I. J. Computer Network and Information Security, **2018, 8, 27-35**
Published Online August 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.08.03

Parallel Prefix Sum Algorithm on Optoelectronic Biswapped Network Hyper Hexa-cell

Ashish Gupta
Birla Institute of Technology, Mesra, Ranchi, Jharkhand India, 835215
E-mail: ashish.parj@gmail.com

Bikash Kanti Sarkar
Birla Institute of Technology, Mesra, Ranchi, Jharkhand India, 835215
E-mail: bk_sarkarbit@hotmail.com

Received: 11 April 2018; Accepted: 03 July 2018; Published: 08 August 2018

***Abstract*—**The biswapped network hyper hexa-cell is recently reported optoelectronic network architecture for delivering excellent performance especially for mapping numerical problems which demands frequent routing and broadcasting. This network contains some important benefits such as *smaller* diameter, higher bisection width, and lower network's total and optical cost as compared to counter-part OTIS hyper hexa-cell network. It is also advantageous as compared to the traditional biswapped network mesh containing smaller diameter and higher minimum node degree. In this paper, we present a parallel algorithm for mapping prefix sum of $2 \times \left(6 \times 2^{\mathrm{d_h}-1}\right)^2$ data elements on a $\mathrm{d_h}$-dimensional biswapped network hyper hexa-cell of$2 \times (6 \times 2^{d_h-1})^2$ processors (assuming each processor contain single data element). It demands total $\left((d_h - 1) \times (d_h + 1)\right) - \frac{(d_h - 1)\times(d_h - 2)}{2} + 5d_h + 10$ intra-cluster (electronic) and *3* inter-cluster (optical) moves.

***Index Terms*—**Optoelectronic, OTIS mesh, OTIS hyper hexa-cell, Biswapped network hyper hexa-cell.

I. Introduction

Recently, the biswapped networks [4] has emerged as a better alternative to the well known OTIS networks [2], since the former has symmetrical structure and comparably larger network size (double). Actually, the biswapped network is a 2-level hierarchal and symmetrical structure that takes any graph as modules and connects them in complete bipartite manner. Thus, a large network can be constructed from its underlying modules, groups or cluster networks. The biswapped networks are Cayley (di) graphs (node-symmetrical) that supports easy embeddings and emulations. Further, efficient routing and mapping of computational algorithms are the other advantages for being a Cayley graph. The biswapped networks' uniform structure eases selection of outgoing channel for shortest path routing (compared to OTIS networks) as it demands fewer parameter comparisons between source and the destination nodes (or processors) [4].

Although biswapped networks have such important and desirable benefits, however it contains some *trade-offs* too when compared to swapped or OTIS networks. For example, increment of a single unit in the network diameter. However, its large network size lowers the overall impact of such a *trade-off*.

The biswapped network hyper hexa-cell [6] is a recently reported variation of biswapped networks [4] especially for mapping numerical problems where frequent routing and broadcasting (among processors) is needed. The network claims some major advantages such as, *smaller diameter*; *higher bisection width*, *higher minimum node-degree* and *lower network total and optical cost* over its counterpart OTIS hyper hexa-cell [3] and traditional biswapped network mesh [5]. In this paper, we present parallel mapping of prefix sum problem of$2 \times (6 \times 2^{d_h-1})^2$ data elements on a $\mathrm{d_h}$-dimensional biswapped network hyper hex-cell (G=2N) of $2 \times (6 \times 2^{d_h-1})^2$ processors (assuming each processor contain single data element). Here G and N are representing total groups (or clusters) and cluster size.

The prefix is the most commonly used numerical problem in parallel processing**.** It is in fact; very useful in designing fast algorithms for polynomial interpolation and generalized divide difference for hermit interpolation**.** It also plays a major contribution in processing numerous problems such as sorting (such as counting sort)**,** carry look-ahead addition**,** the solution of linear recurrences**,** scheduling problems**,** simulating the parallel algorithms for the Parallel Random Access Memory (PRAM) etc**.**

The paper is structured as follows. In Section-II, the literatures review based on the prefix sum is discussed properly. In Section-III, we analyze the computational model (biswapped network hyper hexa-cell). In Section-IV, methodology for prefix problem is presented. Further, Section-V is dedicated to the parallel algorithm for performing prefix sum over $\mathrm{d_h}$-dimensional biswapped network hyper hexa-cell of $2 \times \left(6 \times 2^{\mathrm{d_h}-1}\right)^2$ data elements. Lastly in Section-VI, the suggested work is finally concluded.

