British Journal of Mathematics & Computer Science
2(1): 1-20, 2012

SCIENCEDOMAIN *international*
www.sciencedomain.org

Embedding *n*-Dimensional Crossed Hypercube into Pancake Graphs

M. F. Zerarka[1], S. Femmam[2*] and R. Aschheim[3]

[1]Polytechnic Engineers School Labs E.P.F. 3 bis street Lakanal, 92330 Sceaux, France.
[2]Polytechnic Engineers School Labs E.P.F. 3 bis street Lakanal, 92330 Sceaux, France & University of UHA France.
[3]Polytopics Research Institute, 8 villa Haussmann, 92130 Issy, France.

Research Article

Received 20th September 2011
Accepted 28th December 2011
Online Ready 7th February 2012

Abstract

Among Cayley graphs on the symmetric group, the pancake graph is one as a viable interconnection scheme for parallel computers, which has been examined by a number of researchers. The pancake was proposed as alternatives to the hypercube for interconnecting processors in parallel computers. Some good and attractive properties of this interconnection network include: vertex symmetry, small degree, a sub-logarithmic diameter, extendability, and high connectivity (robustness), easy routing, and regularity of topology, fault tolerance, extensibility and embeddability of other topologies. In this paper, we present the many-to-one dilation 5 embedding of *n*-dimensional crossed hypercube into *n*-dimensional pancake patients. These predictors, however, need further work to validate reliability.

Keywords: Cayley graph; embedding, crossed hypercube networks; pancake networks; dilation.

1 INTRODUCTION

The study of graph embedding arises naturally in a number of computational problems: portability of algorithms across various parallel architectures, layout of circuits in VLSI. Akers and Krishnamurthy (1989) proposed the pancake as alternative to the hypercube and their variations for interconnecting processors in parallel computers.

This network has desirable proprieties: Small diameter and fixed degree, (n-1) regular, high connectivity, vertex symmetry, Hamiltonian, fault tolerance, extensibility, pancyclicity and embeddability of other topologies (Akers and Krishnamurthy, 1989; Kanevsky and Feng, 1995; Hung et al., 2003; Heydari and Sudborough, 1997; Rowley and Bose, 1998; Hsieh et al., 1998), Hsieh and Chen, 2004), Hsieh and Lee, 2009, 2010, Hsieh and Chang, 2006; Hwang and Chen, 2000). The embedding capabilities are important in evaluating an interconnection network. The embedding of the guest graph G into host graph H is a mapping from each vertex of G to one

* *Corresponding author: Email: femmam@ieee.org;*

vertex of H and mapping each edge of G to one path of H. Graph embedding is useful because an algorithm designed for H can be applied to G directly (Bouabdallah et al., 1998; Sengupta, 2003; Menn and Somani, 1992, Fan, 2002; Qiu, 1992; Fang and Hsu, 2000; Hsieh et al., 1999; Rowley and Bose, 1993, Chang et al., 2000; Lin et al., 2008, 2010; Femmam et al., 2012). To compare with crossed hypercube, the pancake graph offers good and simple simulations of other interconnection networks (Miller et al., 1994; Senoussi and Lavault, 1997; Hung et al., 2002).

The paper is organized as follows: In the preliminaries we introduce some definitions and notations, including the definition and proprieties of crossed hypercube and pancake network. In section 3 we present an algorithm of many-to-one embedding crossed hypercube into pancake. In the section 4 we show that a dilation of many-to-one embedding of n-dimensional crossed hypercube embedding into pancake of dimension n is equal to 5. Finally, we give our conclusion in section 5.

2 PRELIMINARIES THEORY ANALYSIS

2.1 Definition 1 Construction

The n-dimensional hypercube Q_n and the crossed hypercube $CQ_n = (V,U)$ have a same set of vertices V. We represent the address of each vertex in Q_n (CQ_n) as a binary string of length n. In such away, we don't distinguish between vertices and their binary address. In Q_n two vertices are adjacent if and only if, their binary labels differ only in one bit position. For the CQ_n n-dimensional crossed hypercube, adjacency requirements are little more involved.

Definition: Two binary strings $x=x_1x_0$ and $y=y_1y_0$ of length two are said pair-related if and only If, $(x, y) \in \{(00,00), (10,10), (01,11), (11,01)\}$.

The n-dimensional crossed hypercube CQ_n is recursively defined as follows: CQ_1 is the complete graph based on two vertices labeled 0 and 1 (Efe, K. 1991, Efe, K. 1992, Aschheim et al., 2012). CQ_n consists of two subcubes $0CQ_{n-1}$ and $1CQ_{n-1}$ the most significant bit of the labels of the vertices in $0CQ_{n-1}$ ($1CQ_{n-1}$) is 0(1).

U is the set of vertices $u=u_{n-1}u_{n-2}\ldots.u_1u_0 \in 0CQ_{n-1}$ *with* $u_{n-1}=0$ and $v=v_{n-1}v_{n-2}\ldots.v_1v_0 \in 1CQ_{n-1}$ with $v_{n-1}=1$ are joined by an edge in CQ_n if and only if*:*

$$\begin{array}{ll} u_{n-2}= v_{n-2} & \text{if n is even} \\ (u_{2i+1}u_{2i},\ u_{2i+1}u_{2i}) & \text{are pair related} \end{array} \qquad (1)$$

Examples of crossed hypercube for n=1, 2, 3 are given in Figure 1.

The n-dimensional crossed hypercube CQ_n as an alternative to the hypercube has the same number of vertices V and degree as the n-dimensional hypercube. The crossed hypercube is one of the variations of hypercube which is derived with some twisted edges. Due to these twisted edges, the diameter of CQ_n is only half of the hypercube one. Nice proprieties include relatively small degree, embedding capabilities, scalability, robustness and the fault-tolerant of hamiltonicity of CQ_n (Huang et al., 2000; Chang et al., 2000; Kulasinghe and Bettayeb, 1995b; Yang et al., 2003; Hsieh et al., 1999). The multiply-twisted hypercube graph is not vertex-transitive for $n \geq 5$ (Kulasinghe and Bettayeb, 1995a).

2.2 Definition 2 Construction

Cayley graphs were originally proposed as a generic theoretic model for analyzing symmetric interconnection network. The most notable feature of Cayley graph is their universality. The Cayley graph represents a class of high performance interconnection network with a small degree and diameter, good connectivity and simple routing algorithms. The pancake is one of the Cayley graph.

Let I=(1,2,3,…,n), p= (p_1,p_2,…..p_n), p_i Є I and $p_i \neq p_j$ for $i \neq j$, where p is the permutation of I.
A pancake graph G_n=(P_n,E_n) of n dimensions is defined as follows:

$$P_n=\{(p_1, p_2...p_n) | p_i \in I, p_i \neq p_j \text{ for } i \neq j\} \quad (2)$$

and

$$E_n=\{((p_1, p_2,........p_{j-1}, p_j, p_{j+1},........., p_n),(p_j, p_{j-1},............, p_2, p_1, p_{j+1},..........., p_n)) | (p_1,p_2,........p_n) \in P_n \text{ for } 2 \leq j \leq n. \quad (3)$$

In other words, the set of P_n of all permutations constitutes the nodes of the vertices of $G_{n.}$

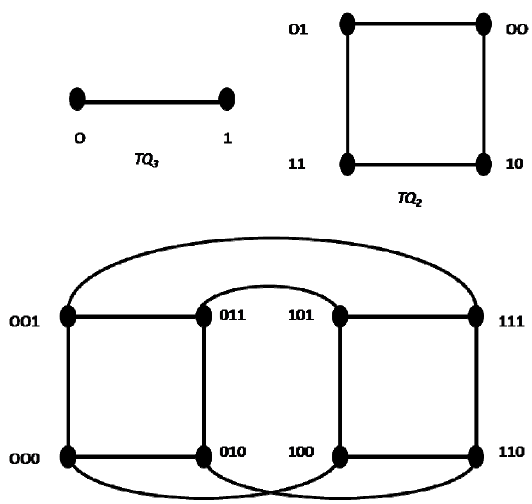

Fig. 1: Crossed hypercube for n=1, 2, 3

Two nodes u and v are joined by an (undirected) edge if and only if, the permutation corresponding to the node v can be obtained from u by flipping the object in positions 1 through j. Since for each permutation we can flip any number of objects between first and jth positions, $2\leq j \leq n$, G_n is a (n-1) regular graph, $|P_n|= n!$, $| E_n |= (n-1)n!/2$. Examples of pancake for n=2, 3, 4 are given in Figure 2 (a) and Figure 2 (b).

