Circle Graphs: A New Class of Fixed Degree Interconnection Networks

Bo Ok Seong #1, Hyeong Ok Lee *2

Dept. Convergence of scientific information., National Univ. of Sunchon, Republic of Korea
Dept. Computer Edu., National Univ. of Sunchon, Republic of Korea
qjrl33@naver.com, oklee@scnu.ac.kr(Corresponding Author)

Article Info
Page Number: 10815-10830
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received:
15 September 2022
Revised: 25 October 2022
Accepted: 14 November 2022
Publication: 21 December 2022

Abstract

This paper suggests a novel graph Circle Graph(CG_n) with fixed degree of three. The node address of the CG_n is expressed with a n-binary number, along with degree of three, 3(2^(n-1)) edges, and 2^n nodes. In this paper, we suggest shortest-path routing algorithm for message transmission, and prove the diameter of CG_n is 2^(n-1) based on the routing algorithm outcome. Our CG_n possess the node symmetry and Hamilton cycle-structure. In addition, in this paper, we developed an algorithm to create parallel paths without node duplication.

Keywords: Algorithm; Graph; Routing; Parallel Path; Shortest Path.

Introduction

Supercomputers play a major role in improving the competitiveness of national science and technology. The competitiveness of supercomputers is the core of the future, and it is used in all industries such as medical care, materials, defense, meteorology, and chemistry, so it has a great influence on the whole country[1]. High-performance computers are evolving throughout the rapidly changing era, offering satisfaction to new market demands and diverse requirements on application domains[2-4].

A parallel computer is a computer system that subdivides an assignment into multiple tasks. It processes the corresponding task by allocating them to hardware resources (*e.g.*, multiple cores) in a parallel fashion. Parallel computers are mainly divided into shared-memory multiprocessors and message passing computers[5]. In a shared-memory multiprocessor system, the memory system directly affects the overall performance[6]. The critical attribute of the message passing computer is the interconnection network, which refers to the connection structure between processors considering the locational property, which is one of the factors that determines the performance of a parallel processing system[7-8]. Therefore, research on interconnection networks is considered fundamental in order to improve the performance and control the execution of parallel processing computers[9].

The network cost is a widely implemented metric to quantitatively assess the interconnection network, mathematically defined as a multiplication between the degree and the diameter where degree refers to the hardware cost and diameter denotes the software cost. In order to construct an efficient network structure with the minimum usage of network cost, the cardinality of both the degree and diameter should be reduced[10]. However, the relationship between the degree and diameter shows a negative correlation, and this trade-off hinders the achievement of decreasing the network cost in a harmonious manner[7,11,12].

The Shortest Path problem refers to a problem of finding an optimal route in which the sum of weights of edges leads to the minimum value that connects the node u to node v in the network G where $(n, v) \in G$. The edge weight is the value of the edge connecting the existing nodes and could be both positive to negative numerical values. This concept offers a numerical representation of resources and factors that evaluate the network (*e.g.* cost, distance, time, etc.), which can have positive or negative values. The shortest path problem in a weighted graph (*i.e.*, edges denoted with weights) computes the shortest path that reaches the destination node v from the initial starting node u. Diverse research was conducted to effectively locate the shortest path concerning this problem. In this paper, we propose a novel degree graph CG_n that presents a degree of three. We analyze the theoretical properties of this graph structure and the shortest path algorithm with respect to the parallel routing algorithmic perspective. Our n-dimensional CG_n indicates the node address through n-bit binary, having degree of three, 2^n number of nodes, and 2^{n-2} diameter.

This paper is organized as follows. Section 2 explores the interconnection network and the types of constant-degree graphs. In section 3, we define interconnection network CG_n and analyze theoretical properties of the suggested graph and its degree, followed by presenting the routing algorithm and parallel path. Finally, we conclude our works in section 4.

Related works

The interconnection networks can be classified into the following three types based on the number of nodes: mesh type with $n(V(G)) = k \times n$, a hypercube type with $n(V(G)) = 2^n$, and a star graph type with $n(V(G)) = n!$[13-14]. The mesh type is a well-known graph structure that has been widely utilized as a planar graph, which has been commercialized in various systems[11,15,16].