II. Related Works

Basel A. Mahafzah [3] presented OTIS hyper hexa-cell network and performed comparison with traditional OTIS mesh. The comparison study shows that the OTIS hyper hexa-cell has advantage over OTIS mesh containing *smaller diameter*, *higher bisection width*, *higher minimum node degree* and *lower optical cost*. Wenjun Xiao [4] proposed a new class of 2-level symmetrical optoelectronic network named biswapped networks. Wei et al. [5] reported biswapped network mesh and its basic parallel algorithms. Gupta and Sarkar [6] presented recursive and symmetrical biswapped network hyper hexa-cell. The proposed network claimed advantages over OTIS hyper hexa-cell containing *smaller diameter*, *higher bisection width*, and *lower network cost*. In addition, this network has benefit of *smaller diameter* and *higher minimum node degree* compared to traditional biswapped network meshes. Wang and Sahni [1] presented two parallel algorithms for N-point prefix computation on an N-processor OTIS mesh network. The first algorithm demands $8\left(\mathrm{N}^{\frac{1}{4}} - 1\right)$ electronic and 2 optical moves, while second algorithm demands $7\left(\mathrm{N}^{\frac{1}{4}} - 1\right)$ electronic and 2 optical moves. Jana and Sinha [7] presented improved parallel algorithm for N-point prefix computation on an N-processor OTIS mesh demands $\left(5.5\mathrm{N}^{\frac{1}{4}} + 3\right)$ electronic and 2 OTIS moves. Mallik and Jana [8] presented prefix algorithm on mesh of trees and OTIS mesh of trees demands $13\log_2 n + O(1)$electronic moves and 2 optical moves. Lukas [9] presented parallel algorithm for prefix computation on SIMD model of OTIS K-Ary 3-cube parallel computers demands O(k) electronic O(1) OTIS moves on K^6 processors. Jana [10] presented improved parallel prefix algorithm on optical multi-trees requires O(logn) electronic and 4 optical moves. Jha presented [11] improved parallel prefix algorithm on n×n mesh that demands $3n + 2$ moves. Jha and Jana [12] presented prefix algorithm for N^2 processors multi mesh of trees needs $3.75\log_2 N$ + 6 time for N data points. Jana [13] presented prefix algorithm on N^2 elements on an n×n extended multi-mesh network takes O($N^{\frac{1}{4}}$) time on N processors (13 $N^{\frac{1}{4}}$ -5 communication steps and $\log N +$ 4) arithmetic steps. Rakesh and Nitin [14] presented prefix algorithm for N data values on N processors multi mesh of trees that demands $\mathrm{O}\left(N^{\frac{1}{4}}\right)$ intrablock moves. Datta, De and Sinha [15] presented prefix algorithm for N data elements on a Multi Mesh network of N processing elements demands $\mathrm{O}\left(N^{\frac{1}{4}}\right)$time for data communication and ($\log N^{\frac{1}{4}}$) time for computation. Gupta and Sarkar [16] presented prefix improved prefix algorithm over n×n BSN mesh in $13(n-1)$ electronic and 3 optical moves.

III. Computational Model

The present Section is dedicated to the detailed analysis of recently reported biswapped network hyper hexa-cell [6], when G=2N, where G and N represents total clusters (or groups) and cluster size respectively.

The Biswapped Network Hyper Hexa-cell (BSN HHC)

The biswapped network hyper hexa-cell [7] is a recently presented optoelectronic recursive and symmetrical optoelectronic network, where its each underlying identical cluster (d_h-dimensional HHC network) is composed of 2^{d_h-1} subclusters (each subcluster is a one-dimensional hyper hexa-cell network) connected among themselves in (d_h-1)-dimensional hypercube pattern. According to the biswapped connectivity rules, every cluster in biswapped network hyper hexa-cell would be connected among themselves in complete bipartite pattern [4]. The Size of each cluster can be computed by formula: N = 6×(2^{d_h-1}) [3].

Labeling of Processors

Each processor in d_h-dimensional biswapped network hyper hexa-cell is labeled by subgroup and processor number, where subgroup number is represented by d_h-1 bits, and processor number is represented by 3 bits. The processor numbers in each identical subcluster is labeled as 00, 01 and 10 in upper and lower triangle, starts from the top node, and then left to the right node. The left most bit for each processor (in each subgroup) is labeled as 0 for the upper triangle and 1 for the lower triangle. Further, processors of the upper triangle would be connected to their corresponding processor in lower triangle in such a way (as shown in Figures 1 and 2) that they form a subcluster (one-dimensional hyper hexa-cell network). For better understanding, the network architectures of 1, 2 and 3-dimensional hyper hexa-cell networks are shown in Figures 1, 2 and 3, respectively.

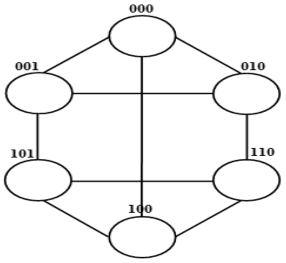

Fig.1. One-dimensional Hyper Hexa-cell Network

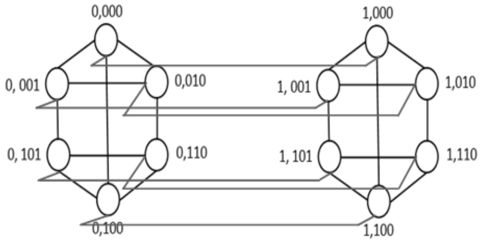

Fig.2. Two-dimensional Hyper Hexa-cell Network

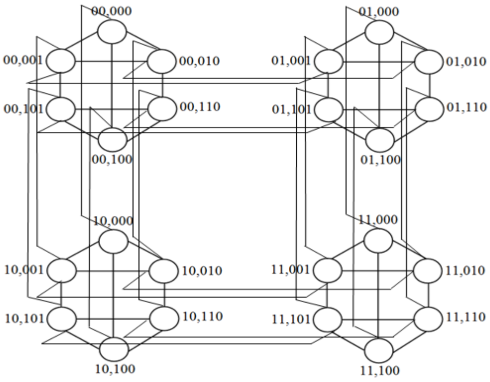

Fig.3. Three-dimensional Hyper Hexa-cell Network

Processor Connectivity: (Biswapped network hyper hexa-cell)

From architectural point of view, d_h-dimensional biswapped network hyper hexa-cell consists of 2 parts namely: part-0 and part-1. Each part contains N clusters and each cluster contains N processors. Therefore, total N^2 processors exist in each part, and $2N^2$ processors in the entire biswapped network hyper hexa-cell. Basically, the biswapped networks comprises of total 2N identical clusters made of its underlying N node basis or component networks. Here N is the size of each identical cluster, where N = 6 ×(2^{d_h-1}).

