n. Thus ζ is merely

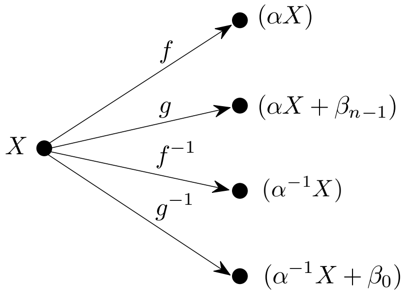

Figure 2.9: The connectivity of the deBruijn graph (DB_n)

an isomorphism or relabeling of the deBruijn nodes. Table 2.5 provides the mapping ζ between the two representations of DB_4.

In order to illustrate the entries in this table, consider mapping of a deBruijn node $(1,1,0,1) \in Z_2^n$ to its new algebraic setting. The dual basis of $GF(2^4)$ given in Table 2.2 is $\langle 1, \alpha, \alpha^2, \alpha^{14} \rangle$. Thus

$$\begin{aligned} \zeta(1,1,0,1) &= 1 + \alpha + \alpha^{14} \\ &= 1 + \alpha + (\alpha^3 + 1) \\ &= \alpha^3 + \alpha \;=\; \alpha^9. \end{aligned}$$

Thus the deBruijn node with binary label $(1,1,0,1)$ is renamed in the new algebraic notation as α^9. The deBruijn graph DB_4 relabeled in the algebraic notation is shown in Fig. 2.10.

Table 2.5: Equivalence between the binary and the algebraic labels of DB_4.

Binary	Algebraic
(0, 0, 0, 0)	0
(0, 0, 0, 1)	α^{14}
(0, 0, 1, 0)	α^{2}
(0, 0, 1, 1)	α^{13}
(0, 1, 0, 0)	α
(0, 1, 0, 1)	α^{7}
(0, 1, 1, 0)	α^{5}
(0, 1, 1, 1)	α^{12}

Binary	Algebraic
(1, 0, 0, 0)	1
(1, 0, 0, 1)	α^{3}
(1, 0, 1, 0)	α^{8}
(1, 0, 1, 1)	α^{6}
(1, 1, 0, 0)	α^{4}
(1, 1, 0, 1)	α^{9}
(1, 1, 1, 0)	α^{10}
(1, 1, 1, 1)	α^{11}

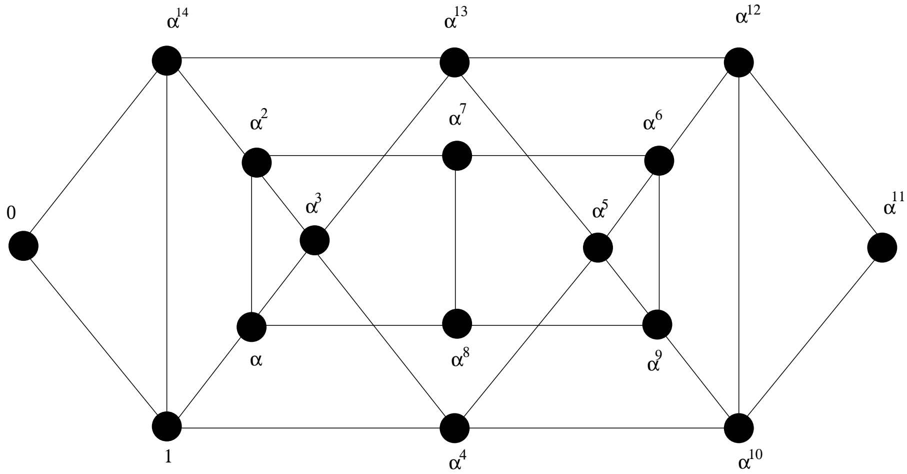

Figure 2.10: Connections of deBruijn DB_4 in Algebraic notation.

Chapter 3

Butterfly Automorphisms

3.1 Introduction

As the quest of high-speed computing resources continues, the physical limitations on uniprocessor speed due to the Moore's law imply a pressing need for parallel processors. These multi-processors exchange information using interconnection networks. Unfortunately, the speed of data transfer between cooperating processors has not kept pace with the increase in the computing speed. Therefore, the choice of the interconnection network affects several characteristics of the system, such as performance, ease of algorithm development, reliability, scalability and complexity of the physical layout. As a result, communication network of a parallel processor dominates its performance.

The *wrap-around butterfly network* represents a good trade-off between the cost and the performance of a parallel machine. It has a large number of processors, fixed node degree, low diameter, symmetry, and an ability to support a variety of parallel algorithms. Cube Connected Cycles is a sub-graph of BF_n [38]. Other extensions of BF_n are also available [39,40]. BF_n supports many parallel algorithms efficiently [1,7–10,27,28,31].

With the advances in the VLSI technology, it is now possible to build parallel machines with a large number of processors. However, larger the machine, higher is the probability that one or more of its processors or links will develop a fault. Thus, for the underlying networks of these large machines, mappings of algorithms on faulty graphs becomes an important design issue.

Previous results about mappings on faulty butterflies include one by Vadapalli and Srimani who have shown that in BF_n, there exists a cycle of length at least $n2^n - 2$ with one faulty node and $n2^n - 4$ with two faulty nodes [41]. Later, Tsai et al., improved this to show that for odd n, cycle length $n2^n - 2$ is possible with two faulty nodes [42]. They also proved that in the presence of one faulty node and one faulty edge, there exists a cycle of length $n2^n - 2$ when n is even, and $n2^n - 1$, when n is odd. Hwang and Chen have shown that the maximal cycle of length $n2^n$ can be embedded in a faulty butterfly even with two edge faults [43]. However, these studies have used the binary representation of the butterfly resulting in rather complex mappings. Their results are limited to either mapping Hamiltonian cycle or the largest possible cycle with limited fault set.

This chapter obtains all the automorphisms of a wrapped butterfly network of degree n using an algebraic model. We show that with this model, it is rather simple to obtain all the automorphisms of the butterfly network. In addition, this chapter uses the powerful algebraic techniques to study the edge transformations due to these automorphisms. This chapter also proposes a new strategy for algorithm mappings on an architecture with faulty edges. This strategy essentially consists of finding an automorphism that would map the faulty edges to the free edges in the graph. Having a set of $n2^{n+1}$ simple well defined automorphisms which translate graph edges deterministically, makes this a very powerful technique for dealing with edge faults. This strategy of avoiding edge faults using automorphisms is quite novel because previously automorphisms have been employed only to avoid the node faults. We illustrate our methods by mapping Hamilton cycle on the butterfly under various edge fault scenarios.

This chapter is organized in these sections. Section 3.2 obtains all the automorphisms of a wrapped butterfly network of degree n using an algebraic model. Section 3.3 investigate the translation of butterfly edges by automorphisms. It proposes a new strategy for algorithm mappings on an architecture with faulty edges. We illustrate in Section 3.4 our methods by mapping Hamiltonian cycle on the butterfly under various edge fault scenarios.

3.2 Automorphisms of the butterfly network

Wagh and Guzide have previously shown that the algebraic model allows efficient mappings of cycles of all possible lengths and trees of largest sizes on the butterfly [10]. The relevant part of that work is summarized in Section 2.5. We extend this work by exploring the automorphisms of butterfly in the same setting.

Since nodes in column m are only connected to nodes in columns $m+1$ and $m-1$, one has only two kinds of automorphisms of BF_n; those which map nodes in column m to nodes in column $m+t$ for an integer t and those which map nodes in column m to nodes in column $t-m$. We denote the automorphisms of the first kind by $\phi(\cdot)$. An automorphism which maps nodes in column m to nodes $-m \bmod n$ is denoted by $\psi(\cdot)$. A product of $\psi(\cdot)$ with the set of $\phi(\cdot)$ automorphisms provides all the automorphisms of the second kind.

We first give the following lemma which relates the edges in a column to edges in any other column. This lemma forms the foundation of the automorphisms of the first kind.

Lemma 4 (connectivity) *Let $K_m, K_{m+1} \in GF(2^n)$ be related as $K_{m+1} = \alpha K_m$ or $K_{m+1} = \alpha K_m + \beta_{n-1}$. For any $X, Y \in GF(2^n)$ and $t \in C_n$, if nodes (m, X) and $(m+1, Y)$ are connected in BF_n, then so are the nodes $(m+t, X+K_m)$ and $(m+1+t, Y+K_{m+1})$.*

Proof. The presence of the edge $(m+t, X+K_m) \to (m+1+t, Y+K_{m+1})$ can be proved by showing that $Y+K_{m+1} = \alpha(X+K_m) + c\beta_{n-1}$ for some $c \in \{0,1\}$. Since $(m,X) \to (m+1,Y)$, the connectivity of BF_n gives $Y = \alpha X + c'\beta_{n-1}$ for $c' \in \{0,1\}$. Further, the given constants K_m and K_{m+1} are related as $K_{m+1} = \alpha K_m + c''\beta_{n-1}$, where $c'' \in \{0,1\}$. Therefore $Y + K_{m+1} = \alpha(X+K_m) + (c'+c'')\beta_{n-1}$. ∎

The connectivity specified by Lemma 4 can be used to obtain the automorphisms of the butterfly network as shown in Theorem 1.

Theorem 1 *If constants $K_0, K_1, \ldots, K_{n-1} \in GF(2^n)$ satisfy*

$$K_i = \begin{cases} \alpha K_{i-1} \;\; or \;\; \alpha K_{i-1} + \beta_{n-1}, & \text{if } 0 < i \leq n-1, \\ \alpha K_{n-1} \;\; or \;\; \alpha K_{n-1} + \beta_{n-1} & \text{if } i = 0, \end{cases}$$

then function $\phi(\cdot) : B_n \to B_n$ defined as

$$\phi((m,\ X)) = (m+t,\ X + K_m) \tag{3.1}$$

for any $t \in C_n$, is an automorphism of BF_n, i.e., it maps nodes of BF_n to nodes and edges to edges.

Proof. The fact that $\phi(\cdot)$ maps edges to edges is clear from Lemma 4. To prove that it is an automorphism we only have to show that it is a one-to-one and onto mapping.

Let $\phi(m, X) = \phi(m', X')$, then from the definition of $\phi(\cdot)$,

$$(m+t, X+K_m) = (m'+t, X'+K_{m'}).$$

From the first components of the two pairs, $m = m'$. From the second components, $X + K_m = X' + K_m$ which implies that $X = X'$. Thus two distinct nodes cannot have the same image under $\phi(\cdot)$, i.e., $\phi(\cdot)$ is one-to-one.

Now consider any node $(m', Y) \in B_n$. It is easy to see that this node is the image of $(m' - t, Y + K_{m'-t})$. Therefore $\phi(\cdot)$ is onto. ∎

Note that constant t merely translates edges in one column to a column t away. As Theorem 1 shows, this t and constant elements $K_i \in GF(2^n)$, $0 \leq i < n$ fully define the automorphism $\phi(\cdot)$. We will henceforth refer to t as the *column offset* and K_is as the *automorphism offsets*,

One can see the simplicity of the automorphism $\phi(\cdot)$ defined in (3.1). Every node in the network is applied the same column offset and every node in the same column is applied the same automorphism offset. Further, the offsets of the two coordinates of a node label are *independent*. This makes use of such an automorphism especially attractive.

Theorem 1 allows one to design such an automorphism under various conditions. For example, suppose one wants an automorphism such that for a given pair of nodes $N_1 = (a,\ U), N_2 = (b,\ V) \in B_n$, the automorphism maps N_1 to N_2, i.e.,

$$\phi(N_1) = N_2. \tag{3.2}$$

(If we can do this for an arbitrary pair of nodes, it would imply that BF$_n$ is a symmetric network.) Such a mapping can be obtained by choosing a column offset t and automorphism offsets $K_0, K_1, \ldots, K_{n-1} \in GF(2^n)$ satisfying condition in Theorem 1) and then defining ϕ as in (3.1). Note that the relations between K_is provide certain flexibility in the choice of the constants. We exploit this flexibility to ensure that (3.2) is satisfied.

Let us rewrite the relations between K_is as

$$K_i = \alpha K_{(i-1)\text{mod }n} + c_i\beta_{n-1}, \qquad 0 \le i \le n-1, \tag{3.3}$$

where each c_i is either 0 or 1. One can use (3.3) repeatedly to express any individual automorphism offset as

$$\begin{aligned} K_a &= \alpha K_{(a-1)\text{mod }n} + c_a\beta_{n-1} \\ &= \alpha^2 K_{(a-2)\text{mod }n} + (c_{(a-1)\text{mod }n}\ \alpha + c_a)\beta_{n-1} \\ &= \alpha^3 K_{(a-3)\text{mod }n} + \\ &\quad (c_{(a-2)\text{mod }n}\ \alpha^2 + c_{(a-1)\text{mod }n}\ \alpha + c_a)\beta_{n-1}. \end{aligned}$$

Proceeding in this fashion, one gets

$$K_a = \alpha^n\ K_a + (\sum_{j=0}^{n-1} c_{(a-j)\text{mod }n}\ \alpha^j)\beta_{n-1},$$

or

$$K_a = (1+\alpha^n)^{-1}(\sum_{j=0}^{n-1} c_{(a-j)\text{mod }n}\ \alpha^j)\beta_{n-1}. \tag{3.4}$$

Further, if $\phi((m,\ X)) = (m+t,\ X+K_m)$, then to satisfy (3.2) requires that

$$\begin{aligned} t &= (b-a)\text{mod } n \qquad \text{and} \\ K_a &= U+V. \end{aligned} \tag{3.5}$$

By combining (3.4) and (3.5), one gets

$$(U+V)(\alpha^n+1)\beta_{n-1}^{-1} = \sum_{j=0}^{n-1} c_{(a-j)\text{mod } n}\alpha^j, \tag{3.6}$$

One can see that the left hand side of (3.6) is an element of $GF(2^n)$ and can therefore be uniquely expressed in the polynomial basis $\langle \alpha^{n-1}, \alpha^{n-2}, \ldots, 1\rangle$. This gives the unique set of values for c_is. One can then use these values in (3.3) to obtain the automorphism offsets $K_{(a+1)\text{mod } n}$, $K_{(a+2)\text{mod } n}$, …, $K_{(a-1)\text{mod } n}$.

One can illustrate this procedure by:

Example 1. Computing an automorphism $\phi(\cdot) : B_4 \to B_4$ which maps node $(3, \alpha^{14})$ to node $(1, \alpha^2)$. For this function, the column offset $t = (1-3)\text{mod } 4 = 2$ and the automorphism offset $K_3 = \alpha^{14} + \alpha^2 = \alpha^{13}$. (see Table 2.2). Further,

$$\sum_{j=0}^{n-1} c_{(3-j)\text{mod } n}\alpha^j = K_3\sigma\beta_3^{-1} = \alpha^3 + 1.$$

Thus one gets $c_0 = 1$, $c_1 = 0$, $c_2 = 0$ and $c_3 = 1$ and consequently, $K_0 = \alpha^3$, $K_1 = \alpha^4$ and $K_2 = \alpha^5$. The resultant automorphism function $\phi(\cdot)$ is given in Table 3.1.

It is easy to verify that the mapping in Table 3.1 preserves connectivity. For example, $(0, \alpha^3)$ and $(1, \alpha^4)$ were connected in the original graph. After mapping, their images $(2, 0)$ and $(3, 0)$ remain connected.

Example 2. Computing an automorphism $\phi(\cdot) : B_3 \to B_3$ which maps node $(1, \alpha^2)$ to node $(0, \alpha^6)$. For this function, the column offset $t = (0-1)\text{mod } 3 = 2$ and the automorphism offset $K_1 = \alpha^2 + \alpha^6 = 1$. (see Table 2.1.) Further,

$$\sum_{j=0}^{n-1} c_{(1-j)\text{mod } n}\alpha^j = K_1\sigma\beta_2^{-1} = 1.$$

Thus one gets $c_0 = 0$, $c_1 = 1$ and $c_2 = 0$ and consequently, $K_0 = \alpha^2$, $K_1 = 1$ and $K_2 = \alpha$. The resultant automorphism function $\phi(\cdot)$ is given in Table 3.2.

Table 3.1: Automorphism $\phi(\cdot) : B_4 \to B_4$ such that $\phi(3, \alpha^{14}) = (1, \alpha^2)$.

(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$
$(0, 0)$	$(2, \alpha^3)$	$(1, 0)$	$(3, \alpha^4)$	$(2, 0)$	$(0, \alpha^5)$	$(3, 0)$	$(1, \alpha^{13})$
$(0, 1)$	$(2, \alpha^{14})$	$(1, 1)$	$(3, \alpha)$	$(2, 1)$	$(0, \alpha^{10})$	$(3, 1)$	$(1, \alpha^6)$
$(0, \alpha)$	$(2, \alpha^9)$	$(1, \alpha)$	$(3, 1)$	$(2, \alpha)$	$(0, \alpha^2)$	$(3, \alpha)$	$(1, \alpha^{12})$
$(0, \alpha^2)$	$(2, \alpha^6)$	$(1, \alpha^2)$	$(3, \alpha^{10})$	$(2, \alpha^2)$	$(0, \alpha)$	$(3, \alpha^2)$	$(1, \alpha^{14})$
$(0, \alpha^3)$	$(2, 0)$	$(1, \alpha^3)$	$(3, \alpha^7)$	$(2, \alpha^3)$	$(0, \alpha^{11})$	$(3, \alpha^3)$	$(1, \alpha^8)$
$(0, \alpha^4)$	$(2, \alpha^7)$	$(1, \alpha^4)$	$(3, 0)$	$(2, \alpha^4)$	$(0, \alpha^8)$	$(3, \alpha^4)$	$(1, \alpha^{11})$
$(0, \alpha^5)$	$(2, \alpha^{11})$	$(1, \alpha^5)$	$(3, \alpha^8)$	$(2, \alpha^5)$	$(0, 0)$	$(3, \alpha^5)$	$(1, \alpha^7)$
$(0, \alpha^6)$	$(2, \alpha^2)$	$(1, \alpha^6)$	$(3, \alpha^{12})$	$(2, \alpha^6)$	$(0, \alpha^9)$	$(3, \alpha^6)$	$(1, 1)$
$(0, \alpha^7)$	$(2, \alpha^4)$	$(1, \alpha^7)$	$(3, \alpha^3)$	$(2, \alpha^7)$	$(0, \alpha^{13})$	$(3, \alpha^7)$	$(1, \alpha^5)$
$(0, \alpha^8)$	$(2, \alpha^{13})$	$(1, \alpha^8)$	$(3, \alpha^5)$	$(2, \alpha^8)$	$(0, \alpha^4)$	$(3, \alpha^8)$	$(1, \alpha^3)$
$(0, \alpha^9)$	$(2, \alpha)$	$(1, \alpha^9)$	$(3, \alpha^{14})$	$(2, \alpha^9)$	$(0, \alpha^6)$	$(3, \alpha^9)$	$(1, \alpha^{10})$
$(0, \alpha^{10})$	$(2, \alpha^{12})$	$(1, \alpha^{10})$	$(3, \alpha^2)$	$(2, \alpha^{10})$	$(0, 1)$	$(3, \alpha^{10})$	$(1, \alpha^9)$
$(0, \alpha^{11})$	$(2, \alpha^5)$	$(1, \alpha^{11})$	$(3, \alpha^{13})$	$(2, \alpha^{11})$	$(0, \alpha^3)$	$(3, \alpha^{11})$	$(1, \alpha^4)$
$(0, \alpha^{12})$	$(2, \alpha^{10})$	$(1, \alpha^{12})$	$(3, \alpha^6)$	$(2, \alpha^{12})$	$(0, \alpha^{14})$	$(3, \alpha^{12})$	$(1, \alpha)$
$(0, \alpha^{13})$	$(2, \alpha^8)$	$(1, \alpha^{13})$	$(3, \alpha^{11})$	$(2, \alpha^{13})$	$(0, \alpha^7)$	$(3, \alpha^{13})$	$(1, 0)$
$(0, \alpha^{14})$	$(2, 1)$	$(1, \alpha^{14})$	$(3, \alpha^9)$	$(2, \alpha^{14})$	$(0, \alpha^{12})$	$(3, \alpha^{14})$	$(1, \alpha^2)$

Table 3.2: Automorphism $\phi(\cdot) : B_3 \to B_3$ such that $\phi(1, \alpha^2) = (0, \alpha^6)$.

(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$
$(0, 0)$	$(2, \alpha^2)$	$(1, 0)$	$(0, 1)$	$(2, 0)$	$(1, \alpha)$
$(0, 1)$	$(2, \alpha^6)$	$(1, 1)$	$(0, 0)$	$(2, 1)$	$(1, \alpha^3)$
$(0, \alpha)$	$(2, \alpha^4)$	$(1, \alpha)$	$(0, \alpha^3)$	$(2, \alpha)$	$(1, 0)$
$(0, \alpha^2)$	$(2, 0)$	$(1, \alpha^2)$	$(0, \alpha^6)$	$(2, \alpha^2)$	$(1, \alpha^4)$
$(0, \alpha^3)$	$(2, \alpha^5)$	$(1, \alpha^3)$	$(0, \alpha)$	$(2, \alpha^3)$	$(1, 1)$
$(0, \alpha^4)$	$(2, \alpha)$	$(1, \alpha^4)$	$(0, \alpha^5)$	$(2, \alpha^4)$	$(1, \alpha^2)$
$(0, \alpha^5)$	$(2, \alpha^3)$	$(1, \alpha^5)$	$(0, \alpha^4)$	$(2, \alpha^5)$	$(1, \alpha^6)$
$(0, \alpha^6)$	$(2, 1)$	$(1, \alpha^6)$	$(0, \alpha^2)$	$(2, \alpha^6)$	$(1, \alpha^5)$

It is easy to verify that the mapping in Table 3.2 preserves connectivity. For example, $(0,\alpha^6)$ and $(1,1)$ were connected in the original graph. After mapping their images $(2,1)$ and $(0,0)$ remain connected.

Alternately, one can design an automorphism $\phi : B_n \to B_n$ given c_is values. To construct ϕ in this case, given constants $c_i \in \{0,1\}$, $0 \leq i < n$, we compute K_0 by (3.6) as

$$K_0(\alpha^n + 1)\beta_{n-1}^{-1} = \sum_{j=0}^{n-1} c_{(-j)\ mod\ n}\alpha^j.$$

The other K_i values can then be inferred from (3.3).

As an example, let constants $c_0 = 0, c_1 = 1, c_2 = 1, c_3 = 1$ in B_4. Automorphism offset K_0 can then be obtained from

$$\sum_{j=0}^{n-1} c_{(-j) \bmod n}\ \alpha^j = K_0(1+\alpha^4)\beta_3^{-1}. \tag{3.7}$$

$$c_0 + c_3\alpha + c_2\alpha^2 + c_1\alpha^3 = K_0(\alpha)(1)^{-1}$$

$$\alpha + \alpha^2 + \alpha^3 = K_0(\alpha)$$

$$\alpha^{11} = K_0(\alpha).$$

Solving (3.7) gives $K_0 = \alpha^{10}$, which, in turn yields $K_1 = \alpha^{12}$, $K_2 = \alpha^6$ and $K_3 = \alpha^9$. The resultant automorphism function $\phi(\cdot)$ is given in Table 3.3.

As is evident from this discussion, all the automorphism offsets for any $\phi(\cdot)$ are related such that choosing any one of them, say, K_0, fixes all the others. On the other hand, distinct K_0 and t values give rise to distinct automorphisms. Thus there are exactly $n2^n$ automorphisms of butterfly BF_n when the first index of all the nodes is translated by the same amount.

Because the automorphism offsets play such a central role in defining the automorphism, we now provide some of their basic properties.

Table 3.3: Automorphism $\phi(\cdot) : B_4 \to B_4$ by choosing c's.

(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$	(m, X)	$\phi(m, X)$
$(0, 0)$	$(0, \alpha^{10})$	$(1, 0)$	$(1, \alpha^{12})$	$(2, 0)$	$(2, \alpha^{6})$	$(3, 0)$	$(3, \alpha^{9})$
$(0, 1)$	$(0, \alpha^{5})$	$(1, 1)$	$(1, \alpha^{11})$	$(2, 1)$	$(2, \alpha^{13})$	$(3, 1)$	$(3, \alpha^{7})$
$(0, \alpha)$	$(0, \alpha^{8})$	$(1, \alpha)$	$(1, \alpha^{13})$	$(2, \alpha)$	$(2, \alpha^{11})$	$(3, \alpha)$	$(3, \alpha^{3})$
$(0, \alpha^{2})$	$(0, \alpha^{4})$	$(1, \alpha^{2})$	$(1, \alpha^{7})$	$(2, \alpha^{2})$	$(2, \alpha^{3})$	$(3, \alpha^{2})$	$(3, \alpha^{11})$
$(0, \alpha^{3})$	$(0, \alpha^{12})$	$(1, \alpha^{3})$	$(1, \alpha^{10})$	$(2, \alpha^{3})$	$(2, \alpha^{2})$	$(3, \alpha^{3})$	$(3, \alpha)$
$(0, \alpha^{4})$	$(0, \alpha^{2})$	$(1, \alpha^{4})$	$(1, \alpha^{6})$	$(2, \alpha^{4})$	$(2, \alpha^{12})$	$(3, \alpha^{4})$	$(3, \alpha^{14})$
$(0, \alpha^{5})$	$(0, 1)$	$(1, \alpha^{5})$	$(1, \alpha^{14})$	$(2, \alpha^{5})$	$(2, \alpha^{9})$	$(3, \alpha^{5})$	$(3, \alpha^{6})$
$(0, \alpha^{6})$	$(0, \alpha^{7})$	$(1, \alpha^{6})$	$(1, \alpha^{4})$	$(2, \alpha^{6})$	$(2, 0)$	$(3, \alpha^{6})$	$(3, \alpha^{5})$
$(0, \alpha^{7})$	$(0, \alpha^{6})$	$(1, \alpha^{7})$	$(1, \alpha^{2})$	$(2, \alpha^{7})$	$(2, \alpha^{10})$	$(3, \alpha^{7})$	$(3, 1)$
$(0, \alpha^{8})$	$(0, \alpha)$	$(1, \alpha^{8})$	$(1, \alpha^{9})$	$(2, \alpha^{8})$	$(2, \alpha^{14})$	$(3, \alpha^{8})$	$(3, \alpha^{12})$
$(0, \alpha^{9})$	$(0, \alpha^{13})$	$(1, \alpha^{9})$	$(1, \alpha^{8})$	$(2, \alpha^{9})$	$(2, \alpha^{5})$	$(3, \alpha^{9})$	$(3, 0)$
$(0, \alpha^{10})$	$(0, 0)$	$(1, \alpha^{10})$	$(1, \alpha^{3})$	$(2, \alpha^{10})$	$(2, \alpha^{7})$	$(3, \alpha^{10})$	$(3, \alpha^{13})$
$(0, \alpha^{11})$	$(0, \alpha^{14})$	$(1, \alpha^{11})$	$(1, 1)$	$(2, \alpha^{11})$	$(2, \alpha)$	$(3, \alpha^{11})$	$(3, \alpha^{2})$
$(0, \alpha^{12})$	$(0, \alpha^{3})$	$(1, \alpha^{12})$	$(1, 0)$	$(2, \alpha^{12})$	$(2, \alpha^{4})$	$(3, \alpha^{12})$	$(3, \alpha^{8})$
$(0, \alpha^{13})$	$(0, \alpha^{9})$	$(1, \alpha^{13})$	$(1, \alpha)$	$(2, \alpha^{13})$	$(2, 1)$	$(3, \alpha^{13})$	$(3, \alpha^{10})$
$(0, \alpha^{14})$	$(0, \alpha^{11})$	$(1, \alpha^{14})$	$(1, \alpha^{5})$	$(2, \alpha^{14})$	$(2, \alpha^{8})$	$(3, \alpha^{14})$	$(3, \alpha^{4})$

Theorem 2 *Let $\phi(\cdot), \phi'(\cdot) : B_n \to B_n$ be any two automorphisms of BF_n based on sets of constants $t, K_0, K_1, \dots K_{n-1}$ and $t', K'_0, K'_1, \dots K'_{n-1}$. Then,*

1. *If any $K_m = 0$, then all $K_i = 0$, $0 \leq i < n$.*

2. *If any $K_m \neq 0$, then all $K_i \neq 0$, $0 \leq i < n$.*

3. *If any $K_m = K'_m$, then all $K_i = K'_i$, $0 \leq i < n$.*

4. *If any $K_m \neq K'_m$, then for every i, $0 \leq i < n$, $K_i \neq K'_i$.*

5. *$\sum_{i=0}^{n-1} K_i$ is either 0 or $(1+\alpha)^{-1}\beta_{n-1}$.*

Proof. From (3.4) one can see that $K_m = 0$ implies that $c_j = 0$, $0 \leq j < n$. Relation (3.3) then shows that each K_i is zero. On the other hand, if any K_m is nonzero, then so is every other K_i or else, any $K_i = 0$ would invalidate any other nonzero K_m. This proves the first two parts of the corollary.