Table 1. Fixed degree graph

Interconnection Network	Number of Nodes	Degree	Diameter	Network Cost
$Mesh$	$k \times n$	4	$2\sqrt{n}$	$O(8\sqrt{n})$
$Honeycomb$	$6t^2$	3	$1.63\sqrt{n}$	$O(4.9\sqrt{n})$
$Torus$	$k \times n$	4	$\sqrt{n}$	$O(4\sqrt{n})$
SEP_n	$n!$	3	$\frac{1}{8}(9n^2 - 22n + 24)$	$O(\frac{27}{8}n^2)$
$NSEP_n$	$n!$	4	$\frac{2}{3}n^2 - \frac{3}{2}n + 1$	$O(\frac{8}{3}n^2)$

CG_n	2^n	3	2^{n-2}	$O(3 \times 2^{n-2})$

The m-dimensional mesh $M_m(N)$ consists of $n(V(G)) = N^m$ and $n\big(E(G)\big) = mN^m - mN^{m-1}$. The address of each node is expressed with an m-dimensional vector, and when the addresses of any two nodes differ by 1 in one dimension, there is an edge between them[17]. The advantage of Low-dimensional meshes is that they are straightforward to design, it is considered as a promising scheme that has been widely implemented to specifically design the network topology for parallel processing computers. Higher-dimensional meshes have smaller diameters and larger bipartite widths. In addition, they tend to perform the parallel algorithms with rapid pace, but requires a high cost[15,18,19]. Hexagonal Mesh[20], Toroidal Mesh, Diagonal Mesh[21], Honeycomb Mesh[22], Hierarchical Petersen[16], Torus are proposed as an alternative structure that shows to have improved general lattice-structured mesh diameter[19,23]. Furthermore, the Shuffle-Exchange Permutation (SEP) was asserted based on permutation groups, which has the advantage of easy conductivity of graph-based simulation (*e.g.*, Cayley graph)[24]. SEP enables to efficiently operate the algorithm with minimal transitions in newly given graphs, and SEP also is a regular network with a degree of three. Moreover, an NSEP (New-SEP) with a degree of four with enhanced SEP diameter and network cost has been proposed[25].

CG_n graph design and analysis

The Graph is formulated based on the Vertex (Node) and Edge. Let node of graph CG_n is S, where the node is expressed with binary n-bit, and the total number of node $n(\bigcup_{\forall i} S_i) = 2^n$. The address of the node S in the CG_n graph is as follows.

$$S = (S_n\, S_{n-1}\, S_{n-2}\, \ldots S_i \ldots\, S_3\, S_2\, S_1),\ 1 \le\ i\ \le\ n, i \in \mathbb{N} \tag{1}$$

Definition 1. *The three edge types connecting to node S constituting the n-dimensional CG_n graph are as follows.*

The symbol % means the remaining operators

Increasing Edge (E_f): Connect a node that is 1 bit higher from the address of the node S.

$$\{(S\ +\ 1) + 2^n\}\,\%\ 2^n \tag{2}$$

Decreasing Edge (E_b): Connect a node that is 1 bit lesser from the address of the node S.

$$\{(S\ -\ 1) + 2^n\}\,\%\ 2^n \tag{3}$$

Complement Edge (E_c): Connect a node that is a complement node of the most significant bit S_n from the address of the node S.

$$(S\ +\ 2^{n-1})\,\%\ 2^n \tag{4}$$

Example 1. *In the n-dimensional CG_n graph, when $n = 4$, the connection relationship of nodes is as follows.*

In G, $n(V(G))$= 2^4 with $E(v_{\forall i}) = 3$. If the node S is 1 (=$0001_{(2)}$) where $x_{(2)}$ denotes the binary with $x \in \{0,1\}$, we substitute a decimal value according to the graph definition, which leads E_f to direct 2 (=$0010_{(2)}$) and E_b to 0 (= $0000_{(2)}$). The final branch (E_c) points to 9 (=$1001_{(2)}$) in which only the leftmost bit is complemented, and this operation is applied equally to all nodes, and the graph of 2^4 has the form as shown in Figure 1.