The pancake graphs proposed by Akers and Krishnamurthy (Akers and Krishnamurthy, 1989) are an important family of interconnection networks. Some interesting properties of the pancake are shown in (Bouabdallah et al., 1998). One of the main proprieties are their symmetric, it is built using Cayley groups with simple routing algorithms. Pancake graphs have many other attractive features, among their hierarchical, maximally fault-tolerant, Hamiltonian (Akers and Krishnamurthy, 1989; Kanevsky and Feng, 1995; Qiu, 1992; Qiu et al., 1991; Hwang and Chen, 2000; Lin et al., 2008), have a small diameter (Morales and Sudborough, 1996).

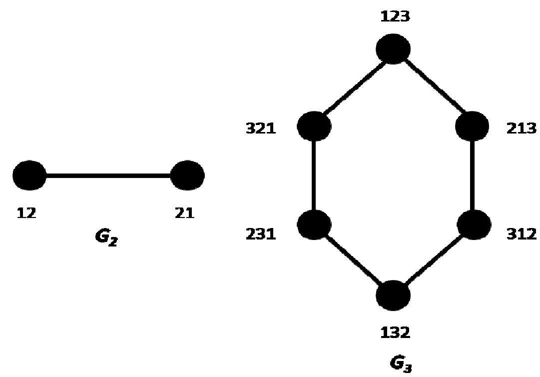

Fig. 2 (a): Example of *n*-pancake graphs *n*=2, 3

The graph G_n is made of n copies of G_{n-1} namely G_n $[n, k]$ for $1 \leq k \leq n$. Considering each G_n $[n, k]$ as a super node. It follows that $G_n[n,s]$, $G_n[n,t]$ are connected by a collection of edges of the form $((t,p_2,p_3,\ldots,p_{n-1},s), (s,p_{n-1},\ldots,p_2,t))$ thus, there are $(n-2)!$ edges connecting $G_n[n,s]$ and $G_n[n,t]$ (Kanevsky and Feng, 1995). G_n is a complete graph on the super nodes connected by the super edges as shown in Figure 3.

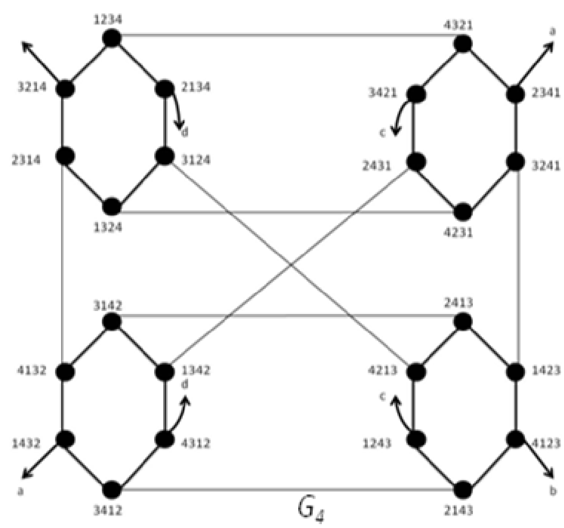

Fig. 2 (b): Example of *n*-pancake graphs *n*= 4

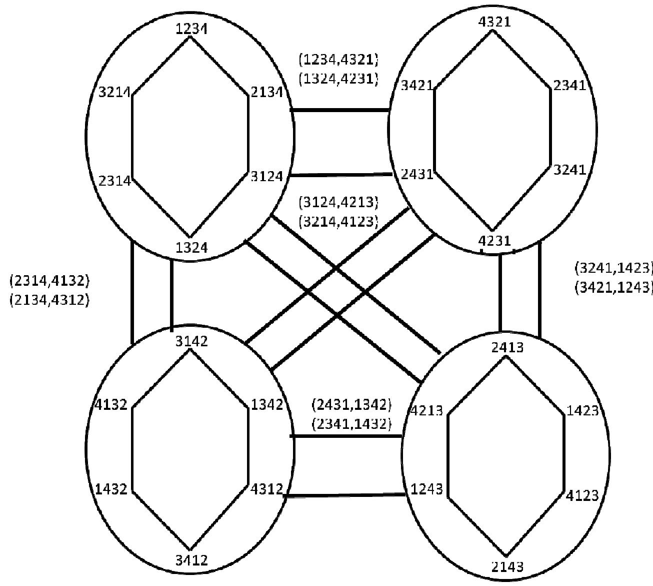

Fig. 3: Recursive structure of G_4

2.3 Definition 3 construction

Let *G* and *H* two simple undirected graphs. An embedding of the graph *G* into the graph *H* is an injective mapping *f* from the vertices of *G* to the vertices of *H.* The dilation of the embedding is the maximum distance between *f*(*x*) and *f*(*y*) taken over all edges (*x, y*) of *G.*

2.4 Notations

Crossed hypercube of *n* dimensions denoted by $CQ_n=(V,U)$, with *V* set of vertices and *U* set of edges.

Pancake of *n* dimensions denoted by $G_n=(P_n,E_n)$, with P_n set of vertices and E_n set of edges.

$A \in V$ such that $A=a_1a_2a_3\ldots\ldots a_{n-3}a_{n-2}a_{n-1}a_n=Pref.a_{n-2}a_{n-1}a_n$, where $Pref=a_1a_2a_3\ldots\ldots a_{n-3}$.
$U1 \subset U$ as $u \in E1$, such that $u=(A,B)$ with *A* and $B \in V$.
$A \in V$, $A=a_1a_2a_3\ldots\ldots a_{n-4}.a_{n-3}a_{n-2}a_{n-1}a_n=Pref.a_{n-4}.a_{n-3}a_{n-2}a_{n-1}a_n$, where $Pref=a_1a_2a_3\ldots\ldots a_{n-5}$.
$U2 \subset U$ as $u \in E2$, $u = (A,B)$ such that *A* and $B \in V$. $P_n' \subset P_n$ is a subset of P_n as $X \in P_n'$, where $X=x_1x_2x_3s_1s_2\ldots\ldots s_{n-l}$, such that $Suffix=s_1s_2\ldots\ldots s_{n-l}$ and $l = (n-2)/2$, for $n > 3$.
$E_n' \subset E_n$ is a subset of paths where all paths (X,Y) beginning by *X* and ending by *Y* with $(X,Y) \in P_n'$.
$P_n'' \subset P_n$ is a subset of P_n such that $X \in P_n''$ and $X=x_1x_2x_3x_4s_1s_2\ldots\ldots s_{n-l}$, such that the number of super node G_4 is equal to $l=(n-2)/2$, for $n > 4$, as $Suffix = s_1s_2\ldots\ldots s_{n-l}$.
$E_n'' \subset E_n$ is a subset of paths where all paths (X,Y) beginning by *X* and ending by *Y,* with $(X,Y) \in P_n''$.

*Suffix*1(*X*) is a function which extracts the *n*-3 characters from a string *X* starting with the character of the lowest weight.

*Suffix*2(*X*) is a function which extracts the *n*-4 characters from a string *X* starting with the character of the lowest weight.

3 EMBEDDING *n*-DIMENSIONAL CROSSED HYPERCUBE GRAPH INTO *n*-DIMENSIONAL PANCAKE GRAPH

In this section, we present a new function, the many-to-one embedding *n*-dimensional crossed hypercube graph denoted by CQ_n into *n*-dimensional pancake graph denoted by G_n.

The main steps of embedding function are as follows:

1. Find the first node of the crossed hypercube and the first node of the pancake. *Example* 000 *of* CQ_3 *and* 123 *of* G_3.
2. Embedding vertex of crossed hypercube of *n* dimensions into pancake of *n* dimensions using the Embed_node(node) algorithm.

Embedding edges of crossed hypercube of *n* dimensions into the path of the pancake of *n* dimensions using the Embed_edge(nodedep,nodearr) algorithm.

3.1 Embed_node(node) Algorithm

Embed_node(node) algorithm is done in the following way:

Case where *n*=3. Embedding crossed hypercube of 3 dimensions into pancake of 3 dimensions as depicted in Figure 4 and Figure 5.
Generally Embed_node(A) algorithm applies all actions specified in the TABLE 1.