To prove the third and fourth parts, it is sufficient to note from (3.3) and (3.4) that any given K_m uniquely determines all the other K_is. If $K_m = K^{'}_m$, then from (3.6) we get the same c values in the two cases, which will generate an equal set of K values. Hence, $K_i = K^{'}_i$, for all i.

Finally, the sum of all K_is can be computed as follows. By applying a summation to both sides of (3.3), one gets

$$\begin{aligned} \sum_{i=0}^{n-1} K_i &= \alpha(\sum_{i=0}^{n-1} K_i) + (\sum_{i=0}^{n-1} c_i)\beta_{n-1} \\ &= (\sum_{i=0}^{n-1} c_i)\beta_{n-1}(1+\alpha)^{-1} \qquad (3.8) \end{aligned}$$

Since $\sum_{i=0}^{n-1} c_i$ in (3.8) is either 0 or 1, the sum of all K_is is as stated in the Corollary. ∎

It is well known that the set of all the automorphisms form a group. In case of BF$_n$, set of automorphisms of the first kind also form a group. In particular it is easy to verify that automorphism $\phi(m, X) = (m, X)$ is the identity of the group. Following theorem specifies inverse of an automorphism.

Theorem 3 *Let $\phi(m, X) = (m + t, X + K_m)$ then the inverse automorphism is given by*

$$\phi^{-1}(m, X) = (m - t, X + K_{m-t}).$$

Proof. It is easy to see that

$$\phi^{-1}\phi(m, X) = \phi(m - t, X + K_{m-t}) = (m, X + K_{m-t} + K_{m-t}) = (m, X)$$

and

$$\phi\phi^{-1}(m, X) = \phi^{-1}(m + t, X + K_m) = (m, X + K_m + K_m) = (m, X).$$

∎

Group formed by all automorphism of the first kind is not commutative. If automorphism ϕ is characterized by t and K_m while another, ϕ', by t' and K'_m, then $\phi\phi'(m, X) = (m+t+t', X+K'_m+K_{m+t'})$ and $\phi'\phi(m, X) = (m+t'+t, X+K_m+K_{m+t})$. Clearly these two expressions cannot be equal always. However, the set of ϕ's with column offset t equal to zero form a commutative group.

We now investigate the automorphism $\psi(\cdot)$ of BF_n that reflects the column index of each node. Our result concerning this automorphism is stated in the following theorem.

Theorem 4 *For every $X \in GF(2^n)$, $X = \sum_{i=0}^{n-1} x_i\beta_i$, let $X' = \sum_{i=0}^{n-1} x_i\beta_{n-1-i}$. Then the mapping*

$$\psi(m, X) = (n - m, X')$$

is an automorphism of BF_n.

Proof. It is simple to see that $\psi(\cdot)$ is one-to-one and onto. We only need to prove that it preserves the edge connectivity of BF_n. In particular, we demonstrate that since vertex (m, X) is connected to the vertices $(m+1, \alpha X + c\beta_{n-1})$, $c \in \{0, 1\}$, $\psi(m, X)$ is also connected to vertices $\psi(m+1, \alpha X + c\beta_{n-1})$. Let $X = \sum_{i=0}^{n-1} x_i\beta_i$. Then using the relationships between the consecutive β_is given in Lemma 1, one gets

$$\begin{aligned} \alpha X + c\beta_{n-1} &= \sum_{i=1}^{n-1}(x_i\beta_{i-1} + x_ip_i\beta_{n-1}) + (c + x_0)\beta_0 \\ &= \sum_{i=0}^{n-2} x_{i+1}\beta_i + (c + \sum_{i=0}^{n-1} p_ix_i)\beta_{n-1}. \end{aligned} \tag{3.9}$$

Thus

$$\psi(m+1, \alpha X + c\beta_{n-1}) = (n - m - 1, Y) \tag{3.10}$$

where,

$$\begin{aligned} Y &= \sum_{i=0}^{n-2} x_{i+1}\beta_{n-1-i} + (c + \sum_{i=0}^{n-1} p_ix_i)\beta_0 \\ &= \sum_{i=1}^{n-1} x_i\beta_{n-i} + (c + \sum_{i=0}^{n-1} p_ix_i)\beta_0 \end{aligned} \tag{3.11}$$

Now,

$$\begin{aligned} \alpha Y &= \sum_{i=1}^{n-1} x_i\alpha\beta_{n-i} + (c + \sum_{i=0}^{n-1} p_ix_i)\beta_{n-1} \\ &= \sum_{i=1}^{n-1}(x_i\beta_{n-1-i} + x_ip_{n-i}\beta_{n-1}) \\ &\quad + (c + \sum_{i=0}^{n-1} p_ix_i)\beta_{n-1} \\ &= \sum_{i=0}^{n-1} x_i\beta_{n-1-i} + c'\beta_{n-1}, \end{aligned} \tag{3.12}$$

where $c' \in \{0, 1\}$ denotes

$$c' = c + \sum_{i=0}^{n-1}(p_i + p_{n-i})x_i. \tag{3.13}$$

Note that

$$\begin{aligned} \psi(m, X) &= (n - m, \sum_{i=0}^{n-1} x_i\beta_{n-1-i}) \\ &= (n - m, \alpha Y + c'\beta_{n-1}). \end{aligned} \tag{3.14}$$

Table 3.4: Automorphism $\psi(\cdot) : B_4 \to B_4$.

(m, X)	$\psi(m, X)$	(m, X)	$\psi(m, X)$	(m, X)	$\psi(m, X)$	(m, X)	$\psi(m, X)$
$(0, 0)$	$(0, 0)$	$(1, 0)$	$(3, 0)$	$(2, 0)$	$(2, 0)$	$(3, 0)$	$(1, 0)$
$(0, 1)$	$(0, \alpha^{14})$	$(1, 1)$	$(3, \alpha^{14})$	$(2, 1)$	$(2, \alpha^{14})$	$(3, 1)$	$(1, \alpha^{14})$
$(0, \alpha)$	$(0, \alpha^2)$	$(1, \alpha)$	$(3, \alpha^2)$	$(2, \alpha)$	$(2, \alpha^2)$	$(3, \alpha)$	$(1, \alpha^2)$
$(0, \alpha^2)$	$(0, \alpha)$	$(1, \alpha^2)$	$(3, \alpha)$	$(2, \alpha^2)$	$(2, \alpha)$	$(3, \alpha^2)$	$(1, \alpha)$
$(0, \alpha^3)$	$(0, \alpha^3)$	$(1, \alpha^3)$	$(3, \alpha^3)$	$(2, \alpha^3)$	$(2, \alpha^3)$	$(3, \alpha^3)$	$(1, \alpha^3)$
$(0, \alpha^4)$	$(0, \alpha^{13})$	$(1, \alpha^4)$	$(3, \alpha^{13})$	$(2, \alpha^4)$	$(2, \alpha^{13})$	$(3, \alpha^4)$	$(1, \alpha^{13})$
$(0, \alpha^5)$	$(0, \alpha^5)$	$(1, \alpha^5)$	$(3, \alpha^5)$	$(2, \alpha^5)$	$(2, \alpha^5)$	$(3, \alpha^5)$	$(1, \alpha^5)$
$(0, \alpha^6)$	$(0, \alpha^9)$	$(1, \alpha^6)$	$(3, \alpha^9)$	$(2, \alpha^6)$	$(2, \alpha^9)$	$(3, \alpha^6)$	$(1, \alpha^9)$
$(0, \alpha^7)$	$(0, \alpha^8)$	$(1, \alpha^7)$	$(3, \alpha^8)$	$(2, \alpha^7)$	$(2, \alpha^8)$	$(3, \alpha^7)$	$(1, \alpha^8)$
$(0, \alpha^8)$	$(0, \alpha^7)$	$(1, \alpha^8)$	$(3, \alpha^7)$	$(2, \alpha^8)$	$(2, \alpha^7)$	$(3, \alpha^8)$	$(1, \alpha^7)$
$(0, \alpha^9)$	$(0, \alpha^6)$	$(1, \alpha^9)$	$(3, \alpha^6)$	$(2, \alpha^9)$	$(2, \alpha^6)$	$(3, \alpha^9)$	$(1, \alpha^6)$
$(0, \alpha^{10})$	$(0, \alpha^{12})$	$(1, \alpha^{10})$	$(3, \alpha^{12})$	$(2, \alpha^{10})$	$(2, \alpha^{12})$	$(3, \alpha^{10})$	$(1, \alpha^{12})$
$(0, \alpha^{11})$	$(0, \alpha^{11})$	$(1, \alpha^{11})$	$(3, \alpha^{11})$	$(2, \alpha^{11})$	$(2, \alpha^{11})$	$(3, \alpha^{11})$	$(1, \alpha^{11})$
$(0, \alpha^{12})$	$(0, \alpha^{10})$	$(1, \alpha^{12})$	$(3, \alpha^{10})$	$(2, \alpha^{12})$	$(2, \alpha^{10})$	$(3, \alpha^{12})$	$(1, \alpha^{10})$
$(0, \alpha^{13})$	$(0, \alpha^4)$	$(1, \alpha^{13})$	$(3, \alpha^4)$	$(2, \alpha^{13})$	$(2, \alpha^4)$	$(3, \alpha^{13})$	$(1, \alpha^4)$
$(0, \alpha^{14})$	$(0, 1)$	$(1, \alpha^{14})$	$(3, 1)$	$(2, \alpha^{14})$	$(2, 1)$	$(3, \alpha^{14})$	$(1, 1)$

From (3.10) and (3.14) it is obvious that vertex $\psi(m, X)$ is connected to vertex $\psi(m + 1, \alpha X + c\beta_{n-1})$, $c \in \{0, 1\}$. ■

When the context is clear, we sometimes write $\psi(X)$ in place of $\psi(m, X)$. Theorem 25 lists some basic properties of $\psi(\cdot)$.

Theorem 5 *1. $\psi(\cdot)$ is an order 2 automorphism.*

2. $\psi(X_1 + X_2) = \psi(X_1) + \psi(X_2)$.

3. $\psi((m, X)) = (n - m, X)$ for exactly $2^{\lceil n/2 \rceil}$ values of $X \in GF(2^n)$.

Proof. The first two properties of $\psi(\cdot)$ are obvious from its definition. For any $X = \sum_{i=0}^{n-1} x_i \beta_i$, $\psi((m, X)) = (n - m, X)$ if and only if $x_i = x_{n-1-i}$, $0 \le i < \lfloor n/2 \rfloor$. From this the third property follows. ■

Automorphism $\psi(\cdot) : B_4 \to B_4$ is shown in Table 3.4.

We end this section with the following theorem enumerating all the automorphisms of BF$_n$.

Theorem 6 *BF$_n$ has a total of $n2^{n+1}$ automorphisms.*

Proof. Note that the product of two automorphisms is also an automorphism. Thus in addition to the $n2^n$ automorphisms defined by Theorem 1, another set of automorphisms can be defined by multiplying each of these $\phi(\cdot)$s by the automorphism $\psi(\cdot)$ in Theorem 4. Since the order of automorphism $\psi(\cdot)$ is 2, these are all the automorphisms of BF$_n$. ∎

3.3 Edge Transformations by automorphisms

This section investigates the effect of an automorphism on the butterfly edges. We call edges $(i-1, X) \rightarrow (i, \alpha X)$ and $(i-1, X) \rightarrow (i, \alpha X + \beta_{n-1})$ for all $X \in GF(2^n)$ as the edges in the ith column of BF$_n$.

The automorphism $\phi(\cdot)$ of Theorem 1 affects all the edges in the same column similarly. This is stated in the following theorem.

Theorem 7 *Let the automorphism offsets be related as:*

$$K_i = \alpha K_{(i-1) mod\ n} + c_i\beta_{n-1}, \qquad 0 \leq i \leq n-1,$$

(a) If $c_i = 1$, then the automorphism $\phi(\cdot)$ maps all f edges of BF_n in column i to g edges and all g edges to f edges.
(b) If $c_i = 0$, then the automorphism $\phi(\cdot)$ maps all f edges of BF_n in column i to f edges and all g edges to g edges.

Proof. Consider an f edge between nodes $N_1 = (i-1, X)$ and $N_2 = (i, \alpha X)$ of the sub-graph of BF$_n$. Now, $\phi(N_1) = (i-1, X + K_{i-1})$ and,

$$\begin{aligned}\phi(N_2) &= (i, \alpha X + K_i) \\ &= (i, \alpha X + \alpha K_{i-1} + c_i\beta_{n-1}) \\ &= (i, \alpha(X + K_{i-1}) + \beta_{n-1})\end{aligned}$$

From this, one can clearly see that the edge between $\phi(N_1)$ and $\phi(N_2)$ is a g edge. The translation of a g edge into an f edge can be similarly proved. ∎

Note that the automorphism $\phi(m,\ X) = (m+t,\ X+K_m)$ also advances the column number m by quantity t. In this case, $c_m = 1$ has the effect of mapping the f edges of the sub-graph between columns $m-1$ and m to g edges and all g edges to f edges; but these transformed edges now appear in column $m+t$. Similarly the edges in mth column are mapped to edges of the same type in column $m+t$ if $c_m = 0$.

We will show in the next section how Theorem 7 is helpful in avoiding faulty edges in a mapping.

To describe the effect of the automorphism $\psi(\cdot)$ on the edges of BF_n, we first define a set S as

$$S = \{X \in GF(2^n) | \psi(X) = \alpha\psi(\alpha X)\} \tag{3.15}$$

The types of edges from any element of S are preserved by ψ. (See Theorem 9). Some of the basic properties of S are listed in the following theorem.

Theorem 8 *Let p_i denote the coefficient of x^i in the primitive polynomial used to generate $GF(2^n)$. Then*

1. *$X = \sum_{i=0}^{n-1} x_i\beta_i \in S$ if and only if $\sum_{i=1}^{n-1}\ x_i\ (p_i + p_{n-i}) = 0$.*

2. *For any $X \notin S$, $\alpha\psi(\alpha X) + \psi(X) = \beta_{n-1}$.*

3. *If $p_i = p_{n-i}$, then $\beta_i \in S$,*

4. *S is a subgroup of $GF(2^n)$ under the operation of addition.*

5. *There are exactly 2^{n-1} elements in S.*

Proof. Using Lemma 1 one gets

$$\alpha X = \sum_{i=0}^{n-2} \alpha x_i\beta_i = x_0\beta_{n-1} + \sum_{i=1}^{n-1} x_i(\beta_{i-1} + p_i\beta_{n-1} = \sum_{i=1}^{n-1} x_i\beta_{n-i} + \beta_{n-1}\sum_{i=0}^{n-1} p_i x_i).$$

Therefore, $$\psi(\alpha X) = \sum_{i=0}^{n-2} x_{i+1}\beta_{n-1-i} + \beta_0 \sum_{i=0}^{n-1} p_i x_i.$$

A multiplication by α gives

$$\begin{aligned}\alpha\psi(\alpha X) &= \sum_{i=1}^{n-1} x_i(\beta_{n-i-1} + p_{n-i}\beta_{n-1}) + \beta_{n-1}\sum_{i=0}^{n-1} p_i x_i \\ &= \sum_{i=1}^{n-1} x_i\beta_{n-1-i} + x_0\beta_{n-1} + \beta_{n-1}(\sum_{i=1}^{n-1} x_i(p_i + p_{n-i})). \qquad (3.16)\end{aligned}$$

Adding (3.16) to

$$\psi(X) = \sum_{i=0}^{n-1} x_i\beta_{n-1-i}$$

gives

$$\alpha\psi(\alpha X) + \psi(X) = \beta_{n-1}\sum_{i=1}^{n-1} x_i(p_i + p_{n-i}). \qquad (3.17)$$

Since all x_i and p_i belong to $GF(2)$, the right hand side of (3.17) is either 0 or β_{n-1}. The first two parts of the theorem therefore follow directly from (3.17).

To prove the third part, note that Lemma 1 implies

$$\begin{aligned}\alpha\psi(\alpha\beta_i) &= \alpha\psi(\beta_{i-1} + p_i\beta_{n-1}) = \alpha(\beta_{n-i} + p_i\beta_0) \\ &= \beta_{n-1-i} + p_{n-i}\beta_{n-1} + p_i\beta_{n-1}. \qquad (3.18)\end{aligned}$$

For $\beta_i \in S$, one needs to have $\psi(\beta_i) = \alpha\psi(\alpha\beta_i)$. Comparing $\psi(\beta_i) = \beta_{n-1-i}$ with (3.18) gives the required result.

To prove that S is a group under addition, all one needs to show is that $0 \in S$ and that S is closed under addition. The first of these is obvious. To show the closure, let $X_1, X_2 \in S$. Then

$$\alpha\psi(\alpha(X_1 + X_2)) = \alpha(\psi(\alpha X_1) + \psi(\alpha X_2)) = \alpha\psi(\alpha X_1) + \alpha\psi(\alpha X_2) = \alpha\psi(\alpha X_1 + \alpha X_2).$$

This implies that $X_1 + X_2 \in S$.

To prove the last part of the theorem, we first show that it is impossible to have a primitive polynomial of degree $n \geq 3$ over $GF(2)$, with all $p_i = p_{n-i}$, $0 \leq i \leq n$. To prove this by contradiction, suppose there is a primitive polynomial $p(x)$ of degree ≥ 3 with coefficients satisfying $p_i = p_{n-i}$ for all $0 \leq i \leq n$. Let α be the primitive root of $p(x)$. Then all its roots of are given as α^{2^i}, $0 \leq i < n$. Now consider a polynomial $g(x) = x^n f(x^{-1})$. Clearly, $g(\alpha^{-1}) = 0$ showing that α^{-1} is a root of $g(x)$. However, because of the assumed relationships between the coefficients of $p(x)$, $g(x) = p(x)$. Thus $\alpha^{-1} = \alpha^{2^i}$, or $\alpha^{2^i+1} = 1$ for some i, $0 \leq i < n$. Now since α is a primitive element, the smallest power of α that gives a 1 is $2^n - 1$. Thus $2^i + 1$ is a multiple of $(2^n - 1)$. But this is impossible for $n \geq 3$ because $i < n$. Therefore it is impossible for a primitive polynomial of degree ≥ 3 to have $p_i = p_{n-i}$ for all $0 \leq i \leq n$. As a result of this, the first part of the theorem implies that there is a linear relationship between the components of X when $X \in S$. Thus one can choose all but one component of X independently. The choice of $n - 1$ independent components, each in $GF(2)$, implies that there are exactly 2^{n-1} elements $X \in S$. ∎

Set S plays an important role in edge transformations of BF_n under $\psi(\cdot)$ as the following theorem shows.

Theorem 9 *When $X \in S$, ψ maps f edges from (m, X) to f edges and g edges to g edges. On the other hand, when $X \notin S$, ψ maps f edges from (m, X) to g edges and g edges to f edges.*

Proof. Consider an edge $(m, X) \rightarrow (m + 1, \alpha X + c\beta_{n-1})$. If $c = 0$, this represents an f edge and if $c = 1$, a g edge. The automorphism maps the first node to $N_1 = (n - m, \psi(X))$ and the second to $N_2 = (n - m - 1, \alpha^{-1}\psi(X) + c\beta_0)$ if $X \in S$. Clearly there is an f edge from N_2 to N_1 when $c = 0$ and a g edge when $c = 1$.

If $X \notin S$, then from the second part of Theorem 8, one can see that the second node maps to $N_2' = (n - m - 1, \alpha^{-1}\psi(X) + \alpha^{-1}\beta_{n-1} + c\beta_0) = (n - m - 1, \alpha^{-1}\psi(X) + (c + 1)\beta_0)$. Thus there is a g edge from N_2' to N_1 when $c = 0$ and an f edge when $c = 1$. ∎

As a consequence of Theorem 8, we have the following result.

Theorem 10 *Automorphism $\psi(\cdot)$ maps edges from exactly half the rows of the butterfly to the edges of the same type.*

Proof. Theorem 9 shows that edges starting from nodes in the same row (i.e., nodes (m, X) having the same X) behave similarly; all of them either map to edges of the same type (when $X \in S$) or map to edges of the other type (when $X \notin S$). The stated result is true because $|S| = 2^{n-1}$ (Theorem 8, Part 5). ∎

3.4 Application of automorphisms to tolerate edge faults

Previously automorphisms have only been used to tolerate node faults. However, Theorems 7 and 9 directly express the effect of an automorphism on the butterfly edges. Consequently, one can now use these automorphisms to tolerate edge faults for many mappings on the butterfly.

The general procedure to obtain a fault free mapping on a faulty butterfly is simple. If some edges used in the mapping are faulty but the edges to which they *can* be mapped by *some* automorphism are free, then applying that automorphism to the mapping will allow it to use only fault-free edges. Note that much of the power of this method is due to the fact that we have $n2^{n+1}$ well-defined and simple automorphisms that map edges in a deterministic fashion. We illustrate this procedure by constructing a Hamilton cycle under various edge fault scenarios.

Theorem 11 *If the edges in one of the columns of BF_n are fault free and the faults in each of the other columns are limited to only one type of edges, then BF_n is Hamiltonian.*

Proof. As shown in [10], it is possible to construct a Hamiltonian cycle in BF_n by first constructing two cycles using only f edges; one linking all nodes (m, X), $X \neq 0$, and another linking all nodes $(m, 0)$. These cycles are merged into a Hamiltonian cycle by using a pair of g edges in column t: $(t-1, 0) \to (t, \beta_{n-1})$ and $(t-1, \beta_0)$

and $(t, 0)$. With $0 \leq t < n$, there are n such independent pairs of g edges that may be used to merge the cycles. We will use the g edges in the column of BF_n that has no faults. We now show that one can design an automorphism $\phi : B_n \to B_n$ which will avoid all faults. To construct ϕ, we compute constants c_i, $0 \leq i < n$ such that

$$c_i = \begin{cases} 1 & \text{if there is a fault in } f \text{ edge in column } i \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

One can then get K_0 by (3.6) as

$$K_0(\alpha^n + 1)\beta_{n-1}^{-1} = \sum_{j=0}^{n-1} c_{(-j)\ mod\ n}\alpha^j.$$

The other K_i values can then be inferred from (3.3). Theorem 7 then shows that the Hamilton cycle will use f edges in columns where f edges are fault free and g edges where f edges have faults. Thus the transformed Hamiltonian cycle will not have any faulty edges. ■

To illustrate Theorem 11, consider a butterfly B_4 shown in Fig. 3.1 with faults in columns 0 and 1 restricted to f edges and in column 2 to g edges. Edges in column 3 are fault free. Clearly in this case, $c_0 = c_1 = 1$ and $c_2 = c_3 = 0$. This gives from (3.19), $K_0 = \alpha^{13}$, $K_1 = \alpha^3$, $K_2 = \alpha^4$ and $K_3 = \alpha^5$. By following the procedure of Theorem 11 we first create the original Hamilton cycle as:

$$\begin{array}{llllll}
(0,1) & \to (1,\alpha) & \to (2,\alpha^2) & \to (3,\alpha^3) & \to (0,\alpha^4) & \to \\
(1,\alpha^5) & \to (2,\alpha^6) & \to (3,\alpha^7) & \to (0,\alpha^8) & \to (1,\alpha^9) & \to \\
(2,\alpha^{10}) & \to (3,\alpha^{11}) & \to (0,\alpha^{12}) & \to (1,\alpha^{13}) & \to (2,\alpha^{14}) & \to \\
(3,0) & \to (0,0) & \to (1,0) & \to (2,0) & \to (3,1) & \to \\
(0,\alpha) & \to (1,\alpha^2) & \to (2,\alpha^3) & \to (3,\alpha^4) & \to (0,\alpha^5) & \to \\
(1,\alpha^6) & \to (2,\alpha^7) & \to (3,\alpha^8) & \to (0,\alpha^9) & \to (1,\alpha^{10}) & \to \\
(2,\alpha^{11}) & \to (3,\alpha^{12}) & \to (0,\alpha^{13}) & \to (1,\alpha^{14}) & \to (2,1) & \to \\
(3,\alpha) & \to (0,\alpha^2) & \to (1,\alpha^3) & \to (2,\alpha^4) & \to (3,\alpha^5) & \to \\
(0,\alpha^6) & \to (1,\alpha^7) & \to (2,\alpha^8) & \to (3,\alpha^9) & \to (0,\alpha^{10}) & \to
\end{array}$$

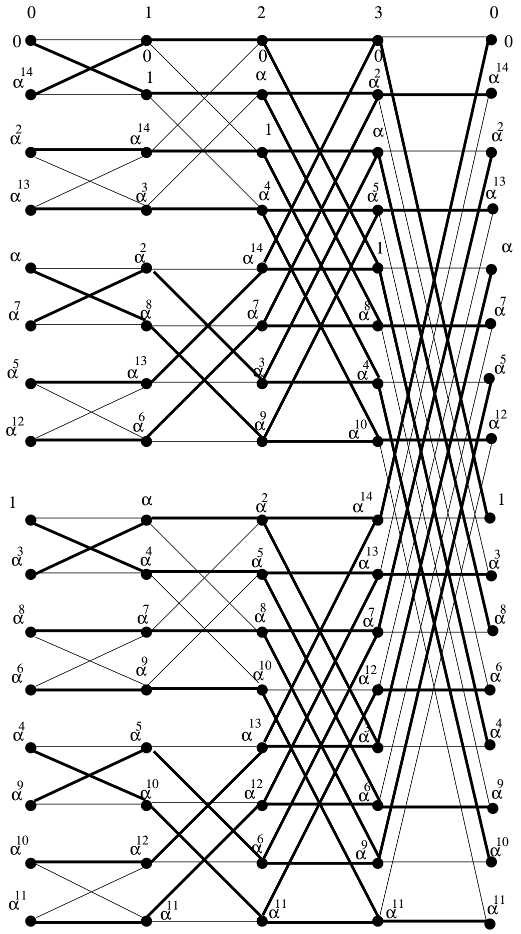

Figure 3.1: Butterfly B_4 with faulty edges marked with light lines and fault-free edges with dark lines. The column numbers are at the top and the row index of each node is marked next to the node.

$$\begin{array}{llllll}
(1,\alpha^{11}) & \rightarrow (2,\alpha^{12}) & \rightarrow (3,\alpha^{13}) & \rightarrow (0,\alpha^{14}) & \rightarrow (1,1) & \rightarrow \\
(2,\alpha) & \rightarrow (3,\alpha^{2}) & \rightarrow (0,\alpha^{3}) & \rightarrow (1,\alpha^{4}) & \rightarrow (2,\alpha^{5}) & \rightarrow \\
(3,\alpha^{6}) & \rightarrow (0,\alpha^{7}) & \rightarrow (1,\alpha^{8}) & \rightarrow (2,\alpha^{9}) & \rightarrow (3,\alpha^{10}) & \rightarrow \\
(0,\alpha^{11}) & \rightarrow (1,\alpha^{12}) & \rightarrow (2,\alpha^{13}) & \rightarrow (3,\alpha^{14}) & \rightarrow (0,1) &
\end{array}$$

By applying the automorphism offsets already calculated, one can then obtain the required fault-free Hamilton cycle as:

$$\begin{array}{llllll}
(0,\alpha^{6}) & \rightarrow (1,\alpha^{9}) & \rightarrow (2,\alpha^{10}) & \rightarrow (3,\alpha^{11}) & \rightarrow (0,\alpha^{11}) & \rightarrow \\
(1,\alpha^{11}) & \rightarrow (2,\alpha^{12}) & \rightarrow (3,\alpha^{13}) & \rightarrow (0,\alpha^{3}) & \rightarrow (1,\alpha) & \rightarrow \\
(2,\alpha^{2}) & \rightarrow (3,\alpha^{3}) & \rightarrow (0,\alpha) & \rightarrow (1,\alpha^{8}) & \rightarrow (2,\alpha^{9}) & \rightarrow \\
(3,\alpha^{5}) & \rightarrow (0,\alpha^{13}) & \rightarrow (1,\alpha^{3}) & \rightarrow (2,\alpha^{4}) & \rightarrow (3,\alpha^{10}) & \rightarrow \\
(0,\alpha^{12}) & \rightarrow (1,\alpha^{6}) & \rightarrow (2,\alpha^{7}) & \rightarrow (3,\alpha^{8}) & \rightarrow (0,\alpha^{7}) & \rightarrow \\
(1,\alpha^{2}) & \rightarrow (2,\alpha^{3}) & \rightarrow (3,\alpha^{4}) & \rightarrow (0,\alpha^{10}) & \rightarrow (1,\alpha^{12}) & \rightarrow \\
(2,\alpha^{13}) & \rightarrow (3,\alpha^{14}) & \rightarrow (0,0) & \rightarrow (1,1) & \rightarrow (2,\alpha) & \rightarrow \\
(3,\alpha^{2}) & \rightarrow (0,\alpha^{14}) & \rightarrow (1,0) & \rightarrow (2,0) & \rightarrow (3,0) & \rightarrow \\
(0,1) & \rightarrow (1,\alpha^{4}) & \rightarrow (2,\alpha^{5}) & \rightarrow (3,\alpha^{6}) & \rightarrow (0,\alpha^{9}) & \rightarrow \\
(1,\alpha^{5}) & \rightarrow (2,\alpha^{6}) & \rightarrow (3,\alpha^{7}) & \rightarrow (0,\alpha^{2}) & \rightarrow (1,\alpha^{14}) & \rightarrow \\
(2,1) & \rightarrow (3,\alpha) & \rightarrow (0,\alpha^{8}) & \rightarrow (1,\alpha^{7}) & \rightarrow (2,\alpha^{8}) & \rightarrow \\
(3,\alpha^{9}) & \rightarrow (0,\alpha^{5}) & \rightarrow (1,\alpha^{13}) & \rightarrow (2,\alpha^{14}) & \rightarrow (3,1) & \rightarrow \\
(0,\alpha^{4}) & \rightarrow (1,\alpha^{10}) & \rightarrow (2,\alpha^{11}) & \rightarrow (3,\alpha^{12}) & \rightarrow (0,\alpha^{6}) &
\end{array}$$

Theorem 11 is interesting because it implies that up to 2^{n-1} edges of the same type may be faulty in up to $n-1$ columns and the faulty butterfly is still Hamiltonian. It is easy to extend this idea to any other mapping also. A direct result of Theorem 11 is the following result.

