Algebraic Models for the Cube Connected Cycles and Shuffle Exchange Graphs

Meghanad D. Wagh and Khadidja Bendjilali

Department of Electrical and Computer Engineering
Lehigh University, Bethlehem, PA 18015, USA
Email: mdw0, khb5@lehigh.edu

***Abstract*—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 a small diameter. This paper develops an 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. This paper exploits these techniques to find optimal paths in the CCC and to explore the relationships between the Cube Connected Cycles, the Shuffle Exchange and the deBruijn networks.**

***Keywords*-Cube Connected Cycles Graph; Interconnection networks; Routing; Shuffle Exchange.**

I. Introduction

Performance of message passing parallel architectures and multi-core chips depends, to a large extent, on the underlying interconnection network. Some of the desirable properties of such networks include the ability to connect a large number of nodes, a small node degree, a small diameter and the symmetry. In addition, the ever increasing demand for computational throughput has necessitated designs of scalable architectures, i.e., architectures which can be built using identical cores or processors irrespective of the size of the network. This implies that the underlying interconnection network graphs of such architectures should have a constant node degree. The prominent networks that have a constant node degree include the Ring, the Wrapped Around Mesh, the Tree, the Shuffle Exchange, the deBruijn, the Cube Connected Cycles and the Wrapped Around Butterfly networks. Of these, Ring and Mesh have a substantially large ($O(N)$) diameter, where N is the number of processors. Thus they do not scale well. The Tree, Shuffle Exchange and deBruijn networks do not possess symmetry, a property that is required for fault tolerance. As N increases, the probability of faulty processors and faulty edges also increases. Thus neither of these two 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. Of these, Wrapped Butterflies have been dealt with exhaustively in literature [1,2]. This paper focuses on the Cube Connected Cycles (CCC).

Previous work on CCC includes VLSI implementation and optimal layout [3, 4], load balancing, routing and one-to-one, one-to-many broadcast strategies [5, 6], mappings of cycles in fault-free and faulty topologies [7] and determination of the forwarding index of the network [8].

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 recently been proposed [9]. Unfortunately that new model does not provide sufficient insight into the graph connectivity. This paper provides a new algebraic model of the cube connected cycles. 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) and the Cube Connected Cycles.

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 [10]. CCC_n has $n2^n$ nodes, each labeled by a pair (m, V) where m is an integer $0 \le m < n$ and V is a binary vector of length n. A node (m, V) of CCC is connected to only three other nodes: $(m+1, V)$, $(m-1, V)$ and $(m, V \oplus 2^m)$, 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 $n > 3$ [11]. This low diameter and the low constant node degree implies that CCC may be very useful for parallel architectures.

Unfortunately 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 paper, we propose new models for the Shuffle Exchange (SE_n) and CCC_n networks using cyclic groups and finite fields. With these new models, 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 [12] and the Wrapped Butterflies [2] have allowed efficient mappings of cycles and trees on the Butterflies and provided insights into intricate structural properties such as the automorphisms [13, 14]. The new models proposed here

978-0-7695-4538-7/11 $26.00 © 2011 IEEE
DOI 10.1109/HPCC.2011.17

help solve similar problems in CCC$_n$. Besides proving this model, this paper demonstrates its use to obtain paths in CCC$_n$. Using the new model of SE$_n$, it also explores the relationship between the Shuffle Exchange and the deBruijn network.

This paper is organized as follows. The necessary mathematical background required for the rest of the paper is presented in Section II. The new algebraic models of the Shuffle Exchange and the Cube Connected Cycles are defined and proved in Sections III and IV. Section V provides optimal path algorithms for the Cube Connected Cycles. Section VI proves that the Shuffle Exchange network is a subgraph of the deBruijn network of the same size. Finally Section VII provides our final remarks.

II. Mathematical Preliminaries

This section provides the basic mathematical concepts required in this paper. In particular, we provide the definition and basic properties of the finite field of characteristic 2. For more detailed description, reader is referred to [15].