We assume P as the processor, and for routing purpose; we consider labeling of processors for each identical cluster by two parameters: c and p, where c represents cluster-id and p shows processor-id. Similarly, labeling of processors in biswapped network hyper hexa-cell contains three parameters: c**,** p and P', which represents the cluster-id**,** processor-id, and the part-id, respectively**.** For better understanding, the network architectures of hyper hexa-cell and biswapped network hyper hexa-cell (G=2N) are displayed in Figures 4 and 5. In Figure 5, blue lines are representing the inter-cluster (optical) connections. In particular, we show inter-cluster connections from cluster-0 of each part only.

Inter-cluster (optical) connections:

a $\forall c$ and $\forall p$, If cluster-id (c) does not match with the processor-id (p)**,** then it implies that the processor**:** $\mathrm{P}(c, p, 0)$ is connected to the processor**:** $\mathrm{P}(p, c, 1)$ and vice versa**.**

b $\forall c$ and $\forall p$, If cluster-id (c) matches with the processor-id (p)**,** then processor**:** P $(c, p, 0)$ is directly connected to the processor**:** $\mathrm{P}(c, p, 1)$ and vice versa**.**

/***It might be noted that no swapping is required between cluster-id and processor-id for case b*/.

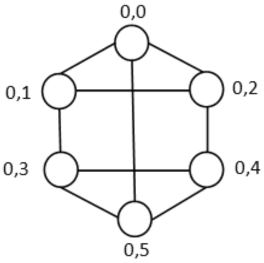

Fig.4. One-dimensional Hyper Hexa-cell Network (cluster-0)

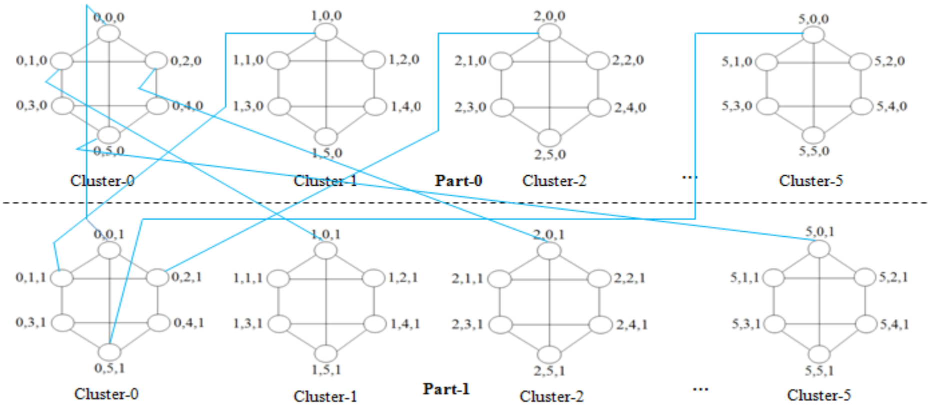

Fig.5. One-dimensional Biswapped Network Hyper Hexa-cell

IV. Methodology

For a given ordered set U of q elements: $U = x_o$, $x_1, x_2, \dots, x_{q-1}$ the resulting prefix would be the binary associative operation O applied as shown below in Eq. (1):

$$x_o, (x_0\ O\ x_1), (x_0\ O\ x_1 O\ x_2), \dots, (x_0\ O\ x_1 O\ x_2, \dots, x_{q-1}) \tag{1}$$

If the binary associative operation O is binary addition ⊕**,** then the resulting prefix sum is as follows. (as shown in Eq. (2)).

$$x_o, (x_0\ \oplus\ x_1), (x_0\ \oplus\ x_1 \oplus\ x_2), \dots, (x_0\ \oplus\ x_1\ \oplus x_2, \dots, x_{q-1}) \tag{2}$$

Assume, $x_o = 5,\ x_1 = 2, x_2 = 8, x_3 = 1\ and\ x_4 = 7.$

Then, the resulting prefix sum would be 5, 7, 15, 16 and 23.

V. Parallel Prefix Sum

This section is dedicated to the computation of prefix sum over d_h-dimensional biswapped network hyper hexa-cell. We first show in subsection A, the parallel algorithm for prefix sum for one-dimensional hyper hexa-cell (subcluster). Thereafter, the parallel algorithm for prefix sum for d_h-dimensional hyper hexa-cell (cluster) is presented in subsection B. Lastly, we perform prefix sum over d_h-dimensional biswapped network hyper hexa-cell in subsection C. Table 1 below shows nomenclatures and their significances.

Table 1. Nomenclatures and their Significances

Nomenclature	Significance
#c	Copy data from source to destination processor.
#b	Broadcast data from source processor.
#s	Store data in destination processor.
#i	Perform inter-part move, and store the result in destination processor.
$\#i^+$	Perform binary addition via inter-part move, and store the result in destination processor.

A. Parallel Prefix Sum for one-dimensional hyper hexa-cell (subcluster)

A d_h-dimensional hyper hexa-cell network (cluster) contains 2^{d_h-1} subclusters, where each subcluster contains six processors and is represented by d_h-*1* bits. For routing purpose, we assume labeling of each processor: P(s, p) by two parameters: s and p. The parameter: s shows subgroup number and p shows processor number, where$0 \leq s \leq 2^{d_h-1}$, $0 \leq p \leq 5$.

Assume, each processor has two registers: $X(s,p)$ and $Y(s,p)$. Initially, registers: $Y(s,p)$ are initialized with the data elements: x_{6s+p} , $0 \leq s \leq 2^{d_h-1}$, $0 \leq p \leq 5$ and registers: $X(s,p)$ are initialized with the value 0.

Parallel Algorithm:

Step 1. $\forall \boldsymbol{s}$, Perform prefix computation on registers: $Y(s,p)$ of upper and the lower triangle in parallel.