Corollary 1 *A butterfly with $n-1$ edge faults distributed one per column is Hamiltonian.*

We now give a new alternate simple proof of a previous known result [43] based on our work discussed in the previous section.

Theorem 12 *Graph BF_n with up to 2 random edge faults is Hamiltonian.*

Proof. If there is only one fault or if there are two faults, both in the same type (f or g) of edges, or if they are in two different columns, then by Theorem 11 one can generate a Hamiltonian cycle for BF_n. Thus we only need to treat cases that involve two faulty edges of different types (one f and one g) in the same column.

Consider now the case of an f and a g faulty edge in the same column m such that they do not share a node. Let the faulty f edge be $(m-1, X) \rightarrow (m, \alpha X)$, $X \neq 0$. In this case, one can first create a cycle containing all the nodes (i, X), $0 \leq i \leq n-1$, $X \neq 0$ using only the f edges. Clearly this cycle avoids the faulty g edge. Further, it can be easily modified to avoid the faulty f edge. To achieve this, add the g edges $(m-1, X) \rightarrow (m, \alpha X + \beta_{n-1})$ and $(m-1, X+\beta_0) \rightarrow (m, \alpha X)$ and remove the f edges $(m-1, X) \rightarrow (m, \alpha X)$ and $(m-1, X+\beta_0) \rightarrow (m, \alpha X + \beta_{n-1})$ as shown in Fig. 3.2. This removes the faulty f edge from the cycle, but partitions it into two disjoint cycles.

We now show that there exist g edges (shown as horizontal lines in Fig. 3.2) connecting the two parts which can be used to rejoin the two halves and create a single cycle of all the nodes (m, X), $X \neq 0$ without any faulty edge. It is easy to see that the number of nodes in each part is a multiple of n, and in fact, is at least $2n$. Let k be any integer between 0 and $n-1$ other than $m-1$ or $m-2 \mathrm{mod}\ n$. This is always possible because $n \geq 3$. Since there are exactly $2^n - 1$ nodes with first index k in the two cycles, one of the cycles will have an odd number of such nodes. Without loss of generality, assume that it is the right cycle. Consider a typical node (k, Y) in this cycle. If node $(k+1, \alpha y + \beta_{n-1})$ also belongs to the same cycle, then the g edge from $(k, Y) \rightarrow (k+1, \alpha y + \beta_{n-1})$ will end up in the same cycle. At the same time, the node $(k, Y+\beta_0)$ which belongs to the same cycle will have a g edge going to $(k+1, \alpha Y)$ in the same cycle. Thus the g edges starting from that cycle and ending up in the same cycle occur in pairs. Since there are odd number of nodes with first index k, one of these nodes, say (k, Y), will have a g edge to the node $(k+1, \alpha Y + \beta_{n-1})$ in the left cycle. Further, the node $(k, Y+\beta_0)$ from the left cycle has a g edge ending up at $(k+1, \alpha Y)$ in the right cycle. Using this pair of g edges, one can create a cycle of all nodes (i, X), $X \neq 0$ without using any faulty edge as shown in Fig. 3.2. Note that because the f and g edge faults are not incident on the same node, none of the g edges used here are faulty.

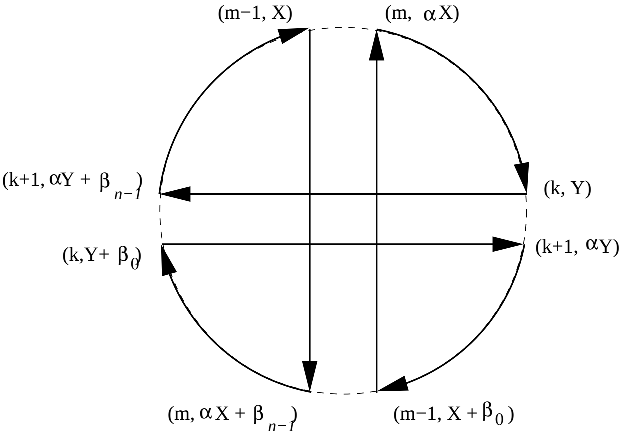

Figure 3.2: Fault free cycle of all nodes (i, X), $0 \leq i \leq n-1$, $X \neq 0$ when an f edge $(m-1, X) \rightarrow (m, \alpha X)$ is faulty.

To add the rest of the BF$_n$ nodes to the Hamiltonian cycle, one can build the cycle of all the nodes $(i, 0)$, $0 \leq i \leq n-1$, using faultless f edges and merge it with the cycle in Fig. 3.2 using g edges in any column other than m.

If the faulty f edge is $(m-1, 0) \rightarrow (m, 0)$, then one can create a cycle of all nodes (t, X), $X \neq 0$ using faultless f edges, and of all nodes $(t, 0)$ using f edges. The faulty f edge will be in the second cycle. Merging the two cycles gets rid of the faulty edge.

Finally, consider the case of the faulty f and g edges in the same column and also sharing a node. We initially construct a Hamiltonian cycle considering that both f and g edges from the node $(0, 0)$ to be faulty. This cycle can then be translated using an appropriate automorphism to one that avoids f and g edges from any node.

We first partition the butterfly nodes into three sets of nodes connected by f edges as follows.

Set 1: $(1, 0) \xrightarrow{f} (2, 0) \xrightarrow{f} (3, 0) \xrightarrow{f} \cdots (0, 0)$.

Set 2: $(0, \beta_0) \xrightarrow{f} (1, \beta_{n-1}) \xrightarrow{f} (2, \alpha\beta_{n-1}) \xrightarrow{f} \cdots (n-1, \beta_0)$.

Set 3: $(0, \beta_{n-1}) \xrightarrow{f} (1, \alpha\beta_{n-1}) \xrightarrow{f} (2, \alpha^2\beta_{n-1}) \xrightarrow{f} \cdots (n-1, \alpha^{-1}\beta_0)$.

Note that sets 2 and 3 when joined together give the cycle of length $n2^n - n$ containing all the nodes with nonzero second coordinates obtained by continuously traveling along the f edges. Set 1 contains all the BF$_n$ nodes with their second coordinate 0. We can connect sets 1 and 2 into a cycle because their endpoints are connected by g edges. In particular, $(0, \beta_0) \xrightarrow{g} (1, 0)$ and $(n-1, \beta_0) \xrightarrow{g} (0, 0)$. The nodes in Set 3 can be incorporated in this cycle if the two end nodes of Set 3 are connected to some two consecutive nodes in the cycle. Note that the end nodes of Set 3 have the following connectivity: $(0, \beta_{n-1}) \xrightarrow{g} (1, \alpha\beta_{n-1} + \beta_{n-1})$ and $(n-1, \alpha^{-1}\beta_0) \xrightarrow{g} (0, \beta_0 + \beta_{n-1})$. One can verify that $(0, \beta_0 + \beta_{n-1}) \xrightarrow{f} (1, \alpha\beta_{n-1} + \beta_{n-1})$. Thus if node $(0, \beta_0 + \beta_{n-1})$ is in Set 2, then one can remove the f edge between this node and the next, and instead connect the nodes of Set 3 into the cycle using the g edges noted here. The resultant cycle is shown in Fig. 3.3.

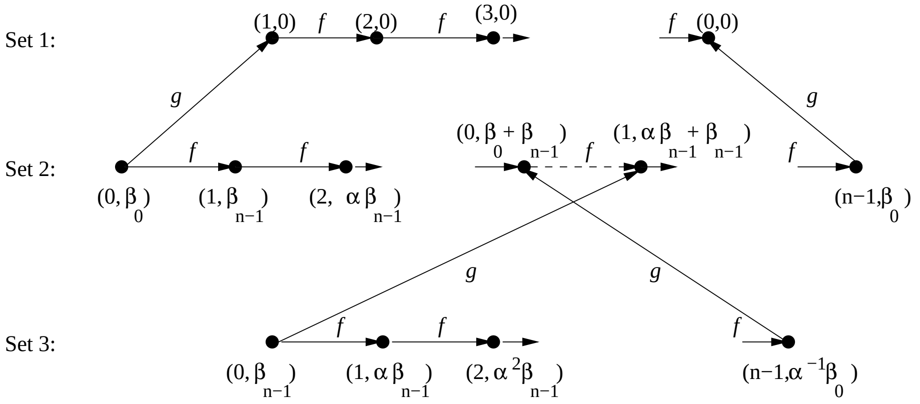

Figure 3.3: The Hamiltonian cycle when the f and g edges from $(0, 0)$ are faulty and the node $(0, \beta_0 + \beta_{n-1})$ is in Set 2. Note that all edges are bidirectional and the dashed f edge is not part of the cycle.

On the other hand, if the node $(0, \beta_0 + \beta_{n-1})$ is in Set 3 rather than in Set 2, then the g edges from the endpoints of Set 3 go to adjacent nodes of Set 3, namely the nodes $(1, \alpha\beta_{n-1} + \beta_{n-1})$ and $(0, \beta_0 + \beta_{n-1})$. By removing the f edge between these adjacent nodes and adding the g edges from the endpoints, one can see that all the nodes of Set 3 form a cycle. To show that this cycle can be merged with the cycle formed by the nodes in Sets 1 and 2, we show that there is some (m, β_0) in cycle 3 with $m \neq 0$ and $m + 1 \neq 0$. Because then, one can drop edge $(m, \beta_0) \xrightarrow{f}$

$(m+1, \beta_{n-1})$ in the cycle of Set 3 and instead use connections to merge this cycle with Set 1 using edges $(m, \beta_0) \xrightarrow{g} (m+1, 0)$ and $(m, 0) \xrightarrow{g} (m+1, \beta_{n-1})$. To see that such a node (m, β_0) exist in Set 3, note that the number of nodes in Set 3 is at least $2^n - 1$. In other words, the second coordinate of the nodes in Set 3 take all possible nonzero values. Consequently, there will be some (m, β_0) present in Set 3. Further, both $(0, \beta_0)$ and $(n-1, \beta_0)$ are in Set 2, showing $m \neq 0, n-1$. Thus nodes in Set 3 can also be merged in the cycle formed by nodes in Sets 1 and 2. This gives the required Hamiltonian cycle. ∎

Theorem 12 can be illustrated by mapping a Hamiltonian cycle in a faulty B_4 in case of the faulty f and g edges in the same column and sharing a node. Assume that the faulty edges are $(1, \alpha^6) \xrightarrow{f} (2, \alpha^7)$ and $(1, \alpha^6) \xrightarrow{g} (2, \alpha^9)$. We first construct a hamiltonian cycle considering that both f and g edges from node $(0,0)$ to be faulty as in Fig.3.3. Then we apply an automorphism $\phi(\cdot) : B_4 \to B_4$ which maps node $(0,0)$ to node $(1, \alpha^6)$. For this ϕ, the column offset $t = (1-0) \text{mod } 4 = 1$ and the automorphism offset $K_0 = 0 + \alpha^6 = \alpha^6$. The cycle obtained is shown in Fig.3.4. Further, from(3.6),

$$\sum_{j=0}^{n-1} c_{(1-j) \text{mod } n} \alpha^j = K_0 \sigma \beta_3^{-1} = \alpha^3 + \alpha + 1.$$

Thus one gets $c_0 = 1$, $c_1 = 1$, $c_2 = 0$ and $c_3 = 1$ and consequently, $K_1 = \alpha^9$, $K_2 = \alpha^{10}$ and $K_3 = \alpha^{12}$. The resultant cycle after applying this automorphism to cycle in Fig.3.4 is shown in Fig.3.4.

Earlier we gave Theorem 11 to deal with faults of the same kind in columns, The following theorem deals with faults of the same kind in rows (Recall that nodes with the same second index are deemed to be in the same row).

The next five theorems use the set S of rows. Equation (3.15) defines S. Theorem 9 shows that ψ does not convert the types of edges.

Theorem 13 *If the edges in rows* 0 *and* β_0 *of* BF_n *are fault-free, the faults in other rows* $X \in S$ *are restricted to* g *edges and those in rows* $X \notin S$ *are restricted to only one type of edges, then* BF_n *is Hamiltonian.*

$$\begin{array}{lllll}
(3,\alpha^{14}) & \to (2,\alpha^{13}) & \to (1,\alpha^{12}) & \to (0,\alpha^{11}) & \to (3,\alpha^{10}) \to \\
(2,\alpha^{9}) & \to (1,\alpha^{8}) & \to (0,\alpha^{7}) & \to (3,\alpha^{6}) & \to (2,\alpha^{5}) \to \\
(1,\alpha^{4}) & \to (0,1) & \to (1,\alpha) & \to (2,\alpha^{2}) & \to (3,\alpha^{3}) \to \\
(0,\alpha^{4}) & \to (1,\alpha^{5}) & \to (2,\alpha^{6}) & \to (3,\alpha^{7}) & \to (0,\alpha^{8}) \to \\
(1,\alpha^{9}) & \to (2,\alpha^{10}) & \to (3,\alpha^{11}) & \to (0,\alpha^{12}) & \to (1,\alpha^{13}) \to \\
(2,\alpha^{14}) & \to (3,1) & \to (0,\alpha) & \to (1,\alpha^{2}) & \to (2,\alpha^{3}) \to \\
(3,\alpha^{4}) & \to (0,\alpha^{5}) & \to (1,\alpha^{6}) & \to (2,\alpha^{7}) & \to (3,\alpha^{8}) \to \\
(0,\alpha^{9}) & \to (1,\alpha^{10}) & \to (2,\alpha^{11}) & \to (3,\alpha^{12}) & \to (0,\alpha^{13}) \to \\
(1,\alpha^{14}) & \to (2,1) & \to (3,\alpha) & \to (0,\alpha^{2}) & \to (1,\alpha^{3}) \to \\
(2,\alpha^{4}) & \to (3,\alpha^{5}) & \to (0,\alpha^{6}) & \to (1,\alpha^{7}) & \to (2,\alpha^{8}) \to \\
(3,\alpha^{9}) & \to (0,\alpha^{10}) & \to (1,\alpha^{11}) & \to (2,\alpha^{12}) & \to (3,\alpha^{13}) \to \\
(0,\alpha^{3}) & \to (3,\alpha^{2}) & \to (2,\alpha) & \to (1,1) & \to (0,\alpha^{14}) \to \\
(1,0) & \to (2,0) & \to (3,0) & \to (0,0) & \to (3,\alpha^{14})
\end{array}$$

Figure 3.4: Hamiltonian cycle in B_4 avoiding faulty f and g edges from $(0,0)$.

$$\begin{array}{lllll}
(0,\alpha^{5}) & \to (3,\alpha^{9}) & \to (2,\alpha^{8}) & \to (1,\alpha) & \to (0,\alpha^{3}) \to \\
(3,\alpha^{13}) & \to (2,\alpha^{12}) & \to (1,\alpha^{10}) & \to (0,\alpha^{14}) & \to (3,1) \to \\
(2,\alpha^{14}) & \to (1,\alpha^{13}) & \to (2,\alpha^{3}) & \to (3,\alpha^{4}) & \to (0,\alpha^{10}) \to \\
(1,\alpha^{12}) & \to (2,\alpha^{6}) & \to (3,\alpha^{7}) & \to (0,\alpha^{2}) & \to (1,\alpha^{14}) \to \\
(2,0) & \to (3,0) & \to (0,1) & \to (1,\alpha^{4}) & \to (2,\alpha^{10}) \to \\
(3,\alpha^{11}) & \to (0,\alpha^{11}) & \to (1,\alpha^{11}) & \to (2,\alpha^{11}) & \to (3,\alpha^{12}) \to \\
(0,\alpha^{6}) & \to (1,\alpha^{9}) & \to (2,\alpha^{5}) & \to (3,\alpha^{6}) & \to (0,\alpha^{9}) \to \\
(1,\alpha^{5}) & \to (2,\alpha^{13}) & \to (3,\alpha^{14}) & \to (0,0) & \to (1,1) \to \\
(2,\alpha^{4}) & \to (3,\alpha^{5}) & \to (0,\alpha^{13}) & \to (1,\alpha^{3}) & \to (2,\alpha) \to \\
(3,\alpha^{2}) & \to (0,\alpha^{14}) & \to (1,0) & \to (2,1) & \to (3,\alpha) \to \\
(0,\alpha^{8}) & \to (1,\alpha^{7}) & \to (2,\alpha^{2}) & \to (3,\alpha^{3}) & \to (0,\alpha) \to \\
(1,\alpha^{2}) & \to (0,\alpha^{7}) & \to (3,\alpha^{8}) & \to (2,\alpha^{7}) & \to (1,\alpha^{8}) \to \\
(2,\alpha^{9}) & \to (3,\alpha^{10}) & \to (0,\alpha^{12}) & \to (1,\alpha^{6}) & \to (0,\alpha^{5})
\end{array}$$

Figure 3.5: Hamiltonian cycle in B_4 avoiding faulty f and g edges from $(1,\alpha^6)$.

Proof. We prove the theorem by constructing a Hamiltonian cycle in BF_n using only fault-free edges.

We begin with a cycle containing all nodes $(m, X) \in B_n$, $X \neq 0$ linked by f edges and another containing all nodes $(m, 0) \in B_n$, again using only f edges. (This is a procedure similar to that in Section 2.5.) We then merge these two cycles into a Hamiltonian cycle using g edges in rows 0 and β_0: $(t, 0) \rightarrow (t+1, \beta_{n-1})$ and $(t, \beta_0) \rightarrow (t+1, 0)$ for some t. If the faults in rows other than 0 and β_0 are only in g edges, then we already have the fault-free Hamiltonian cycle. If the faults in rows $X \notin S$ are in f edges, then applying the automorphism ψ to BF_n would map them to fault-free g edges as stated in Theorem 9. Note that ψ maps the f edges in rows $X \in S$ to fault-free f edges, thus giving the required Hamiltonian cycle. ∎

Application of this theorem is illustrated in Fig.3.6 which assumes that the butterfly B_4 has a large number of faults in the category specified by Theorem 13. Note that in B_4, $S = \{0, \alpha, \alpha^6, \alpha^7, \alpha^8, \alpha^{10}, \alpha^{11}, \alpha^{14}\}$.

The fault-free Hamiltonian cycle is then obtained as:

$$\begin{array}{lllll}
(0,\alpha^{14}) & \rightarrow (1,1) & \rightarrow (2,\alpha^4) & \rightarrow (3,\alpha^{10}) & \rightarrow (0,\alpha^{11}) \rightarrow \\
(1,\alpha^{12}) & \rightarrow (2,\alpha^6) & \rightarrow (3,\alpha^7) & \rightarrow (0,\alpha^8) & \rightarrow (1,\alpha^9) \rightarrow \\
(2,\alpha^5) & \rightarrow (3,\alpha^{13}) & \rightarrow (0,\alpha^3) & \rightarrow (1,\alpha) & \rightarrow (2,\alpha^2) \rightarrow \\
(3,\alpha^{14}) & \rightarrow (0,0) & \rightarrow (1,0) & \rightarrow (2,0) & \rightarrow (3,0) \rightarrow \\
(0,1) & \rightarrow (1,\alpha^4) & \rightarrow (2,\alpha^{10}) & \rightarrow (3,\alpha^{11}) & \rightarrow (0,\alpha^{12}) \rightarrow \\
(1,\alpha^6) & \rightarrow (2,\alpha^7) & \rightarrow (3,\alpha^8) & \rightarrow (0,\alpha^9) & \rightarrow (1,\alpha^5) \rightarrow \\
(2,\alpha^{13}) & \rightarrow (3,\alpha^3) & \rightarrow (0,\alpha) & \rightarrow (1,\alpha^2) & \rightarrow (2,\alpha^{14}) \rightarrow \\
(3,1) & \rightarrow (0,\alpha^4) & \rightarrow (1,\alpha^{10}) & \rightarrow (2,\alpha^{11}) & \rightarrow (3,\alpha^{12}) \rightarrow \\
(0,\alpha^6) & \rightarrow (1,\alpha^7) & \rightarrow (2,\alpha^8) & \rightarrow (3,\alpha^9) & \rightarrow (0,\alpha^5) \rightarrow \\
(1,\alpha^{13}) & \rightarrow (2,\alpha^3) & \rightarrow (3,\alpha) & \rightarrow (0,\alpha^2) & \rightarrow (1,\alpha^{14}) \rightarrow \\
(2,1) & \rightarrow (3,\alpha^4) & \rightarrow (0,\alpha^{10}) & \rightarrow (1,\alpha^{11}) & \rightarrow (2,\alpha^{12}) \rightarrow \\
(3,\alpha^6) & \rightarrow (0,\alpha^7) & \rightarrow (1,\alpha^8) & \rightarrow (2,\alpha^9) & \rightarrow (3,\alpha^5) \rightarrow \\
(0,\alpha^{13}) & \rightarrow (1,\alpha^3) & \rightarrow (2,\alpha) & \rightarrow (3,\alpha^2) & \rightarrow (0,\alpha^{14})
\end{array}$$

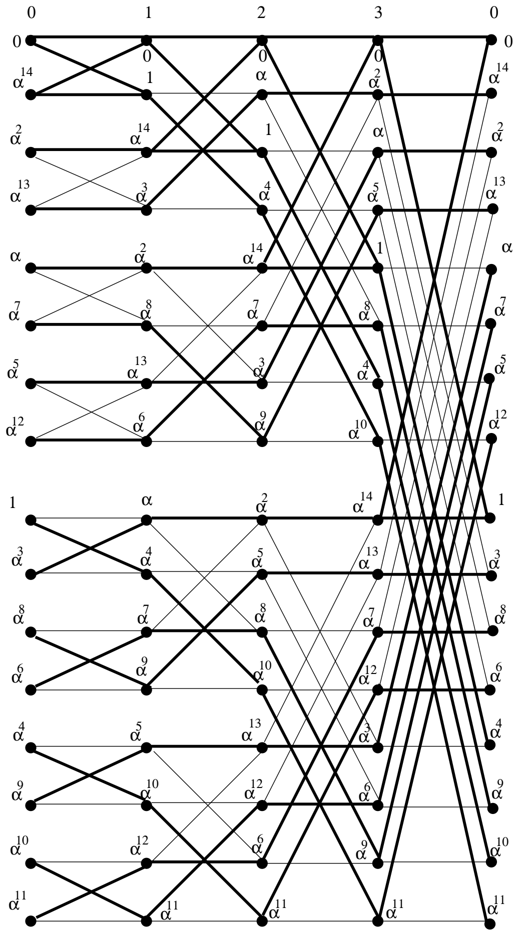
0 1 2 3 0
0 α^{14} α^{2} α^{13} α α^{7} α^{5} α^{12} 1 α^{3} α^{8} α^{6} α^{4} α^{9} α^{10} α^{11}
0 1 α^{14} α^{3} α^{2} α^{8} α^{13} α^{6} α α^{4} α^{7} α^{9} α^{5} α^{10} α^{12} α^{11}
0 α 1 α^{4} α^{14} α^{7} α^{3} α^{9} α^{2} α^{5} α^{8} α^{10} α^{13} α^{12} α^{6} α^{11}
0 α^{2} α α^{5} 1 α^{8} α^{4} α^{10} α^{14} α^{13} α^{7} α^{12} α^{3} α^{6} α^{9} α^{11}
0 α^{14} α^{2} α^{13} α α^{7} α^{5} α^{12} 1 α^{3} α^{8} α^{6} α^{4} α^{9} α^{10} α^{11}

Figure 3.6: Butterfly B_4 with faulty edges marked with light lines and fault-free edges with dark lines. The column numbers are at the top and the row index of each node is marked next to the node.

If the edge faults are located differently, then one can use the automorphism $\psi(\cdot)$ with a different Hamiltonian cycle to obtain a fault-free mapping as the following theorem shows.

Theorem 14 *If the edges in rows $\sigma = (1+\alpha)^{-1}\beta_{n-1}$ and $\sigma + \beta_0$ of BF_n are fault-free, the faults in other rows $X \in S$ are restricted to f edges and those in rows $X \notin S$, restricted to only one type of edges, then BF_n is Hamiltonian.*

Proof. We use the original Hamiltonian cycle of Theorem 13. By applying a $\phi(\cdot)$ which flips edges in every column (Theorem 7), we get a Hamiltonian cycle which uses only g edges in all rows other than rows σ and $\sigma + \beta_0$. The rest of the proof runs parallel to the proof of Theorem 13. ▮

To illustrate Theorem 14, consider a butterfly B_4 shown in Fig. 3.7 where rows $\sigma = \alpha^{11}$ and $\sigma + \beta_0 = \alpha^{10}$ of B_4 are fault-free, the faults in other rows $X \in S = \{0, \alpha, \alpha^6, \alpha^7, \alpha^8, \alpha^{10}, \alpha^{11}, \alpha^{14}\}$. are restricted to f edges and those in rows $X \notin S$, restricted to g edges.

We start by constructing the hamiltonian cycle of Theorem 13. By applying $\phi(\cdot)$ which flips edges in every column, we get a Hamiltonian cycle which uses only g edges in all rows other than rows α^{11} and α^{10}. Then applying $\psi(\cdot)$ should get rid of all faults. The Hamiltonian cycle obtained from this is shown below. The fault-free Hamiltonian cycle is then obtained as:

$$\begin{array}{lllll}
(0,\alpha^7) & \to (1,\alpha^2) & \to (2,\alpha^3) & \to (3,\alpha^4) & \to (0,\alpha^5) \to \\
(1,\alpha^6) & \to (2,\alpha^9) & \to (3,\alpha^{10}) & \to (0,\alpha^{11}) & \to (1,\alpha^{11}) \to \\
(2,\alpha^{11}) & \to (3,\alpha^{11}) & \to (0,\alpha^{12}) & \to (1,\alpha^{13}) & \to (2,\alpha^{14}) \to \\
(3,0) & \to (0,1) & \to (1,\alpha) & \to (2,\alpha^8) & \to (3,\alpha^7) \to \\
(0,\alpha^2) & \to (1,\alpha^3) & \to (2,\alpha^4) & \to (3,\alpha^5) & \to (0,\alpha^6) \to \\
(1,\alpha^9) & \to (2,\alpha^{10}) & \to (3,\alpha^{12}) & \to (0,\alpha^{13}) & \to (1,\alpha^{14}) \to \\
(2,0) & \to (3,1) & \to (0,\alpha) & \to (1,\alpha^8) & \to (2,\alpha^7) \to \\
(3,\alpha^2) & \to (0,\alpha^3) & \to (1,\alpha^4) & \to (2,\alpha^5) & \to (3,\alpha^6) \to
\end{array}$$

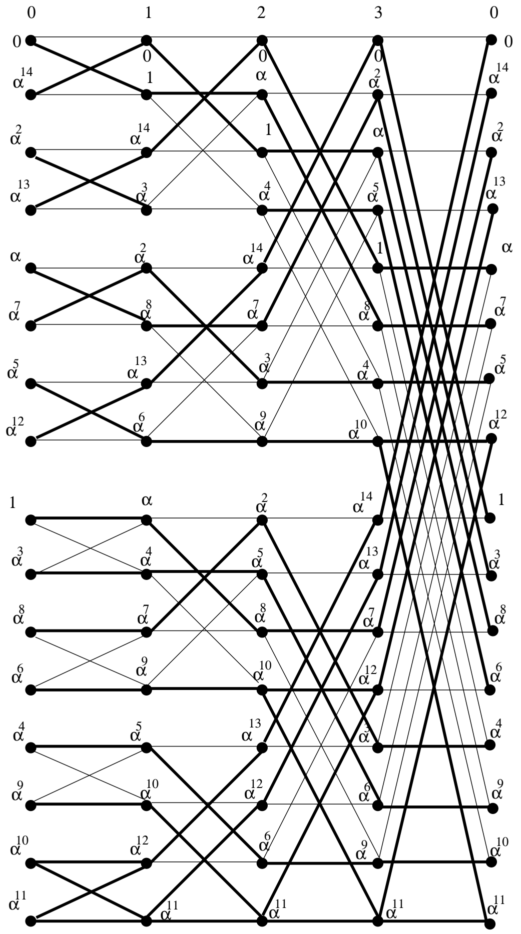

Figure 3.7: Butterfly B_4 with faulty edges marked with light lines and fault-free edges with dark lines. The column numbers are at the top and the row index of each node is marked next to the node.

$$\begin{array}{llllll}
(0,\alpha^9) & \to (1,\alpha^{10}) & \to (2,\alpha^{12}) & \to (3,\alpha^{13}) & \to (0,\alpha^{14}) & \to \\
(1,0) & \to (2,1) & \to (3,\alpha) & \to (0,\alpha^8) & \to (1,\alpha^7) & \to \\
(2,\alpha^2) & \to (3,\alpha^3) & \to (0,\alpha^4) & \to (1,\alpha^5) & \to (2,\alpha^6) & \to \\
(3,\alpha^9) & \to (0,\alpha^{10}) & \to (1,\alpha^{12}) & \to (2,\alpha^{13}) & \to (3,\alpha^{14}) & \to \\
(0,0) & \to (1,1) & \to (2,\alpha) & \to (3,\alpha^8) & \to (0,\alpha^7) &
\end{array}$$

Theorems 13 and 14 require that two rows of BF_n be fault-free. As shown in the next two theorems, this condition may be dropped if n is odd.