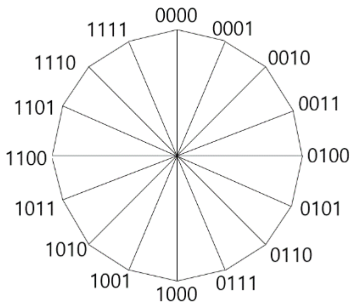

Figure 1. CG_4 graph

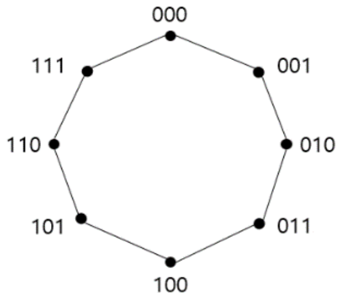

Figure 2. H_3 cycle

Definition 2. *In CG_n, a cycle of length 2^n consisting only of increasing or decreasing edges is nominated as H_n cycle.*

Example 2. When $n = 3$, H_3 cycle is a subgraph that the length is 8 (Fig. 2).

A cycle that includes all vertices in the connected graph G is called a Hamiltonian cycle. If the network has a Hamilton pass or a Hamilton cycle, a ring or a linear array can be easily implemented, thus it is used as a pipeline for parallel processing. Theorem 1 shows that the internal CG_n network has a Hamiltonian cycle[26-27].

Theorem 1. *CG_n contains Hamilton cycle.*

Proof of Theorem 1. Let S be an arbitrary node. By definition 2, there is always a cycle starting from S and returning to S by repeating increment or decrement operations. Therefore, CG_n has a Hamiltonian cycle. □

Theorem 2. *CG_n is a symmetric graph.*

Proof of Theorem 2. A symmetrical relationship proves a one-to-one relationship between adjacent nodes of each existing node when mapping two arbitrary nodes. Let U be an arbitrary departure node and V be a destination node $(0 \le U, V \le 2^n - 1, U \ne V)$. By definition 2, arbitrary two nodes of CG_n can be reached via increasing or decreasing edges. Therefore, U and V have the following relationship.

$$(S + 2^{n-1}) \% 2^n \tag{5}$$

The adjacent nodes of node U and V are $\{U+1, U-1, (U+2^n)\%2^n\}$, $\{V+1, V-1, (V+2^n)\%2^n\}$ respectively. When mapping U to V, the neighbor node is $\{(U+\alpha)+1, (U+\alpha)-1, ((U+\alpha)+2^n)\%2^n\}$, and for $U+\alpha=V$, we conclude that $\{V+1, V-1, (V+2^n)\%2^n\}$. Therefore CG_n is a symmetric graph. □

Routing Algorithm

Routing is the process of efficiently and quickly determining the route of two points among one or multiple networks. We can obtain an effective solution to Routing problems in diverse domains via network techniques. A path is a route from one node to another through the edge of the graph. A practical routing algorithm should a path that is optimal and simple[28]. We introduce the routing algorithm and parallel path of the CG_n. Moreover, we prove the shortest path through the routing algorithm and parallel path algorithm.

Definition 3. *$D's$ location represents the set of destination nodes within the range when the starting node u is 0.*

Our ultimate objective is to compute the optimal length, and the preliminary step is to determine the range which the destination node is located. Our approach is to divide the graph into two sections, and in the corresponding section, we compute the optimal length. We have set u as an initial starting node, and we subdivided the section range of destination node v into Section A and Section B. Note that this study only divides and analyzes the right side of the graph while the routing algorithm is being processed.

$$\text{Section A: } 0 < D's\ location < 2^{n-2}+1 \tag{6}$$

$$\text{Section B: } 2^{n-2} < D's\ location < 2^{n-1}+1 \tag{7}$$

Example 3. *In a graph where n is 3, the nodes in parentheses (001, 111), (010, 110), (011, 101) have the identical number of edges used to arrive at the optimal length from the start node to the destination node when the initial node is 0.*

(001, 111): $n(E(001,111)) = 1$, (010, 110): $n(E(010,110)) = 2$, (011, 101): $n(E(011,101)) = 3$

The main reason for this is that the graph is symmetric by definition. That is, when obtaining the optimal length with (010) as the target node, the length of the edges leading to (110) enables the successful path.

Theorem 3. *Let D be an arbitrary destination node of CG_n. The optimal length can be obtained through D' that satisfies $D + D' = 2^n$.*

Proof of Theorem 3. Let D be an arbitrary destination node of CG_n. With the formula: $D' = 2^n - D$, a node D' is being used as the destination node. The range is divided into two, and the operation selected first differs depending on the section in Sections A and B. Suppose the node belonging to Section A is the destination node. In that case, the increment operation is optimal, and if the node belonging to Section B is the destination node, the decrement operation is preferred. We verify this assertion through the routing algorithm. □

Example 4. *When $n = 4$, Section A, B is defined as follows.*

Section A: Since it is a set of nodes greater than 0 and less than $2^{n-2} + 1$, it has a value of {1, 2, 3, 4} in decimal.