Step 2. $\forall \boldsymbol{s}$, $X(s,2) \xleftarrow{\#c} Y(s,2)$.

Step 3. $\forall \boldsymbol{s}$, Broadcast data element in upper triangle from register: $X(s,2)$.

Step 4. $\forall \boldsymbol{s}$, Perform the steps 4.1, 4.2 and 4.3 in parallel.

4.1. $X(s,3) \xleftarrow{\#s} X(s,1) \oplus X(s,3)$.

4.2. $X(s,4) \xleftarrow{\#s} X(s,2) \oplus X(s,4)$.

4.3. $X(s,5) \xleftarrow{\#s} X(s,0) \oplus X(s,5)$.

Step 5. $\forall \boldsymbol{s}$ and $\forall p$, $Y(s,p) \xleftarrow{\#s} X(s,p) \oplus Y(s,p)$.

//The register: $Y(s,p)$ yields the final result as prefix sum for one-dimensional hyper hexa-cell network.//

For better understanding, final result as prefix sum for subcuster-0 is shown in Figure 6. Here, for instance, base 0-5 representing prefix sum of six data elements ranging from 0 to 5.

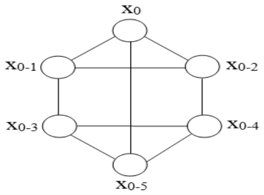

Fig.6. The Content of Registers: Y(s, p) for Subcluster 0

Time Complexity Analysis

In performing prefix sum for a one-dimensional hyper hexa-cell, Step1 demand 2 electronic moves**.** Further, Step 3 demands 2 electronic moves for broadcasting in the upper triangle of one-dimensional hyper hexa-cell network. Step 2, 4 and 5 need constant moves**.** Therefore**,** total 4 electronic moves are necessary to perform parallel prefix over one-dimensional hyper hexa-cell network**.**

B. Parallel Prefix Sum for dh-dimensional hyper hexa-cell ($d_h > 1$)

The computation of prefix sum for a d_h-dimensional hyper hexa-cell network requires d_h stages ($d_h > 1$). For example, a three-dimensional hyper hexa-cell requires 3 stages for computing parallel prefix sum.

Parallel Algorithm:

Stage1. Perform prefix computation (applying the algorithm presented in Subsection A) on each of the 2^{d_h-1} partitions in parallel, where each partition is one-dimensional hyper hexa-cell network**.**

Stage2. Compute prefix sum on each of the 2^{d_h-2} partitions in parallel, where each partition is two-dimensional hyper hexa-cell network.

/* *Inter-sub cluster communication flow for the computation of prefix sum in Stage 2 of d_h-dimensional hyper hexa-cell network is displayed in Figure 7* */.

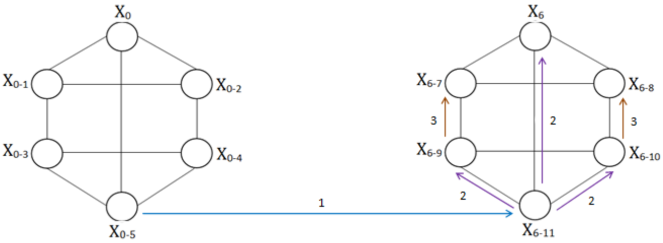

Fig.7. Inter-subcluster Communication moves for Prefix Computation in stage 2 in Dh-dimensional Hyper Hexa-cell.

Stage3. Perform prefix computation in parallel over each of the 2^{d_h-3} partitions in parallel, where each partition is three-dimensional hyper hexa-cell network.

/ Inter-subcluster communication flow for the computation of prefix sum of Stage 3 of d_h-dimensional hyper hexa-cell network is displayed in Figure 8*/.*

… … ….. … ..
… … ….. … ..

Stage d_h. Find prefix computation over entire d_h-dimensional hyper hexa-cell network**.**

Time Complexity Analysis

At the lowest level (Stage-1)**,** 4 electronic steps are required for prefix computation in each partition (each partition is a one-dimensional hyper hexa-cell network) according to the algorithm for prefix computation in Subsection A. Note that, after stage-1, each stage demands $d_h + 1$ electronic moves for computing prefix. Such as in Stage-2**,** parallel prefix computation over each partition (each partition is a two-dimensional hyper hexa-cell network) of d_h-dimensional hyper hexa-cell demands 3 electronic moves**.** This process continues further up to stage-d_h. At final stage (stage-d_h), $d_h + 1$ electronic moves will be required. The total number of electronic moves required for performing parallel prefix after d_h stages are shown below in Eq. (3)**.**

$$\text{Parallel Prefix Sum} = (d_h + 1) + ((d_h - 1) + 1) + ((d_h - 2) + 1) + , \dots , + 4 \qquad (3)$$

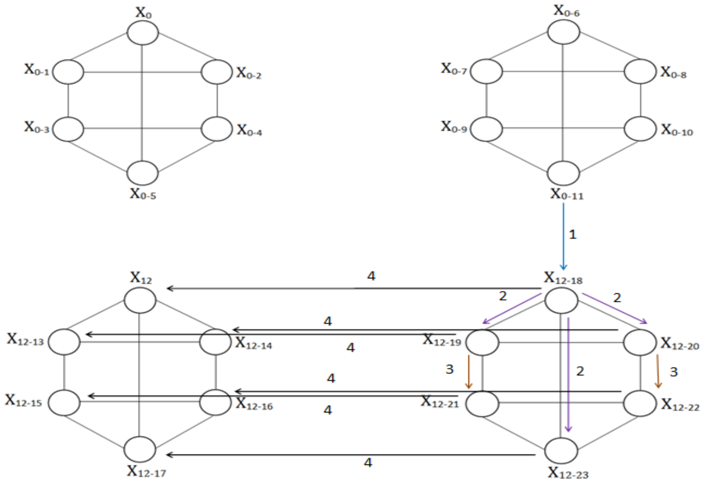

Fig.8. Inter-subcluster Communication moves for Prefix Computation in stage 3 in d_h-dimensional Hyper Hexa-cell.