Theorem 15 *Let n be odd. If the faults in rows 0 and β_0 of BF_n are restricted to the f edges, those in the other rows $X \in S$ to the g edges, and in rows $X \notin S$ to faults of only one type, then BF_n is Hamiltonian.*

Proof. We first construct a Hamiltonian cycle as follows. Start from any node of the butterfly and choose the next node from a current node $(m, X) \in B_n$ using:

$$\text{next node} = \begin{cases} (m+1, 0) & \text{if } X = \beta_0, \\ (m+1, \beta_{n-1}) & \text{if } X = 0 \text{ and} \\ (m+1, \alpha X) & \text{otherwise.} \end{cases} \tag{3.20}$$

It is easy to prove that the cycle from (3.20) is a Hamilton cycle. Further, nodes in rows 0 and β_0 in this cycle use only fault-free g edges. The rest of nodes use f edges. If they are fault free, we already have the fault-free Hamiltonian cycle. If faults in rows $X \notin S$ are restricted to edges of type f, then applying automorphism $\psi(\cdot)$ to this cycle will give the fault-free Hamiltonian cycle. ∎

Theorem 15 can be illustrated by mapping a Hamiltonian cycle in a faulty B_3 shown in Fig. 3.8. Note that in B_3, $S = \{0, 1, \alpha^4, \alpha^5\}$.

The Hamiltonian cycle obtained from Theorem 15 is shown below.

$$\begin{array}{llllll}
(0,\alpha^2) & \to (1,1) & \to (2,0) & \to (0,\alpha) & \to (1,\alpha^4) & \to \\
(2,\alpha^5) & \to (0,\alpha^6) & \to (1,\alpha^3) & \to (2,\alpha^2) & \to (0,1) & \to \\
(1,0) & \to (2,\alpha) & \to (0,\alpha^4) & \to (1,\alpha^5) & \to (2,\alpha^6) & \to
\end{array}$$

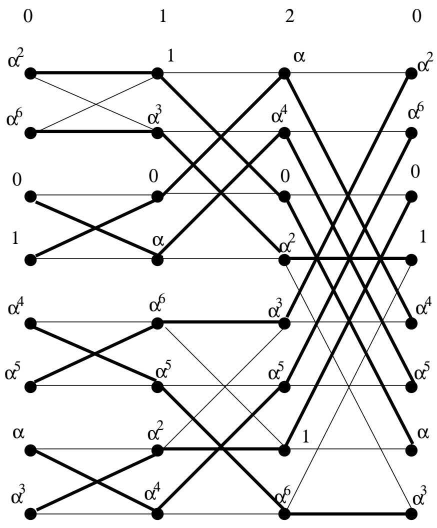

Figure 3.8: Butterfly B_3 with faulty edges marked with light lines and fault-free edges with dark lines. The column numbers are at the top and the row index of each node is marked next to the node.

$$\begin{array}{llllll} (0,\alpha^3) & \to (1,\alpha^2) & \to (2,1) & \to (0,0) & \to (1,\alpha) & \to \\ (2,\alpha^4) & \to (0,\alpha^5) & \to (1,\alpha^6) & \to (2,\alpha^3) & \to (0,\alpha^2) & \end{array}$$

A similar result can also be derived by applying $\phi(\cdot)$ which flips all the edges of the cycle (3.20) to get the starting Hamiltonian cycle used in Theorem 15. We state the result below without proof.

Theorem 16 *Let n be odd. If the faults in rows σ and $\sigma + \beta_0$ of BF_n are restricted to the g edges, those in the other rows $X \in S$ to the f edges, and in rows $X \notin S$ to faults of only one type, then BF_n is Hamiltonian.*

Note that the symmetry of BF$_n$ will allow further generalization of Theorems 13 - 16.

We end this section by showing that one can also employ automorphisms $\phi(\cdot)$ and $\psi(\cdot)$ together to get even more powerful results.

Theorem 17 *If the edges in one of the columns of BF_n are fault free, and the faults in each of the other columns are such that edges from $X \in S$ have one type of fault and those from $X \notin S$ have another type of fault. Then BF_n is Hamiltonian.*

Proof. If the faulty edges from (m, X), are of type g when $X \in S$ and of type f if $X \notin S$, then applying ψ will map all of these faulty edges to type g in column $n - m$. On the other hand if the faulty edges from (m, X), are of type f when $X \in S$ and of type g if $X \notin S$, then applying ψ will map all of these faulty edges to type f in column $n - m$. Thus, after applying ψ, all the faulty edges in any column will be limited to only one type and there will be no faulty edges in one column. Theorem 11 can then be used to build the required Hamiltonian cycle using fault free edges. ■

Application of this theorem is illustrated in the next example which assumes that faults are limited to g edges from $X \in S$ and to f edges from $X \notin S$ in columns $1, 3$ in B_4; and for column 0 faults are limited to f edges from $X \in S$ and to g edges from $X \notin S$, where $S = \{0, \alpha, \alpha^6, \alpha^7, \alpha^8, \alpha^{10}, \alpha^{11}, \alpha^{14}\}$. We need to choose an automorphism with $c_0 = 0, c_1 = 1, c_2 = 0, c_3 = 0$. Automorphism offset K_0 can then be obtained from

$$K_0(\alpha^4 + 1)\beta_3^{-1} = \sum_{j=0}^{n-1} c_{(-j)\ mod\ n}\alpha^j. \tag{3.21}$$

Solving (3.21) gives $K_0 = \alpha^2$, which in turn yields $K_1 = \alpha^{14}$, $K_2 = 1$ and $K_3 = \alpha$ from (3.3). Using an automorphism based on these offsets provides a re-mapping of the cycle on the faulty butterfly as:

$$\begin{array}{llllll}
(0,\alpha^8) & \rightarrow (1,\alpha^7) & \rightarrow (2,\alpha^8) & \rightarrow (3,\alpha^9) & \rightarrow (0,\alpha^{10}) & \rightarrow \\
(1,\alpha^{12}) & \rightarrow (2,\alpha^{13}) & \rightarrow (3,\alpha^{14}) & \rightarrow (0,1) & \rightarrow (1,\alpha^4) & \rightarrow \\
(2,\alpha^5) & \rightarrow (3,\alpha^6) & \rightarrow (0,\alpha^7) & \rightarrow (1,\alpha^2) & \rightarrow (2,\alpha^3) & \rightarrow \\
(3,\alpha) & \rightarrow (0,\alpha^2) & \rightarrow (1,\alpha^{14}) & \rightarrow (2,1) & \rightarrow (3,\alpha^4) & \rightarrow \\
(0,\alpha^5) & \rightarrow (1,\alpha^{13}) & \rightarrow (2,\alpha^{14}) & \rightarrow (3,1) & \rightarrow (0,\alpha) & \rightarrow \\
(1,\alpha^8) & \rightarrow (2,\alpha^9) & \rightarrow (3,\alpha^{10}) & \rightarrow (0,\alpha^{11}) & \rightarrow (1,\alpha^{11}) & \rightarrow \\
(2,\alpha^{12}) & \rightarrow (3,\alpha^{13}) & \rightarrow (0,\alpha^{14}) & \rightarrow (1,0) & \rightarrow (2,0) & \rightarrow \\
(3,0) & \rightarrow (0,0) & \rightarrow (1,1) & \rightarrow (2,\alpha) & \rightarrow (3,\alpha^2) & \rightarrow \\
(0,\alpha^3) & \rightarrow (1,\alpha) & \rightarrow (2,\alpha^2) & \rightarrow (3,\alpha^3) & \rightarrow (0,\alpha^4) & \rightarrow \\
(1,\alpha^{10}) & \rightarrow (2,\alpha^{11}) & \rightarrow (3,\alpha^{12}) & \rightarrow (0,\alpha^{13}) & \rightarrow (1,\alpha^3) & \rightarrow
\end{array}$$

$$\begin{array}{lllll}
(2,\alpha^{4}) & \to (3,\alpha^{5}) & \to (0,\alpha^{6}) & \to (1,\alpha^{9}) & \to (2,\alpha^{10}) \to \\
(3,\alpha^{11}) & \to (0,\alpha^{12}) & \to (1,\alpha^{6}) & \to (2,\alpha^{7}) & \to (3,\alpha^{8}) \to \\
(0,\alpha^{9}) & \to (1,\alpha^{5}) & \to (2,\alpha^{6}) & \to (3,\alpha^{7}) & \to (0,\alpha^{8})
\end{array}$$

Applying ψ, one gets the cycle:

$$\begin{array}{lllll}
(0,\alpha^{7}) & \to (1,\alpha^{8}) & \to (2,\alpha^{9}) & \to (3,\alpha^{5}) & \to (0,\alpha^{6}) \to \\
(1,\alpha^{7}) & \to (2,\alpha^{8}) & \to (3,\alpha^{9}) & \to (0,\alpha^{10}) & \to (1,\alpha^{11}) \to \\
(2,\alpha^{12}) & \to (3,\alpha^{6}) & \to (0,\alpha^{9}) & \to (1,\alpha^{5}) & \to (2,\alpha^{13}) \to \\
(3,\alpha^{3}) & \to (0,\alpha^{4}) & \to (1,\alpha^{10}) & \to (2,\alpha^{11}) & \to (3,\alpha^{12}) \to \\
(0,\alpha^{13}) & \to (1,\alpha^{3}) & \to (2,\alpha) & \to (3,\alpha^{2}) & \to (0,\alpha^{3}) \to \\
(1,\alpha) & \to (2,\alpha^{2}) & \to (3,\alpha^{14}) & \to (0,0) & \to (1,0) \to \\
(2,0) & \to (3,0) & \to (0,1) & \to (1,\alpha^{4}) & \to (2,\alpha^{10}) \to \\
(3,\alpha^{11}) & \to (0,\alpha^{11}) & \to (1,\alpha^{12}) & \to (2,\alpha^{6}) & \to (3,\alpha^{7}) \to \\
(0,\alpha^{2}) & \to (1,\alpha^{14}) & \to (2,1) & \to (3,\alpha^{4}) & \to (0,\alpha^{5}) \to \\
(1,\alpha^{13}) & \to (2,\alpha^{14}) & \to (3,1) & \to (0,\alpha) & \to (1,\alpha^{2}) \to \\
(2,\alpha^{3}) & \to (3,\alpha) & \to (0,\alpha^{8}) & \to (1,\alpha^{9}) & \to (2,\alpha^{5}) \to \\
(3,\alpha^{13}) & \to (0,\alpha^{14}) & \to (1,1) & \to (2,\alpha^{4}) & \to (3,\alpha^{10}) \to \\
(0,\alpha^{12}) & \to (1,\alpha) & \to (2,\alpha^{7}) & \to (3,\alpha^{8}) & \to (0,\alpha^{7})
\end{array}$$

One can check that this Hamiltonian cycle does not contain any of the faulty edges.

3.5 Conclusion

This Chapter has focused on exploring structural properties and fault tolerant mappings on BF_n using an algebraic model based on finite fields. This Chapter has also provided all the $n2^{n+1}$ automorphisms of a wrapped butterfly network of degree n

using the direct product of a cyclic group and a finite field. In the past, automorphisms have been used to map algorithms on architecture with generally one faulty node. This Chapter investigated for the first time the translation of butterfly edges by automorphisms. A new strategy for algorithm mappings on an architecture with faulty edges has been proposed. We have illustrated our technique by mapping Hamilton cycle on the butterfly under various edge fault scenarios.

Chapter 4

Shuffle Exchange Networks

4.1 Introduction

Interconnection networks which are used to move data between multiple cores within a chip or between computers in a parallel machine often constrain the performance of the machine. The Shuffle Exchange (SE) is an interconnection network that is non-symmetric but still considered one of the most fundamental interconnection networks for parallel computation due to their small, fixed node degree and small (logarithmic) diameter [3,44,45]. Previous work on SE includes VLSI implementation and design of optimal layout [46]. The Shuffle Exchange network is proven to be a coset graph of the CCC. It was shown in [47] that the Shuffle Exchange network contains a Hamiltonian cycle. Feldmann and Unger [38] have proved that the Shuffle Exchange network is a subgraph of deBruijn network. The problem of designing fault-tolerant networks for Shuffle Exchange was addressed by [48, 49].

One drawback to this class of networks lies in its unwieldy model. Unfortunately the connectivity of Shuffle Exchange using the binary model is much too complex to obtain many of the useful properties of the network. This Chapter is organized as follows. Section 4.2 develops an algebraic model for the Shuffle Exchange network based on a finite field. With this model, one can avail of the powerful algebraic

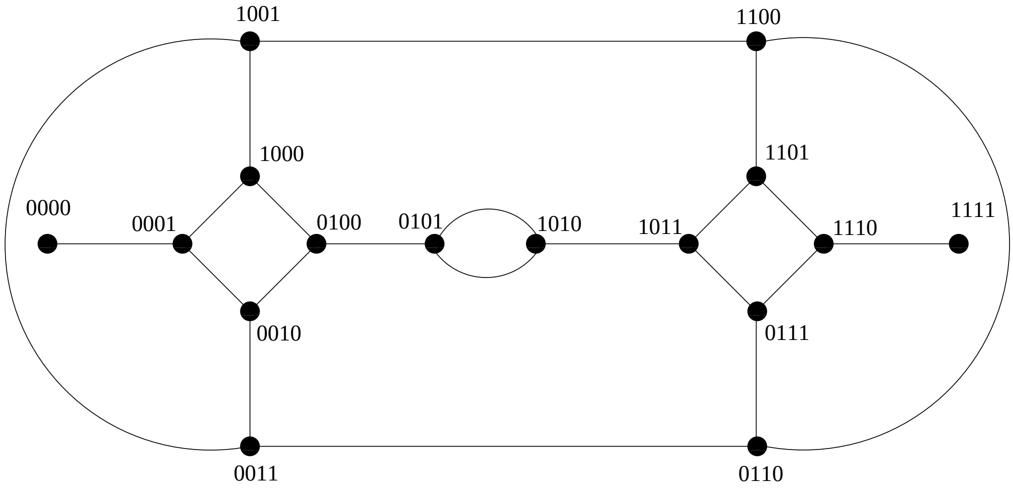

Figure 4.1: An 16-node Shuffle Exchange network (SE_4) in Binary notation

techniques to investigate the structural properties of this network. Section 4.3 exploits these techniques to find paths in the Shuffle Exchange network. Section 4.4 explores the relationships between the Shuffle Exchange and deBruijn networks.

4.2 An Algebraic model of the Shuffle Exchange Network

Even though non-symmetric, Shuffle Exchange is a popular interconnection network [45]. A Shuffle Exchange graph of degree n, SE$_n$ has 2^n nodes, each with a maximum node degree of 3. Traditionally, one uses a set Z_2^n of n-bit binary strings to label the nodes of SE$_n$. A node $\langle v_{n-1}, v_{n-2}, \ldots, v_0\rangle$ is connected to nodes $\langle v_{n-2}, v_{n-3}, \ldots, v_0, v_{n-1}\rangle$, $\langle v_0, v_{n-1}, v_{n-2}, \ldots, v_1\rangle$ (shuffle edges) and $\langle v_{n-1}, v_{n-2}, \ldots, \overline{v}_0\rangle$ (exchange edge). SE$_4$ graph labeled in binary is shown in Fig. 4.1.

In this section we show that the nodes of SE$_n$ may be labeled with elements of the finite field $GF(2^n)$ such that the node connectivity is expressed through an algebraic relationship between these labels.

Theorem 18 *The nodes of the Shuffle Exchange graph SE_n can be labeled by the elements of the finite field $GF(2^n)$ such that a graph node X is connected to the nodes $(\alpha X + \beta_{n-1}Tr(\sigma X))$, $(\alpha^{-1}X + \beta_0 Tr(\sigma\alpha^{-1}X))$ and $(X + \beta_0)$.*

Proof. Consider a mapping $\zeta(\cdot) : Z_2^n \rightarrow GF(2^n)$ defined as

$$\zeta(\langle v_{n-1}, v_{n-2}, \ldots, v_0\rangle) = \sum_{i=0}^{n-1} v_i\beta_i \tag{4.1}$$

We now show that the correspondence expressed by (4.1) relabels the graph nodes in such a manner that the graph connectivity is expressed as in the theorem.

Let X denote the algebraic label of the node $V = \langle v_{n-1}, v_{n-2}, \ldots, v_0\rangle$, i.e.,

$$X = \zeta(V) = \sum_{i=0}^{n-1} v_i\beta_i \tag{4.2}$$

The neighbors of V are $V_1 = \langle v_{n-2}, v_{n-3}, \ldots, v_0, v_{n-1}\rangle$, $V_2 = \langle v_0, v_{n-1}, v_{n-2}, \ldots, v_1\rangle$ and $V_3 = \langle v_{n-1}, v_{n-2}, \ldots, \overline{v}_0\rangle$.

The relabeling of node V_1 gives

$$\zeta(V_1) = \sum_{i=0}^{n-1} v_i\beta_{i+1} \tag{4.3}$$

where the index of β is considered modulo n. Using Lemma 1, one can write (4.3) as

$$\begin{aligned} \zeta(V_1) &= \sum_{i=0}^{n-2} v_i\alpha^{-1}(\beta_i + p_{i+1}\beta_{n-1}) + u_{n-1}\alpha^{-1}\beta_{n-1} \\ &= \alpha^{-1}X + \beta_0 \sum_{i=0}^{n-2} p_{i+1}v_i. \end{aligned} \tag{4.4}$$

However, from Lemma 2,

$$Tr(\alpha^{-1}\sigma X) = \sum_{i=0}^{n-1} v_i Tr(\alpha^{-1}\sigma\beta_i) = \sum_{i=0}^{n-2} v_i p_{i+1}. \tag{4.5}$$

Comparing (4.4) and (4.5) we get

$$\zeta(V_1) = \alpha^{-1}X + \beta_0 Tr(\alpha^{-1}\sigma X).$$

Similarly, the relabeling of node V_2 gives

$$\zeta(V_2) = \sum_{i=0}^{n-1} v_i \beta_{i-1} \tag{4.6}$$

where the index of β_{i-1} is considered modulo n. Using Lemma 1, (4.6) may be rewritten as

$$\begin{aligned} \zeta(V_2) &= \sum_{i=1}^{n-1} v_i(\alpha\beta_i + p_i\beta_{n-1}) + v_0\alpha\beta_0 \\ &= \alpha X + \beta_{n-1} \sum_{i=1}^{n-1} p_i v_i. \end{aligned} \tag{4.7}$$

But from Lemma 2,

$$Tr(\sigma X) = \sum_{i=0}^{n-1} v_i Tr(\sigma\beta_i) = \sum_{i=1}^{n-1} v_i p_i. \tag{4.8}$$

From (4.7) and (4.8) one gets

$$\zeta(V_2) = \alpha X + \beta_{n-1} Tr(\sigma X).$$

Finally, recognizing that the value of $\overline{v}_0$ can be expressed in $GF(2^n)$ as $v_0 + 1$, the relabeling of V_3 gives

$$\begin{aligned} \zeta(V_3) &= (v_0 + 1)\beta_0 + \sum_{i=1}^{n-1} v_i\beta_i \\ &= \beta_0 + \sum_{i=0}^{n-1} v_i\beta_i = X + \beta_0. \end{aligned}$$

Thus all the three edges of SE$_n$ in binary notation may be expressed in terms of their algebraic relationship. ∎

The translation of binary labels of graph SE_4 to their algebraic values using (4.1) is illustrated in Table 4.1. The SE$_3$ and SE$_4$ relabeled in the algebraic notation is shown in Fig. 4.2 and Fig. 4.3.

The connectivity of SE_4 using the new algebraic model is shown in Fig. 4.4. As indicated in this figure, we will refer to the three edges as f, f^{-1} and g in this path.

Table 4.1: Equivalence between the binary and the algebraic labels of SE_4.

Binary	Algebraic	Binary	Algebraic
(0000)	0	(1000)	1
(0001)	α^{14}	(1001)	α^3
(0010)	α^2	(1010)	α^8
(0011)	α^{13}	(1011)	α^6
(0100)	α	(1100)	α^4
(0101)	α^7	(1101)	α^9
(0110)	α^5	(1110)	α^{10}
(0111)	α^{12}	(1111)	α^{11}

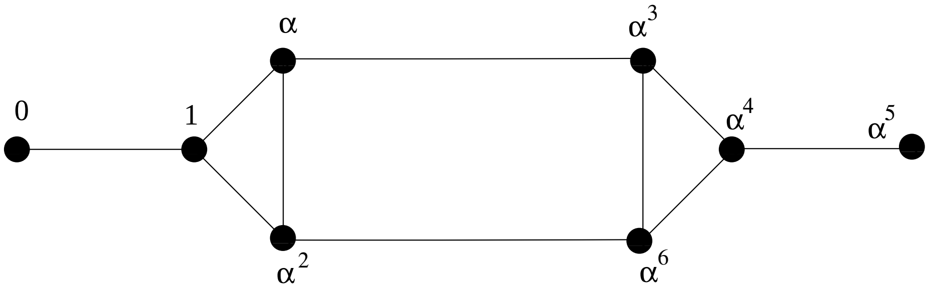

Figure 4.2: An 8-node Shuffle Exchange network (SE_3) in Algebraic notation

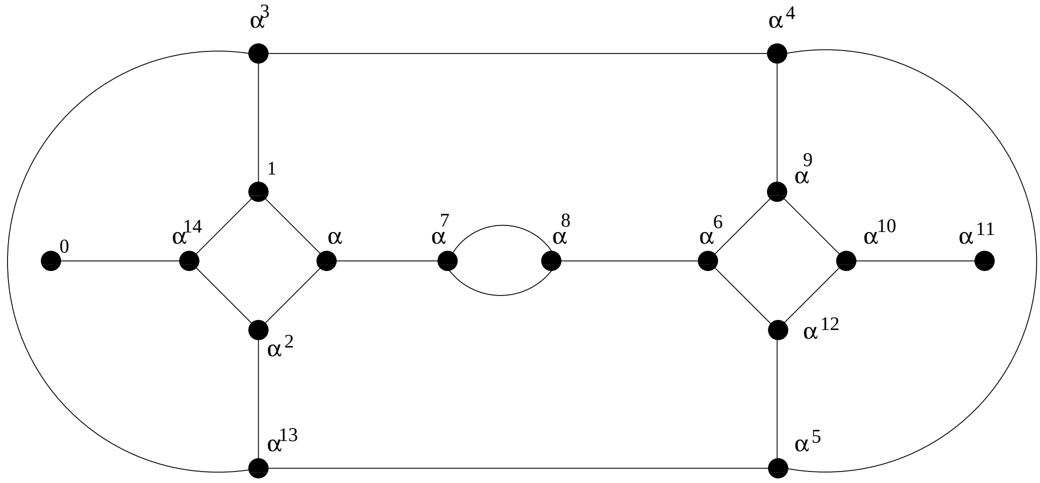

Figure 4.3: An 16-node Shuffle Exchange network (SE_4) in Algebraic notation

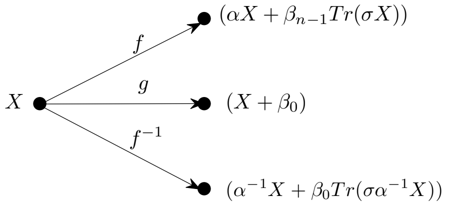
$(\alpha X + \beta_{n-1} Tr(\sigma X))$
f
g
X
$(X + \beta_0)$
f^{-1}
$(\alpha^{-1} X + \beta_0 Tr(\sigma \alpha^{-1} X))$

Figure 4.4: The connectivity of the Shuffle Exchange graph SE$_n$.

Not only is the connectivity expression in this model is simple, but because of the linearity of the trace function, one can make the following interesting observation.
If $X_1 \xrightarrow{f} Y_1$ and $X_2 \xrightarrow{f} Y_2$, then,
$X_1 + X_2 \xrightarrow{f} Y_1 + Y_2$.
Similarly, If $X_1 \xrightarrow{f^{-1}} Y_1$ and $X_2 \xrightarrow{f^{-1}} Y_2$, then,
$X_1 + X_2 \xrightarrow{f^{-1}} Y_1 + Y_2$.
We will call this observation as the *linearity in source property* of the f and f^{-1} edges and use it later when we embed SE$_n$ in the deBruijn graph.

We now show the use of algebraic machinery to chart a path from node X to node Yin the shuffle exchange graph. We will only use edges f and g.

4.3 Path algorithm for Shuffle Exchange Network

This section deals with designing a path to travel between any two nodes of the Shuffle Exchange network.

Algorithm 1 (Path to go from X to Y in SE$_\mathbf{n}$)

```
Compute $c_i = Tr((X+Y)\alpha^i)$, $0 \leq i < n$.
Compute destination nodes $D_i$ as follows. Let $D_0 = X$.
for $i = 0$ to $n-1$ do
        From $D_i$ go to node $D'_i = D_i + c_i\beta_0$.
        From $D'_i$ go to node $D_{i+1} = \alpha D'_i + \beta_{n-1}Tr(\sigma D'_i)$
Then $D_n = Y$.
```

Note the simplicity of Algorithm 1 which is a direct consequence of the algebraic model specified in Theorem 18. The path given here is not necessarily optimal since we constrain ourselves to only two of the edges. However, it shows that the diameter of SE$_n$ cannot be more than $2n$. Further, when $c_i = 0$, the first step in iteration i is not really a move from a node to the next node on the path. Thus the length of the path in the algorithm is $n+$ number of nonzero c_is. The proof of the correctness of the algorithm is provided below.

Proof. We first show (by mathematical induction) that

$$D_i = \alpha^i X + \sum_{j=0}^{i-1} t_j \beta_{n-i+j}, \tag{4.9}$$

where $t_j = c_j + Tr(\alpha^j \sigma X)$.

When $i = 1$, One can demonstrate that (4.9) is true by direct calculation. Now assume that (4.9) is true for some i. Its truth for $i+1$ can be established as follows. From Algorithm 1,

$$D'_i = \alpha^i X + \sum_{j=0}^{i-1} t_j \beta_{n-i+j} + c_i \beta_0,$$

and consequently,

$$\begin{aligned} D_{i+1} &= \alpha^{i+1} X + \sum_{j=0}^{i-1} t_j \alpha \beta_{n-i+j} + c_i \beta_{n-1} + \beta_{n-1} Tr(\alpha^i \sigma X) + \sum_{j=0}^{i-1} t_j Tr(\sigma \beta_{n-i-j}) \\ &= \alpha^{i+1} X + t_i \beta_{n-1} + \sum_{j=0}^{i-1} t_j [\alpha \beta_{n-i+j} + \beta_{n-1} p_{n-i+j}] \\ &= \alpha^{i+1} X + t_i \beta_{n-1} + \sum_{j=0}^{i-1} t_j \beta_{n-i+j} \\ &= \alpha^{i+1} X + \sum_{j=0}^{i} t_j \beta_{n-i-1+j}. \end{aligned} \tag{4.10}$$

Step 2 of (4.10) uses Lemma 2 and step 3 uses Lemma 1. Equation (4.10) shows that if (4.9) is true for i then it is also true for $i+1$. Therefore by mathematical induction, it is true for all $0 \leq i < n$.

Clearly the final destination D_n is dependent upon the values of c_is. We now show that to achieve $D_n = Y$, c_is should have the values specified in the algorithm. Using (4.10), one gets

$$Tr((X+Y)\alpha^i) = Tr((X + \alpha^n X + \sum_{j=0}^{n-1} t_j \beta_j)\alpha^i. \tag{4.11}$$

Using linearity of the trace function and the fact that $Tr(\beta_j \alpha^i) = 1$ only if $j = i$ and is 0 otherwise, one can simplify (4.11) as

$$\begin{aligned} Tr((X+Y)\alpha^i) &= Tr(\alpha^i \sigma X) + \sum_{j=0}^{n-1} t_j Tr(\beta_j \alpha^i) \\ &= Tr(\alpha^i \sigma X) + t_i \\ &= c_i. \end{aligned} \tag{4.12}$$

This proves that the c_is used in the algorithm do indeed lead the path to the desired final destination $D_n = Y$. ▮

As an illustration, consider the path from node $X = 0$ to node $Y = \alpha^6$ in SE_4. One can compute $c_0 = 1$, $c_1 = 1$, $c_2 = 0$ and $c_3 = 1$, The path is then given by $0 \to \alpha^{14} \to 1 \to \alpha^3 \to \alpha^4 \to \alpha^5 \to \alpha^{12} \to \alpha^6$.