Section B: Since it is a set of vertices greater than 2^{n-2} and less than $2^{n-2} + 1$, it has a value of {5, 6, 7, 8} in decimal.

Definition 4. The terms used in the routing algorithm are defined as follows.

Table 2. Notation definition

Notation	Definition
S	Starting node (Initial node)
D	Destination node
Goal	Destination node
RN	Current node
FD, FD_s	Verify whether functions move_f() and move_b() were previously applied
D'	$2^n - D$ $(D > 2^{n-1})$
move_f(a)	Apply edge E_f, Shift status by $\left((a+1) + 2^n\right)\%2^n$
move_b(a)	Apply edge E_b, Shift status by $\left((a-1) + 2^n\right)\%2^n$
move_c(a)	Apply edge E_c, Shift status by $(a + 2^{n-1})\%2^n$

Example 5. *When $n = 5$, if the initial node is 0 and the destination node is 1, the path is as follows.*

Section A = {1, 2, 3, 4, 5, 6, 7, 8}

Section B = {9, 10, 11, 12, 13, 14, 15, 16}

Increment operation is conducted until it reaches the destination node since the destination node is an element of Section A.

$$0 \rightarrow 1$$

Example 6. *When* $n = 5$, *if the initial node is 30 and the destination node is 15, the path is as follows.*

Since destination node 30 is larger than 2^5, it utilizes the value of D'.

$$D' = 2^5 - 30 = 32 - 30 = 2$$

Since D' is affiliated in Section A, it conducts decreasing operation until it reaches the destination node.

$$3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow 32 \rightarrow 31 \rightarrow 30$$

Example 7. *When* $n = 5$, *if the initial node is 30 and the destination node is 15, the path is as follows.*

Since destination node 15 is smaller than 2^n and affiliated in Section B, it conducts the incremental operation after the complement operation.

$$30 \rightarrow 14 \rightarrow 15$$

Algorithm 1: Routing Algorithm

```
Input: Starting node S, Destination node D
Output: int RN
Node move_f(node RN)
    return ((RN + 1) + 2^n)%2^n
Node move_b(node RN)
    return ((RN − 1) + 2^n)%2^n
Node move_f(node RN)
    return (RN + 2^(n−1))%2^n
Initialize int RN ← S, int (FD, FD_s) ← 0, int goal ← D
if (D < 2^(n−1)) then
 | FD ← 1
else
 | FD_s ← 1, D' ← 2^n − D
 | D ← D'
if D ∈ Section A then
 | if FD > FD then
 | | RN ← move_f(RN)
 | | while (True) do
 | | | if RN = goal then break
 | else
 | | RN ← move_b(RN)
 | | while (True)
 | | | if RN = goal then
 | | | | break
 | | RN ← move_b(RN)
```

```
else if D ∈ Section B then
    RN ← move_c(RN)
    if FD > FD_s then
        while (True) do
            if RN = goal then
                break
            RN ← move_f(RN)
    else
        while (True) do
            if RN = goal then
                break
            RN ← move_b(RN)
```

Parallel path

Node disjoint paths between any two nodes in a network are critical to transmitting fast-forward large amounts of data or to provide alternative paths when a node fails. There are three parallel paths within the CG_n graph from the start node to the target node, and the pseudocode of the Parallel Path algorithm is as follows.

CG_n is a graph with a degree number of 3, and there can be three parallel paths. From the starting node to the target node, there is a path using only Front() operations, a path using only Back() operations, and finally, a path using Cross() operations first and then Front() or Back() operations. In the routing algorithm, the Front() and Back() operations continued to increase or decrease until reaching the target node. However, in the parallel path, when increasing or decreasing, it continuously checks through repair operations to verify whether it is a target node and goes through the process of returning.

Theorem 4. *The CG_n has three parallel paths.*

Proof of Theorem 4. CG_n is analyzed only when it is less than 2^{n-1} according to definition 3. If the target node is Section A and Section B, it is divided into two categories.

① Front() priority: Since the target node is less than 2^{n-1}, it is terminated before reaching 2^{n-1} through an increase operation.

② Back() priority: Since the target node is less than 2^{n-1}, the target node is reached through a repair operation before reaching 2^{n-1} while decreasing.