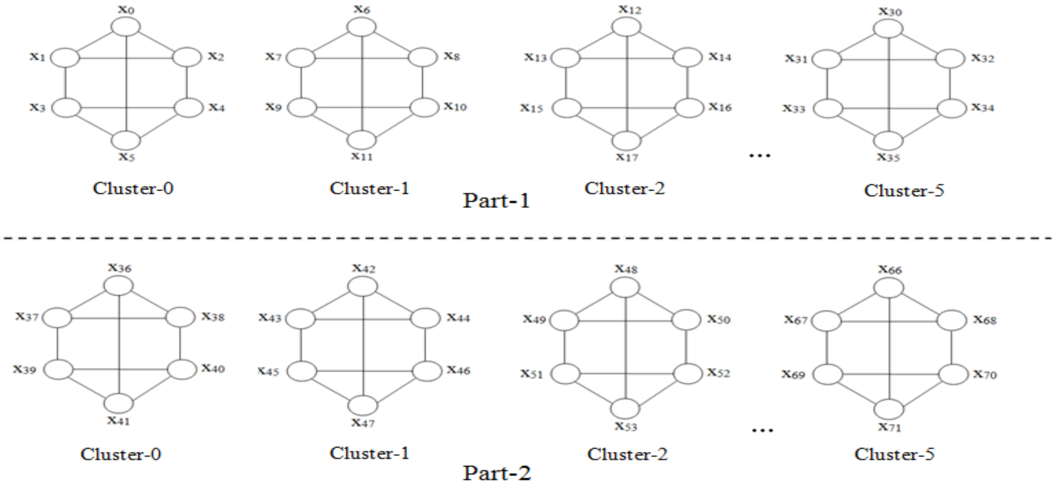

Fig.9. Initialization of Data Elements in Register: Y(c, p, P')

Hence**,** the required communication moves (electronic) for prefix computation over the d_h-dimensional hyper hexa-cell is as follows (as shown below in Eq. (4)).

$$\text{Parallel Prefix} = ((d_h - 1)(d_h + 1)) - \frac{(d_h - 1)(d_h - 2)}{2} + 4 \qquad (4)$$

C. Parallel Prefix Sum for dh-dimensional Biswapped network hyper hexa-cell

Assume**,** each processor has three registers namely: $X(c, p, P'),\ Y(c, p, P')\ and\ Z(c, p, P')$**.**

∀c, ∀p and ∀P’**,** registers: $Y(c,p,P')$ are initialized with the data element: $x_{p+cN+P'N^2}$, and registers**:** $Z(c,p,P')$ are initialized with the value 0**.** The initialization of $Y(c,p,P')$ registers is displayed in Figure. 9.

Parallel Algorithm:

Step 1. Perform parallel prefix sum on initial data elements stored in $Y(c,p,P')$ registers for each cluster (d_h-dimensional hyper hexa-cell network) applying the prefix sum algorithm presented in Subsection B. For better understanding, output of prefix sum for each cluster is displayed in Figure 10.

Step 2. $\forall P'$ and $\forall c$, $X(c,\ N-1,\ P') \xleftarrow{\#c} Y(c,\ N-1,\ P')$.

Step 3. Perform parallel data broadcast in each cluster. //Call Subroutine *C.1* for parallel data broadcast for each cluster// $\forall P'$ and $\forall c$, $\xrightarrow{\#b}$ $X(c,\ N-1,\ P')$.

Step 4. Perform the following inter-cluster moves in parallel for each of the Steps-4.1 and 4.2.

4.1. $\forall c$, $Z(c+k,\ c,\ 1) \xleftarrow{\#i} X(c,\ c+k,\ 0)$, $0 \le k \le (N-1)$-c.

4.2. $\forall c$, $Z(c+k,\ c,\ 0) \xleftarrow{\#i} X(c,\ c+K,\ 1)$, $0 \le k \le (N-1)$-c.

//The data elements of registers: $Z(c,\ p,\ P')$ after Step 4 is displayed in Figure 11.//