4.4 Relation with deBruijn network

We now show that the Shuffle Exchange network SE$_n$ is related to the deBruijn network DB$_n$.

A deBruijn network of degree n, DB$_n$, is a graph with 2^n nodes, each labeled with an n bit binary string. A node with label $(a_{n-1}, a_{n-2}, \ldots, a_0)$ is connected to four nodes $(0, a_{n-1}, a_{n-2}, \ldots, a_1)$, $(1, a_{n-1}, a_{n-2}, \ldots, a_1)$, $(a_{n-2}, a_{n-3}, \ldots, a_0, 0)$ and $(a_{n-2}, a_{n-3}, \ldots, a_0, 1)$. DB$_n$ is attractive because it has a small constant node degree and a small diameter, n. However, it is not symmetric and is not seen prominently in commercial world because of its lack of algorithm mappings.

An algebraic model for DB$_n$ is available in literature [15]. In this model, the nodes of DB$_n$ are labeled using the elements of the finite field $GF(2^n)$. A node with label $X \in GF(2^n)$ is connected to nodes αX, $\alpha X + \beta_{n-1}$, $\alpha^{-1} X$ and $\alpha^{-1} X + \beta_0$. These connections are indicated in Fig. 4.5.

Following theorem shows that SE$_n$ can be embedded in DB$_n$.

Theorem 19 *SE$_n$ is a subgraph of DB$_n$.*

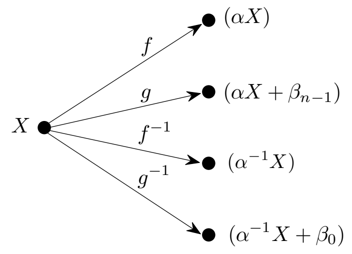

Figure 4.5: The connectivity of the deBruijn graph (DB$_n$)

Proof. Use the mapping $\lambda(\cdot) : \mathrm{SE}_n \to \mathrm{DB}_n$ defined as

$$\lambda(X) = \alpha^{f(X)} X + f(X)\beta_{n-1} Tr(\sigma X), \quad X \in GF(2^n),$$

where,

$$f(X) = Tr(\sigma(1+\alpha)^{-1}X).$$

We first show that $\lambda(\cdot)$ is a one-to-one mapping between the nodes of SE$_n$ and DB$_n$ by contradiction. Assume $\lambda(X) = \lambda(Y)$ for some $X, Y \in GF(2^n)$. If $f(X) = f(Y) = 0$, then $X = Y$. If $f(X) = f(Y) = 1$, then it leads to $\alpha(X + Y) + \beta_{n-1} Tr(\sigma(X + Y)) = 0$. However, we know from the CCC$_n$ connectivity (see Fig. 5.3) that $\alpha(X + Y) + \beta_{n-1} Tr(\sigma(X + Y))$ is the destination of $(X + Y)$ along the f edge. Since this destination is 0, the source $X + Y = 0$ as well. Thus, again we have $X = Y$. Finally, if $f(X)$ and $f(Y)$ are different, say if $f(X) = 1$ and $f(Y) = 0$, then $\lambda(X) = \lambda(Y)$ gives $\alpha X + \beta_{n-1} Tr(\sigma X) = Y$. One then gets $f(Y) = f(X)$, which is a contradiction. Thus any time $\lambda(X) = \lambda(Y)$, $X = Y$, i.e., function $\lambda(\cdot)$ is one-to-one.

We now prove that the edges of SE$_n$ are preserved by $\lambda(\cdot)$. Because of the linearity in source property of the f edges of SE$_n$, one only needs to show the preservation of f and f^{-1} edges starting from β_i, $0 \le i < n$. Consider the edge $\beta_i \xrightarrow{f} \beta_{i-1}$. One can show that $f(\beta_i) = f(\beta_{i-1}) = 1$. Thus $\lambda(\beta_i) = \beta_i + \beta_{n-1} Tr(\sigma\beta_i)$. From Lemmas 1 and 2, one can then see that $\lambda(\beta_i) = \beta_{i-1}$. Similarly $\lambda(\beta_{i-1}) = \beta_{i-2}$. Thus the edge $\beta_i \xrightarrow{f} \beta_{i-1}$ of SE$_n$ translates to the edge $\beta_{i-1} \longrightarrow \beta_{i-2}$ of DB$_n$ (see Fig. 4.5) (it is an f edge in DB$_n$ if $p_{i-1} = 0$ and a g edge otherwise).

Finally, to see the preservation of the g edge of SE$_n$, consider edge $X \xrightarrow{g} X+\beta_0$. One has, $f(X+\beta_0) = f(X) + 1$. Thus if $f(X) = 0$, then $f(X+\beta_0) = 1$. Thus the edge is transformed by $\lambda(\cdot)$ to $X \to \alpha(X+\beta_0) + \beta_{n-1}Tr(\sigma(X+\beta_0)) = \alpha X + \beta_{n-1} + \beta_{n-1}Tr(\sigma X)$. This is clearly either edge f or g of DB$_n$. On the other hand, if $f(X) = 1$, then $f(X+\beta_0) = 0$. Thus the SE$_n$ edge is transformed to $\alpha X + \beta_{n-1}Tr(\sigma X) \to X + \beta_0$. This is either edge f^{-1} or g^{-1} of DB$_n$. ∎

4.5 Conclusion

This Chapter has developed a new algebraic model for the Shuffle Exchange network that is used in parallel architecture. This model allows the use of powerful algebraic techniques to study the structural properties of the network. Our strategy exploits these techniques to find paths in the Shuffle Exchange network and to explore the relationship between Shuffle Exchange and deBruijn networks.

Chapter 5

Cube Connected Cycles

5.1 Introduction

Interconnection networks often constrain the performance of multi-cores chips or parallel computers. Cube Connected Cycles (CCC) is an attractive interconnection network because of its symmetry, small constant node degree and small diameter.

Previous work on CCC includes VLSI implementation and optimal layout [50,51], load balancing, routing and one-to-one, one-to-many broadcast strategies [52, 53], mappings of cycles in fault-free and faulty topologies [54] and determination of the forwarding index of the network [55].

One of the drawbacks of the CCC network is its unwieldy model which complicates mappings of algorithms on these architectures. As a result, even though this network is scalable and has attractive topological properties, its utility in applications is somewhat constrained. With this in mind, a new addressing scheme for CCC using Cayley graphs over permutation groups has been proposed [56]. Unfortunately even that new model does not provide sufficient insight into the graph connectivity. This Chapter provides a new algebraic model of the Cube Connected Cycles using cyclic groups and finite fields. Our model allows one to harness powerful algebraic techniques to explore the topological properties and mappings on the Cube Connected Cycles graph. It also illuminates the relationships between graphs as diverse as Shuffle Exchange, deBruijn (both non-symmetric), Wrapped Butterflies and the Cube Connected Cycles.

With this new model, one can avail of powerful algebraic techniques to investigate the structure and mappings of these networks. Similar algebraic models developed previously for the deBruijn network [15] and the Wrapped Butterflies [10] have allowed efficient mappings of cycles and trees on the Butterflies and provided insights into intricate structural properties such as the automorphisms [23, 24]. The new model proposed here helps solve similar problems in Cube Connected Cycles.

This Chapter is organized as follows. The new algebraic model of the Cube Connected Cycles is defined and proved in Section 5.2. Section 5.3 provides optimal path algorithms for the Cube Connected Cycles. Section 5.4 obtains all the automorphisms of the Cube Connected Cycles using an algebraic model. We explore the edge transformation in Cube Connected Cycles networks due to automorphisms in Section 5.5. Section 5.6 proves that the Cube Connected Cycles is a subgraph of the Butterfly network of the same size.

5.2 An Algebraic model of the Cube Connected Cycles

The cube connected cycles network of dimension n (CCC$_n$) has $n2^n$ nodes, each of which is labeled by a pair (m, V) where $m \in Z_n$, a group of integers $\{0, 1, \cdots, n-1\}$ and $V \in Z_2^n$, a set of n bit binary strings. A node (m, V) is connected to nodes $(m+1, V)$, $(m-1, V)$ and $(m, V \oplus 2^m)$ as shown in Fig. 5.1, where $V \oplus 2^m$ is the string V with mth bit complemented. The diameter of CCC is 6 when $n = 3$ and $2n + \lfloor n/2 \rfloor - 2$ when $n > 3$ [6]. This low diameter and the low constant node degree implies that CCC may be very useful for parallel architectures. CCC$_4$ graph labeled in binary is shown in Fig. 5.2.

In this section we provide a new model for the CCC$_n$ defined over the structure $C_n \times GF(2^n)$. In particular, following theorem shows that if the nodes of CCC$_n$ are labeled by the elements of the structure $C_n \times GF(2^n)$, then the edges can be

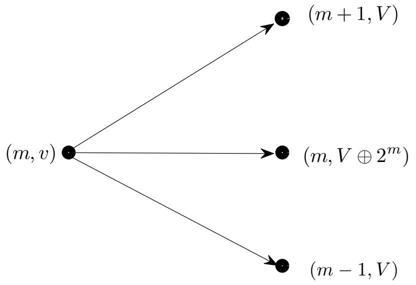

Figure 5.1: The connectivity of the Cube Connected Cycles graph CCC$_n$.

expressed by a simple algebraic relationship between the labels.

Theorem 20 *The nodes of the cube connected cycles graph CCC_n can be labeled by the elements of $C_n \times GF(2^n)$ in such a fashion that the graph connectivity can be expressed as follows. A node (m,X) is connected to the three nodes $(m+1,\alpha X + \beta_{n-1}Tr(\sigma X))$, $(m-1,\alpha^{-1}X+\beta_0 Tr(\sigma\alpha^{-1}X))$ and $(m,X+\beta_0)$.*

Proof. Let $V = \langle v_{n-1}, v_{n-2}, \ldots, v_0\rangle$. Consider the mapping $\zeta : Z_n \times Z_2^n \to C_n \times GF(2^n)$ defined by

$$\zeta((m,V)) = (m,X), \qquad \text{where } X = \sum_{i=0}^{n-1} v_{m+i}\beta_i. \tag{5.1}$$

We now show that the correspondence expressed by (5.1) relabels the graph nodes allowing the graph connectivity as stated in the theorem.

The three neighbors of (m,V) are $(m+1,V)$, $(m-1,V)$ and (m,V_1), where $V_1 = \langle v_{n-1}, v_{n-2}, \ldots, v_{m+1}, \overline{v}_m, v_{m-1}, \ldots, v_0\rangle$.

The image of the first neighbor of (m,V) is

$$\begin{aligned} \zeta(m+1,V) &= (m+1, \sum_{i=0}^{n-1} v_{m+1+i}\beta_i) \\ &= (m+1, \sum_{i=0}^{n-1} v_{m+i}\beta_{i-1}), \end{aligned} \tag{5.2}$$

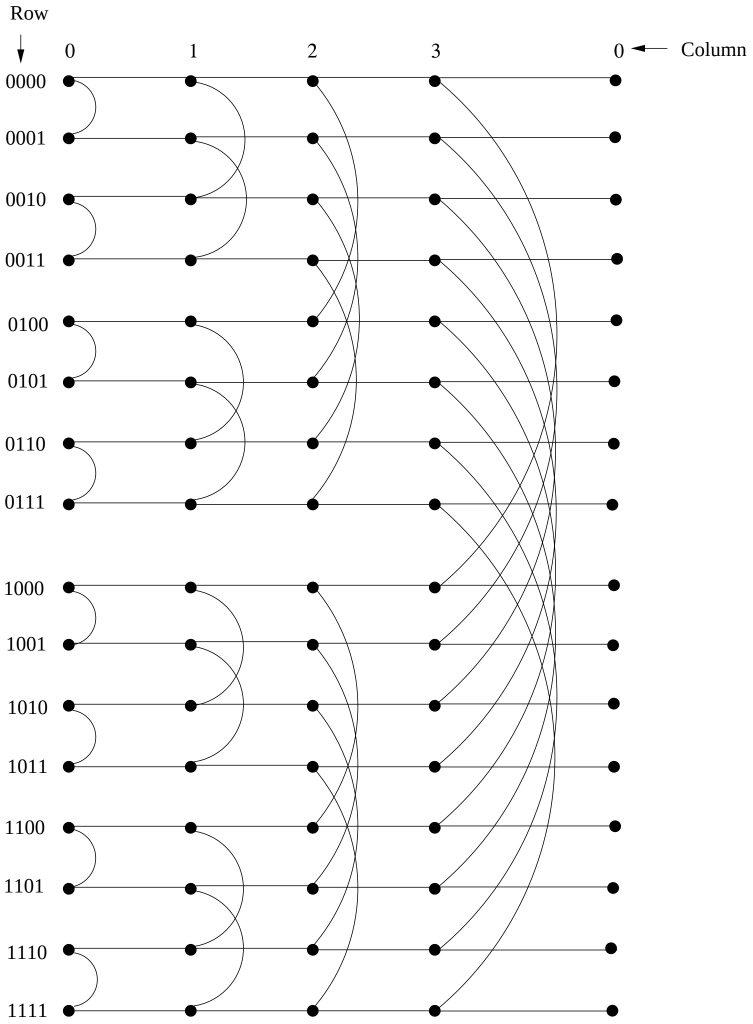

Figure 5.2: Connections of Cube Connected Cycles CCC_4 in Binary notation. To make the drawing simpler, m in $(m,\ V)$ is written as a column heading and nodes in column 0 are repeated.

where the index of β is considered modulo n. From Lemma 1, one can express β_{i-1} in (5.2) in terms of β_i to get

$$\begin{aligned}\zeta(m+1,V) &= (m+1,\alpha\sum_{i=1}^{n-1} v_{m+i}\beta_i + \\ &\qquad \beta_{n-1}\sum_{i=0}^{n-1} v_{m+i}p_i + v_m\alpha\beta_0) \\ &= (m+1,\alpha X + \beta_{n-1}\sum_{i=1}^{n-1} v_{m+i}p_i). \qquad (5.3)\end{aligned}$$

Now, using Lemma 2 and definition (5.1),

$$\begin{aligned}Tr(\sigma X) &= \sum_{i=0}^{n-1} v_{m+i}Tr(\sigma\beta_i) \\ &= \sum_{i=0}^{n-1} v_{m+i}p_i. \qquad (5.4)\end{aligned}$$

Combining (5.3) and (5.4) one can see that

$$\zeta(m+1,V) = (m+1,\alpha X + \beta_{n-1}Tr(\sigma X)).$$

Thus (m,X) is connected to $(m+1,\alpha X + \beta_{n-1}Tr(\sigma X))$.

Similarly, The image of the second neighbor of (m,V) is

$$\zeta(m-1,V) = (m-1,\sum_{i=0}^{n-1} v_{m+i}\beta_{i+1}), \qquad (5.5)$$

where the index of β is taken modulo n. Using Lemma 1, one gets

$$\begin{aligned}\zeta(m-1,V) &= (m-1,\alpha^{-1}\sum_{i=0}^{n-2} v_{m+i}\beta_i + \\ &\qquad \beta_0\sum_{i=0}^{n-2} v_{m+i}p_{i+1} + v_{m+n-1}\beta_{n-1}\alpha^{-1}) \\ &= (m+1,\alpha^{-1}X + \beta_0\sum_{i=0}^{n-2} v_{m+i}p_{i+1}). \qquad (5.6)\end{aligned}$$

However, Lemma 2 and definition (5.1) give

$$\begin{aligned}Tr(\alpha^{-1}\sigma X) &= \sum_{i=0}^{n-1} v_{m+i}Tr(\alpha^{-1}\sigma\beta_i) \\ &= \sum_{i=0}^{n-2} v_{m+i}p_{i+1}. \qquad (5.7)\end{aligned}$$

From (5.6) and (5.7) one can see that

$$\zeta(m+1, V) = (m+1, \alpha^{-1}X + \beta_0 Tr(\alpha^{-1}\sigma X)).$$

Thus (m, X) is connected to $(m-1, \alpha^{-1}X + \beta_0 Tr(\alpha^{-1}\sigma X))$.

Finally, since $\overline{v}_m = v_m + 1$, the image of the third neighbor of (m, V) is given by

$$\begin{aligned} \zeta(m, V_1) &= (m, \sum_{i=0}^{n-1} v_{m+i}\beta_i) + \beta_0 \\ &= (m, X + \beta_0). \end{aligned}$$

Thus (m, X) is connected to $(m, X + \beta_0)$. ∎

Fig. 5.3 shows the connectivity of the algebraic model for CCC given by Theorem 20.

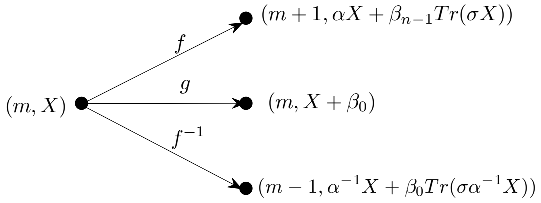
$(m+1, \alpha X + \beta_{n-1} Tr(\sigma X))$
f
g
(m, X)
$(m, X + \beta_0)$
f^{-1}
$(m-1, \alpha^{-1}X + \beta_0 Tr(\sigma\alpha^{-1}X))$

Figure 5.3: The connectivity of the Cube Connected Cycles graph CCC$_n$.

Note that unlike its binary counterpart, this connectivity is amenable to algebraic manipulation. Recall also that in binary representation, an edge from node (m, V) ended on node $(m, V \oplus 2^m)$. Thus the second coordinate of the destination depends on both, the first and the second, coordinates of the source. On the other hand, in the new algebraic model, each coordinate of a destination depends only on the corresponding coordinate of the source (see Fig. 5.3). This, in our opinion, would greatly simplify explorations of the CCC$_n$ network. Finally, note that similar to the Shuffle Exchange graphs, within the context of the algebraic model of CCC$_n$, one can make the following observation. If

Table 5.1: Equivalence between the nodes of CCC_4 and graph $C_4 \times GF(2^4)$.

label	$(m,\ X)$	label	$(m,\ X)$	label	$(m,\ X)$	label	$(m,\ X)$
(0, 0000)	$(0,0)$	(1, 0000)	$(1,0)$	(2, 0000)	$(2,0)$	(3, 0000)	$(3,0)$
(0, 0001)	$(0,\alpha^{14})$	(1, 0001)	$(1,1)$	(2, 0001)	$(2,\alpha)$	(3, 0001)	$(3,\alpha^{2})$
(0, 0010)	$(0,\alpha^{2})$	(1, 0010)	$(1,\alpha^{14})$	(2, 0010)	$(2,1)$	(3, 0010)	$(3,\alpha)$
(0, 0011)	$(0,\alpha^{13})$	(1, 0011)	$(1,\alpha^{3})$	(2, 0011)	$(2,\alpha^{4})$	(3, 0011)	$(3,\alpha^{5})$
(0, 0100)	$(0,\alpha)$	(1, 0100)	$(1,\alpha^{2})$	(2, 0100)	$(2,\alpha^{14})$	(3, 0100)	$(3,1)$
(0, 0101)	$(0,\alpha^{7})$	(1, 0101)	$(1,\alpha^{8})$	(2, 0101)	$(2,\alpha^{7})$	(3, 0101)	$(3,\alpha^{8})$
(0, 0110)	$(0,\alpha^{5})$	(1, 0110)	$(1,\alpha^{13})$	(2, 0110)	$(2,\alpha^{3})$	(3, 0110)	$(3,\alpha^{4})$
(0, 0111)	$(0,\alpha^{12})$	(1, 0111)	$(1,\alpha^{6})$	(2, 0111)	$(2,\alpha^{9})$	(3, 0111)	$(3,\alpha^{10})$
(0, 1000)	$(0,1)$	(1, 1000)	$(1,\alpha)$	(2, 1000)	$(2,\alpha^{2})$	(3, 1000)	$(3,\alpha^{14})$
(0, 1001)	$(0,\alpha^{3})$	(1, 1001)	$(1,\alpha^{4})$	(2, 1001)	$(2,\alpha^{5})$	(3, 1001)	$(3,\alpha^{13})$
(0, 1010)	$(0,\alpha^{8})$	(1, 1010)	$(1,\alpha^{7})$	(2, 1010)	$(2,\alpha^{8})$	(3, 1010)	$(3,\alpha^{7})$
(0, 1011)	$(0,\alpha^{6})$	(1, 1011)	$(1,\alpha^{9})$	(2, 1011)	$(2,\alpha^{10})$	(3, 1011)	$(3,\alpha^{12})$
(0, 1100)	$(0,\alpha^{4})$	(1, 1100)	$(1,\alpha^{5})$	(2, 1100)	$(2,\alpha^{13})$	(3, 1100)	$(3,\alpha^{3})$
(0, 1101)	$(0,\alpha^{9})$	(1, 1101)	$(1,\alpha^{10})$	(2, 1101)	$(2,\alpha^{12})$	(3, 1101)	$(3,\alpha^{6})$
(0, 1110)	$(0,\alpha^{10})$	(1, 1110)	$(1,\alpha^{12})$	(2, 1110)	$(2,\alpha^{6})$	(3, 1110)	$(3,\alpha^{9})$
(0, 1111)	$(0,\alpha^{11})$	(1, 1111)	$(1,\alpha^{11})$	(2, 1111)	$(2,\alpha^{11})$	(3, 1111)	$(3,\alpha^{11})$

$(m, X_1) \xrightarrow{f} (m+1, Y_1)$ and $(m, X_2) \xrightarrow{f} (m+1, Y_2)$,
then, $(m, X_1 + X_2) \xrightarrow{f} (m+1, Y_1 + Y_2)$.
Similarly, If
$(m, X_1) \xrightarrow{f^{-1}} (m-1, Y_1)$ and $(m, X_2) \xrightarrow{f^{-1}} (m-1, Y_2)$,
then, $(m, X_1 + X_2) \xrightarrow{f^{-1}} (m-1, Y_1 + Y_2)$.
We will refer to this observation as the *linearity in source property* of the f and f^{-1} edges of the CCC$_n$ graph.

Table 5.1 provides the mapping ζ between the two representations of CCC_4. In order to illustrate the mapping from Binary to Algebraic notation, consider mapping of a Cube Connected Cycles node (1, 0110) $\in Z_n \times Z_2^n$ to its new algebraic setting. The dual basis of $GF(2^4)$ given in Table 2.2 is $\langle \beta_3, \beta_2, \beta_1, \beta_0 \rangle = \langle 1, \alpha, \alpha^2, \alpha^{14} \rangle$. Thus

$$\begin{aligned} \zeta(1,\ 0110) &= (1,\ \beta_0 + \beta_1) \\ &= (1,\ \alpha^{14} + \alpha^2) \\ &= \alpha^{13}. \end{aligned}$$

Thus the Cube Connected Cycles node with binary label (1, 0110) is renamed in the new algebraic notation as $(1,\ \alpha^{13})$. The CCC$_4$ relabeled in the algebraic notation is shown in Fig.5.4.

5.3 Path Algorithms for Cube Connected Cycles

This section deals with designing a path to travel between any two nodes of the Cube Connected Cycles network. We start by stating a result that will help us minimize the path length.

Lemma 5 *A path with n consecutive f edges forms a cycle in CCC_n.*

Proof. We have

$$\begin{aligned}(m, \beta_i) \quad &\xrightarrow{f} \quad (m+1, \alpha\beta_i + \beta_{n-1}Tr(\sigma\beta_i)) \\ &= \quad (m+1, \alpha\beta_i + p_i\beta_{n-1}) \text{ from Lemma 2} \\ &= \quad (m+1, \beta_{i-1}) \text{ from Lemma 1.}\end{aligned}$$

Consequently, starting from any (m, β_i) and traversing n f edges will bring one back to the starting node. Since any $X \in GF(2^n)$ can be decomposed into a sum of β_is, the linearity of the f edges (see discussion after Theorem 22) implies that the cycle characteristics is also true of any starting node (m, X). ∎

We are now ready to use the algebraic machinery to chart a path from a node $(0, X)$ to the node $(a, 0)$ in CCC$_n$ for any given $a \in C_n$ and $X \in GF(2^n)$. Because of the symmetry of CCC$_n$, one can transform the problem of finding the path between any two arbitrary nodes to the one of finding a path between such a node pair. We develop two strategies to determine such a path.

In our first strategy, we employ the edges f and g only. Since the g edge is its own inverse, it can be followed only by an f edge. Thus there are only two possible paths to go from the mth column of CCC$_n$ to the $(m+1)$th column. In first of these paths, $(m, D) \xrightarrow{f} (m+1, D')$, where $D' = \alpha D + \beta_{n-1}Tr(\sigma D)$, while for the second

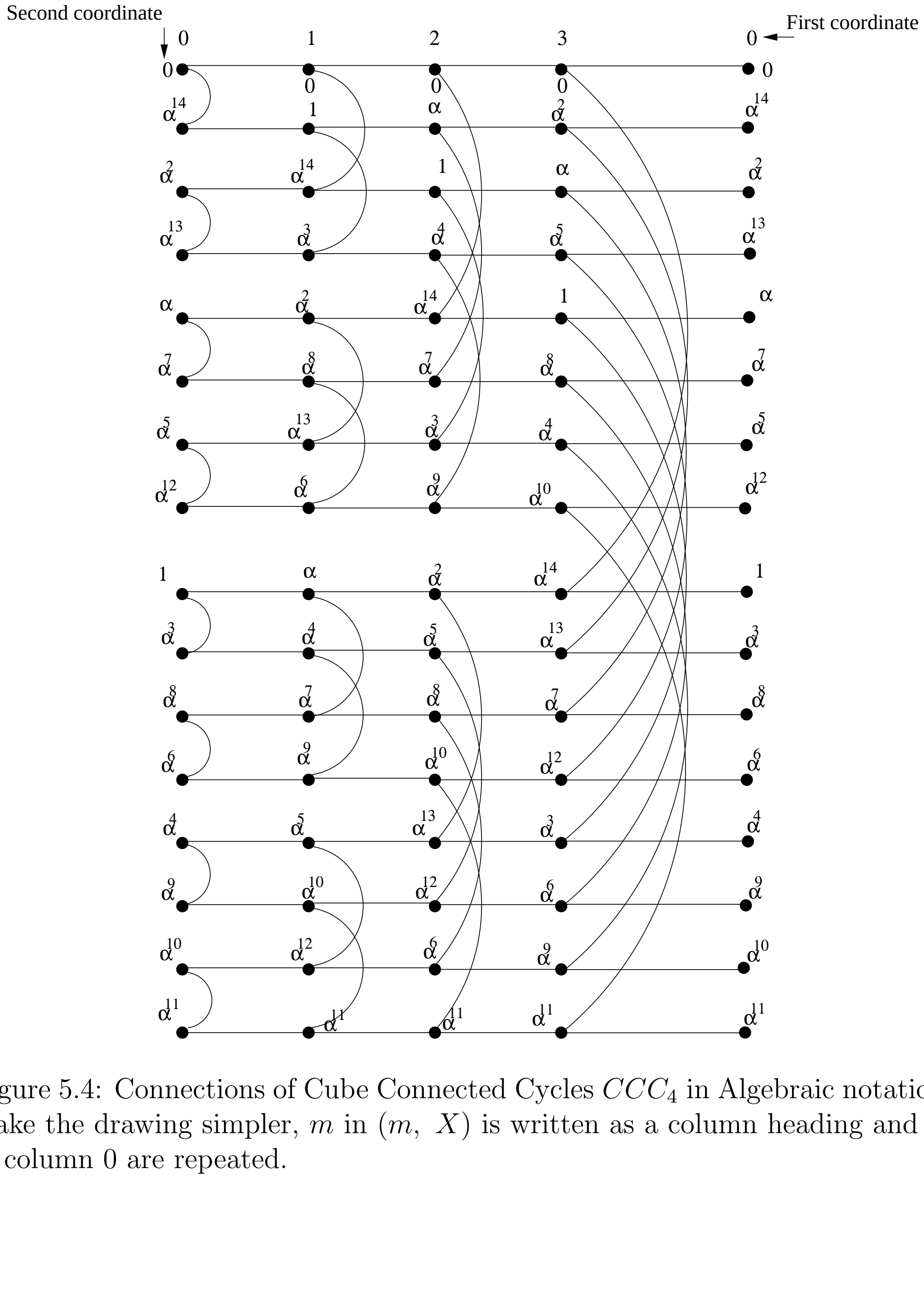

Figure 5.4: Connections of Cube Connected Cycles CCC_4 in Algebraic notation. To make the drawing simpler, m in $(m,\ X)$ is written as a column heading and nodes in column 0 are repeated.

path, $(m, D) \xrightarrow{g} (m, D + \beta_0) \xrightarrow{f} (m+1, D')$, where $D' = \alpha D + \beta_{n-1} + \beta_{n-1} Tr(\sigma D + \sigma\beta_0) = \alpha D + \beta_{n-1}(1 + Tr(\sigma D))$. This last simplification uses Lemma 2. Thus in both cases, the second coordinate of the destination node, $(m+1, D')$, can be expressed as

$$D' = \alpha D + \beta_{n-1}(c + Tr(\sigma D)), \tag{5.8}$$

where c is either 0 or 1. We will refer to the path going from (m, D) to $(m+1, D')$ as a path segment. Clearly, each path segment in this strategy is made of either an f edge or a g edge followed by an f edge.