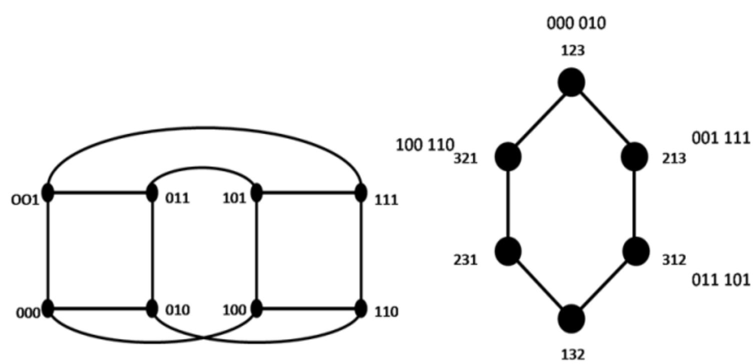

Fig. 4: Crossed hypercube and pancake of 3 dimensions

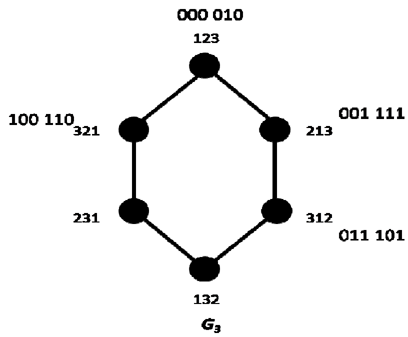

Fig. 5: The embedding graph of CQ_3 into G_3

Table 1: Embed_node(A) algorithm for $A=A1PREFA_{N-2}A_{N-1}A_N$, where $A1$=00,01,10,11.

Nodes of CQ_n *prefixed* by 00	**Nodes of 1st $G_n[n,n]$**	**Nodes of CQ_n *prefixed by* 10**	**Nodes of 2nd $G_n[n,1]$**
00*Pref*000	$x_1x_2x_3Suf1$	10*Pref*000	$x_3x_2x_1Suf2$
00*Pref*001	$x_2x_1x_3Suf1$	10*Pref*001	$x_2x_3x_1Suf2$
00*Pref*010	$x_1x_2x_3Suf1$	10*Pref*010	$x_3x_2x_1Suf2$
00*Pref*011	$x_3x_1x_2Suf1$	10*Pref*011	$x_1x_3x_2Suf2$
00*Pref*100	$x_3x_2x_1Suf1$	10*Pref*100	$x_1x_2x_3Suf2$
00*Pref*101	$x_3x_1x_2Suf1$	10*Pref*101	$x_1x_3x_2Suf2$
00*Pref*110	$x_3x_2x_1Suf1$	10*Pref*110	$x_1x_2x_3Suf2$
00*Pref*111	$x_2x_1x_3Suf1$	10*Pref*111	$x_2x_3x_1Suf2$
Nodes of CQ_n *prefixed* by 01	**Nodes of 3rd $G_n[n,3]$**	**Nodes of CQ_n *prefixed* by 11**	**Nodes of 4th $G_n[n,2]$**
01*Pref*000	$x_3x_1x_2Suf3$	11*Pref*000	$x_2x_1x_3Suf4$
01*Pref*001	$x_3x_1x_2Suf3$	11*Pref*001	$x_1x_2x_3Suf4$
01*Pref*010	$x_3x_1x_2Suf3$	11*Pref*010	$x_2x_1x_3Suf4$
01*Pref*011	$x_1x_3x_2Suf3$	11*Pref*011	$x_2x_1x_3Suf4$
01*Pref*100	$x_1x_3x_2Suf3$	11*Pref*100	$x_1x_2x_3Suf4$
01*Pref*101	$x_1x_3x_2Suf3$	11*Pref*101	$x_2x_1x_3Suf4$
01*Pref*110	$x_1x_3x_2Suf3$	11*Pref*110	$x_1x_2x_3Suf4$
01*Pref*111	$x_3x_1x_2Suf3$	11*Pref*111	$x_1x_2x_3Suf4$

The variable *Sufi* with (i=1…4) is *Suffix*1(X), where $X \in P_i$, such that $G_n(n,k)=(P_i, E)$, where ($k=n$,1…3).

Case where n =4. The embedding nodes of CQ_4 in G_4 are produced as follows: CQ_4 is made recursively by two copies of CQ_3, one copy is prefixed by 0(0 CQ_3) and the other one prefixed by 1(1CQ_3). The G_4 is made recursively by four copies of G_3 named $G_4[4,k]$ for k=1,4. We used in this case two copies of $G_4[4,k]$, for k=1,4. The first is $G_4[4,4]$ used to embed all nodes of CQ_4 prefixing by 0(0CQ_3) and the second component $G_4[4,1]$ to embed all nodes prefixing by 1(1CQ_3). The embedding is done by using the basic function of embedding of CQ_3 into G_3 as depicted in Figure 6. The embedding is done by using the rules specified in TABLE 2.

Table 2: Embedding all nodes of CQ_4 into G_4

0CQ_3	**G_4[4,4]**	**1CQ_3**	**G_4[4,1]**
0000	$x_1x_2x_3x_4$	1000	$x_4x_3x_2x_1$
0001	$x_2x_1x_3x_4$	1001	$x_3x_4x_2x_1$
0010	$x_1x_2x_3x_4$	1010	$x_4x_3x_2x_1$
0011	$x_3x_1x_2x_4$	1011	$x_2x_4x_3x_1$
0100	$x_3x_2x_1x_4$	1100	$x_2x_3x_4x_1$
0101	$x_3x_1x_2x_4$	1101	$x_2x_4x_3x_1$
0110	$x_3x_2x_1x_4$	1110	$x_2x_3x_4x_1$
0111	$x_2x_1x_3x_4$	1111	$x_3x_4x_2x_1$

Fig. 6: Embedding graph of CQ_5 into G_5

The case where *n*=5. The embedded nodes of CQ_5 are produced as follows: CQ_5 is made recursively by prefixing the two copies of CQ_4 one by 0(0 CQ_4) and the other by 1(1 CQ_4), in other words, 00CQ_3, 01CQ_3, 10CQ_3, 11CQ_3. The G_4 is made recursively by four copies of G_3 named $G_4[4,k]$, where k=1,4. The first component is G_4[4,4] used for embedding nodes of CQ_5 prefixing by 00CQ_3, the second G_4[4,1] for embedding all nodes 01CQ_3, the third component G_4[4,3] and the last component G_4[4,2] are used for embedding nodes of 11CQ_3 as shown in Figure 7. The embedding is done by using the rules specified in TABLE 3.

Table 3: Embedding all nodes of CQ_5 into G_5

00CQ_3	**G_4[4,4]**	**10CQ_3**	**G_4[4,1]**	**01CQ_3**	**G_4[4,3]**	**11CQ_3**	**G_4[4,2]**
00000	$x_1x_2x_3x_4$	10000	$x_4x_3x_2x_1$	01000	$x_3x_4x_1x_2$	11000	$x_2x_1x_4x_3$
00001	$x_2x_1x_3x_4$	10001	$x_3x_4x_2x_1$	01001	$x_4x_3x_1x_2$	11001	$x_1x_2x_4x_3$
00010	$x_1x_2x_3x_4$	10010	$x_4x_3x_2x_1$	01010	$x_3x_4x_1x_2$	11010	$x_2x_1x_4x_3$
00011	$x_3x_1x_2x_4$	10011	$x_2x_4x_3x_1$	01011	$x_1x_3x_2x_4$	11011	$x_4x_2x_1x_3$
00100	$x_3x_2x_1x_4$	10100	$x_2x_3x_4x_1$	01100	$x_1x_4x_3x_2$	11100	$x_4x_1x_2x_3$
00101	$x_3x_1x_2x_4$	10101	$x_2x_4x_3x_1$	01101	$x_1x_3x_4x_2$	11101	$x_4x_2x_1x_3$
00110	$x_3x_2x_1x_4$	10110	$x_2x_3x_4x_1$	01110	$x_1x_4x_1x_2$	11110	$x_4x_1x_2x_3$
00111	$x_2x_1x_3x_4$	10111	$x_3x_4x_2x_1$	01111	$x_4x_3x_1x_2$	11111	$x_1x_2x_4x_3$