③ Cross() priority: After moving to 2^{n-1}, since the target node is smaller than 2^{n-1}, the target node is reached through a reduction operation.

Therefore, the three parallel paths of CG_n are not duplicated. □

ISSN: 2094-0343
2326-9865

Algorithm 2: Parallel Path Algorithm

Input: Starting node S, Destination node D
Output: int RN

```
int RN_f, RN_b, RN_c;
int RT_f = 0, RT_b = 0, RT_c = 0;
int Flag_F = 0, Flag_b = 0;

(Front priority Path)
RN_f ← S
while(1)
    RN_f ← move_f(RN_f )
    RT_f++
    if (RN_f == d) then break
    else
        RN_ ← move_c(RN_f)
        if (RN_f == d)
            RN_f++
            Flag_F ← 1
            break
        else
            RN_f ← move_c(RN_f)

(Back priority Path)
RN_b ← S
while (1)
    RN_b ← move_b(RN_b)
    RT_b++
    if (RN_b == d) then break
    else
        RN_b ← move_c(RN_b)
        if (RN_b == d)
            RT_b++
            Flag_B ← 1
            break
        else
            RN_b ← move_c(RN_b)

(Cross priority Path)
RN_c  ← S
RN_c  ← move_c(RN_c)
RT_c++
if (Flag_F > Flag_B)
    while (1)
```

ISSN: 2094-0343
2326-9865

```
    | | if (RN_c == d) then break
    | | else
    | | | RN_c ← move_f(RN_c)
    | | | RT_c++
else
    | while (1)
    | | if (RN_c == d) then break
    | | else
    | | | RN_c ← move_b(RN_c)
    | | | RT_c++
```

When n=5, and when the start node is 0 and the target node is 1, the front priority path is as follows.

$$0 \rightarrow 1$$

When n=5, the start node 0 and the target node 1 are as follows. It decreases and reaches the target node while checking the node through E_C in the middle.

$$(31 \rightarrow 15 \rightarrow 31) \rightarrow (30 \rightarrow 14 \rightarrow 30) \rightarrow (29 \rightarrow 13 \rightarrow 29) \rightarrow (28 \rightarrow 12 \rightarrow 28)$$

$$\rightarrow (27 \rightarrow 11 \rightarrow 27) \rightarrow \cdots \rightarrow (18 \rightarrow 2 \rightarrow 18) \rightarrow 17 \rightarrow 1$$

When n=5, when the start node 0 and the target node 1 are 1, the Cross priority path is as follows. After first executing the edge E_C, an increase operation is performed when the target node is greater than 2^n, and a decrease operation is performed when the target node is smaller.

$$0 \rightarrow 16 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow \cdots \rightarrow 3 \rightarrow 2 \rightarrow 1$$

Optimal Length

In this section, we mathematically prove the formula that computes the Optimal Routing Path Length. In a graph with n bits, the optimal length from the single starting node S to the destination node D when the section is divided into two subparts according to the routing algorithm is as follows.

Section A: $|S - (D \ or \ D')|$

Section B: $|C(S) - (D \ or \ D')|$

In Fig. 1, we assume that the starting node S is 0 and the destination node D is 15. According to the formula $D' = 2^n - D$, D' becomes 1, and it is affiliated in Section A. Therefore, substituting (0-1), the absolute value is 1, which leads the optimal length also being 1. Assuming that the start node S is 0 and the target node 1 is 1, and the number of bits n is up to 21 (number of nodes: 2^{21}), the time for n for all cases is shown in the following graph.

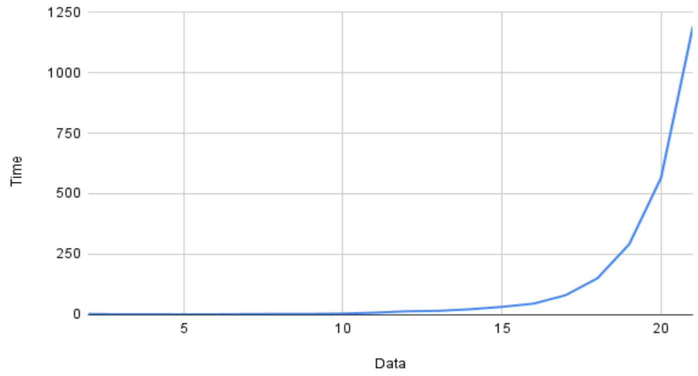

Figure 3. The value of time for n

As a result of implementing which is a parallel path algorithm in C language, it was found that the optimal length obtained from the routing algorithm and the length of the edge of the optimal path obtained through Parallel Path match. Through this, we derived the result that the optimal length exists among the three non-overlapping paths of the Parallel Path.