The finite field of 2^n elements, denoted by $GF(2^n)$, is an extension of $GF(2)$. Similar to $GF(2)$, $GF(2^n)$ uses modulo 2 addition, i.e, for any $X \in GF(2^n)$, $X + X = 0$. Elements of $GF(2^n)$ can be expressed as $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^n-2}\}$, where the element α is known as the *primitive element*. Note that $\alpha^{2^n-1} = 1$. The minimum degree polynomial (over $GF(2)$) of which α is a root, is called the *primitive polynomial*. Since α is a root of this polynomial of degree n , the elements of $GF(2^n)$ may also be expressed as polynomials (of degree at most $n-1$) in α over $GF(2)$. One can therefore view $GF(2^n)$ as a vector space over $GF(2)$ with basis $\langle \alpha^{n-1}, \alpha^{n-2}, \ldots, \alpha, 1\rangle$.

In this paper the primitive polynomial of $GF(2^n)$ is denoted by $p(x)$. $p(x)$ plays a central role in the design of $GF(2^n)$ and can be expressed as

$$p(x) = 1 + p_1x^1 + p_2x^2 + \cdots + x^n = \sum_{i=0}^{n} p_i x^i \quad (1)$$

Field $GF(2^4)$ is illustrated in Table I. This table shows expression of Each element of $GF(2^4)$ in basis $\langle \alpha^3, \alpha^2, \alpha, 1\rangle$. The expressions for the successive higher powers of α are obtained by multiplying the expressions for the lower powers by α and replacing any α^4, thus created, by $\alpha + 1$. This is because α being the root of the primitive polynomial, satisfies $\alpha^4 + \alpha + 1 = 0$, or $\alpha^4 = \alpha + 1$. (Recall that $GF(2^n)$ uses modulo 2 additions.) Table I can be used for additions between field elements. For example, $\alpha^{10} + \alpha^{11} = (\alpha^2 + \alpha + 1) + (\alpha^3 + \alpha^2 + \alpha) = \alpha^3 + 1 = \alpha^{14}$.

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 basis is unique and its component β_i is defined as that element of $GF(2^n)$ which

Table I: 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 $\langle \beta_3, \beta_2, \beta_1, \beta_0\rangle$ = $\langle 1, \alpha, \alpha^2, \alpha^{14}\rangle$.	

satisfies

$$Tr(\alpha^j \beta_i) = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where, the *Trace* function $Tr(\cdot) : GF(2^n) \rightarrow GF(2)$ is computed as [15]:

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

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, $0 \le i < n$ 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

$$\beta_i = \begin{cases} \alpha\beta_0 & \text{if } i = n-1 \\ \alpha\beta_{i+1} + p_{i+1}\beta_{n-1} & \text{otherwise,} \end{cases} \quad (3)$$

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.

Let $\sigma = (\alpha^n + 1)$. 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: Dual base elements β_is and σ are related as

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

and

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

The proofs of both lemmas are omitted for brevity.

III. An Algebraic model of the Shuffle Exchange Network

Even though non-symmetric, Shuffle Exchange is a popular interconnection network [16]. 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 C_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).

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 1: 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 $\psi(\cdot) : C_2^n \rightarrow GF(2^n)$ defined as

$$\psi(\langle v_{n-1}, v_{n-2}, \ldots, v_0 \rangle) = \sum_{i=0}^{n-1} v_i \beta_i \tag{4}$$

We now show that the correspondence expressed by (4) 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.,

$$X = \psi(V) = \sum_{i=0}^{n-1} v_i \beta_i \tag{5}$$

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

$$\psi(V_1) = \sum_{i=0}^{n-1} v_i \beta_{i+1} \tag{6}$$

where the index of β is considered modulo n. Using Lemma 1, one can write (6) as

$$\begin{aligned} \psi(V_1) &= \sum_{i=0}^{n-2} v_i \alpha^{-1} (\beta_i + p_{i+1} \beta_{n-1}) + u_{n-1} \alpha^{-1} \beta_{n-1} \\ &= \alpha^{-1} X + \beta_0 \sum_{i=0}^{n-2} p_{i+1} v_i. \end{aligned} \tag{7}$$