The case for n > 5. The crossed hypercube of n dimensions is produced by the composition of two copies of crossed hypercube of (n-1)-dimensions. The first is prefixed by 0(0CQ_{n-1}) and the second is prefixed by 1(1CQ_{n-1}). The pancake of n-1dimensions is made by i copies of $G_{n-1}[n\text{-}1,k]$, for k=1,i. In other words, I super nodes containing 2^l components G_4, with i = 2 if n is even, i=4 if n is odd and l = (n-1)/2. There are two stated situations: the first one is when n is even, we use two components: the super node $G_{n-1}[n\text{-}1,\ n\text{-}1]$ and the super node $G_{n-1}[n\text{-}1,\ 1]$, the first for embedding all nodes of $0CQ_{n-1}$ and the second one for embedding all nodes of $1CQ_{n-1}$. The second situation is when n is odd or n=2m+1 (m € ℕ), the CQ_N nodes are $0CQ_{2m}$, $1CQ_{2m}$. For N=2m the CQ_N nodes are $0CQ_N$, $1CQ_N$, that is to say $00CQ_{N\text{-}1}$, $01CQ_{N\text{-}1}$ and $10CQ_{N\text{-}1}$, $11CQ_{N\text{-}1}$. In other words, we use 4 super nodes $G_{N\text{-}1}[N\text{-}1,N\text{-}1]$, $G_{N\text{-}1}[N\text{-}1,1]$, $G_{n\text{-}1}[N\text{-}1,2]$, $G_{N\text{-}1}[N\text{-}1,3]$.

The first node for embedding all nodes of $00CQ_{N\text{-}1}$, the second one for embedding all nodes of $10CQ_{N\text{-}1}$, the third one for embedding all nodes of $11CQ_{N\text{-}1}$ and the last node for embedding all nodes of $01CQ_{N\text{-}1}$ by using the rules specified in TABLE 4.

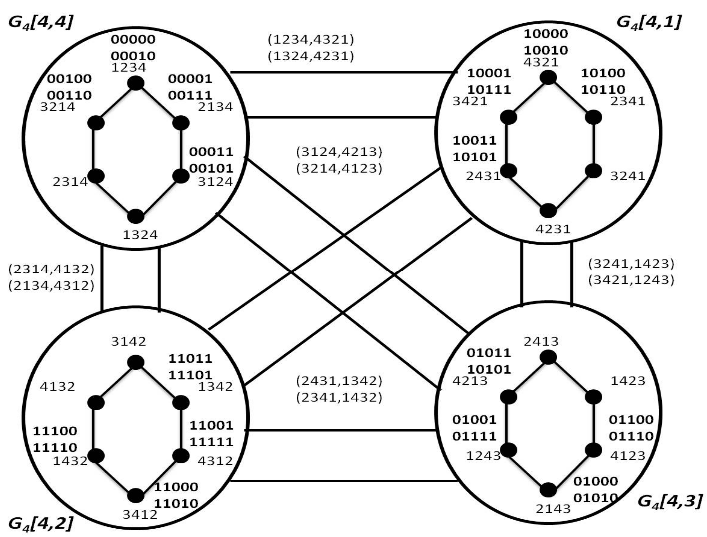

Fig. 7: Embedding graph of CQ_5 into G_5

3.2 Embed_edge(nodedep_nodearr) Algorithm

The Embed_edge(nodedep,nodearr) algorithm is given as follows:

```
Begin;
S1:=Suffix1(nodedep); S2:=Suffix1(nodearr); S3:=Suffix2(nodedep); S4:=Suffix2(nodearr);
If  S1=S2 then Embed1_edge(nodedep,nodearr)
Else
      If  S3=S4  then Embed2_edge(nodedep,nodearr)
      Else
           Embed3_edge(nodedep,nodarr)
      Endif;
   Endif;
End;
```

Table 4: Embedding all nodes of CQ_n in G_5 for n>5

***APref*00CQ_3**	**$G_{n-1}[n-1,n-1]$**	***APref*10CQ_3**	**$G_{n-1}[n-1,1]$**
*APref*000000	$x_1x_2x_3x_4suf1$	*APref*100000	$x_4x_3x_2x_1Suf2$
*APref*000001	$x_2x_1x_3x_4suf1$	*APref*100001	$x_3x_4x_2x_1Suf2$
*APref*000010	$x_1x_2x_3x_4Suf1$	*APref*100010	$x_4x_3x_2x_1Suf2$
*APref*000011	$x_3x_1x_2x_4suf1$	*APref*100011	$x_2x_4x_3x_1Suf2$
*APref*000100	$x_3x_2x_1x_4suf1$	*APref*100100	$x_2x_3x_4x_1Suf2$
*APref*000101	$x_3x_1x_2x_4suf1$	*APref*100101	$x_2x_4x_3x_1Suf2$
*APref*000110	$x_3x_2x_1x_4suf1$	*APref*100110	$x_2x_3x_4x_1Suf2$
*APref*000111	$x_2x_1x_3x_4suf1$	*APref*100111	$x_3x_4x_2x_1Suf2$
***APref*01CQ_3**	**$G_{n-1}[n-1,3]$**	***APref*11CQ_3**	**$G_{n-1}[n-1,2]$**
*APref*010000	$x_3x_4x_1x_2Suf3$	*APref*110000	$x_2x_1x_4x_3Suf4$
*APref*010001	$x_4x_3x_1x_2Suf3$	*APref*110001	$x_1x_2x_4x_3Suf4$
*APref*010010	$x_3x_4x_1x_2Suf3$	*APref*110010	$x_2x_1x_4x_3Suf4$
*APref*010011	$x_1x_3x_2x_4Suf3$	*APref*110011	$x_4x_2x_1x_3Suf4$
*APref*010100	$x_1x_4x_3x_2Suf3$	*APref*110100	$x_4x_1x_2x_3Suf4$
*APref*010101	$x_1x_3x_4x_2Suf3$	*APref*110101	$x_4x_2x_1x_3Suf4$
*APref*010110	$x_1x_4x_1x_2Suf3$	*APref*110110	$x_4x_1x_2x_3Suf4$
*APref*010111	$x_4x_3x_1x_2Suf3$	*APref*110111	$x_1x_2x_4x_3Suf4$

3.3 Embed1_edge(nodedep, nodearr) Algorithm

The Embed1_edge(nodedep, nodearr) algorithm is used when the paths are in the same G_3 of a super node. This procedure applies exactly the different cases outlined in TABLE 5, for A= 00 or 10 and the symmetric paths are shown in TABLE 6 for A= 01 or 11. Note that the function *Suffix* is *Suffix*2(X).

3.4 Embed2_edge(nodedep,nodearr) Algorithm

This procedure is used when the paths are in the same G_4 of a super node. The Embed2_edge(nodedep,nodearr) algorithm realizes the embedding of the edge of crossed hypercube into pancake, if the suffix of nodedep and the suffix of nodearr differ exactly in the fourth position.

Four cases arise in this situation. In the first case, the edge of the crossed hypercube is $Pref00a_{n-2}a_{n-1}a_n$-$Pref01a_{n-2}a_{n-1}a_n$, in the second is $Pref00a_{n-2}a_{n-1}a_n$-$Pref10a_{n-2}a_{n-1}a_n$, in the third case is $Pref01a_{n-2}a_{n-1}a_n$-$Pref11a_{n-2}a_{n-1}a_n$, and finally in the last case is $Pref10a_{n-2}a_{n-1}a_n$-$Pref11a_{n-2}a_{n-1}a_n$. The Embed2_edge(nodedep,nodearr) algorithm applies exactly the actions outlined in TABLE 7.