As a result of implementing a parallel path algorithm through programming using C language, it was found that the optimal length obtained from the routing algorithm matched the length of the edge of the optimal path obtained through Parallel Path. Through this, we derive the result that the optimal length exists among the three non-overlapping paths of the Parallel Path.

Theorem 5. *The diameter of CG_n is 2^{n-2}.*

Proof of Theorem 5. According to Theorem 3, the distance is equivalent when the initial node is 0, and the destination node is 2^n. Note that the case of less than 2^n was divided into Section A and Section B. Since there is an edge E_c that connects the 2^{n-1} nodes with the farthest distance on the cycle H_n, the node with the farthest distance from node 0 is 2^{n-2} and 2^{n-1} in Section B. According to the algorithm, the 2^{n-2} applies the edge E_f and 2^{n-1} implements the edge E_b, followed by utilizing E_b to reach the destination node. In this case, the distance to 2^{n-2} and 2^{n-1} is equivalent to 2^{n-2}. Therefore, the diameter of CG_n is 2^{n-2}.
□

Example 8. *When $n = 5$, set the initial node as 0, and the destination node as 7.*

Since destination node 7 is associated with Section A, only increment operation is performed to reach the destination node.

$$0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7$$

Therefore, the diameter of CG_5 is 8.

Graph cycle properties

According to the graph definition, it is obvious that the target node becomes the starting node by moving from the starting node to the edge E_f (or E_b) 2^n times. Therefore, CG_n has a

Hamiltonian cycle. To describe this, we analyzed the length of various cycles present in CG_n and their relationship .

Property 1. *if the cycle existing in CG_n is $Cycle_n$ and the length of $Cycle_n$ is m_n, it is* $4 \leq m_n \leq 2^n$. *(However, when $m_n < 2^{n-1}$, there is no odd-length cycle.)*

Proof of property 1.

case1) m_n = even:

When starting node moves α times to the edge E_f, and starting node move once to the edge E_c, a cycle with a length of 2 α+2 that symmetrically moves the edge E_b α times and moves to the edge E_c to return to the starting node. Since the range of α is $1 \leq \alpha \leq 2^{n-1} - 1$, $Cycle_n$ where m_n is an even number is 4, 6, 8, ⋯, 2^{n-2}, 2^{n-1}, 2^n.

case2) m_n = odd:

2-1) $m_n < 2^{n-1}$

We can make the shortest cycle when m_n = 4. This is because the path of moving to the edge E_f (or E_b) and again moving to the edge E_c is the path that has a shortest distance with a symmetry. When $m_n < 2^{n-1}$, the cycle cannot be generated without symmetry. That is, when m_n is less than 2^{n-1}, there is no odd-length cycle.

2-2) $m_n = 2^{n-1} + 1$

The odd-length cycle of $Cycle_n$ exists from the cycle with m_n = $2^{n-1} + 1$. A cycle in which m_n = $2^{n-1} + 1$ is moved to the edge E_c and then again moved to the edge E_f (or E_b) 2^{n-1} times.

2-3) $m_n > 2^{n-1} + 1$

Move once to edge E_c and then $\alpha(1 < \alpha < 2^{n-1} - 1)$ times to edge E_f (or E_b). Move back to edge E_c once and then to edge E_f (or E_b) $\beta(1 < \beta < 2^{n-1} - \alpha)$ times. Move to the edge E_c once again and then to the edge E_f (or E_b) $\gamma(\gamma = 2^{n-1} - \alpha - \beta)$ times. In this way, edge E_c is used even times for symmetry except for the first time, and it can move up to 2^{n-1}-2 times, so it can move a total of $2^{n-1} - 1$. The total length of the cycle is $2^{n-1} - 1 + \alpha + \beta + \gamma$, and since it is α+β+γ=2^{n-1}, it is odd as $2^{n-1} - 1 + 2^{n-1}$, or $2(2^{n-1}) + 1$.