However, from Lemma 2,

$$Tr(\alpha^{-1} \sigma X) = \sum_{i=0}^{n-1} v_i Tr(\alpha^{-1} \sigma \beta_i) = \sum_{i=0}^{n-2} v_i p_{i+1}. \tag{8}$$

Comparing (7) and (8) we get

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

Similarly, the relabeling of node V_2 gives

$$\psi(V_2) = \sum_{i=0}^{n-1} v_i \beta_{i-1} \tag{9}$$

where the index of β_{i-1} is considered modulo n. Using Lemma 1, (9) may be rewritten as

$$\begin{aligned} \psi(V_2) &= \sum_{i=1}^{n-1} v_i (\alpha \beta_i + p_i \beta_{n-1}) + v_0 \alpha \beta_0 \\ &= \alpha X + \beta_{n-1} \sum_{i=1}^{n-1} p_i v_i. \end{aligned} \tag{10}$$

But from Lemma 2,

$$Tr(\sigma X) = \sum_{i=0}^{n-1} v_i Tr(\sigma \beta_i) = \sum_{i=1}^{n-1} v_i p_i. \tag{11}$$

From (10) and (11) one gets

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

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

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

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) is illustrated in Table II.

Table II. Equivalence between the binary and the algebraic labels of SE_4.

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

The connectivity of SE_4 using the new algebraic model is shown in Fig. 1. As indicated in this figure, we will refer to the three edges as f, f^{-1} and g in this path. Not only

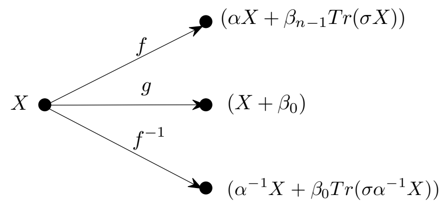

Fig. 1. The connectivity of the Shuffle Exchange graph SE_n.

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 $X_1 \stackrel{f}{\longrightarrow} Y_1$ and $X_2 \stackrel{f}{\longrightarrow} Y_2$, then,
$X_1 + X_2 \stackrel{f}{\longrightarrow} Y_1 + Y_2$.
Similarly, If $X_1 \stackrel{f^{-1}}{\longrightarrow} Y_1$ and $X_2 \stackrel{f^{-1}}{\longrightarrow} Y_2$, then,
$X_1 + X_2 \stackrel{f^{-1}}{\longrightarrow} Y_1 + Y_2$.
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.

IV. 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 $m \in C_n$, a group of integers $\{0, 1, \cdots, n-1\}$ and $V \in C_2^n$, a set of n bit binary strings. A node (m, V) is connected to nodes $(m+1, V)$, $(m-1, V)$ and $(m, V \oplus 2^m)$. 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 expressed by a simple algebraic relationship between the labels.

Theorem 2: 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 $\psi : C_n \times C_2^n \to C_n \times GF(2^n)$ defined by

$$\psi((m, V)) = (m, X), \quad \text{where} X = \sum_{i=0}^{n-1} v_{m+i}\beta_i. \quad (12)$$

One can now use this ψ function to prove the model in a fashion similar to the proof of Theorem 1. ∎

Fig. 2 shows the connectivity of the algebraic model for CCC given by Theorem 2. Note that unlike its binary

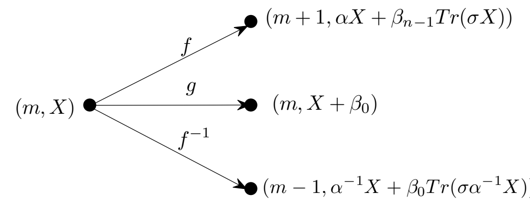

Fig. 2. The connectivity of the Cube Connected Cycles graph CCC_n.

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. 2). 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
$(m, X_1) \stackrel{f}{\longrightarrow} (m+1, Y_1)$ and $(m, X_2) \stackrel{f}{\longrightarrow} (m+1, Y_2)$, then, $(m, X_1 + X_2) \stackrel{f}{\longrightarrow} (m+1, Y_1 + Y_2)$.
Similarly, If
$(m, X_1) \stackrel{f^{-1}}{\longrightarrow} (m-1, Y_1)$ and $(m, X_2) \stackrel{f^{-1}}{\longrightarrow} (m-1, Y_2)$, then, $(m, X_1 + X_2) \stackrel{f^{-1}}{\longrightarrow} (m-1, Y_1 + Y_2)$.
We will refer to this observation as the *linearity in source property* of the f and f^{-1} edges of the CCC_n graph.