Table 5: Embedding edges with label format 00*Pref* $x_1x_2x_3$-00*Pref* $y_1y_2y_3$ of crossed hypercube into pancake

Crossed hypercube edge	Pancake path(*S*1=*Suffix*)	Dilation
*APref*000-*APref001*	$x_1x_2x_3x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	1
*APref*000-*APref*010	$x_1x_2x_3x_4$*Suffix*-$x_1x_2x_3x_4$*Suffix*	1
*APref*000-*APref*100	$x_1x_2x_3x_4$*Suffix*-$x_3x_2x_1x_4$*Suffix*	1
*APref*001-*APref*011	$x_2x_1x_3x_4$*Suffix*-$x_3x_1x_2x_4$*Suffix*	1
*APref*001-*APref*111	$x_2x_1x_3x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	1
*APref*010-*APref*011	$x_1x_2x_3x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*- $x_3x_1x_2x_4$*Suffix*	2
*APref*010-*APref*110	$x_1x_2x_3x_4$*Suffix*-$x_3x_2x_1x_4$*Suffix*	1
*APref*011-*APref*101	$x_3x_1x_2x_4$*Suffix*-$x_3x_1x_2x_4$*Suffix*	1
*APref*100-*APref*101	$x_3x_2x_1x_4$*Suffix*-$x_1x_2x_3x_4$*Suffix*- $x_2x_1x_3x_4$*Suffix*-$x_3x_1x_2x_4$*Suffix*	3
*APref*101-*APref*111	$x_3x_1x_2x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	1
*APref*110-*APref*111	$x_3x_2x_1x_4$*Suffix*-$x_1x_2x_3x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	2

Table 6: Embedding edges with label format 10*Pref* $x_1x_2x_3$-10*Pref* $y_1y_2y_3$ of crossed hypercube into pancake

Crossed hypercube edge	Pancake path (S1=x_4*Suffix*)	Dilation
*APref*000-*APref001*	$x_1x_3x_2x_4$*Suffix*-$x_3x_1x_2x_4$*Suffix*	1
*APref*000-*APref*010	$x_1x_3x_2x_4$*Suffix*-$x_1x_3x_2x_4$*Suffix*	1
*APref*000-*APref*100	$x_1x_3x_2x_4$*Suffix*-$x_2x_3x_1x_4$*Suffix*	1
*APref*001-*APref*011	$x_3x_1x_2x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	1
*APref*001-*APref*111	$x_1x_3x_2x_4$*Suffix*-$x_1x_3x_2x_4$*Suffix*	1
*APref*010-*APref*011	$x_1x_3x_2x_4$*Suffix*-$x_3x_1x_2x_4$*Suffix*- $x_2x_1x_3x_4$*Suffix*	2
*APref*010-*APref*110	$x_1x_3x_2x_4$*Suffix*-$x_2x_3x_1x_4$*Suffix*	1
*APref*011-*APref*101	$x_2x_1x_3x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	1
*APref*100-*APref*101	$x_2x_3x_1x_4$*Suffix*-$x_1x_3x_2x_4$*Suffix*- $x_3x_1x_2x_4$*Suffix*-$x_2x_1x_3x_4$*Suffix*	3
*APref*101-*APref*111	$x_2x_1x_3x_4$*Suffix*-$x_1x_3x_2x_4$*Suffix*	1
*APref*110-*APref*111	$x_2x_3x_1x_4$*Suffix*-$x_1x_3x_2x_4$*Suffix*- $x_1x_3x_2x_4$*Suffix*	2

Table 7: Cases of embedding CQ_n into G_n, when the path is in the same G_4 of any super node

Crossed hypercube edge	Pancake path	Dilation
*APref*00000-*BPref*00000 *APref*00010-*BPref*00010	$x_1x_2x_3x_4x_5Suffix-x_3x_2x_1x_4x_5Suffix$	1
*APref*00001-*BPref*00011 *APref*00101-*BPref*00111	$x_1x_2x_3x_4x_5Suffix-x_2x_3x_1x_4x_5Suffix-x_5x_4x_1x_3x_2Suffix-x_4x_5x_1x_3x_2Suffix-x_1x_5x_4x_3x_2Suffix$	4
*APref*00100-*BPref*00100 *APref*00110-*BPref*00110	$x_1x_2x_3x_4x_5Suffix-x_3x_2x_1x_4x_5Suffix-x_5x_4x_1x_2x_3Suffix-x_1x_4x_5x_2x_3Suffix$	3
*APref*00011-*BPref*00001 *APref*00111-*BPref*00101	$x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_5x_4x_3x_1x_2Suffix-x_3x_4x_5x_1x_2Suffix-x_1x_5x_4x_3x_2Suffix$	4
*APref*01000-*BPref*01000 *APref*01010-*BPref*01010	$x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_5x_1x_2x_3x_4Suffix$	2
*APref*01001-*BPref*01011 *APref*01101-*BPref*01111	$x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_4x_3x_1x_2x_5Suffix-x_5x_2x_1x_3x_4Suffix-x_2x_5x_1x_3x_4Suffix$	4
*APref*01100-*BPref*01100 *APref*01110-*BPref*01110	$x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_5x_1x_2x_3x_4Suffix-x_4x_3x_2x_1x_5Suffix$	3
*APref*01011-*BPref*01001 *APref*01111-*BPref*01101	$x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_4x_3x_1x_2x_5Suffix-x_5x_2x_1x_3x_4Suffix-x_3x_1x_2x_5x_4Suffix$	4
*APref*10000-*BPref*10000 *APref*10010-*BPref*10010	$x_1x_2x_3x_4x_5Suffix-x_3x_2x_1x_4x_5Suffix-x_5x_4x_1x_2x_3Suffix-x_1x_4x_5x_2x_3Suffix$	3
*APref*10001-*BPref*10011 *APref*10101-*BPref*10111	$x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_5x_4x_3x_1x_2Suffix-x_3x_4x_5x_1x_2Suffix-x_4x_3x_5x_1x_2Suffix-x_4x_3x_5x_1x_2Suffix$	5
*APref*10100-*BPref*10100 *APref*10110-*BPref*10110	$x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix$	1
*APref*10011-*BPref*10001 *APref*10111-*BPref*10101	$x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix-x_4x_5x_3x_2x_1Suffixx_3x_5x_4x_2x_1Suffix-x_2x_4x_5x_3x_1Suffix-x_4x_2x_5x_3x_1Suffix$	5
*APref*11000-*BPref*11000 *APref*11010-*BPref*11010	$x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_2x_3x_4x_1x_5Suffix-x_5x_1x_4x_3x_2Suffix-x_4x_1x_5x_3x_2Suffix-x_3x_5x_1x_4x_2Suffix$	5
*APref*11001-*BPref*11011 *APref*11101-*BPref*11111	$x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix-x_4x_5x_3x_2x_1Suffix-x_2x_3x_5x_4x_1Suffix$	3
*APref*11100-*BPref*11100 *APref*11110-*BPref*11110	$x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_2x_3x_4x_1x_5Suffix-x_5x_1x_4x_3x_2Suffix-x_4x_1x_5x_3x_2Suffix-x_3x_5x_1x_4x_2Suffix$	5
*APref*11011-*BPref*11001 *APref*11111-*BPref*11101	$x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix-x_4x_5x_3x_2x_1Suffix-x_2x_3x_5x_4x_1Suffix$	3