To express the coordinates of any node along the path, one can apply (5.8) repeatedly. We begin by designating the starting node as (m, D_0) and a node reached after i path segments as $(m+i, D_i)$. Let c_i denote the value of binary constant c used in the ith path segment. From (5.8) one gets,

$$D_1 = \alpha D_0 + \beta_{n-1}(c_0 + Tr(\sigma D_0)). \tag{5.9}$$

Using (5.8) repeatedly and simplifying the result each time using Lemmas 2 and 1 gives the destination $(m+k, D_k)$ after k path segments as

$$(m + k, \alpha^k D_0 + \sum_{j=0}^{k-1} \beta_{n-k+j}(c_j + Tr(\alpha^j \sigma D_0)). \tag{5.10}$$

Assuming the starting node $(m, D_0) = (0, X)$ and the destination node $(m+k, D_k) = (a, 0)$, then

$$\begin{aligned} k &= a \bmod n \quad \text{and} \\ 0 &= \alpha^k X + \sum_{j=0}^{k-1} \beta_{n-k+j}(c_j + Tr(\alpha^j \sigma X)). \end{aligned} \tag{5.11}$$

Values of k and c_j, $0 \leq j < k$ satisfying (5.11) give the required path.

To solve (5.11), first note that for any $k \geq n$, the summation in (5.11) goes over all the β_j, $0 \leq j < n$. Since $\alpha^k X$ has a unique decomposition in the dual basis, one can always find c_is to satisfy (5.11) in this case. For smallest such k, $k = n + a$, (5.11) becomes

$$\sum_{j=0}^{n-1} Tr(\alpha^{n+a+j} X)\beta_j = \sum_{j=0}^{n+a-1} \beta_{-a+j}(c_j + Tr(\alpha^j \sigma X)), \tag{5.12}$$

where we have expressed $\alpha^{n+a}X$ on the left hand side of the expression in its dual basis. Comparing the coefficients of β_j, $0 \leq j < n-a$, on both sides of (5.12) gives

$$Tr(\alpha^{n+a+j}X) = c_{j+a} + Tr(\alpha^{j+a}\sigma X).$$

By using the linearity of the trace function and the fact that $\sigma = 1 + \alpha^n$ gives

$$\begin{aligned} c_{j+a} &= Tr(\alpha^{j+a}X),\ 0 \leq j < n-a \ \text{ or,} \\ c_j &= Tr(\alpha^j X),\ a \leq j < n. \end{aligned} \tag{5.13}$$

Similarly, comparing the coefficients of β_j, $n-a \leq j < n$, in (5.12) gives

$$\begin{aligned} Tr(\alpha^{n+a+j}X) &= c_{j+a} + Tr(\alpha^{j+a}\sigma X) + c_{j-n+a} + \\ &\qquad Tr(\alpha^{j-n+a}\sigma X). \end{aligned}$$

Simplifying this as before gives

$$\begin{aligned} c_{j+a} + c_{j+a-n} &= Tr(\alpha^{j+a-n}X),\ n-a \leq j < n \ \text{ or} \\ c_j + c_{n+j} &= Tr(\alpha^j X),\ 0 \leq j < a. \end{aligned} \tag{5.14}$$

For a smaller $k = a$, the summation in (5.11) does not span all the β_j, $0 \leq j < n$ of the dual basis. Therefore all X values may not yield a solution to (5.11). In particular, with $k = a$, (5.11) becomes

$$\sum_{j=0}^{n-1} Tr(\alpha^{a+j}X)\beta_j = \sum_{j=0}^{a-1} \beta_{-a+j}(c_j + Tr(\alpha^j \sigma X)). \tag{5.15}$$

All the path segments as described here end with an f edge. In order to provide a greater flexibility at designing the path, we allow a last g edge (if required) after the a path segments to reach the destination node. Using the last g edge has the effect of adding β_0 to the expression on the right hand side of (5.15). By comparing the coefficients of various β_js on both sides of this equation as before, one gets

$$\begin{aligned} & c_j = Tr(\alpha^j X),\ 0 \leq j < a, \\ & \text{last } g \text{ edge to be used if } \ Tr(\alpha^a X) = 1 \text{ and} \\ & Tr(\alpha^j X) = 0,\ a < j < n. \end{aligned} \tag{5.16}$$

The discussion above, including the computation of c_is from (5.13), (5.14) and (5.16), provide the following path algorithm.

Algorithm 2 (Path to go from $(\mathbf{0}, \mathbf{X})$ to $(\mathbf{a}, \mathbf{0})$ in $\mathbf{CCC_n}$ using edges f and g.

```
If $Tr(\alpha^i X) = 0$, for all $a < i < n$, then
    Set PathSegments to $a$, LastGEdge $= Tr(\alpha^a X)$ and
    choose binary values $c_i = Tr(\alpha^i X)$, $0 \leq i < a$.
Else Set PathSegments to $a + n$,
    choose binary values $c_i$, $0 \leq i < a + n$, as
    $c_i + c_{i+n} = Tr(\alpha^i X)$, $0 \leq i < a$ and
    $c_i = Tr(\alpha^i X)$, $a \leq i < n$.
    (Note: $c_i$, $c_{i+n}$, $0 \leq i < a$ are not unique.)
Start from the node $(0, X)$.
For i from 0 to PathSegments do
    If $c_i = 1$, proceed along a $g$ followed an $f$ edge.
    If $c_i = 0$, proceed along an $f$ edge.
If PathSegments = $a$ and LastGEdge = 1,
    proceed along the $g$ edge.
```

Note that the path obtained by this algorithm can sometimes be shortened. Because of Lemma 5, any time $t > \lfloor n/2 \rfloor$ consecutive f edges are indicated by the algorithm, they can be replaced by $n - t$ f^{-1} edges.

We illustrate the algorithm with the following examples.

Example 1. (path from $(0, \alpha^7)$ to $(2, 0)$ in CCC_4).

In this case, $Tr(\alpha^3 \alpha^7) = 0$. Therefore one needs only 2 path segments in this path. By using appropriate traces, one has: $c_0 = Tr(\alpha^0 \alpha^7) = 1$, $c_1 = Tr(\alpha \alpha^7) = 0$ and the last g edge is to be used because $Tr(\alpha^2 \alpha^7) = 1$. The required path then uses the edge sequence gf, f, g (We have separated path segments by commas for clarity). The actual path is given by: $(0, \alpha^7) \xrightarrow{g} (0, \alpha) \xrightarrow{f} (1, \alpha^2) \xrightarrow{f} (2, \alpha^{14}) \xrightarrow{g} (2, 0)$.

Example 2. (path from $(0, \alpha^6)$ to $(2, 0)$ in CCC_4).

In this case, one needs 6 path segments. By following the procedure of the algorithm,

$c_0+c_4=1$, $c_1+c_5=1$, $c_2=0$ and $c_3=1$. To satisfy the first two of these equations, we choose $c_0=c_1=1$ and $c_4=c_5=0$. The path will then use the edge sequence gf, gf, f, gf, f, f. Since in CCC_4, four consecutive f edges from any node return one to the same node, $fff \equiv f^{-1}$. Thus, in this case, a shorter path to the destination is given by the edge sequence gf, gf, f, gf^{-1}. The actual path is given by:
$(0,\alpha^6) \xrightarrow{g} (0,\alpha^8) \xrightarrow{f} (1,\alpha^7) \xrightarrow{g} (1,\alpha) \xrightarrow{f}$
$(2,\alpha^2) \xrightarrow{f} (3,\alpha^{14}) \xrightarrow{g} (3,0) \xrightarrow{f^{-1}} (2,0)$

We can also create a path from $(0,X)$ to $(a,0)$ using the f^{-1} and g edges. As before, since g edges cannot follow each other, the path segments going from a column m to a column $m-1$ will be made up of edges f^{-1} or gf^{-1}. Let the starting node be (m,D). The destination of the first path segment can be computed to be the node $(m-1,D')$, where $D' = \alpha^{-1}D + \beta_0 Tr(\alpha^{-1}\sigma D) + c_0\beta_1$, where the binary value c_0 equals 0 if the path segment is f^{-1} and 1, if it is gf^{-1}. The node on the path after going through k such path segments is given by

$$(m-k, \alpha^{-k}D + \sum_{j=1}^{k} \beta_{k-j} Tr(\alpha^{-j}\sigma D) + \sum_{j=0}^{k-1} c_j \beta_{k-j}), \tag{5.17}$$

where c_j, $0 \le j < k$, is the binary constant used in the jth path segment.

With the starting node $(0,X)$, (5.17) will give the destination node $(a,0)$ after k path segments if

$$\begin{aligned} a &= -k \bmod n \quad \text{and} \\ 0 &= \alpha^k X + \sum_{j=1}^{k} \beta_{k-j} Tr(\alpha^{-j}\sigma X) + \sum_{j=0}^{k-1} c_j \beta_{k-j}). \end{aligned} \tag{5.18}$$

As before, we need to consider only two cases; $k = (n-a) \bmod n$ and $k = (n-a) \bmod n + n$.

When $k = n + (n-a) \bmod n$, (5.18) has a solution for every X because all the basis vectors of the dual base, β_i, $0 \le i < n$ are available on the right hand side. By matching the coefficients of each β_i on both the sides of (5.18), one can obtain relationships between c_js. Comparing coefficients of β_0, one gets

$$Tr(\alpha^{-k}X) = Tr(\alpha^{-k}\sigma X) + Tr(\alpha^{n-k}\sigma X) + c_{k-n}.$$

On simplification, this yields

$$c_{n-a} = Tr(\alpha^a X). \tag{5.19}$$

Similarly, coefficients of β_i, $1 \leq i < k-n$, one gets

$$\begin{aligned} Tr(\alpha^{i-k}X) &= Tr(\alpha^{i-k}\sigma X) + Tr(\alpha^{i-k+n}\sigma X) + \\ & \qquad c_{k-i} + c_{k-i-n}. \end{aligned}$$

This equation can be simplified to yield

$$c_i + c + i + n = Tr(\alpha^{n-i}X), \ 0 < i < n-a. \tag{5.20}$$

Similarly, comparing coefficients of β_{k-n}, one gets

$$Tr(\alpha^{-n}X) = Tr(\alpha^n \sigma X) + c_0 + c_n,$$

which simplifies to

$$c_0 + c + n = Tr(X). \tag{5.21}$$

Finally, comparing coefficients of β_i, $k-n < i < n$, one gets

$$Tr(\alpha^{i-k}X) = Tr(\alpha^{i-k}\sigma X) + c_{k-i},$$

which gives

$$c_i = Tr(\alpha^{n-i}X), \ n-a < i < n. \tag{5.22}$$

When $k = (n-a)\text{mod } n$, (5.18) may not have a solution for all x values. In this case, $a > 0$ as is obvious from (5.18). For this $k = n-a$, (5.18) becomes

$$\begin{aligned} \sum_{j=0}^{n-1} Tr(\alpha^{n-a+j}X)\beta_j &= \sum_{j=1}^{n-a} \beta_{n-a-j} Tr(\alpha^{-j}\sigma X) \\ & \quad + \sum_{j=0}^{n-a-1} c_j \beta_{k-j}. \end{aligned} \tag{5.23}$$

The path described here necessarily ends in an f^{-1} edge. To make the strategy more flexible, we allow for a last g edge which may reach the destination node in the same column, a. With this, the expression on the right hand side of (5.23) gets added with an additional β_0. Solution of this equation gives

$$\begin{aligned}
&c_0 = Tr(X),\\
&c_j = Tr(\alpha^{n-j}X),\ 1 \le j < n-a,\\
&\text{last } g \text{ edge to be used if } \ Tr(\alpha^a X) = 1 \text{ and}\\
&Tr(\alpha^j X) = 0,\ n-a < j < n. \qquad (5.24)
\end{aligned}$$

This discussion gives the following path algorithm using f^{-1} and g edges.

Algorithm 3 (Path to go from $(\mathbf{0},\mathbf{X})$ to $(\mathbf{a},\mathbf{0})$ in $\mathbf{CCC_n}$ using edges $\mathbf{f^{-1}}$ and g).

```
If $a = 0$, Set PathSegments to $n - a$,
    choose binary values $c_0 = Tr(X)$ and
    $c_i = Tr(\alpha^{n-i}X)$, $1 \le i < n - a$.
Else If $a > 0$ and $Tr(\alpha^i X) = 0$, for all $0 < i < a$, then
    Set PathSegments to $n - a$, LastGEdge $= Tr(\alpha^a X)$
    and choose binary values $c_0 = Tr(X)$,
    $c_i = Tr(\alpha^{n-i}X)$, $1 \le i < n - a$.
Else Set PathSegments to $n + (n - a) mod\ n$,
    choose binary values $c_i$, $0 \le i < a + n$, as
    $c_0 + c_n = Tr(X)$, $c_i + c_{i+n} = Tr(\alpha^{n-i}X)$,
    $0 < i < n - a$ and $c_i = Tr(\alpha^{n-i}X)$, $n - a \le i < n$.
    (Note: $c_i$, $c_{i+n}$, $0 \le i < n - a$ are not unique.)
Start from node $(0, X)$.
For i from 0 to PathSegments do
    If $c_i = 1$, proceed along a $g$ followed by an $f^{-1}$ edge.
    If $c_i = 0$, proceed along an $f^{-1}$ edge.
If PathSegments $= n - a$ and LastGEdge = 1,
    proceed along the $g$ edge.
```

Note that, as in the case of the first algorithm, the path obtained by this algorithm can sometimes be shortened using Lemma 5. Any time $t > \lfloor n/2 \rfloor$ consecutive f^{-1} edges are indicated by this algorithm, they can be replaced by $n - t$ f edges.

Following example illustrates the algorithm.

Example 3. (path from $(0, \alpha^{11})$ to $(1, 0)$ in CCC_4).
In this case, since $a = 1$, the condition $Tr(\alpha^j X) = 0$, $n - a < j < n$ in (5.21) is obviously satisfied. Thus we can use only $n - a = 3$ path segments. From step 1 of algorithm 3, one gets $c_0 = c_1 = c_2 = 1$. Further, since $Tr(\alpha\alpha^{11}) = 1$, one should use one extra g edge at the end. The edge sequence is therefore $gf^{-1}, gf^{-1}, gf^{-1}, g$.
The actual path is given by:
$(0, \alpha^{11}) \xrightarrow{g} (0, \alpha^{10}) \xrightarrow{f^{-1}} (3, \alpha^9) \xrightarrow{g} (3, \alpha^4) \xrightarrow{f^{-1}}$
$(2, \alpha^3) \xrightarrow{g} (2, 1) \xrightarrow{f^{-1}} (1, \alpha^{14}) \xrightarrow{g} (1, 0)$.

Example 4. (path from $(0, \alpha^5)$ to $(2, 0)$ in CCC_4).
In this case, one has $c_0 + c_4 = Tr(\alpha^5) = 0$, $c_1 + c_5 = Tr(\alpha^8) = 0$, $c_2 = Tr(\alpha^7) = 1$ and $c_3 = Tr(\alpha^6) = 1$, We use $c_0 = c_4 = c_1 = c_5 = 0$ to satisfy the relationships between c_is. Thus the edge sequence of the path is $f^{-1}, f^{-1}, gf^{-1}, gf^{-1}, f^{-1}, f^{-1}$. Since $f^{-1}, f^{-1}, f^{-1} = f$ in CCC_4, one can use a shorter edge sequence $f^{-1}f^{-1}, gf^{-1}, gf$.
The actual path is given by:
$(0, \alpha^5) \xrightarrow{f^{-1}} (3, \alpha^4) \xrightarrow{f^{-1}} (2, \alpha^3) \xrightarrow{g} (2, 1) \xrightarrow{f^{-1}}$
$(1, \alpha^{14}) \xrightarrow{g} (1, 0) \xrightarrow{f} (2, 0)$.

Example 5. (path from $(0, \alpha^5)$ to $(0, 0)$ in CCC_4).
In this case, since $a = 0$ there will be no extra g edge at the end. Thus, we can use only $n - a = 4$ path segments. From step 1 of algorithm 3, one gets $c_0 = c_1 = 0, c_2 = c_3 = 1$. Thus the edge sequence of the path is $f^{-1}, f^{-1}, gf^{-1}, gf^{-1}$.
The actual path is given by:
$(0, \alpha^5) \xrightarrow{f^{-1}} (3, \alpha^4) \xrightarrow{f^{-1}} (2, \alpha^3) \xrightarrow{g} (2, 1) \xrightarrow{f^{-1}}$
$(1, \alpha^{14}) \xrightarrow{g} (1, 0) \xrightarrow{f^{-1}} (0, 0)$.

One can show that the algorithms 2 and 3 provide paths which are less than the diameter of the Cube Connected Cycles graph as given in the following theorem.

Theorem 21 *The path algorithms 2 and 3 provide a path less than the diameter of*

CCC$_n$.

Proof. The diameter of CCC$_n$ is 6 if $n = 3$ and $2n + \lfloor n/2 \rfloor - 2$ if $n > 3$ [6]. We show that the path obtained by one of the two algorithms is always less than the diameter.

Because of symmetry of CCC$_n$, the path between any pair of nodes in CCC$_n$ is isomorphic to a path between $(0, X)$ and $(a, 0)$ with appropriately chosen $a \in C_n$ and $X \in GF(2^n)$. We therefore only focus on these paths using algorithms 2 and 3.

The theorem for $n = 3$ can be proved from algorithm 2 rather easily. for $a = 0$, the constants c_0, c_1 and c_2 are either 0 or 1. Since $c_i = 0$ implies an f edge and $c_i = 1$, an edge sequence gf, even when each c_i is 1, the path length is at most 6. For $a = 1$, even if all traces that give the c_is are 1, one can choose $c_0 = c_1 = c_2 = 1$ and $c_3 = 0$. This results in the edge sequence gf, gf, gf, f, which, from Lemma 5 equals $gfgfgf^{-1}$, a path of length 6. Finally, when $a = 2$, in the worst case (of all trace functions are 1), one can choose $c_0 = c_1 = c_2 = 1$ and $c_3 = c_4 = 0$, giving the edge sequence $gf, gf, gf, f, f = gfgfg$, a path of length 5.

When $a > 3$, the choice of algorithm can be based on a (for the purpose of this proof). If $\lceil n/2 \rceil \leq a < n$, one can use algorithm 2. If the number of path segments equals a, then the path length is at most $2a \leq 2(n-1)$. If the number of path segments equal $a + n$, then $c_i + c_{i+n}$, $0 \leq i < a$ are fixed, but individual $c_i, c_{i+n} \in GF(2)$ are not. We choose $c_n = c_{n+1} = \cdots = c_{n+a-1} = 0$. Value c_{n-1} may be either a 0 or a 1. Since each 0 value of c_i implies an f edge, while a 1, gf edges, at least $a+1$ edges at the end of the path are f edges. Using Lemma 5, these consecutive $a + 1$ f edges can be replaced with $(n - a - 1)$ f^{-1} edges. The path length is then given by the number of edges due to c_i, $0 \leq i < n - 1$, at most one g edge due to c_{n-1} and $(n - a - 1)$ f^{-1} edges at the end. We therefore have path length $\leq 2(n-1) + 1 + (n - a - 1) \leq 2n + \lfloor n/2 \rfloor - 2$.

On the other hand, if $0 \leq a < \lceil n/2 \rceil$, we use algorithm 3. If the number of path segments equals $n - a$, then the path length is at most $2(n - a) \leq 2n$ because each path segment is made up of at most two edges, This also covers the case when $a = 0$. If $a \neq 0$ and the number of path segments equal $(2n - a)$, then $c_i + c_{i+n}$,

$0 \leq i < n - a$ are fixed, but individual $c_i, c_{i+n} \in GF(2)$ are not. As before, we choose $c_n = c_{n+1} = \cdots = c_{2n-a-1} = 0$. Value c_{n-1} may be either a 0 or a 1. Since each 0 value of c_i implies an f^{-1} edge, while a 1, gf^{-1} edges, at least $(n - a + 1)$ edges at the end of the path are f^{-1} edges. Using Lemma 5 again, these consecutive $n - a + 1$ f^{-1} edges can be replaced with $(a - 1)$ f edges. Thus the path length in this case satisfies path length $\leq 2(n - 1) + 1 + (a - 1) \leq 2n + \lfloor n/2 \rfloor = 2$.

Finally, when $X = \beta_0 + \beta_1 + \cdots + \beta_{n-1}$, $Tr(\alpha^i X) = 1$, $0 \leq i < n$. If $a = \lfloor n/2 \rfloor$, then using similar arguments, it can be shown that either of the two algorithms give the minimum path length from $(X, 0)$ to $(a, 0)$ to be $2n + \lfloor n/2 \rfloor - 2$. Therefore this is the diameter of CCC$_n$. ■

5.4 Automorphisms of the Cube Connected Cycles Graph

We now explore the automorphisms of the cube connected cycles graph CCC$_n$. For this, consider the constants K_0, K_1, …, $K_{n-1} \in GF(2^n)$ related to each other as

$$K_{m+1} = \alpha K_m + \beta_{n-1} Tr(\sigma K_m), \qquad 0 \leq m < n, \tag{5.25}$$

where, as will be shown later, the index of K can be considered modulo n.

Constants K_0 through K_{n-1} play a central role in characterizing the automorphisms. It is therefore worthwhile considering their interdependance first. Because each K_{m+1} is related to K_m, it is natural to expect that each of these constants can be obtained from K_0. The explicit dependence of K_m on K_0 is given by

$$K_m = \alpha^m K_0 + \sum_{j=1}^{m} Tr(\alpha^{m-j} \sigma K_0) \beta_{n-j}, \quad 0 \leq j < n. \tag{5.26}$$

Equation (5.26) can be proved by mathematical induction. It is obvious for $m = 0$. If it is true for some K_m, then its truth for K_{m+1} can be established using (5.25) and (5.26) as follows.

$$K_{m+1} \;\; = \;\; \alpha^{m+1} K_0 + \sum_{j=1}^{m} Tr(\alpha^{m-j} \sigma K_0) \beta_{n-j} \alpha + \beta_{n-1} Tr(\sigma \alpha^m K_0)$$

$$\begin{aligned} & \quad + \beta_{n-1} Tr(\sigma \sum_{j=1}^{m} Tr(\alpha^{m-j} \sigma K_0) \beta_{n-1}) \\ &= \alpha^{m+1} K_0 + \sum_{j=1}^{m} Tr(\alpha^{m-j} \sigma K_0)(\beta_{n-j-1} + p_{n-j} \beta_{n-1}) + \beta_{n-1} Tr(\sigma \alpha^m K_0) \\ & \quad + \beta_{n-1} \sum_{j=1}^{m} Tr(\alpha^{m-j} \sigma K_0) p_{n-j} \\ &= \alpha^{m+1} K_0 + \sum_{j=1}^{m} Tr(\alpha^{m-j} \sigma K_0) \beta_{n-j-1} + \beta_{n-1} Tr(\sigma \alpha^m K_0) \\ &= \alpha^{m+1} K_0 + \sum_{j=1}^{m+1} Tr(\alpha^{m+1-j} \sigma K_0) \beta_{n-j}, \end{aligned}$$

Let σK_0 be expanded in dual basis as

$$\sigma K_0 = \sum_{i=0}^{n-1} a_i \beta_i.$$

Then the relation (5.26) for $m = n$ gives

$$\begin{aligned} K_n &= \alpha^n + \sum_{j=1}^{n} \sum_{i=0}^{n-1} a_i Tr(\alpha^{n-j} \beta_i) \beta_{n-j} \\ &= \alpha^n + \sum_{j=1}^{n} a_{n-j} \beta_{n-j} \\ &= \alpha^n + \sigma K_0 = K_0. \end{aligned} \tag{5.27}$$

From this, one can clearly see that the index of K_m in (5.25) is modulo n.

Theorem 22 now states a set of automorphisms of the CCC graph.

Theorem 22 *A mapping $\phi(\cdot) : C_n \times GF(2^n) \to C_n \times GF(2^n)$ defined by*

$$\phi(m, X) = (m + t, X + K_m), \tag{5.28}$$

for arbitrary $t \in C_n$ and constants K_m's related to each other as in (5.25) is an automorphism of graph CCC_n.

Proof. It is clear that ϕ is a one-to-one onto mapping. We show that if two nodes N and N' are connected in CCC$_n$, then so are the nodes $\phi(N)$ and $\phi(N')$. Let $N = (m, X)$. Then we have

$$\phi(N) = (m + t, X + K_m) \tag{5.29}$$

There may be 3 kinds of edges between N and N'. We prove the result for each separately.

Case 1 $N' = (m+1, \alpha X + \beta_{n-1} Tr(\sigma X))$.

In this case we have

$$\begin{aligned} \phi(N') &= (m+t+1, \alpha X + \beta_{n-1} Tr(\sigma X) + K_{m+1}) \\ &= (m+t+1, \alpha X + \alpha K_m + \beta_{n-1} Tr(\sigma(X+K_m))) \\ &= (m+t+1, \alpha(X+K_m) + \beta_{n-1} Tr(\sigma(X+K_m))). \qquad (5.30) \end{aligned}$$

Eq.s (5.29) and (5.30) show that $\phi(N)$ is connected to $\phi(N')$.

Case 2 $N' = (m, \alpha X + \beta_0)$.

We now have

$$\phi(N') = (m+t, \alpha X + K_m + \beta_0). \qquad (5.31)$$

Clearly from eq.s (5.29) and (5.31) $\phi(N)$ is connected to $\phi(N')$ in this case also.

Case 3 $N' = (m-1, \alpha^{-1} X + \beta_0 Tr(\sigma\alpha^{-1} X))$.

The image of this N' under ϕ is given by

$$\phi(N') = (m+t-1, \alpha^{-1} X + \beta_0 Tr(\sigma\alpha^{-1} X) + K_{m-1}) \qquad (5.32)$$

Now from the relationship (5.25) between K_m and K_{m-1}, we get

$$\sigma\alpha^{-1} K_m = \sigma K_{m-1} + \sigma\beta_0 Tr(\sigma K_{m-1}), \qquad \text{or}$$

$$Tr(\sigma\alpha^{-1} K_m) = Tr(\sigma K_{m-1}).$$

Thus,

$$K_{m-1} = \alpha^{-1} K_m + \alpha^{-1}\beta_{n-1} Tr(\sigma\alpha^{-1} K_m). \qquad (5.33)$$

Combining (5.32) and (5.33) gives

$$\begin{aligned} \phi(N') &= (m+k-1, \alpha^{-1} X + \beta_0 Tr(\sigma\alpha^{-1} X) + \alpha^{-1} K_m + \beta_0 Tr(\sigma\alpha^{-1} K_m)) \\ &= (m+k-1, \alpha^{-1}(X+K_m) + \beta_0 Tr(\sigma\alpha^{-1}(X+K_m))). \qquad (5.34) \end{aligned}$$

From (5.29) and (5.34) one can see that $\phi(N)$ is connected to $\phi(N')$ in the third case also. ∎

We will refer to the automorphisms specified by Theorem 22 as the *automorphisms of the first kind.* The parameters K_0 through K_{n-1} characterizing the modification of the second coordinate of a node label will be called the *automorphism constants* the change in the first coordinate t would be called the automorphism offset.

We now demonstrate the use of automorphisms of first kind to prove the symmetry of CCC$_n$.

Corollary 2 *Cube connected cycles graph is symmetric.*

Proof. We prove the symmetry by showing that given any two nodes $N_1 = (a, X_1)$ and $N_2 = (b, X_2)$, there exists an automorphism $\phi(\cdot)$ of CCC$_n$ which maps N_1 to N_2.

Choose $t = (b - a) \text{mod } n$, constant $K_a = X_1 + X_2$ and other constants K_i, $0 \leq i < n$, $i \neq a$ obtained using (5.25). Define the automorphism $\phi(\cdot)$ as in Theorem 22. Clearly, $\phi(N_1) = N_2$. ∎

As an illustration, we provide in Table 5.2 an example of an automorphism that maps $(1, \alpha^3)$ to $(2, \alpha^7)$ in CCC$_4$. First, we calculate the column offset $t = 2 - 1 = 1$. This first calls for computing constants K_0 through K_3 which will provide the automorphism as in Theorem 22. Note that in $GF(2^4)$, $\beta_{n-1} = \beta_3 = 1$ and $\sigma = (\alpha^4 + 1) = \alpha$. Using (5.25), we then get

$$\begin{aligned} K_1 &= \alpha^3 + \alpha^7 &&= \alpha^4 \\ K_2 &= \alpha K_1 + Tr(\sigma K_1) &&= \alpha^5 \\ K_3 &= \alpha K_2 + Tr(\sigma K_2) &&= \alpha^{13} \quad \text{and} \\ K_0 &= \alpha K_3 + Tr(\sigma K_3) &&= \alpha^3 \end{aligned}$$

Which leads to the automorphism mapping shown in Table 5.2.