V. Path Algorithms for Cube Connected Cycles

We start by stating a result that will help us minimize the path length.

Lemma 3: A path with n consecutive f edges forms a cycle in CCC_n.

Proof. By direct computation. ∎

We are now ready to use the algebraic machinery to chart a path from a node $(0, X)$ to 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 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

$$D' = \alpha D + \beta_{n-1}(c + Tr(\sigma D)), \tag{13}$$

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 (13) 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 (13) one gets,

$$D_1 = \alpha D_0 + \beta_{n-1}(c_0 + Tr(\sigma D_0)). \tag{14}$$

Using (13) repeatedly and simplifying the result each time using Lemmas 2 and 1 gives the destination $(m+k, D_k)$ after k path segments as

$$(m+k, \alpha^k D_0 + \sum_{j=0}^{k-1} \beta_{n-k+j}(c_j + Tr(\alpha^j \sigma D_0)). \tag{15}$$

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

$$\begin{aligned} k &= a \mod n \quad \text{and} \\ 0 &= \alpha^k X + \sum_{j=0}^{k-1} \beta_{n-k+j}(c_j + Tr(\alpha^j \sigma X)). \end{aligned} \tag{16}$$

Values of k and c_j, $0 \le j < k$ satisfying (16) give the required path.

To solve (16), first note that for any $k \ge n$, the summation in (16) goes over all the β_j, $0 \le j < n$. Since $\alpha^k X$ has a unique decomposition in the dual basis, one can always find c_is to satisfy (16) in this case. For smallest such k, $k = n+a$, (16) becomes

$$\sum_{j=0}^{n-1} Tr(\alpha^{n+a+j} X)\beta_j = \sum_{j=0}^{n+a-1} \beta_{-a+j}(c_j + Tr(\alpha^j \sigma X)), \tag{17}$$

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, $0 \le j < n-a$, on both sides of (17) 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

$$\begin{aligned} c_{j+a} &= Tr(\alpha^{j+a} X),\ 0 \le j < n-a \quad \text{or,} \\ c_j &= Tr(\alpha^j X),\ a \le j < n. \end{aligned} \tag{18}$$

Similarly, comparing the coefficients of β_j, $n-a \le j < n$, in (17) gives

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

Simplifying this as before gives

$$\begin{aligned} c_{j+a} + c_{j+a-n} &= Tr(\alpha^{j+a-n} X),\ n-a \le j < n \quad \text{or} \\ c_j + c_{n+j} &= Tr(\alpha^j X),\ 0 \le j < a. \end{aligned} \tag{19}$$

For a smaller $k = a$, the summation in (16) does not span all the β_j, $0 \le j < n$ of the dual basis. Therefore all X values may not yield a solution to (16). In particular, with $k = a$, (16) becomes

$$\sum_{j=0}^{n-1} Tr(\alpha^{a+j} X)\beta_j = \sum_{j=0}^{a-1} \beta_{-a+j}(c_j + Tr(\alpha^j \sigma X)). \tag{20}$$

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 (20). By comparing the coefficients of various β_js on both sides of this equation as before, one gets

$$\begin{aligned} & c_j = Tr(\alpha^j X),\ 0 \le j < a, \\ & \text{last } g \text{ edge to be used if } \ Tr(\alpha^a X) = 1 \text{ and} \\ & Tr(\alpha^j X) = 0,\ a < j < n. \end{aligned} \tag{21}$$

The discussion above, including the computation of c_is from (18), (19) and (21), provide the following path algorithm.