Table 8: Case 1 for *A*=00 and *B*=01

Case	Crossed hypercube edge	Pancake path	Dilation
	*Pref*00000-*Pref*01000 *Pref*00010-*Pref*01010	$x_1x_2x_3x_4\mathit{Suffix}$-$x_3x_2x_1x_4\mathit{Suffix}$- $x_2x_1x_4x_3\mathit{Suffix}$-$x_2x_1x_4x_3\mathit{Suffix}$	3
1	*Pref*00001-*Pref*01011 *Pref*00101-*Pref*01111	$x_1x_2x_3x_4\mathit{Suffix}$-$x_2x_1x_3x_4\mathit{Suffix}$- $x_4x_3x_2x_1\mathit{Suffix}$	2
	*Pref*00100-*Pref*01100 *Pref*00110-*Pref*01110	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$	1
	*Pref*00011-*Pref*01001 *Pref*00111-*Pref*01101	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$- $x_2x_3x_4x_1\mathit{Suffix}$	2
	*Pref*00000-*Pref*10000 *Pref*00010-*Pref*10010	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$	1
2	*Pref*00001-*Pref*10011 *Pref*00101-*Pref*10111	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$- $x_2x_3x_4x_1\mathit{Suffix}$-$x_1x_4x_3x_2\mathit{Suffix}$	3
	*Pref*00100-*Pref*10100 *Pref*00110-*Pref*10110	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$- $x_3x_4x_2x_1\mathit{Suffix}$-$x_1x_2x_4x_3\mathit{Suffix}$	3
	*Pref*00011-*Pref*10001 *Pref*00111-*Pref*10101	$x_1x_2x_3x_4\mathit{Suffix}$-$x_3x_2x_1x_4\mathit{Suffix}$- $x_4x_1x_2x_3\mathit{Suffix}$-$x_1x_4x_2x_3\mathit{Suffix}$	3
	*Pref*01000-*Pref*11000 *Pref*01010-*Pref*11010	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$	1
	*Pref*01001-*Pref*11011 *Pref*01101-*Pref*11111	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$- $x_2x_3x_4x_1\mathit{Suffix}$-$x_1x_4x_3x_2\mathit{Suffix}$	3
3	*Pref*01100-*Pref*11100 *Pref*01110-*Pref*11110	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$- $x_3x_4x_2x_1\mathit{Suffix}$-$x_1x_2x_4x_3\mathit{Suffix}$	3
	*Pref*01011-*Pref*11001 *Pref*01111-*Pref*11101	$x_1x_2x_3x_4\mathit{Suffix}$-$x_3x_2x_1x_4\mathit{Suffix}$- $x_4x_1x_2x_3\mathit{Suffix}$-$x_1x_4x_2x_3\mathit{Suffix}$	3
	*Pref*10000-*Pref*11000 *Pref*10010-*Pref*11010	$x_1x_2x_3x_4\mathit{Suffix}$-$x_3x_2x_1x_4\mathit{Suffix}$- $x_2x_1x_4x_3\mathit{Suffix}$-$x_2x_1x_4x_3\mathit{Suffix}$	3
	*Pref*10001-*Pref*11011 *Pref*10101-*Pref*11111	$x_1x_2x_3x_4\mathit{Suffix}$-$x_2x_1x_3x_4\mathit{Suffix}$- $x_4x_3x_2x_1\mathit{Suffix}$	2
4	*Pref*10100-*Pref*11100 *Pref*10110-*Pref*11110	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$	1
	*Pref*10011-*Pref*11001 *Pref*10111-*Pref*11101	$x_1x_2x_3x_4\mathit{Suffix}$-$x_4x_3x_2x_1\mathit{Suffix}$- $x_2x_3x_4x_1\mathit{Suffix}$	2

Table 9: Case 2 for *A*= 00 and *B*=10

Crossed hypercube edge	Pancake path	Dilation
*APref*00000-*BPref*00000 *APref*00010-*BPref*00010	$x_1x_2x_3x_4x_5Suffix$-$x_3x_2x_1x_4x_5Suffix$-$x_5x_4x_1x_2x_3Suffix$	2
*APref*00001-*BPref*00011 *APref*00101-*BPref*00111	$x_1x_2x_3x_4x_5Suffix$-$x_3x_2x_1x_4x_5Suffix$-$x_5x_4x_1x_2x_3Suffix$-$x_1x_4x_5x_2x_3Suffix$-$x_2x_5x_4x_1x_3Suffix$	4
*APref*00100-*BPref*00100 *APref*00110-*BPref*00110	$x_1x_2x_3x_4x_5Suffix$-$x_5x_4x_1x_2x_3Suffix$-$x_1x_4x_5x_2x_3Suffix$	2
*APref*00011-*BPref*00001 *APref*00111-*BPref*00101	$x_1x_2x_3x_4x_5Suffix$-$x_5x_4x_3x_2x_1Suffix$-$x_3x_4x_5x_2x_1Suffix$-$x_2x_5x_4x_3x_1Suffix$-$x_4x_2x_5x_3x_1Suffix$	4
*APref*01000-*BPref*01000 *APref*01010-*BPref*01010	$x_1x_2x_3x_4x_5Suffix$-$x_4x_3x_2x_1x_5Suffix$-$x_2x_3x_4x_5x_1Suffix$-$x_1x_5x_4x_3x_2Suffix$-$x_3x_4x_5x_1x_2Suffix$	4
*APref*01001-*BPref*01011 *APref*01101-*BPref*01111	$x_1x_2x_3x_4x_5Suffix$-$x_5x_4x_3x_2x_1Suffix$-$x_3x_4x_5x_2x_1Suffix$-$x_2x_5x_4x_3x_1Suffix$	3
*APref*01100-*BPref*01100 *APref*01110-*BPref*01110	$x_1x_2x_3x_4x_5Suffix$-$x_2x_1x_3x_4x_5Suffix$-$x_5x_4x_3x_1x_2Suffix$-$x_1x_3x_4x_5x_2Suffix$-$x_4x_3x_1x_5x_2Suffix$-$x_3x_4x_1x_5x_2Suffix$	5
*APref*01011-*BPref*01001 *APref*01111-*BPref*01101	$x_1x_2x_3x_4x_5Suffix$-$x_3x_2x_1x_4x_5Suffix$-$x_5x_4x_1x_2x_3Suffix$-$x_2x_1x_4x_5x_3Suffix$-$x_1x_2x_4x_5x_3Suffix$-$x_4x_5x_4x_1x_3Suffix$	3
*APref*10000-*BPref*10000 *APref*10010-*BPref*10010	$x_1x_2x_3x_4x_5Suffix$-$x_5x_4x_3x_2x_1Suffix$-$x_3x_4x_5x_2x_3Suffix$	2
*APref*10001-*BPref*10011 *APref*10101-*BPref*10111	$x_1x_2x_3x_4x_5Suffix$-$x_2x_1x_3x_4x_5Suffix$-$x_5x_4x_3x_1x_2Suffix$-$x_3x_4x_5x_1x_2Suffix$-$x_4x_3x_5x_1x_2Suffix$-$x_5x_3x_4x_1x_2Suffix$	5
*APref*10100-*BPref*10100 *APref*10110-*BPref*10110	$x_1x_2x_3x_4x_5Suffix$-$x_3x_2x_1x_4x_5Suffix$-$x_5x_4x_1x_2x_3Suffix$	2
*APref*10011-*BPref*10001 *APref*10111-*BPref*10101	$x_1x_2x_3x_4x_5Suffix$-$x_2x_1x_3x_4x_5Suffix$-$x_5x_4x_3x_1x_2Suffix$-$x_3x_4x_5x_1x_2Suffix$-$x_1x_5x_4x_3x_2Suffix$-$x_5x_1x_4x_3x_2Suffix$	5
*APref*11000-*BPref*11000 *APref*11010-*BPref*11010	$x_1x_2x_3x_4x_5Suffix$-$x_4x_3x_2x_1x_5Suffix$-$x_5x_1x_2x_3x_4Suffix$-$x_2x_1x_5x_3x_4Suffix$-$x_3x_5x_1x_2x_4Suffix$	4
*APref*11001-*BPref*11011 *APref*11101-*BPref*11111	$x_1x_2x_3x_4x_5Suffix$-$x_3x_2x_1x_4x_5Suffix$-$x_4x_1x_2x_3x_5Suffix$-$x_5x_3x_2x_1x_4Suffix$-$x_2x_3x_5x_1x_4Suffix$	4
*APref*11100-*BPref*11100 *APref*11110-*BPref*11110	$x_1x_2x_3x_4x_5Suffix$-$x_4x_3x_2x_1x_5Suffix$-$x_5x_1x_2x_3x_4Suffix$-$x_2x_1x_5x_3x_4Suffix$-$x_3x_5x_1x_2x_4Suffix$	4
*APref*11011-*BPref*11001 *APref*11111-*BPref*11101	$x_1x_2x_3x_4x_5Suffix$-$x_4x_3x_2x_1x_5Suffix$-$x_5x_1x_2x_3x_4Suffix$	2

3.5 Embed3_edge(nodedep, nodarr) Algorithm

This procedure is used when n>5 and all paths are between two different G_4 in the different super nodes.

Let $A=a_1a_2$ and $B=b_1b_2$, where (a_1a_2, b_1b_2)=(00,01), (00,10), (01,11), (10,11). For n>5, Embed3_edge(nodedep,nodarr) algorithm performs the different actions specified in the four stated following cases. Excepting the case when n=6, *APref* is reduced to 0, *BPref* is reduced to 1.

For the case *n*=7, (*APref, BPref*)=(00,01), (00,10), (01,11), (10,11).

For the sake of simplicity the cases 3 and 4 are not given in the paper.