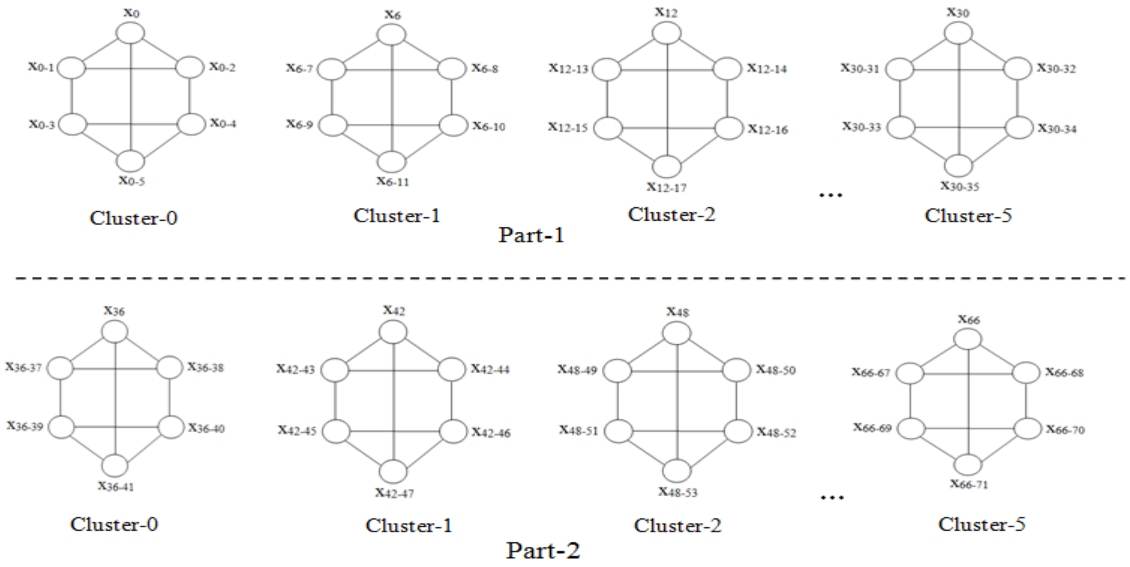

Fig.10. The data Elements as Prefix sum for each Cluster Stored in Y(c, p, P’) Registers after Step-1

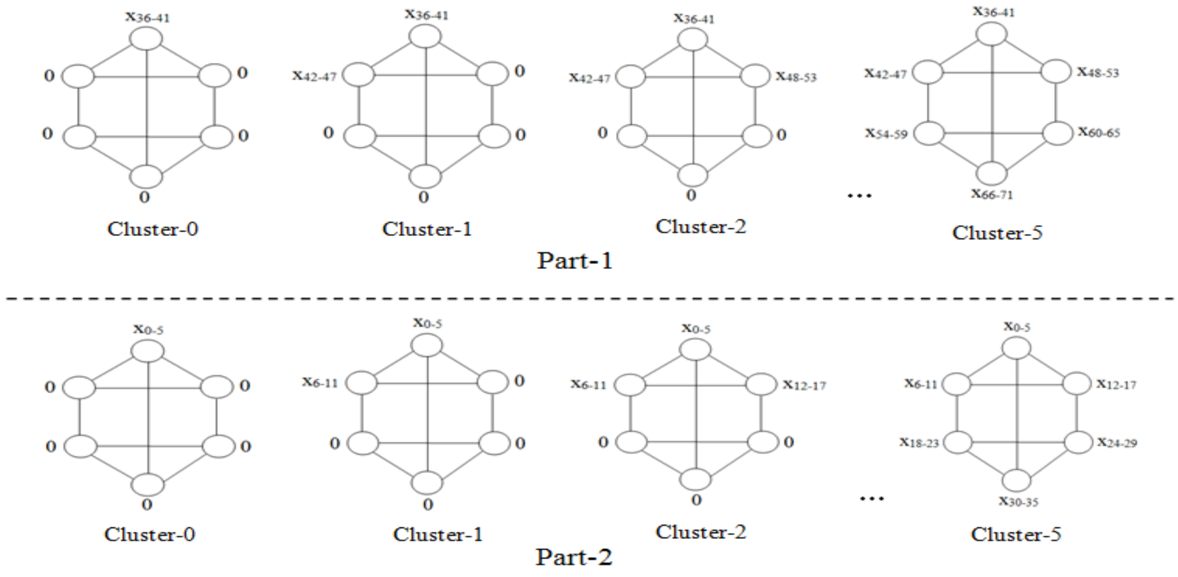

Fig.11. The data Elements Stored in Register: Z(c, p, P’) after Step-4

Step 5. $\forall P'$, Perform parallel data sum on each cluster and store the result in registers: $Z(N-1,\ N-1,\ P')$. //Call Subroutine *C.2* for parallel data sum for each cluster//

Step 6. $\xrightarrow{\#b} Z(N-1,\ N-1,\ 1)$.

//Call Subroutine *C.1* for parallel data broadcast//

Step 7. $\forall p, Z(p,\ N-1,\ 0) \xleftarrow{\#i^{+}} Z(N-1,\ p,\ 0)$.

Step 8. $\forall P'$ and $\forall c, \xrightarrow{\#b} Z(c,\ N-1,\ P')$.
//Call Subroutine *C.1* for parallel data broadcast//

Step 9. Perform the following inter-cluster moves in parallel for each of the Steps-9.1 and 9.2, where c<N-1.

9.1. $\forall c, X(c+1,\ c,\ 1) \xleftarrow{\#i} Z(c,\ c+1,\ 0)$, c<N-1.

9.2. $\forall c, X(c+1,\ c,\ 0) \xleftarrow{\#i} Z(c,\ c+1,\ 1)$, c<N-1.

Step 10. $\forall P'$ and $\forall c, \xrightarrow{\#b} X(c,\ c-1,\ P')$, 0<c<N.

//Call Subroutine *C.1* for parallel data broadcast//

Step 11. $\forall P'$, $\forall c$ and $\forall$p, $\mathrm{Y}(c,\ p,\ P') \xleftarrow{\#s} \mathrm{X}(c,\ p,\ P') \oplus \mathrm{Y}(c,\ p,\ P')$, where symbol $\oplus$, represents binary summation.

Note: $\forall$P', $\forall$c and $\forall$p**,** the final result emerges from register**:** $\mathrm{Y}(c,\ p,\ P')$**.** The data values stored in register: $\mathrm{Y}(c,\ p,\ P')$ is displayed in Figure 12.

Time Complexity Analysis

In Step-1, $\big((d_h-1)(d_h+1)\big) - \frac{(d_h-1)(d_h-2)}{2} + 4$ electronic moves are required to perform parallel for d_h-dimensional hyper hexa-cell. Step-3 demands d_h+1 intra-cluster (electronic) moves for parallel data broadcast in each cluster. Steps-4, 7 and 9 needs single inter-cluster move (optical) each. Further, Step-5 demands d_h+2 electronic moves for performing parallel data sum in each cluster. Step-6 again requires d_h+1 electronic moves for performing the parallel broadcast in cluster-(N-1) of part-1.