Algorithm 1: **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 $c_i = Tr(\alpha^i X)$, $0 \le i < a$.
Else Set PathSegments to $a+n$,
  choose binary values c_i, $0 \le i < a+n$, as
  $c_i + c_{i+n} = Tr(\alpha^i X)$, $0 \le i < a$ and
  $c_i = Tr(\alpha^i X)$, $a \le i < n$.
  (Note: c_i, c_{i+n}, $0 \le i < a$ 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 3, 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

$$(m-k, \alpha^{-k}D + \sum_{j=1}^{k} \beta_{k-j} Tr(\alpha^{-j}\sigma D) + \sum_{j=0}^{k-1} c_j \beta_{k-j}), \quad (22)$$

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

With the starting node $(0, X)$, (22) will give the destination node $(a, 0)$ after k path segments if

$$\begin{aligned} a &= -k \mod n \quad \text{and} \\ 0 &= \alpha^k X + \sum_{j=1}^{k} \beta_{k-j} Tr(\alpha^{-j}\sigma X) + \sum_{j=0}^{k-1} c_j \beta_{k-j}) \end{aligned} \quad (23)$$

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

When $k = n + (n - a) \mod n$, (23) 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 (23), 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

$$c_{n-a} = Tr(\alpha^a X). \quad (24)$$

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

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

This equation can be simplified to yield

$$c_i + c + i + n = Tr(\alpha^{n-i}X), \; 0 < i < n - a. \quad (25)$$

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

$$c_0 + c + n = Tr(X). \quad (26)$$

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

$$c_i = Tr(\alpha^{n-i}X), \; n - a < i < n. \quad (27)$$

When $k = (n - a) \mod n$, (23) may not have a solution for all x values. In this case, $a > 0$ as is obvious from (23). For this $k = n - a$, (23) becomes

$$\begin{aligned} \sum_{j=0}^{n-1} Tr(\alpha^{n-a+j}X)\beta_j \;&=\; \sum_{j=1}^{n-a} \beta_{n-a-j} Tr(\alpha^{-j}\sigma X) \\ &\quad + \sum_{j=0}^{n-a-1} c_j \beta_{k-j}. \end{aligned} \quad (28)$$

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 (28) gets added with an additional β_0. Solution of this equation gives

$$\begin{aligned} &c_0 = Tr(X), \\ &c_j = Tr(\alpha^{n-j}X), \; 1 \le j < n - a, \\ &\text{last } g \text{ edge to be used if } \; Tr(\alpha^a X) = 1 \text{ and} \\ &Tr(\alpha^j X) = 0, \; n - a < j < n. \end{aligned} \quad (29)$$

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

Algorithm 2: **(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
$c_i = Tr(\alpha^{n-i}X)$, $1 \leq i < n - a$.
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 $c_0 = Tr(X)$,
$c_i = Tr(\alpha^{n-i}X)$, $1 \leq i < n - a$.
Else Set PathSegments to $n + (n - a) \mod n$,
choose binary values c_i, $0 \leq i < a + n$, as
$c_0 + c_n = Tr(X)$, $c_i + c_{i+n} = Tr(\alpha^{n-i}X)$,
$0 < i < n - a$ and $c_i = Tr(\alpha^{n-i}X)$, $n - a \leq i < n$.
(Note: c_i, c_{i+n}, $0 \leq i < n - a$ 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 3. 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 $(0, \alpha^5)$ 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 corresponding path is
$(0, \alpha^5) \xrightarrow{f^{-1}} (3, \alpha^4) \xrightarrow{f^{-1}} (2, \alpha^3) \xrightarrow{g} (2, 1) \xrightarrow{f^{-1}}$
$(1, \alpha^{14}) \xrightarrow{g} (1, 0) \xrightarrow{f} (2, 0)$.

Algorithms 1 and 2 are also useful to compute the diameter of the cube connected cycles graph.

Theorem 3: The diameter of CCC$_n$ is 6 if $n = 3$ and $2n + \lfloor n/2 \rfloor - 2$ if $n > 3$.

Proof. 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 1 and 2.

The theorem for $n = 3$ can be proved from algorithm 1 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 3 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 1. 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 3, 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 2. 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 3 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$_n$. ∎

VI. RELATIONS BETWEEN INTERCONNECTION NETWORKS

We now show that using the new model, the Shuffle Exchange network SE$_n$ can be easily embedded in the deBruijn network DB$_n$.