Thus, m_n of the cycle $Cycle_n$ present in CGn is $4 \leq m_n \leq 2^n$. □

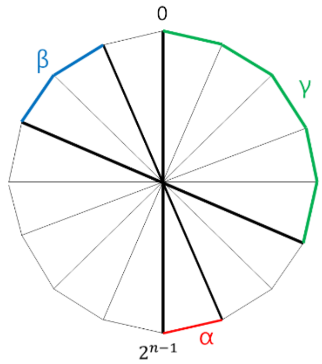

Figure 4. $Cycle_n$ with odd m_n

when n=4, $Cycle_4$ is as follows.

$Cycle_4$ has a cycle of m_4 = {4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16}. Depending on property 1, the even-length cycle is m_4={4, 6, 8, 10, 12, 14, 16} and the odd-length cycle is m_4={9, 11, 13, 15}. In Figure 5, the case of m_4={4, 9, 11} is confirmed sequentially.

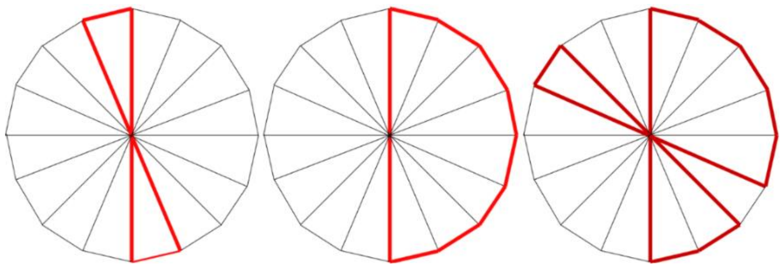

Figure 5. $Cycle_4$ with m_4 = {4, 9, 11}

Algorithm 3: $Cycle_n$ Algorithm

Input: Starting node S, Destination node D
Output: int CN
int CN, E_c, E_f, E_b, E, E'

CN ← S
if (CN % 2 == 0)
  X ← (CN) / 2 - 1
  E_c
  for (i = 1; i ≤ X; i++)
    E_f
  E_c
  for (i = 1; i ≤ X; i++)
    E_b
else
  for (i = 1; i ≤ CN; i++)
    E

```
        E_f
    for ( i = 1; i ≤ 2^n −CN; i++)
        E_f
if (CN % 2 == 1)
    for (i = 1; i ≤ CN −2^(n−1); i++)
        E'
        E_f
    for ( i = 1; i ≤ 2 × 2^(n−1) −CN; i++)
        E_f
```

Conclusions

In this paper, we proposed an interconnection network structure with a constant degree of three and analyzed its theoretical property. We defined the novel three-constant degree graph CG_n , also proposed the shortest path routing algorithm in propounded CG_n . The n-dimensional CG_n proposed in this paper represents a node address with n bits of a binary number, with a numerical degree of three, the 2^n number of nodes, and a diameter of 2^{n-2}. We show that the Hamilton cycle exists in internal CG_n , also has a maximum fault-tolerance. This novel graph topology and its property offer the viability of future implementation of a parallel processor system to enable high-performance computing operations, as well as providing insights and a reference to be applied in a real-world network system.

Acknowledgment

This research was funded by National Research Foundation of KOREA(NRF) grant funded by the Korea government(MSIT) (No. 2020R1A2C1012363).

MIST: Ministry of Science and ICT

References

[1] C.H. Im, “Supercomputer architecture trend and applied technology analysis,” Journal of Korean Institute of Next Generation Computing, Vol. 18, no. 3, pp. 62-74, 2022.

[2] S. J. Ezell and R. D. Atkinson, “The Vital Importance of High-Performance Computing to U.S. Competitiveness,” ITIF, April, 2016

[3] P-J. Lu, M-C. Lai, and J-S. Chang, “A Survey of High-Performance Interconnection Networks in High-Performance Computer Systems,” electronics, Vol. 11(9), 2022.

[4] Z. Cao, X. An, X. Liu, Y. Su, and Z. Wang, “Status and Prospect of High Performance Computer Interconnection Network,” Inf. Technol. Lett, Vol. 10, pp. 12–28, 2012.

[5] G. Bell, Ultracomputers: A teraflop before its time, Communications of the ACM, 1992, pp. 27-47.

[6] D. Culler, J. Singh, A. Gupta, Parallel computer architecture-a hardware/software approach, 1999.