Further, Steps-8 and 10 needs d_h+1 electronic moves to perform parallel broadcasting of data in each cluster of both parts. Step-2 and 12 requires constant moves. Therefore, total $\big((d_h-1)(d_h+1)\big) - \frac{(d_h-1)(d_h-2)}{2} + 5d_h + 10$ intra-cluster (electronic) and 3 inter-cluster (optical) moves are necessary to perform parallel prefix over d_h-dimensional biswapped network hyper hexa-cell.

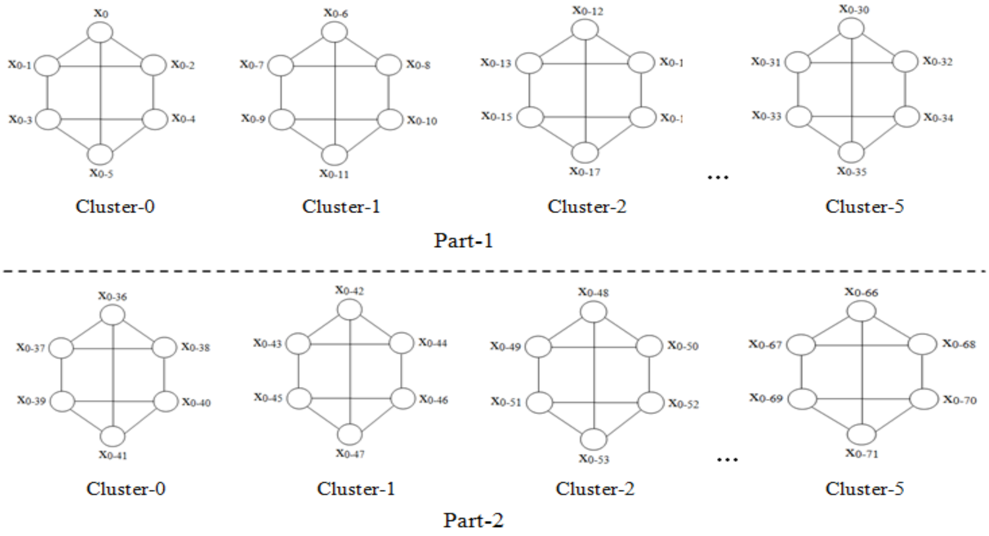

Fig.12. The data Elements Stored in Register: Y(c, p, P') after Step-11

Subroutine C.1:

Parallel Data Broadcast on d_h-dimensional hyper hexa-cell (cluster)

Parallel Algorithm:

Step 1. Perform parallel broadcast of data element (that need to be broadcast) in source subcluster.

Step 2. Broadcast in the entire cluster the identical copies of data elements obtained at processors of source subcluster (in Step1).

Time Complexity Analysis

In Step 1, parallel data broadcast in source processors' subcluster needs 2 electronic communication moves (as shown by flow of data by arrows in Figure 13). In Step 2, d_h-1 electronic communication moves will be required to broadcast the identical copies of data element obtained at each processor of source processors' subcluster (as shown by flow of data blue arrows in step-3 in Figure 14)**.** Let us assume, Processor: [0,100]) as source processor in two-dimensional hyper hexa-cell network. Figure 14 show the flow of data from source processor (Processor: [0,100]) in two-dimensional hyper hexa-cell network**.** Therefore, total d_h+1 electronic communication moves will be required (as shown in Eq. (4)) for data broadcast in d_h-dimensional hyper hexa-cell network (cluster).

$$\text{Parallel Data Broadcast} = d_h + 1 \tag{4}$$

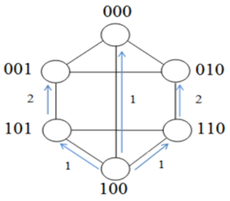

Fig.13. Parallel Data Broadcast (One-dimensional HHC network)

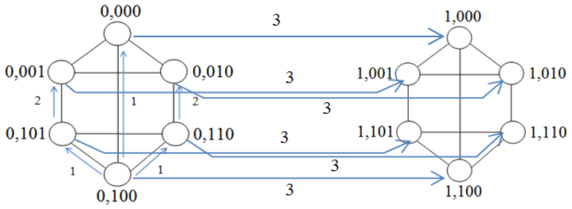

Fig.14. Parallel Data Broadcast (Two-dimensional HHC network)

Subroutine C.2:

Parallel Data Sum on d_h-dimensional hyper hexa-cell (cluster)

Parallel Algorithm:

Step 1. Perform parallel summation of data elements in each subcluster (one-dimensional HHC network) scattered at different processors and store their outcomes at particular processors.

Step 2. Perform data communication to collect partial results (as data sum) at source subcluster from other subclusters.