The automorphisms constants can also be determined from the traces $Tr(\sigma K_i)$, $0 \leq i < n$ as shown in the following Theorem. This result would be used later to develop mappings on faulty CCC$_n$ graphs.

Table 5.2: an automorphism ϕ that maps $(1, \alpha^3)$ to $(2, \alpha^7)$ in CCC$_4$.

node N	$\phi(N)$	node N	$\phi(N)$	node N	$\phi(N)$	node N	$\phi(N)$
$(0, 0)$	$(1, \alpha^3)$	$(1, 0)$	$(2, \alpha^{14})$	$(2, 0)$	$(3, \alpha^5)$	$(3, 0)$	$(0, \alpha^{13})$
$(0, 1)$	$(1, \alpha^{14})$	$(1, 1)$	$(2, \alpha)$	$(2, 1)$	$(3, \alpha^{10})$	$(3, 1)$	$(0, \alpha^6)$
$(0, \alpha)$	$(1, \alpha^9)$	$(1, \alpha)$	$(2, 1)$	$(2, \alpha)$	$(3, \alpha^2)$	$(3, \alpha)$	$(0, \alpha^{12})$
$(0, \alpha^2)$	$(1, \alpha^6)$	$(1, \alpha^2)$	$(2, \alpha^{10})$	$(2, \alpha^2)$	$(3, \alpha)$	$(3, \alpha^2)$	$(0, \alpha^{14})$
$(0, \alpha^3)$	$(1, 0)$	$(1, \alpha^3)$	$(2, \alpha^7)$	$(2, \alpha^3)$	$(3, \alpha^{11})$	$(3, \alpha^3)$	$(0, \alpha^8)$
$(0, \alpha^4)$	$(1, \alpha^7)$	$(1, \alpha^4)$	$(2, 0)$	$(2, \alpha^4)$	$(3, \alpha^8)$	$(3, \alpha^4)$	$(0, \alpha^{11})$
$(0, \alpha^5)$	$(1, \alpha^{11})$	$(1, \alpha^5)$	$(2, \alpha^8)$	$(2, \alpha^5)$	$(3, 0)$	$(3, \alpha^5)$	$(0, \alpha^7)$
$(0, \alpha^6)$	$(1, \alpha^2)$	$(1, \alpha^6)$	$(2, \alpha^{12})$	$(2, \alpha^6)$	$(3, \alpha^9)$	$(3, \alpha^6)$	$(0, 1)$
$(0, \alpha^7)$	$(1, \alpha^4)$	$(1, \alpha^7)$	$(2, \alpha^3)$	$(2, \alpha^7)$	$(3, \alpha^{13})$	$(3, \alpha^7)$	$(0, \alpha^5)$
$(0, \alpha^8)$	$(1, \alpha^{13})$	$(1, \alpha^8)$	$(2, \alpha^5)$	$(2, \alpha^8)$	$(3, \alpha^4)$	$(3, \alpha^8)$	$(0, \alpha^3)$
$(0, \alpha^9)$	$(1, \alpha)$	$(1, \alpha^9)$	$(2, \alpha^{14})$	$(2, \alpha^9)$	$(3, \alpha^6)$	$(3, \alpha^9)$	$(0, \alpha^{10})$
$(0, \alpha^{10})$	$(1, \alpha^{12})$	$(1, \alpha^{10})$	$(2, \alpha^2)$	$(2, \alpha^{10})$	$(3, 1)$	$(3, \alpha^{10})$	$(0, \alpha^9)$
$(0, \alpha^{11})$	$(1, \alpha^5)$	$(1, \alpha^{11})$	$(2, \alpha^{13})$	$(2, \alpha^{11})$	$(3, \alpha^3)$	$(3, \alpha^{11})$	$(0, \alpha^4)$
$(0, \alpha^{12})$	$(1, \alpha^{10})$	$(1, \alpha^{12})$	$(2, \alpha^6)$	$(2, \alpha^{12})$	$(3, \alpha^{14})$	$(3, \alpha^{12})$	$(0, \alpha)$
$(0, \alpha^{13})$	$(1, \alpha^8)$	$(1, \alpha^{13})$	$(2, \alpha^{11})$	$(2, \alpha^{13})$	$(3, \alpha^7)$	$(3, \alpha^{13})$	$(0, 0)$
$(0, \alpha^{14})$	$(1, 1)$	$(1, \alpha^{14})$	$(2, \alpha^9)$	$(2, \alpha^{14})$	$(3, \alpha^{12})$	$(3, \alpha^{14})$	$(0, \alpha^2)$

Theorem 23 *The n binary values $c_i = Tr(\sigma K_i)$, $0 \leq i < n$, uniquely determine the automorphism constants K_i, $0 \leq i < n$.*

Proof. From (5.25), one gets, $K_1 = \alpha K_0 + c_0 \beta_{n-1}$, $K_2 = \alpha^2 K_0 + (\alpha c_0 + c_1)\beta_{n-1}$, etc. After applying (5.25) n times, one obtains

$$K_0 = \alpha^n K_0 + \beta_{n-1} \sum_{i=0}^{n-1} c_i \alpha^{n-1-i}.$$

From this, one gets

$$K_0 = \sigma^{-1} \beta_{n-1} \sum_{i=0}^{n-1} c_i \alpha^{n-1-i}. \tag{5.35}$$

Once K_0 is determined, other K_i, $1 \leq i < n$ are fixed by (5.26). ∎

The symmetry of a network is important for designing mappings, avoiding faults and readjusting mapping templates to specific situations. For example, Algorithm 2 provides a path to travel from $(0, X)$ to $(a, 0)$. If we want to travel from (m_1, X_1) to (m_2, X_2) we can use symmetry to first transform the problem to one suitable for Algorithm 2 as follows. We first find an automorphism ϕ such that for some $X \in GF(2^n)$ and $a \in C_n$, one has

$$\phi(0, X) = (m_1, X_1) \tag{5.36}$$

and

$$\phi(a, 0) = (m_2, X_2) \tag{5.37}$$

From (5.36) one gets the column offset $t = m_1$. But from (5.37), $t = m_2 - a$, giving $a = m_2 - m_1$. Now, (5.37) gives $K_a = X_2$. From K_a, one can obtain K_0 as in (5.25). Then using Equation (5.36), one gets $X_1 = X + K_0$ or $X = X_1 + K_0$. Once X and a are thus obtained, one can use Algorithm 2 (or 3) to obtain a path from $(0, X)$ to $(a, 0)$. Applying automorphism ϕ to this path transforms it to one from (m_1, X_1) to (m_2, X_2).

We illustrate this procedure by finding a path from $(2, \alpha^6)$ to $(1, \alpha^8)$ in CCC$_4$. We start by defining an automorphism ϕ such that

$$\phi(0, X) = (2, \alpha^6) \tag{5.38}$$

and

$$(a, 0) = (1, \alpha^8) \tag{5.39}$$

The column offset from (5.38) is $t = 2$, and $t = 1 - a$ from (5.39). Thus, $a = 3$. From (5.39) gives $K_3 = \alpha^8$, which in turn yields $K_0 = \alpha^7$, $K_1 = \alpha^8$, $K_2 = \alpha^7$. Then, using Equation (5.38) one gets $K_0 = X + \alpha^6$ and $X = \alpha^6 + \alpha^7 = \alpha^{10}$. Therefore, finding a path from $(2, \alpha^6)$ to $(1, \alpha^8)$ becames a problem of finding a path between $(0, \alpha^{10})$ and $(3, 0)$ using Algorithm 2. The path obtained from the algorithm is: $(0, \alpha^{10}) \xrightarrow{f} (1, \alpha^{12}) \xrightarrow{g} (1, \alpha^5) \xrightarrow{f} (2, \alpha^{13}) \xrightarrow{g} (2, \alpha^2) \xrightarrow{f} (3, \alpha^{14}) \xrightarrow{g} (3, 0)$.

Applying ϕ to every node, one can then obtain the required path from $(2, \alpha^6)$ to $(1, \alpha^8)$ as:
$(2, \alpha^6) \xrightarrow{f} (3, \alpha^9) \xrightarrow{g} (3, \alpha^4) \xrightarrow{f} (0, \alpha^5) \xrightarrow{g} (0, \alpha^{12}) \xrightarrow{f} (1, \alpha^6) \xrightarrow{g} (1, \alpha^8)$.

We now count the total number of automorphisms of the first kind. Note that $K_0 \in GF(2^n)$ can have 2^n distinct values. Once K_0 is chosen, all other automorphism constants K_1 through K_{n-1} are fixed by (5.26). Similarly, the automorphism offset t can be chosen from n distinct values. Thus there are $n2^n$ different automorphisms of the first kind.

In order to describe the remaining automorphisms we define a function $\mu(\cdot) : GF(2^n) \rightarrow GF(2^n)$ as follows.

$$\mu(X) = \sum_{i=0}^{n-1} x_{n-i}\beta_i, \quad \text{where } X = \sum_{i=0}^{n-1} x_i\beta_i. \tag{5.40}$$

Note that the index of x_i is considered modulo n. When expressed in the dual basis, the components x_1 through x_{n-1} of X are reflected to obtain $\mu(X)$. Following Lemma specifies some basic properties of μ.

Lemma 6 *The function μ defined in (5.40) has following properties.*

1. *μ is a one-to-one mapping.*

2. *$\mu(\mu(X)) = X$ for all $X \in GF(2^n)$.*

3. *$\mu(X + Y) = \mu(X) + \mu(Y)$, for any $X, Y \in GF(2^n)$.*

4. *$\mu(X+\beta_0) = \mu(X) + \beta_0$, for any $X \in GF(2^n)$.*

5. *$\mu(\alpha X) = x_1\beta_0 + \sum_{i=0}^{n-1} x_i p_i \beta_1 + \sum_{i=2}^{n-1} x_i \beta_{n-i+1}$, for any $X \in GF(2^n)$, where p_i is the coefficient of α^i in the primitive polynomial.*

Proof. The first four assertions are obvious. We prove the last one using Lemma 1.

$$\begin{aligned}\mu(\alpha X) &= \mu\left(\sum_{i=0}^{n-1} x_i \alpha \beta_i\right) \\ &= \mu\left(x_0\beta_{n-1} + \sum_{i=1}^{n-1} x_i(\beta_{i-1} + p_i\beta_{n-1})\right) \\ &= x_1\beta_0 + \sum_{i=0}^{n-1} x_i p_i \beta_1 + \sum_{i=2}^{n-1} x_i \beta_{n-i+1}.\end{aligned}$$

∎

We can now specify a new automorphism of CCC$_n$ not covered by Theorem 22.

Theorem 24 *Mapping $\psi(m, X) = (n-m, \mu(X))$ is an automorphism of CCC_n.*

Proof. From Lemma 6 one can see that mapping ψ is one-to-one. To show that it preserves connectivity, consider the f edge between (m, X) and $(m+1, \alpha X + \beta_{n-1}Tr(\sigma X))$. The images of these nodes under ψ are: $N_1 = (n-m, \mu(X))$ and $N_2 = (n-m-1, Y)$ where, $Y = \mu(\alpha X + \beta_{n-1}Tr(\sigma X))$. We now show that there is an f edge from N_2 to N_1. Note that the column of N_1 is one higher than that of N_2. To show that their rows are related as required, we will show that $\alpha Y + \beta_{n-1}Tr(\sigma Y) = \mu(X)$. One has,

$$\alpha Y + \beta_{n-1}Tr(\sigma Y) = \alpha\mu(\alpha X) + \alpha\beta_1 Tr(\sigma X) + \beta_{n-1}(Tr(\sigma\mu(\alpha X)) + p_1 Tr(\sigma X)). \tag{5.41}$$

However,

$$\begin{aligned}\alpha\mu(\alpha X) + \alpha\beta_1 Tr(\sigma X) &= \alpha\mu(\alpha X) + \alpha\beta_1 \sum_{i=0}^{n-1} Tr(x_i\sigma\beta_i) \\ &= \alpha\mu(\alpha X) + \alpha\beta_1 \sum_{i=1}^{n-1} x_i p_i\end{aligned}$$

$$\begin{aligned} &= x_1\beta_{n-1} + x_0\alpha\beta_1 + \sum_{i=2}^{n-1} x_i\alpha\beta_{n-i+1} \\ &= x_1\beta_{n-1} + x_0\beta_0 + x_0p_1\beta_{n-1} + \sum_{i=2}^{n-1} x_i\beta_{n-i} \\ &+ \sum_{i=2}^{n-1} x_ip_{n-i+1}\beta_{n-1}. \end{aligned} \tag{5.42}$$

$$\begin{aligned} \beta_{n-1}Tr(\sigma\mu(\alpha X)) &= \beta_{n-1}[Tr(x_1\sigma\beta_0) + \sum_{i=2}^{n-1} x_iTr(\sigma\beta_{n-i+1}) + Tr(\sigma\beta_1)\sum_{i=0}^{n-1} x_ip_i] \\ &= \beta_{n-1}\sum_{i=2}^{n-1} x_ip_{n-i+1} + p_1\beta_{n-1}\sum_{i=0}^{n-1} x_ip_i. \end{aligned} \tag{5.43}$$

And finally,

$$\begin{aligned} p_1\beta_{n-1}Tr(\sigma X) &= p_1\beta_{n-1}\sum_{i=0}^{n-1} x_i\sigma\beta_i \\ &= p_1\beta_{n-1}\sum_{i=1}^{n-1} x_ip_i. \end{aligned} \tag{5.44}$$

By adding (5.42) through (5.44) gives from (5.41),

$$\begin{aligned} \alpha Y + \beta_{n-1}Tr(\sigma Y) &= x_1\beta_{n-1} + x_0\beta_0 + \sum_{i=2}^{n-1} x_i\beta_{n-i} \\ &= \mu(X) \end{aligned} \tag{5.45}$$

From (5.45) it is clear that there is an f edge from N_2 to N_1.

Similarly, nodes (m, X) and $(m, X + \beta_0)$ which are connected by a g edge have images $(n - m, \mu(X))$ and $(n - m, \mu(X + \beta_{n-1})) = (n - m, \mu(X) + \beta_{n-1})$. Clearly, these images are also connected by a g edge.

Thus the one-to-one mapping ψ preserves connectivity of CCC$_n$. It is therefore an automorphism of CCC$_n$. ∎

Automorphism $\psi(\cdot) : CCC_4 \to CCC_4$ is shown in Table 5.3.

Theorem 25 lists some basic properties of the automorphism $\psi(\cdot)$.

Theorem 25 *1. $\psi(\cdot)$ is an order 2 automorphism.*

Table 5.3: Automorphism $\psi(\cdot) : CCC_4 \rightarrow CCC_4$.

(m,X)	$\psi(m,X)$	(m,X)	$\psi(m,X)$	(m,X)	$\psi(m,X)$	(m,X)	$\psi(m,X)$
$(0,0)$	$(0,0)$	$(1,0)$	$(3,0)$	$(2,0)$	$(2,0)$	$(3,0)$	$(1,0)$
$(0,1)$	$(0,\alpha^2)$	$(1,1)$	$(3,\alpha^2)$	$(2,1)$	$(2,\alpha^2)$	$(3,1)$	$(1,\alpha^2)$
$(0,\alpha)$	$(0,\alpha)$	$(1,\alpha)$	$(3,\alpha)$	$(2,\alpha)$	$(2,\alpha)$	$(3,\alpha)$	$(1,\alpha)$
$(0,\alpha^2)$	$(0,1)$	$(1,\alpha^2)$	$(3,1)$	$(2,\alpha^2)$	$(2,1)$	$(3,\alpha^2)$	$(1,1)$
$(0,\alpha^3)$	$(0,\alpha^{13})$	$(1,\alpha^3)$	$(3,\alpha^{13})$	$(2,\alpha^3)$	$(2,\alpha^{13})$	$(3,\alpha^3)$	$(1,\alpha^{13})$
$(0,\alpha^4)$	$(0,\alpha^5)$	$(1,\alpha^4)$	$(3,\alpha^5)$	$(2,\alpha^4)$	$(2,\alpha^5)$	$(3,\alpha^4)$	$(1,\alpha^5)$
$(0,\alpha^5)$	$(0,\alpha^4)$	$(1,\alpha^5)$	$(3,\alpha^4)$	$(2,\alpha^5)$	$(2,\alpha^4)$	$(3,\alpha^5)$	$(1,\alpha^4)$
$(0,\alpha^6)$	$(0,\alpha^6)$	$(1,\alpha^6)$	$(3,\alpha^6)$	$(2,\alpha^6)$	$(2,\alpha^6)$	$(3,\alpha^6)$	$(1,\alpha^6)$
$(0,\alpha^7)$	$(0,\alpha^7)$	$(1,\alpha^7)$	$(3,\alpha^7)$	$(2,\alpha^7)$	$(2,\alpha^7)$	$(3,\alpha^7)$	$(1,\alpha^7)$
$(0,\alpha^8)$	$(0,\alpha^8)$	$(1,\alpha^8)$	$(3,\alpha^8)$	$(2,\alpha^8)$	$(2,\alpha^8)$	$(3,\alpha^8)$	$(1,\alpha^8)$
$(0,\alpha^9)$	$(0,\alpha^{12})$	$(1,\alpha^9)$	$(3,\alpha^{12})$	$(2,\alpha^9)$	$(2,\alpha^{12})$	$(3,\alpha^9)$	$(1,\alpha^{12})$
$(0,\alpha^{10})$	$(0,\alpha^{10})$	$(1,\alpha^{10})$	$(3,\alpha^{10})$	$(2,\alpha^{10})$	$(2,\alpha^{10})$	$(3,\alpha^{10})$	$(1,\alpha^{10})$
$(0,\alpha^{11})$	$(0,\alpha^{11})$	$(1,\alpha^{11})$	$(3,\alpha^{11})$	$(2,\alpha^{11})$	$(2,\alpha^{11})$	$(3,\alpha^{11})$	$(1,\alpha^{11})$
$(0,\alpha^{12})$	$(0,\alpha^9)$	$(1,\alpha^{12})$	$(3,\alpha^9)$	$(2,\alpha^{12})$	$(2,\alpha^9)$	$(3,\alpha^{12})$	$(1,\alpha^9)$
$(0,\alpha^{13})$	$(0,\alpha^3)$	$(1,\alpha^{13})$	$(3,\alpha^3)$	$(2,\alpha^{13})$	$(2,\alpha^3)$	$(3,\alpha^{13})$	$(1,\alpha^3)$
$(0,\alpha^{14})$	$(0,\alpha^{14})$	$(1,\alpha^{14})$	$(3,\alpha^{14})$	$(2,\alpha^{14})$	$(2,\alpha^{14})$	$(3,\alpha^{14})$	$(1,\alpha^{14})$

2. $\psi(m, X_1 + X_2) = \psi(m, X_1) + \psi(m, X_2)$.

3. $\psi(m, X) = (n - m, X)$ *for exactly* $2^{\lceil (n+1)/2 \rceil}$ *values of* $X \in GF(2^n)$.

Proof. The first two properties of $\psi(\cdot)$ are obvious from its definition. For any $X = \sum_{i=0}^{n-1} x_i \beta_i$, $\psi(m, X) = (n - m, X)$ if and only if $x_i = x_{n-i}$, $1 \le i \le \lfloor (n-1)/2 \rfloor$. From this, the third property follows. ∎

Since ψ is an order 2 automorphism and it is independent of the ϕ automorphisms, $\psi \cdot \phi$ for each ϕ, is also an automorphism. Further, $\psi \cdot \phi = \phi' \cdot \psi$ where, the two automorphisms defined by pairs (t, K_0) and (t', K_0') respectively are related as $t' = -t$ and $K_0' = \mu(K_0)$. Therefore $\psi \cdot \phi$ cannot generate any new automorphisms that are not generated by $\phi' \cdot \psi$. Thus the total number of automorphisms of CCC$_n$ is $n2^{n+1}$.

5.5 Edge transformations by automorphisms in CCC$_n$

This section investigates the effect of the automorphisms described in Sec. 5.4 on the edges of CCC$_n$. It is easy to see that all the automorphisms of CCC$_n$ map g edges to g edges. In particular, given any automorphism of the first kind, $\phi(\cdot)$ with automorphism offset t and automorphism constants K_i, the g edge $(m, X) \xrightarrow{g} (m, X + \beta_0)$ is mapped to the edge $(m + t, X + K_m) \xrightarrow{g} (m + t, X + \beta_0 + K_m)$. Similarly, the automorphism of the second kind, $\psi(\cdot)$, maps the g edge $(m, X) \xrightarrow{g} (m, X + \beta_0)$ to another g edge $(n - m, \mu(X)) \xrightarrow{g} (n - m, \mu(X) + \beta_0)$.

To study the effect of the automorphisms on the f edges of CCC$_n$, we classify them as either of type 0 or type 1 based on the nodes from which they originate. We define an f edge from (m, X) to be of type 0 if $Tr(\sigma X) = 0$, and of type 1, if $Tr(\sigma X) = 1$. This classification of f edges is equivalent to partitioning elements of $GF(2^n)$ into two sets, E_0 and E_1 such that when the f edge from (m, X) is of type 0, $X \in E_0$ and when that edge is of type 1, $X \in E_1$. Clearly, $Tr(\sigma X) = 0$ if $X \in E_0$, and $Tr(\sigma X) = 1$ if $X \in E_1$.

We then have the following result about the sets E_0 and E_1.

Theorem 26 *E_0 is a subgroup of the additive group $GF(2^n)$ and $|E_0| = 2^{n-1}$. Further, E_1 is a coset of E_0.*

Proof. Note that if $X_1, X_2 \in E_0$, then $Tr(\sigma(X_1 + X_2)) = Tr(\sigma X_1) + Tr(\sigma X_2) = 0$ implying that $X_1 + X_2 \in E_0$ as well. Element $0 \in E_0$ is the identity of E_0 and inverse of any $X \in E_0$ is X itself. Thus, E_0 is a group. To find the number of elements in E_0, note that when X goes over all the elements of $GF(2^n)$, so does σX. Since exactly half the elements of $GF(2^n)$ have a 0 trace, $|E_0| = 2^{n-1}$. E_1 being the coset of E_0 is obvious. ∎

Theorem 26 shows that exactly half the f edges in any column of CCC$_n$ are of type 0 and the remaining half, of type 1. Theorem 27 explores the effect of an automorphism of the first kind on these edges.

Theorem 27 *Let $\phi(\cdot)$ be an automorphism of the first kind defined by the automorphism offset t and the automorphism constants K_i, $0 \leq i < n$. $\phi(\cdot)$ maps each f edge in column m to an edges of the same type if $Tr(\sigma K_m) = 0$ and to an edge of the opposite type if $Tr(\sigma K_m) = 1$.*

Proof. Consider an f edge from a node $N_1 = (m, X)$. The image of this edge under the automorphism ϕ is an f edge from node $N_2 = \phi(m, X) = (m + t, X + K_m)$. The classification of the f edge from node N_1 into type 0 or 1 depends on the value of $Tr(\sigma X)$ while that of the edge from N_2 depends on $Tr(\sigma(X + K_m)) = Tr(\sigma X) + Tr(\sigma K_m)$. The theorem immediately follows from this. ∎

Note that $Tr(\sigma K_m)$ has been previously denoted by c_m as in Theorem 23.

As Theorem 27 shows, the effect of an automorphism of the first kind on the type (0 or 1) of an f edge from (m, X) depends only on the value of m. In other words, the automorphism either preserves or alters the type of *all* f edges in a column. On the other hand, the effect of the automorphism of the second kind, $\psi(\cdot)$, on the type of an f edge from (m, X) depends only upon X. In other words, $\psi(\cdot)$ either preserves or alters the type of *all* f edges in a row of CCC$_n$. To see this, consider an f edge $(m, X) \xrightarrow{f} (m + 1, Y)$, where $Y = \alpha X + \beta_{n-1} Tr(\sigma X)$. Under the automorphism $\psi(\cdot)$, the image of this edge is $(n - m, \mu(X)) \xleftarrow{f} (n - m - 1, \mu(Y))$ where function μ is defined in (5.40). Clearly, the types of these two f edges depends upon $Tr(\sigma X)$ and $Tr(\sigma\mu(Y))$ respectively. Since these values are independent of m, one can see that the transformation of the edge type by $\psi(\cdot)$ does not depend on m.

To study the effect of $\psi(\cdot)$ on the edges of CCC$_n$, we define a set S of the elements of $GF(2^n)$ such that an $X \in S$ if and only if the type of f edge from (m, X) is preserved under $\psi(\cdot)$. It is not difficult to find elements of the set S.

An $X \in S$ can happen in two cases. when the f edge from (m, X) is of type 0, i.e, $(m, X) \xrightarrow{f} (m+1, \alpha X)$ and its image $(n-m, \mu(X)) \xleftarrow{f} (n-m-1, \mu(\alpha X))$ is also of type 0. This case requires that $Tr(\sigma X) = 0$ and $Tr(\sigma\mu(\alpha X)) = 0$. In the second case, the f edge from (m, X) is of type 1, i.e, $(m, X) \xrightarrow{f} (m + 1, \alpha X + \beta_{n-1})$ and its image $(n - m, \mu(X)) \xleftarrow{f} (n - m - 1, \mu(\alpha X + \beta_{n-1}))$ is also of type 1. This case requires that $Tr(\sigma X) = 1$ and $Tr(\sigma\mu(\alpha X+\beta_{n-1})) = 1$. However, the linearity of the trace and the $\mu(\cdot)$ functions and Lemma 2 allows one to write $Tr(\sigma\mu(\alpha X+\beta_{n-1})) = Tr(\sigma\mu(\alpha X) + Tr(\sigma\mu(\beta_{n-1})) = Tr(\sigma\mu(\alpha X)) + p_1$. Therefore, one gets $X \in S$ if and only if

$$Tr(\sigma X) = Tr(\sigma\mu(\alpha X)) = 0, \text{ or } Tr(\sigma X) = Tr(\sigma\mu(\alpha X)) + p_1 = 1. \tag{5.46}$$

Condition (5.46) can be written more compactly as

$$X \in S \text{ if and only if } Tr(\sigma(\overline{p}_1 X + \mu(\alpha X))) = 0, \tag{5.47}$$

where, $\overline{p}_1$ denotes the complement of p_1, i.e., when $p_1 = 0$, $\overline{p}_1 = 1$ and vice versa.

With the set S characterized by (5.47), one can now state Theorem 28 which specifies its properties.

Theorem 28 *1. S is a subgroup of the additive group of $GF(2^n)$.*

2. If the primitive polynomial $p(x)$ is such that

$$p(x) + x^{n+1}p(x^{-1}) = (1+x)(1+x^n), \tag{5.48}$$

then $|S| = 2^n$, otherwise, $|S| = 2^{n-1}$.

3. Exactly half the elements of S are in E_0 and the other half in E_1.

4. Amongst the elements of $GF(2^n)$ that are not in S, exactly half are in E_0 and the other half in E_1.

Proof. From (5.47), it is obvious that $0 \in S$ and S is closed under addition because of the linearity of Tr and μ functions. Additive inverse of each X is X itself. Thus S is a group.

To determine the number of elements in S and their distribution amongst sets E_0 and E_1, consider first the case when $p(x)$ satisfies (5.48). Let $p(x) = 1 + x^n + \sum_{i=1}^{n-1} p_i x^i$. Then a direct computation gives

$$p(x) + x^{n+1}p(x^{-1}) = (1+x)(1+x^n) + p_1(x+x^n) + \sum_{i=2}^{n-1} x^i(p_i + p_{n-i+1}), \tag{5.49}$$

Thus if the primitive polynomial $p(x)$ satisfies (5.48), then its coefficients must satisfy $p_1 = 0$ and for every i, $2 \leq i < n$, $p_i + p_{n-i+1} = 0$. Equation (5.47) for $X = \beta_i$, $1 \leq i < n$, then gives

$$\begin{aligned} Tr(\sigma(\beta_i + \mu(\alpha\beta_i))) &= Tr(\sigma(\beta_i + \mu(\beta_{i-1} + p_i\beta_{n-1}))) \\ &= Tr(\sigma(\beta_i + \beta_{n-i+1} + p_i\beta_1)) = p_i + p_{n-i+1} + p_ip_1 = 0. \end{aligned}$$

This shows that $\beta_i \in S$, $1 \leq i < n$. Similarly,

$$Tr(\sigma(\beta_0 + \mu(\alpha\beta_0))) = Tr(\sigma(\beta_0 + \beta_1) = p_1 = 0,$$

showing that $\beta_0 \in S$. Since S is a group and every $X \in GF(2^n)$ can be decomposed into the dual basis, every $X \in S$ and $|S| = 2^n$. Further, from Theorem 26, exactly half of these elements are in E_0 and the rest in E_1.