[7] Parhami. B., Rakov. M, “Perfect difference networks and related interconnection structures for parallel and distributed systems,” IEEE transactions on parallel and distributed systems, Vol. 16(8), pp. 714-724. 2005.

[8] Gao, J. Lu, H. He, W. Ren, X. Chen, S. Si, T. Zhou, Z. Hu, S. Yu, K. Wei, D. “The Interconnection Network and Message Machinasim of Sunway Exascale Prototype System,” Chin. J. Comput. Vol. 44, pp. 222–234, 2021.

[9] R. Trobec, R. Vasiljevic, M. Tomasevic, V. Milutinovic, R. Beivide, M. Valero, “Interconnection Networks in Petascale Computer Systems: A Survey,” ACM Comput. Surv. Vol. 49, pp. 1–24, 2016.

[10] Robin, J. W., John, J. W. Graphs an introductory approach, Sons, USA, 1990

[11] E.A.Monakhova, “A survey on undirected circulant graphs, Discrete mathematics, Algorithms and Applications,” vol. 4, No. 1, 2012.

[12] W. Dally, “Performance Analysis of k-Ary n-Cube Interconnection Networks,” IEEE Trans. Computers, Vol. 39, No. 6, pp. 775-785, 1990.

[13] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays Trees and Hypercubes. 1992.

[14] I. Navid, S-A. Hamid, and G.A. Selim, “Some topological properties of star graphs: The surface area and volume,” Discrete Mathematics, Vol. 309, No. 3, pp. 560-569, 2009.

[15] Leighton. F. T. Introduction to parallel algorithms and architectures: arrays, hypercube, Morgan Kaufmann Publishers, 1992.

[16] J-H. Seo, H-O. Lee, “Design and analysis of rotational binary graphs as an alternative to hypercube and torus,” The Journal of supercomputing, Vol. 76, No. 9, pp. 7161-7176, 2020.

[17] V. Bokka, H. Gurla, S. Olariu, and J. L. Schwing, “Podality-based time-optimal computations on enhanced meshes,” IEEE Trans. Parallel and distributed systems. Vol. 8, No. 10, pp. 1019-1035, 1997.

[18] Adve. V. S, Vernon. M. K, “Performance analysis of mesh interconnection networks with deterministic routing,” IEEE Transactions on Parallel and Distributed Systems, Vol. 5(3), pp. 225-246, 1994.

[19] Bokka. V. H, Gurla. S. Olariu and J. L. Schwing, “Podality-based time-optimal computations on enhanced meshes,” IEEE Trans, Parallel and Distributed System. Vol. 8(10), pp. 1019-1035, 1997.

[20] M.S. Chen, K.G. Shin and D.D. Kandlur, "Addressing Routing and Broadcasting in Hexagonal Mesh Multiprocessors," IEEE Trans. Computers, vol. 39, no. 1, pp. 10-18, Jan. 1990.

[21] K.W.Tang, S.A.Padubidri, “Diagonal and Toroidal mesh networks,” IEEE Trans. On Computers, Vol. 43, pp. 815-826, July, 1994.

[22] I. Stojmenovic, “Honeycomb Networks: Topological properties and communication algorithms,” IEEE Trans, on Parallel and Distributed Systems, Vol. 8, No. 10, pp. 1036-1042, 1997.

[23] Chen. J. L, Shin. K. G. “Addressing, routing, and broadcasting in hexagonal mesh multiprocessors,” IEEE trans. comput, vol. 39, no. 1, pp. 10-18, 1990.

[24] Latifi, S., Srimani, P. K. “A new fixed degree regular network for parallel processing,” IEEE Computer Society Press. pp. 152-159, 1996.

[25] B-O. Seong, H-O. Lee, J-S. Kim, and J-H. Seo, “A novel interconnection network with improved network cost through shuffle-exchange permutation graph,” Electronics, Vol. 10, no. 8, 2021.
[26] J-H. Seo, J-S. Kim. H-J. Chang, and H-O. Lee, “The Hierarchical Petersen Network: A New Interconnection Network with Fixed Degree,” The Journal of Supercomputing, Vol. 74, no. 14, pp. 1636-1654, 2018.
[27] Sharma, P., Khurana, N. “Study of optimal path finding techniques,” International Journal of Advancements in Technology, vol. 4, no. 2, 2013.
[28] Tolkien, J. R., Medhi, D., Ramasamy, K. Networking and network routing: an introduction. San Francisco: Elsevier. 2007.