4 DILATIONS OF MANY-TO-ONE *n*-DIMENSIONAL CROSSED HYPERCUBE EMBEDDED INTO *n*-DIMENSIONAL PANCAKE

4.1 Lemma 1

The n-dimensional crossed hypercube $CQ_n^{'}$=($V,U1$) has many-to-one dilation 3 embedding into $G_n^{'}$=($P_n^{'}$, $E_n^{'}$) for any n>3.

Proof
We prove this lemma by induction.

Base
For n = 3, TABLE 1 presents all paths between the embedded nodes of CQ_3 into G_3 with dilation 3.

Induction hypothesis
Suppose that for $k \leq n$-1, $CQ_{k-1}^{'}$ embedding many-to-one dilation 3 into $G^{'}_{k-1}$ is true. Let us now prove that is true for k=n.
We have the following cases:

Case* 1:* k is even

$CQ_n^{'}$=($V,U1$) is constructed by two copies of $CQ_{n-1}^{'}$, one copy is prefixed by 0(0$CQ_{k-1}^{'}$), the second one is prefixed by 1(1$CQ_{k-1}^{'}$). All nodes $A \in V$, such that, $A=0Prefa_{n-3}a_{n-2}a_{n-1}= Pref_1a_{k-3}a_{k-2}a_{k-1}$ are embedded by Embed_node(A) algorithm as shown in TABLE 4 into the first super node or into the projection $G_k^{'}[k,k]$.

All nodes $A \in V$, $A=1prefa_{k-3}a_{k-2}a_{k-1}$ *or* $A=Pref_2a_{k-3}a_{k-2}a_{k-1}$ are embedded into the second super node or into the projection $G_k^{'}[k,1]$ as shown in TABLE 5. That is to say, they are embedded into $G_{k-1}^{'}$. However, the dilation of embedding into $G_{k-1}^{'}$ is 3 (hypothesis of induction). □

Case* 2*: *k* is odd

Let k=2m+1, where $m \in \mathbb{N}$, and CQ_n is obtained from two copies of $0CQ'_{2m}$ and $1CQ'_{2m}$, and suppose that for N=2m we have $0CQ'_N$ and $1CQ'_N$, that is to say, $00CQ'_{N-1}$, $01CQ'_{N-1}$ and $10CQ'_{N-1}$, $11CQ'_{N-1}$.
The Embed-node(A) algorithm as shown in TABLE 1, embed all nodes A=$00Prefa_{N-3}a_{N-2}a_{N-1}$ ($A \in V$) into the first super node or into the projection $G'_N[N,N]$, all nodes A=$10Prefa_{N-3}a_{N-2}a_{N-1}$ into $G'_N[N,1]$, all nodes A=$01Prefa_{N-3}a_{N-2}a_{N-1}$ *into* $G'_N[N,3]$, and all nodes A=$11Prefa_{N-3}a_{N-2}a_{N-1}$ *into* $G'_N[N,2]$. In other words, we use only four super nodes among the *k* projections or super nodes. G'_N is a (n-1)-dimensional pancake graphs and the embedding many-to-one dilation 3 into G'_N (hypothesis of induction). □

4.2 Lemma 2

The n-dimensional crossed hypercube CQ''_n=(V,U2) has many-to-one dilation 4 embedding into G''_n=(P''_n, E''_n) for any n>4.

Proof
We use the same method to prove lemma 2, except that the embedding of the edges of CQ''_k is defined in TABLE 7 for the case where k is even and TABLE 7 for the case where k is odd.

Theorem
The n-dimensional crossed hypercube CQ_n=(V,U) has many-to-one dilation 5 embedding into G_n=(P_n,E_n) for any n>5.

Proof
Base: For n = 6, TABLE 9 presents the case of different actions of embedding all edges of CQ_6 into G_6 with dilation 5.

For n = 7, TABLE 8, TABLE 9 and include the non given Tables for case 3 and case 4 present the different actions of embedding all edges of CQ_7 into G_7 with dilation 5.

Induction hypothesis
Assume that this lemma holds for $k \leq n$-1. That is CQ_{k-1} embedding many-to-one dilation 5 into G_{k-1} is true.

Now we prove that this is true for k=n.

Case 1*:* k is even. There are two sub-cases

Case a

As the crossed hypercube is defined to be CQ_k=(V,U). Let A and $B \in V$, where A = $0Prefa_{k-4}a_{k-3}a_{k-2}a_{k-1}$=$Pref_1\ a_{k-4}a_{k-3}a_{k-2}a_{k-1}$ as $Pref_1$ =$0Pref$ and B = $Pref_1b_{k-4}b_{k-3}b_{k-2}b_{k-1}$. The embedding of ($A,B$) $\in U$ into the first super node or into the projection $G_k[k,k]$. All edges (A,B) $\in$ *U such that,* A=$1prefa_{k-4}a_{k-3}a_{k-2}a_{k-1}$ *or* A=$Pref_2a_{k-4}a_{k-3}a_{k-2}a_{k-1}$ where $Pref_2$=$1Pref$, and *the node* B=$Pref_2b_{k-4}b_{k-3}b_{k-2}b_{k-1}$ are embedded into the second super node or into the projection $G_k[k,1]$ in other words, into G_{k-1}. However, the dilation of embedding into G_{k-1} is 5 hypothesis of induction. □

Case b

As the crossed hypercube is defined to be $CQ_k = (V,U)$. Let A and $B \in V$, $A=0Prefa_{k-4}a_{k-3}a_{k-2}a_{k-1}$ or $A=Pref_1\ a_{k-4}a_{k-3}a_{k-2}a_{k-1}$ as $Pref_1=0Pref$ and $B=Pref_2b_{k-4}b_{k-3}b_{k-2}b_{k-1}$.

If we use Embed_node(A) algorithm, all nodes A are embedded into a super node $G_k[k,k]$ and all nodes B are embedded into a super node $G_k[k,1]$. The different edges of CQ_k are embedded into different paths. The first node of every path is embedded into the super node $G_k[k,k]$ and the ending node is embedded into the super node $G_k[k,1]$, that is to say, we use the different embedding edge*s* outlined in case 1, case 2, cases 3 (not given in the paper) and case 4. In all cases the dilation is 5. ▭

Case 2: k is odd. There are two sub-cases

Case a

Let $k = 2m+1$, where $m \in \mathbb{N}$, CQ_k is produced by two copies of $0CQ'_{2k}$ and $1CQ'_{2k}$. Suppose that for $N=2k$ we have $0CQ'_N$, $1CQ'_N$, in other words, $00CQ'_{N-1}$, $01CQ'_{N-1}$, $10CQ'_{N-1}$ and $11CQ'_{N-1}$. Let A *and* $B \in V$ where $A=A_1A_2$, such that $A_1=(00,01,10,11)$, $A_2=Prefa_{N-4}a_{N-3}a_{N-2}a_{N-1}$ *as* $Pref_1=A_1Pref$, hence, $A=Pref_1a_{N-4}a_{N-3}a_{N-2}a_{N-1}$ and $B=Pref_1b_{N-3}b_{N-2}b_{N-1}b_N$. The embedding of $(A,B) \in U$ is into the first super node $G_N[N,N]$ if $A_1=00$, it is into the second super node $G_N[N,1]$ if $A_1=10$, it is into the third super node $G_N[N,3]$ if $A_1=01$, and it is into the fourth super node $G_N[N,2]$ if $A_1=11$. The dilation in all super nodes is 5 (hypothesis induction). ▭

Case b

Let A *and* $B \in V$, and $A=A_1A_2$, $B=B_1B_2$ as $(A_1,B_1)=(00,01)$, $(00,10)$, $(01,11)$, $(10,11)$ and $A_2=Pref_1a_{N-4}a_{N-3}a_{N-2}a_{N-1}$, $B_2=Pref_1b_{N-3}b_{N-2}b_{N-1}b_N$. The embedding of $(A,B) \in U$ are into a different paths between two super nodes $(G_N[N,N],G_N[N,3])$, $(G_N[N,N],G_N[N,1])$, $(G_N[N,3],G_N[N,2])$, $(G_N[N,1],G_N[N,2])$. Each super node contains exactly $2^{l-1}G_4$. In other words, case 1 or case 2 is used, because the first node of the different paths is located in one node of G_4 of the super node $G_N[N,N]$, and the ending node is located in one node of G_4 of the super node $G_N[N,3]$. Or for all edges of CQ_N having the first extremity a node prefixed by $00Pref$, and the second extremity a node prefixed by $01Pref$ for instance case 1, case 2, case 3 and case 4 (cases 3 and 4 are not given in the paper) are used. In all cases the dilation is 5. ▭

5 CONCLUSION

It is both practically significant and theoretically interesting to investigate the embeddability of different architecture into pancake (Miller, Z., et al., 1994, Senoussi, H., Lavault, C., 1997, Hung, C.N., et al., 2002). In this paper, the main purpose is the many-to-one 5 dilation embedding of n-dimensional crossed hypercube into pancake of n dimensions. The study of the dilation of this new function many-to-one embedding is explained in three steps. The first step is the embedding many-to-one dilation 3 of all edges in paths in the same G_3 components of a super node as proved by lemma 1. The second step is that for all paths results of many-to-one dilation 4 embedding graph are in the same G_4 components of a super node, in other words, the path is between two G_3 of the same G_4 as proved by lemma 2, and the latter step is the general embedding many-to-one dilation 5 of all edges of the n-dimensional crossed hypercube CQ_n in the paths between two different super nodes.