When $p(x)$ does not satisfy (5.48), either $p_1 = 1$ or for some $2 \leq i < n$, $p_i + p_{n-i+1} = 1$. If $p_1 = 1$, the condition (5.47) shows that $X \in S$ if and only if $Tr(\sigma\mu(\alpha X)) = 0$. But because $\mu(\cdot)$ is an one-to-one onto function from $GF(2^n)$ to $GF(2^n)$, as X varies over all the elements of $GF(2^n)$, so does $\sigma\mu(\alpha X)$. Since exactly half of the field elements have a trace equal to 0, $|S| = 2^{n-1}$. Now consider the pair of elements X and $(X + \beta_0 + \beta_1)$. One has

$$\begin{aligned} Tr(\sigma\mu(\alpha X)) + Tr(\sigma\mu(\alpha(X + \beta_0 + \beta_1))) &= Tr(\sigma\mu(\alpha(\beta_0 + \beta_1))) \\ &= Tr(\sigma\mu(\beta_0)) = Tr(\sigma\beta_0) = 0. \end{aligned}$$

Using (5.47) one can thus infer that either both X and $(X + \beta_0 + \beta_1)$ are in S or both are not in S, However, only one of these is in E_0 because

$$Tr(\sigma X) + Tr(\sigma(X + \beta_0 + \beta_1)) = Tr(\sigma(\beta_0 + \beta_1)) = p_1 = 1.$$

Therefore in this case, exactly half the elements in S are in E_0 and the rest in E_1. In addition, amongst the elements that are not in S, exactly half are in E_0 and the rest in E_1.

Finally, when $p(x)$ does not satisfy (5.48), but $p_1 = 0$, there must be an i, $2 \leq i < n$ such that $p_i + p_{n-i+1} = 1$. For a pair of elements X and $X + \beta_i$, one has

$$\begin{aligned} Tr(\sigma(X + \mu(\alpha X))) + Tr(\sigma(X + \beta_i + \mu(\alpha(X + \beta_i)))) = Tr(\sigma(\beta_i + \mu(\alpha\beta_i))) \\ = Tr(\sigma(\beta_i + \mu(\beta_{i-1} + p_i\beta_{n-1}))) = p_ip_{n-i+1} + p_ip_1 = 1. \end{aligned}$$

Thus from (5.47), exactly one of X and $X+\beta_i$ is in S, giving $|S|=2^{n-1}$. To find out how many of these elements in S are in E_0, we examine X and $(X+\beta_i+\beta_{n-i+1})$. We have,

$$\begin{aligned}
&Tr(\sigma(X+\mu(\alpha X)))+Tr(\sigma(X+\beta_i+\beta_{n-i+1}+\mu(\alpha(X+\beta_i+\beta_{n-i+1}))))\\
&\qquad = Tr(\sigma(\beta_i+\beta_{n-i+1}+\mu(\alpha(\beta_i+\beta_{n-i+1}))))\\
&\qquad = Tr(\sigma(\beta_i+\beta_{n-i+1}+\mu(\beta_{i-1}+\beta_{n-i}+(p_i+p_{n-i+1})\beta_{n-1})))\\
&\qquad = Tr(\sigma(p_i+p_{n-i+1})\beta_1) = (p_i+p_{n-i+1})p_1 = 0.
\end{aligned}$$

Thus from (5.47), both X and $X+\beta_i$ are in S or not in S. However, exactly one of them is in E_0 because

$$Tr(\sigma X)+Tr(\sigma(X+\beta_i+\beta_{n-i+1})) = Tr(\sigma(\beta_i+\beta_{n-i+1})) = p_i+p_{n-i+1} = 1.$$

Therefore exactly half the elements in S are in E_0 and the rest in E_1. In addition, exactly half of elements that are not in S are in E_0 and the rest in E_1. ∎

Theorem 28 shows that the edge transformation in CCC$_n$ because of the automorphism of the second kind, $\psi(\cdot)$ is highly regular. Either none of the f edges in the graph change their type (when $p(x)$ satisfies (5.48)), or edges in exactly half the rows of the graph change their type (when $p(x)$ does not satisfy (5.48)) because of ψ. Further, within the sets of edges that change or do not change, exactly half are of type 0 and the others, of type 1.

It is clear from the theorem that the characteristics of the primitive polynomial used to build the field and model the network are important. In particular, relation (5.48) is critical in determining the sizes of sets of edges that change types because of the automorphism $\psi(\cdot)$. One can show that to satisfy (5.48), the degree of the polynomial should be odd and its coefficient $p_{(n+1)/2}$ should be 1. Even though condition (5.48) seems rather artificial, there are many primitive polynomials which satisfy it. Primitive polynomials x^3+x^2+1 and $x^5+x^4+x^3+x^2+1$ can be cited as examples.

For applications to mappings on faulty networks, it is preferred to have smaller sets of edges with predictable transformations due to an automorphism. It is therefore preferred to use a primitive polynomial $p(x)$ which does not conform to (5.48). This would ensure that f edges in only half the rows in CCC$_n$ would change type, while other half would not. Fortunately, it is possible to show that there does exist at least one primitive polynomial which does not satisfy (5.48) for every degree n. Let a primitive polynomial $p(x)$ of degree n satisfy (5.48). Then its reciprocal polynomial, $\tilde{p}(x) = x^n p(x^{-1})$ is also a primitive polynomial. $\tilde{p}(x)$ cannot also satisfy (5.48), because otherwise, one would have

$$\tilde{p}(x) + x^{n+1}\tilde{p}(x^{-1}) = (1+x)(1+x^n). \tag{5.50}$$

Expressing $\tilde{p}(x)$ in terms of $p(x)$ in (5.50) and simplifying gives $p(x) = 1+x^n$, which is impossible as $p(x)$ is primitive.

5.6 CCC$_n$ as a subgraph of BF$_n$

It is known that the cube connected cycles is a subgraph of the butterfly graph [38]. We show in this section how this can be derived in our algebraic model.

Consider a function $f(\cdot) : GF(2^n) \to GF(2)$ defined as

$$f(X) = Tr(\sigma(\alpha+1)^{-1}X). \tag{5.51}$$

Following lemma states important properties of f.

Lemma 7 *The function $f(\cdot) : GF(2^n) \to GF(2)$ defined by (5.51) is linear and has the following properties.*

$$\begin{aligned} f(X+\beta_0) &= f(X)+1 \\ f(\alpha X + \beta_{n-1}Tr(\sigma X)) &= f(X) \quad \textit{and} \\ f(\alpha^{-1}X + \beta_0 Tr(\sigma\alpha^{-1}X)) &= f(X) \end{aligned}$$

Proof. Linearity of f is obvious from the linearity of the trace function. Let $X' = X + \beta_0$. Then,

$$\begin{aligned} f(X') &= Tr(\sigma(\alpha+1)^{-1}X) + Tr(\sigma(\alpha+1)^{-1}\beta_0) \\ &= Tr(\sigma(\alpha+1)^{-1}X) + 1. \end{aligned}$$

To prove the other two parts of the lemma, note that $\sigma(1+\alpha)^{-1} = 1 + \alpha + \alpha^2 + \cdots \alpha^{n-1}$. Let $X' = \alpha X + \beta_{n-1}Tr(\sigma X)$. Then,

$$\begin{aligned} f(X') &= Tr(\alpha X + \cdots + \alpha^n X) + Tr(\sigma X)Tr(\beta_{n-1} + \alpha\beta_{n-1} + \cdots + \alpha^{n-1}\beta_{n-1}) \\ &= f(X) + Tr(X) + Tr(\alpha^n X) + Tr(\sigma X) \\ &= f(X). \end{aligned}$$

Similarly, when $X' = \alpha^{-1}X + \beta_0 Tr(\sigma\alpha^{-1}X)$, one has

$$\begin{aligned} f(X') &= Tr(\alpha^{-1}X + \cdots + \alpha^{n-2}X) + Tr(\sigma\alpha^{-1}X)Tr(\beta_0 + \alpha\beta_0 + \cdots + \alpha^{n-1}\beta_0) \\ &= f(X) + Tr(\alpha^{-1}X) + Tr(\alpha^{n-1}X) + Tr(\sigma\alpha^{-1}X) \\ &= f(X). \end{aligned}$$

∎

We now state the central result of this section.

Theorem 29 *CCC_n is a subgraph of BF_n.*

Proof. We show that the function $\phi : CCC_n \to BF_n$ defined by

$$\phi(m, X) = (m + f(X),\ \alpha^{f(X)}X + f(X)\beta_{n-1}Tr(\sigma X)), \tag{5.52}$$

maps all the edges of CCC$_n$ to distinct edges of BF_n.

First note that ϕ is a one-to-one function. This is because if $\phi(m_1, X_1) = \phi(m_2, X_2)$, then

$$\alpha^{f(X_1)}X_1 + f(X_1)\beta_{n-1}Tr(\sigma X_1)) = \alpha^{f(X_2)}X_2 + f(X_2)\beta_{n-1}Tr(\sigma X_2)). \tag{5.53}$$

If $f(X_1) = f(X_2)$, then (5.53) gives $X_1 = X_2$. Without loss of generality, assume $f(X_1) = 1$ and $f(X_2) = 0$. Then (5.53) can be rewritten as

$$\alpha X_1 + \beta_{n-1} Tr(\sigma X_1) = X_2. \tag{5.54}$$

Thus,

$$\begin{aligned} f(X_2) &= Tr(\sigma(\alpha+1)^{-1}X_2) \\ &= Tr(\sigma\alpha(\alpha+1)^{-1}X_1) + Tr(\sigma X_1)Tr(\sigma(\alpha+1)^{-1}\beta_{n-1}). \end{aligned} \tag{5.55}$$

Using $\alpha(\alpha+1)^{-1} = 1 + (\alpha+1)^{-1}$ in the first term of (5.55) and $\sigma(\alpha+1)^{-1} = 1 + \alpha + \alpha^2 + \cdots + \alpha^{n-1}$ in the second, one gets

$$f(X_2) = Tr(\sigma X_1) + f(X_1) + Tr(\sigma X_1) = f(X_1).$$

But this contradictory to the assumption that $f(X_1) = 1$ and $f(X_2) = 0$. Therefore ϕ is a one-to-one function.

We now show that ϕ maps the edges of CCC$_n$ to edges of BF_n through the following three cases.

Case 1. Edge between (m, X) and $(m+1, X')$ in CCC$_n$, where $X' = \alpha X + \beta_{n-1}Tr(\sigma X))$.

When $f(X) = 0$, because of Lemma 7, $f(X') = 0$ as well. The two vertices map in BF_n to

$$\begin{aligned} \phi(m, X) &= (m, X), \quad \text{and,} \\ \phi(m+1, X') &= (m+1, \alpha X + \beta_{n-1}Tr(\sigma X)). \end{aligned}$$

Because $Tr(\sigma X)$ is either 0 or 1, there is clearly an edge between these two vertices in BF_n.

When $f(X') = f(X) = 1$, vertices (m, X) and $(m+1, X')$ in CCC$_n$ map to the following BF_n vertices.

$$\begin{aligned} \phi(m, X) &= (m+1, X')), \quad \text{and,} \\ \phi(m+1, X') &= (m+2, \alpha X' + \beta_{n-1}Tr(\sigma X')). \end{aligned}$$

Since $Tr(\sigma X')$ is either 0 or 1, there is an edge in BF_n between these two vertices.
Case 2. Edge between (m, X) and $(m-1, X')$ in CCC$_n$, where $X' = \alpha^{-1}X + \beta_0 Tr(\sigma\alpha^{-1}X))$.
As in the previous case, let $f(X) = f(X') = 0$, Then the two vertices map in BF_n to

$$\begin{aligned} \phi(m, X) &= (m, X), \quad \text{and,} \\ \phi(m-1, X') &= (m-1, X') = (m-1, \alpha^{-1}X + \beta_0 Tr(\sigma\alpha^{-1}X)). \end{aligned}$$

Since $Tr(\sigma\alpha^{-1}X)$ is either 0 or 1, there is a direct link between these two vertices of BF_n.

On the other hand, when $f(X) = f(X') = 1$, one has

$$\begin{aligned} \phi(m, X) &= (m+1, \alpha X + \beta_{n-1} Tr(\sigma X)) \quad \text{and,} \\ \phi(m-1, X') &= (m, \alpha X + \beta_{n-1} Tr(\sigma X)) = (m, X). \end{aligned}$$

Thus in this case also, there is an edge between these two vertices in BF_n.
Case 3. Edge between (m, X) and $(m, X + \beta_0)$ in CCC$_n$.
In this case, when $f(X) = 0$, $f(X + \beta_0) = 1$ from Lemma 7. Thus the two vertices map to

$$\begin{aligned} \phi(m, X) &= (m, X) \quad \text{and,} \\ \phi(m, X + \beta_0) &= (m+1, \alpha(X + \beta_0) + \beta_{n-1} Tr(\sigma(X + \beta_0))) \\ &= (m+1, \alpha X + \beta_{n-1}(Tr(\sigma X) + 1)), \end{aligned}$$

showing that the two vertices are connected in BF_n.

Similarly, when $f(X) = 1$, $f(X + \beta_0) = 0$. Thus the edge maps to

$$\begin{aligned} \phi(m, X) &= (m+1, \alpha X + \beta_{n-1} Tr(\sigma X)) \quad \text{and,} \\ \phi(m, X + \beta_0) &= (m, X + \beta_0). \end{aligned}$$

Clearly, in this case also, there is an edge between these two images in BF_n. ∎

5.7 Conclusion

This Chapter has provided a new algebraic model for the CCC using the direct product of a cyclic group and a finite field. This model allows the use of powerful algebraic techniques to study the structural properties of the network. We exploited these techniques to find optimal paths in the CCC and explore the relationships between the Cube Connected Cycles, the Shuffle Exchange and the deBruijn networks. We have shown that the total number of automorphisms of a the CCC network of degree n is $n2^{n+1}$ and have obtained explicit expressions for these automorphisms.

Chapter 6

Conclusion

Wrap-around Butterfly Network (BF_n), Cube Connected Cycles (CCC_n), Shuffle Exchange (SE_n) and deBruijn (DB_n) are some of the most popular interconnection networks on which many of the existing parallel machines are based. This dissertation focused on exploring structural properties of Wrapped Butterflies (BF_n) and Cube Connected Cycles (CCC_n) using algebraic models based on finite fields.

We have obtained simple expressions for all the $n2^{n+1}$ automorphisms of BF_n. Automorphisms have been used in the past to deal with single node faults and for tight VLSI layout of single chip implementations of interconnection networks. Even though the number of automorphisms could be computed from the packages such as Nauty, there is no prior work on obtaining these automorphisms themselves. We have explored useful properties and interactions of these automorphisms. We have investigated, for the first time, the effect of automorphisms on graph edges. This, in turn, can be used to map algorithm on Butterfly architectures with faulty edges. To achieve a fault free mapping, one only has to choose an appropriate automorphism to map the set of faulty edges to free edges. Since an automorphism set is complete and since each of these automorphisms are simple; it helps in this choice.

We have illustrated our technique by mapping a Hamilton cycle on a Butterfly under various edge fault scenarios. Previously, Hamilton cycle mappings were possible only for two node or one node and one edge or two edge faults. Our work shows that Butterfly (BF_n) supports a Hamiltonian cycle even when it has up to

2^n faulty edges of the same type in each column except one (Theorem 11) or even when it has faulty edges in all but two of its rows as long as the faults in a given set of rows are constrained to one type and those outside to one type as well (Theorems 13, 14). Further, the requirement of two fault-free rows can be lifted when n is odd (Theorems 15, 16, 17). As a corollary, we have shown that BF_n is Hamiltonian with up to $n-1$ random edge faults distributed one per column (Corollary 1). We also give much simpler construction than [43] for building Hamilton cycles in BF_n with up to two random edge faults.

This dissertation has provided new algebraic models for the Shuffle Exchange (SE_n) and Cube Connected Cycles (CCC_n) networks. Because of fixed node degree and small diameter, these networks are scalable.

Our models use finite fields and are much simpler to deal with than the usual binary models. We show the power of the algebraic model by proving that CCC_n is a subgraph of BF_n in a manner much simpler than the prior proof [38]. We also use the models to design path algorithms to travel between any two nodes of the Cube Connected Cycles and Shuffle Exchange networks (Algorithms 1, 2, 3). Paths in CCC_n have been studied before [56, 57], but our algorithms are much simpler and provide many alternatives which might be a useful characteristics if some edges in the graph are faulty.

This research has obtained all the $n2^{n+1}$ automorphisms of the Cube Connected Cycles of dimension n for the first time (Theorem 22). Even though the number of automorphisms of CCC_n could be computed earlier using the Naulty package, the automorphisms themselves were not determined earlier.

Similar to our work on BF_n, we have also investigated the effect of automorphisms on the edges of CCC_n (Theorems 26, 27, 28). Our work shows that four quite distinct interconnection networks, SE_n, DB_n, CCC_n and BF_n all share very similar algebraic models and are subject to similar mathematical exploits.

6.1 Future Research

This research proposed a new approach to mappings on Butterfly network with faulty edges. Even though we have limited our mappings to Hamiltonian cycle, we believe that the techniques developed in this research hold a lot of promise for other parallel algorithm mappings on BF_n under a larger set of edge faults and mapping under node faults. The algebraic model shows very promising results in analyzing constant node degree networks. In this dissertation we have proposed a new algebraic model for Cube Connected Cycles and Shuffle Exchange networks. In particular we show that these networks can be very effectively described by a finite field. One may identify the nodes in the interconnection graph with the elements of the abstract algebraic structure in such a fashion that the connectivity between nodes is expressed as a simple algebraic relation between the field elements. This allows one to exploit the rich properties of the finite fields to develop good mappings on these networks. We demonstrated the power of these techniques to find optimal paths in the CCC_n and SE_n. Our work could be extended to map cycles and trees on Cube Connected Cycles (CCC_n). Finally, we also believe that similar algebraic approach could be used to model other constant node interconnection networks as well.

Bibliography

[1] F. Leighton, *Introduction to parallel algorithms and architectures: Arrays, trees, hypercubes.* M. Kaufman Pub., 1992.

[2] S. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Computers*, vol. 38, no. 4, pp. 555–566, 1989.

[3] Z. Chen, Z.-J. Liu, and Z.-L. Qiu, "Bidirectional shuffle-exchange network and tag-based routing algorithm," *Communications Letters, IEEE*, vol. 7, pp. 121 – 123, march 2003.

[4] M. Samatham and D. Pradhan, "The de bruijn multiprocessor network: A versatile parallel processing and sorting network for vsli," *IEEE Trans. Computers*, vol. 38, no. 4, pp. 567–581, 1989.

[5] F. Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," *Comm. ACM*, vol. 24, no. 5, pp. 30–39, 1991.

[6] I. Friš, I. Havel, and P. Liebl, "The diameter of the cube-connected cycles," *Inf. Process. Lett.*, vol. 61, no. 3, pp. 157–160, 1997.

[7] W. Lin, T. Sheu, C. R. Das, T. Feng, and C. Wu, "Fast data selection and broadcast on the butterfly network," in *Proc Workshop Future Trends Distrib Comput Syst 1990s*, pp. 65–72, 1990.

[8] A. Ranade, "Optimal speedup for backtrack search on a butterfly network," *Mathematical systems theory*, vol. 27, pp. 85–102, Jan 1994.

[9] E. J. Schwabe, “Constant-slowdown simulations of normal hypercube algorithms on the butterfly network,” *Information processing letters*, vol. 45, p. 295, Apr 1993.

[10] M. D. Wagh and O. Guzide, “Mapping cycles and trees on wrap-around butterfly graphs,” *SIAM J. on Comput.*, vol. 35, pp. 741–765, 2006.

[11] F. Annexstein, M. Baumslag, and A. Rosenberg, “Group action graphs and parallel architectures,” *SIAM J. Compt.*, vol. 19, no. 3, pp. 544–569, 1990.

[12] S. Akers and B. Krishnamurthy, “The star graph: An attractive alternative to n-cube,” in *Int'l Conf. Parallel Processing (ICPP 87)* (I. St. Charles, ed.), pp. 393–400, 1987.

[13] B. Arden and K. Tang, “Representation and routing of cayley graphs,” *IEEE Trans. Comm.*, vol. 39, pp. 1,533–1,537, 1991.

[14] J. Mo, *Interconnection networks based on finite fields and finite groups.* PhD thesis, Lehigh University, 1996.

[15] M. D. Wagh and J. C. Mo, “An analytical setting and mappings on the product of generalized de bruijn graphs,” in *Proc. of the 10th Int. Conference on Parallel and Distributed Computing Systems*, (New Orleans), pp. 253–257, October 1–3 1997.

[16] C.-H. Tsai, “Cycles embedding in hypercubes with node failures,” *Inf. Process. Lett.*, vol. 102, no. 6, pp. 242–246, 2007.

[17] C.-H. Tsai and Y.-C. Lai, “Conditional edge-fault-tolerant edge-bipancyclicity of hypercubes,” *Inf. Sci.*, vol. 177, no. 24, pp. 5590–5597, 2007.

[18] J.-S. Fu, “Fault-tolerant cycle embedding in the hypercube,” *Parallel Comput.*, vol. 29, no. 6, pp. 821–832, 2003.

[19] T.-L. Kueng, T. Liang, L.-H. Hsu, and J. J. M.Tan, “Long paths in hypercubes with conditional node-faults,” *Inf. Sci.*, vol. 179, no. 5, pp. 667–681, 2009.

[20] S.-Y. Hsieh, "Fault-tolerant cycle embedding in the hypercube with more both faulty vertices and faulty edges," *Parallel Comput.*, vol. 32, no. 1, pp. 84–91, 2006.

[21] A. Avior, T. Calamoneri, S. Even, A. Litman, and A. L. Rosenberg, "A tight layout of the butterfly network," *Theory of Computing Systems*, vol. 31, pp. 475–488, Dec 1998.

[22] C.-H. Yeh, B. Parhami, E. A. Varvarigos, and H. Lee, "VLSI layout and packaging of butterfly networks," in *Proc. of ACM symp. on Parallel algorithms and architectures*, (Bar Harbor, Maine), pp. 196–205, 2000.

[23] M. D. Wagh and K. Bendjilali, "Butterfly automorphisms and edge faults," in *Proc. of Int. Symp. on Parallel and Distr. Comput.*, (Istanbul, Turkey), July 2010.

[24] M. D. Wagh and K. Bendjilali, "Conquering edge faults in butterfly with automorphisms," in *Proc. of Int. Conf. on Theoretical and Mathematical Foundations of Comp. Sc.*, (Orlando, FL), pp. 57–64, July 2010.

[25] E. R. Berlekamp, *Algebraic coding theory.* New York, NY: Mc-Graw Hill, 1968.

[26] P. A. Grillet, *Algebra.* A Wiley-Interscience publication, 1999.

[27] C. T. Gray, W. Liu, T. Hughes, and R. Cavin, "The design of a high-performance scalable architecture for image processing applications," in *Proc 90 Int. Conf. Appl. Specif. Array Processors*, pp. 722–733, 1991.

[28] J. H. Reif and S. Sen, "Randomized algorithms for binary search and load balancing on fixed connection networks with geometric applications," *SIAM Journal on Computing*, vol. 23, pp. 633–651, Jun 1994.

[29] A. L. Rosenberg, "Cycles in networks," Tech. Rep. Tech Rep 91-20, Univ. of Mass., 1993.

[30] A. Gupta and S. E. Hambrusch, "Embedding complete binary trees into butterfly networks," *IEEE Trans. Computers*, vol. 40, pp. 853–863, Jul 1991.

[31] S. Bhatt, F. R. K. Chung, J. Hong, F. Leighton, B. Obrenic, A. L. Rosenberg, and E. J. Schwabe, "Optimal emulation by butterfly-like networks," *JACM*, vol. 43, pp. 293–330, Mar 1996.

[32] N. G. de Bruijn, "A combinatiorial problem," *Proc. Koninklijke Nederlandse Acad. van Wetenschappen*, vol. A49, pp. 758–764, 1949.

[33] J. Bermond and P. Fraigniaud, "Broadcasting and gossiping in de bruijn networks.," *SIAM J. on Comput.*, vol. 23, pp. 212–225, Feb 1994.

[34] D. Z. Du, D. F. Hsu, F. K. Hwang, and X. M. Zhang, "The hamiltonian property of generalized de bruijn digraphs.," *J. of Combinatorial Theory. Series B.*, vol. 52, pp. 1–8, May 1991.

[35] M. C. Heydemann, J. Opatrny, and D. Sotteau, "Broadcasting and spanning trees in de bruijn and kautz networks," *Discrete applied math.*, vol. 37–38, pp. 297–317, July 1992.

[36] M. Heydemann, J. Opatrny, and D. Sotteau, "Embedding of hypercubes and grids into de bruijn graphs," *Journal of Parallel and Distributed Computing*, vol. 23, pp. 104–111, 1994.

[37] M. Baumslag, "An algebraic analysis of the connectivity of debruijn and shuffle exchange diagraphs," *Discrete Applied Math*, vol. 61, no. 3, pp. 213–227, 1995.

[38] R. Feldmann and W. Unger, "The cube-connected-cycle is a subgraph of the butterfly network," *Parallel Processing Letters*, vol. 2, no. 1, pp. 13–19, 1992.

[39] O. Guzide and M. D. Wagh, "Extended butterfly networks," in *Proc. of the The 18th Int. Conf. on Parallel and Distributed Computing Systems*, (Las Vegas, NV), pp. 109–113, 2005.

[40] O. Guzide and M. D. Wagh, "Enhanced butterfly: A Cayley graph with node degree 5," in *Proc. of the The 20th Int. Conf. on Parallel and Distributed Computing Systems*, (Las Vegas, NV), pp. 224–229, 2007.

[41] P. Vadapalli and P. Srimani, "Fault tolerant ring embedding in tetravalent cayley network graphs," *J Circuits Syst Comput*, vol. 10, pp. 527–536, 1996.

[42] C.-H. Tsai, T. Liang, L.-H. Hsu, and M.-Y. Lin, "Cycle embedding in faulty wrapped butterfly graphs," *Networks*, vol. 42, no. 2, pp. 85–96, 2003.

[43] S.-C. Hwang and G.-H. Chen, "Cycles in butterfly graphs," *Networks*, vol. 35, no. 2, pp. 161–171, 2000.

[44] H. S. Stone, "Parallel processing with the perfect shuffle," *IEEE Trans. Comput.*, vol. 20, pp. 153–161, Feb 1971.

[45] C. Chen and J.-K. Lou, "An efficient tag-based routing algorithm for the backward network of a bidirectional general shuffle-exchange network," *Communications Letters, IEEE*, vol. 10, pp. 296–298, apr 2006.

[46] F. Leighton, *Complexity issues in VLSI: optimal layouts for the shuffle-exchange graph and other networks.* Cambridge, MA, USA: MIT Press, 1983.

[47] R. Feldmann and P. Mysliwietz, "The shuffle exchange network has a hamiltonian path," in *Proc. of the 17th Int. Symp. on Mathematical Foundations of Computer Science*, (London, UK), pp. 246–254, Springer-Verlag, 1992.

[48] J. Bruck, R. Cypher, and C. T. Ho, "Fault-tolerant de bruijn and shuffle-exchange networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, no. 5, pp. 548–553, 1994.

[49] M. Baumslag, "Fault-tolerance properties of debruijn and shuffle-exchange networks," in *SPDP'93*, pp. 556–563, 1993.

[50] G. Chen and F. C. M. Lau, "Layout of the cube-connected cycles without long wires," *Comput. J.*, pp. 374–383, 2001.

[51] Y. Tanaka and Y. Shibata, "On the pagenumber of the cube-connected cycles," *Math. in CS*, vol. 3, pp. 109–117, 2010.

[52] S. Jianping, H. Zifeng, and S. Yuntao, "An optimal multicast algorithm for cube-connected cycles," *J. Comput. Sci. Technol.*, vol. 15, pp. 572–583, November 2000.

[53] G. E. Jan, S. W. Leu, C. H. Li, and X. Dong, "A perfect load balancing algorithm on cube-connected cycles," in *Proc. of the 5th WSEAS Int. Conf. on Comp. Intelligence, Man-Machine Systems and Cybernetics*, pp. 345–350, 2006.

[54] A. Germa, M. Heydemann, and D. Sotteau, "Cycles in the cube-connected cycles graph," *Discrete Appl. Math.*, vol. 83, pp. 135–155, March 1998.

[55] J. Yan, J. M. Xu, and C. Yang, "Forwarding index of cube-connected cycles," *Discrete Appl. Math.*, vol. 157, pp. 1–7, January 2009.

[56] L. Peng, W. Wei, and J. Xiang, "A new addressing scheme for cube-connected cycles network," in *Proc. of the 3rd IEEE Conf. on Industrial Electronics and Applications*, (Singapore), pp. 586 – 590, June 3-5 2008.

[57] D. Meliksetian and C. R. Chen, "Optimal routing algorithm and the diameter of the cube-connected cycles," *IEEE Transactions on parallel and Distributed Systems*, vol. 4, pp. 1172–1178, 1993.

Vita

Khadidja Bendjilali recieved her PhD. degree from the department of Computer Engineering at Lehigh University, USA in January 2012. She received her M.S. degree from the Computer Science department at Lehigh University, USA in January 2005. She received her Bachelor of Science degree in Computer Science from University of Petra. Her research interests include parallel processing, and interconnection networks.