We use an algebraic model for DB$_n$ [12] which labels the nodes of DB$_n$ with the elements of $GF(2^n)$. A node with label $X \in GF(2^n)$ is connected to nodes αX, $\alpha X + \beta_{n-1}$, $\alpha^{-1}X$ and $\alpha^{-1}X + \beta_0$. Theorem 4 states the main result of this section.

Theorem 4: SE$_n$ is a subgraph of DB$_n$.
Proof. Use the mapping $\phi(\cdot) : \text{SE}_n \to \text{DB}_n$ defined as

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

where, the binary function $b(X) = Tr(\sigma(1 + \alpha)^{-1}X)$. To show that $\phi(\cdot)$ is one-to-one, assume to the contrary that

$\phi(X) = \phi(Y)$ for some $X, Y \in GF(2^n)$. If $b(X) = b(Y) = 0$, then $X = Y$. If $b(X) = b(Y) = 1$, then $\phi(X) = \phi(Y)$ leads to $\alpha(X+Y)+\beta_{n-1}Tr(\sigma(X+Y)) = 0$. However, we know from the SE$_n$ connectivity (see Fig. 1) 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 $b(X)$ and $b(Y)$ are different, say, $b(X) = 1$ and $b(Y) = 0$, then $\phi(X) = \phi(Y)$ gives $\alpha X + \beta_{n-1}Tr(\sigma X) = Y$. This gives $b(Y) = b(X)$, which is a contradiction. Thus any time $\phi(X) = \phi(Y)$, $X = Y$, i.e., function $\phi(\cdot)$ is one-to-one.

We now prove that the edges of SE$_n$ are preserved by $\phi(\cdot)$. 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, $0 \leq i < n$. Consider the edge $\beta_i \xrightarrow{f} \beta_{i-1}$. One can show that $b(\beta_i) = b(\beta_{i-1}) = 1$. Thus $\phi(\beta_i) = \alpha\beta_i + \beta_{n-1}Tr(\sigma\beta_i)$. From Lemmas 1 and 2, one can then see that $\phi(\beta_i) = \beta_{i-1}$. Similarly $\phi(\beta_{i-1}) = \beta_{i-2}$. Thus the edge $\beta_i \xrightarrow{f} \beta_{i-1}$ of SE$_n$ translates to the edge $\beta_{i-1} \xrightarrow{f} \beta_{i-2}$ of DB$_n$.

Finally, to see the preservation of the g edge of SE$_n$, consider edge $X \xrightarrow{g} X+\beta_0$. One has, $b(X+\beta_0) = b(X)+1$. Thus if $b(X) = 0$, then $b(X + \beta_0) = 1$. Thus the edge is transformed by $\phi(\cdot)$ 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 $b(X) = 1$, then $b(X + \beta_0) = 0$. 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$_n$. ∎

VII. Conclusion

This paper has provided new models for the Shuffle Exchange (SE$_n$) and Cube Connected Cycles (CCC$_n$) networks that are used in parallel architectures. 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. For example, in case of CCC$_n$, the nodes in both cases are labeled by pairs. But in the binary model, the coordinates of an edge destination node depend on both the coordinates of the source node. In our new model, each coordinate of the destination node depends only on the corresponding coordinate of the source node. This greatly simplifies development of paths and mappings on these networks. In addition, this new model allows one to exploit the powerful results in finite fields that have been developed over a century.

A similar algebraic model for Wrapped Butterflies has previously allowed highly efficient mappings of cycles and trees [2] as well as investigations into automorphisms and fault avoidance [13, 14]. The new models proposed here can be used to obtain similar results in case of the Cube Connected Cycles networks.

References

[1] 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.

[2] 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.

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

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

[5] 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.

[6] 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.

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

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

[9] 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.

[10] F. P. Preparata and J. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation," *Communications of the ACM*, vol. 24, pp. 300–309, May 1981.

[11] 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.

[12] 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. Conf. on Parallel and Dist. Comput. Sys.*, (New Orleans), pp. 253–257, October 1–3 1997.

[13] 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.

[14] 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.

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

[16] 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, April 2006.