In the feature of this work, it is more interesting to study the one-to-one embedding case and the fault-tolerant embedding of n-dimensional crossed hypercube into n-dimensional pancake graph.

References

Akers, S. B., Krishnamurthy, B. (1989). A group-theoritic model for symmetric interconnection networks. IEEE transactions on Computers. Vol. 4, pp. 555-566.

Aschheim, R., Femmam, S., Zerarka, M.F. (2012). New “Graphiton” Model: a Computational Discrete Space, Self-Encoded as a Trivalent Graph. Computer and Information Science Journal, Vol. 5, N° 1, pp. 2-12.

Bouabdallah, A., Heydemann, M.C., Opatrny, J., Sotteau, D. (1998). Embedding complete binary trees into star and pancake graphs. Theory Computer System Journal, pp. 279-305.

Chang, C. P., Sung, T. Y., Hsu, L. H. (2000). Edge congestion and topological properties of crossed cubes. IEEE Transactions Parallel and Distributed Systems, vol. 4 pp. 64-80.

Efe, K. (1991). A variation on the hypercube with lower diameter. IEEE Trans. Comput. vol. 40, n°. 11, pp. 1312-1316.

Efe, K. (1992). The crossed cube architecture for parallel computing. IEEE Trans. Parallel Distributed Systems, vol. 3, n° 5, pp. 513-524.

Fan, J. (2002). Hamilton-connectivity and cycle-embedding of the mobius cubes. Information Processing Letters, Vol. 82, pp. 113-117.

Fang, W.C., Hsu, C.C. (2000). On the fault-tolerant embedding of complete binary trees in the pancake graph interconnection network. Information Sciences, Elsevier, pp. 191-204.

Femmam, S., Zerarka, M.F., Benakila, M.I., Ouahabi, A. (2012). New Approach Construction for Wireless ZigBee Sensor Based on Embedding Pancake Graphs. Networks and Communication Technologies Journal, to appear in June 2012.

Heydari, M.H., Sudborough, I.H. (1997). On the diameter of Pancake network. Algorithms Journal pp. 67-94.

Huang, W.T., Lin, M.Y., Tan, J.M., Hsu, L.H. (2000). Fault-tolerant ring embedding in faulty crossed cubes. World Multiconf.Sys. Cybertetics and Infor. SCI’2000, Vol. IV, pp. 97-102.

Hung, C.N., Liang, K.Y., Hsu, L.H. (2002). Embedding Hamiltonian Paths and Hamiltonian Cycles in Faulty Pancake Graphs. International Symposium on International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '02).

Hung, C.N., Hsu, H., Liang, K., Hsu, L. (2003). Ring embedding in faulty pancake graphs. Information Processing Letters, Elsevier, Vol. 86, pp. 271-275.

Hsieh, S.Y., Chen, G.H., Ho, C.W. (1998). Embed longest rings onto star graphs with vertex faults. International Conference on Parallel Processing, Proceeding, pp. 140-147.

Hsieh, S.Y., Chen, G.H., Ho, C.W. (1999). Fault-free Hamilton-cycles in faulty arrangement graphs. IEEE Transactions on Parallel Distribution System, Vol. 10, pp. 223-237.

Hsieh, S.Y., Chen, C.H. (2004). Pancyclicity on Möbius cubes with maximal edge faults. Parallel Computing, vol. 30, no. 3, pp. 407-421.

Hsieh, S.Y., Chang, N.W. (2006). Hamiltonian path embedding and pancyclicity on the Mobius cube with faulty nodes and faulty edges. IEEE Transactions on Computers, vol. 55, no. 7, pp. 854-863.

Hsieh, S.Y., Lee, C.W. (2009). Conditional edge-fault hamiltonicity of matching composition networks. IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 4, pp. 581-592.

Hsieh, S.Y., Lee, C.W. (2010). Pancyclicity of restricted hypercube-like networks under the conditional fault model. SIAM Journal on Discrete Mathematics, vol. 23, no. 4, pp. 2010-2019.

Hwang, S.C., Chen, G.H. (2000). Cycles in butterfly graphs. Networks Journal, vol. 35, pp. 161-171.

Kanevsky, A., Feng, C. (1995). On the embedding of cycles in pancake graphs. Parallel Computer Journal, pp. 923-926.

Kulasinghe, P., Bettayeb, S. (1995a). Multiply-Twisted Hypercube with Five or More Dimensions is not Vertex-Transitive. Information Processing Letters, vol. 5, pp. 33-36.

Kulasinghe, P., Bettayeb, S. (1995b). Embedding binary trees into crossed cubes. IEEE Trans. Comput., vol. 44, pp. 923-929.

Lin, C.K., Tan, J.M., Huang, H.M., Hsu, D.F., Hsu, L.H. (2008). Mutually Independent Hamiltonianicity of Pancake Graphs and Star Graphs. International Symposium on Parallel Architectures, Algorithms, and Networks (i-span), pp. 151-158.

Lin, J.C., Yang, J.S., Hsu, C.C., Chang, J.M. (2010). Independent spanning trees vs edge-disjoint spanning trees in locally twisted cubes. Information Processing Letters Journal, Elsevier North-Holland, Inc. Amsterdam, The Netherlands, Vol. 110 Issue 10, pp. 414-419.

Menn, A., Somani, A.K. (1992). An efficient sorting algorithm for the star graph interconnection network. 19th International Conference In Parallel Computation, Proceeding, Ed. Cambridge University Press.

Miller, Z., Pritikin, D., Sudborough, I.H. (1994). Near embedding of hypercubes into cayley graphs on the symmetric group. IEEE Transactions on Computers, Vol. 43, 1994, pp. 13-22.

Morales, L., Sudborough, I. (1996). Comparing Star and Pancake Networks. Symposium Parallel and Distributed Processing.

Rowley, R.A., Bose, B. (1993). On ring embedding in de-Bruijn networks. IEEE Transaction on Computer Journal, Vol. 12, pp. 1480-1486.

Rowley, R.A., Bose, B. (1998). Fault-tolerant ring embedding in de Bruijn networks. IEEE Transactions on Computer. Vol. 12, pp. 1480-1486.

Sengupta, A. (2003).On ring embedding in Hypercubes with faulty nodes and links. Information Processing Letters, Elsevier, Vol. 68, pp. 207-214.

Senoussi, H., Lavault, C. (1997). Embeddings into the pancake interconnection network. High Performance Computing and Grid in Asia Pacific Region, International Conference on High-Performance Computing on the Information Superhighway, HPC-Asia '97.

Qiu, K., Akl, S.G., Stojmenovic, D.I. (1991). Data communication and computational geometry on the star interconnection networks. 3rd IEEE Symp. On parallel and Distributed Processing, Proceeding, pp. 125-129.

Qiu, K. (1992). The star and Pancake interconnection networks: proprieties and algorithms. PhD thesis, Queens University, Kingston, Ontario, Canada.

Yang, M. C., Li, T. K., Tan, J. M., Hsu, L. H. (2003). Fault-tolerant cycles embedding of crossed cubes. Elsevier, Information Processing Letters, vol. 88, pp. 149-154.