Time Complexity Analysis

In Step 1, parallel data summation of data elements in a subcluster demands 3 electronic moves (as shown by flow of data by blue arrows in Figure 15). Here we assume top processor of the upper triangle to store the outcome of data sum in subcluster. In Step 2, parallel data summation in a cluster demands d_h-1 electronic moves to collect partial results (data sum obtained in each subcluster). The blue arrow in step-4 in Figure 16 indicates the flow of data to collect the partial results (as data sum) obtained at each one-dimensional hyper hexa-cell network (subcluster) of two-dimensional HHC network (cluster). Therefore, total d_h+2 electronic moves will be required (as shown in Eq. (5) to perform parallel data summation over d_h-dimensional hyper hexa-cell network (cluster).

$$\text{Parallel Data Summation} = d_h + 2 \tag{5}$$

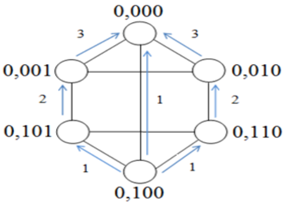

Fig.15. Parallel Data Summation (One-dimensional HHC Network) (Cluster-0)

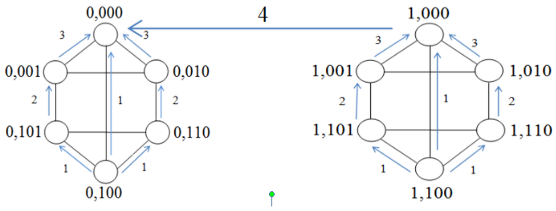

Fig.16. Parallel Data Summation (Two-dimensional HHC network)

VI. Conclusion

In this paper, we have presented a novel parallel approach for mapping an important numerical problem of prefix sum on recently presented biswapped network hyper hexa-cell network [6] (a variation of biswapped network [4]). This algorithm mapped for $2 \times \left(6 \times 2^{d_h - 1}\right)^2$ data elements on d_h-dimensional biswapped network hyper hex-cell of $2 \times \left(6 \times 2^{d_h - 1}\right)^2$ processors. It demands total $\left((d_h - 1) \times (d_h + 1)\right) - \frac{(d_h - 1)\times(d_h - 2)}{2} + 5d_h + 10$ intra-cluster (electronic) and 3 inter-cluster (optical) moves.

References

[1] C.F. Wang and S. Sahni, Basic operations on the OTIS mesh optoelectronic computer, IEEE trans on parallel and distributed systems, Vol 9, No 12, 1998.

[2] Marsden G, Marchand P, Harvey P, Esener S, Optical transpose interconnection system architecture, Opt Lett 18(13):1083–1085, 1993.

[3] Basel A. Mahafzah, Azzam Sleit, Nesreen A. Hamad, Elham F. Ahmad, Tasneem M, Abu-Kabeer, The OTIS hyper hexa-cell optoelectronic architecture, Journal of computing, Springer, 2012, 94:411–432.

[4] Wenjun Xiaoa, Behrooz Parhamib*, Weidong Chena, Mingxin Hea and WenhongWeia, Biswapped networks: a family of interconnection architectures with advantages over swapped or OTIS networks, "International Journal of Computer Mathematics", Vol. 88, No. 13, September 2011, 2669–2684, Taylor and francis.

[5] Wenhong Wei, Qingxia Li, Ming Tao, BSN-mesh and its basic parallel algorithms", International Journal of Grid and Utility Computing, Vol 6, Issue 3-4, pp 213-220, 2015.

[6] A. Gupta, B. K. Sarkar, The recursive and symmetrical optoelectronic network architecture: biswapped network hyper hexa-cell, Arabian Journal of Science and Engineering, Mar 2018.

[7] P.K.Jana and B.P.Sinha, "An improved parallel prefix algorithm on OTIS mesh", Parallel processing letters (World Scientific), pp. 429-440, Vol.16, No. 4, 2006.

[8] Dheeresh K. Mallic, Prasanta K. Jana, In Proc:"Parallel prefix on mesh of trees and OTIS mesh of trees", PDPTA 08, 2012, Solan, Himanchal Pradesh, India.

[9] K.T.Lukas, "Parallel Algorithm for Prefix Computation on OTIS K-Ary 3-cube parallel computers", International journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009.

[10] P.K. Jana, "Improved parallel prefix computation on optical multi trees", IEEE INDICON 2004.

[11] S. K. Jha, "An Improved Parallel Prefix Computation on 2D-Mesh Network", In Proc: International Conference on

Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013, pp. 919-926

[12] S.K.Jha, P.K.Jana, “Fast Parallel Prefix on Multi-Mesh of Trees”, In Proc: IEEE international conference on computer and communication technology, Motilal Nehru National Institute of Technology, Allahabad, 17-19 September 2010, pp: 641-646.

[13] P.K.Jana, B.D.Naidu, S.Kumar, M.arora, B.P.Sinha, “Parallel prefix computation on extended multi- mesh network”, Information processing letters (Elsevier Science), pp, 295-303, Vol, 84, No. 6, Oct 2002.

[14] N Rakesh, N. Nitin, “Parallel Prefix Sum Computation on Multi Mesh of Trees”, In Proc: IEEE INDICON 2009.

[15] A. Datta, Mallika De, B.P. Sinha,”Fast Parallel Algorithm for Prefix Computation in Multi-Mesh Architecture”, Parallel Processing Letters, 2017.

[16] A Gupta, B. K. Sarkar, “A new parallel approach for prefix sum on BSN mesh”, In Proc:Next Generation Computing Technologies, Apr 2016, pp. 393-396, dehradun, uttrakhand, INDIA.

Author’s Profiles

Ashish Gupta received his B. Tech Degree in Information technology from Uttar Pradesh Technical University Lucknow in 2008, M. Tech Degree in Computer Sciene from Indian Institute of Technology, Dhanbad, India in 2011. Currently, he is pursuing Ph. D in Computer Science from Birla Institute of Technology, Mesra, Ranchi (India). His research interests include Parallel and Distributed computing, Optoelectronic parallel network architectures, Designing of parallel algorithms. He has published many research papers in reputed SCI and Scopus indexed journals and Conferences.

Bikash kanti Sarkar received his M.Sc in Mathematics from Indian Institute of Technology, Kharagpur, India in 1993, Master in Computer Application from Bengal Engineering College (D.U.), Kolkata, India in 1999 and M.Phil in Computer Science from Annamalai University, Chennai, India in 2008. In 2013, he obtained his Ph.D in Computer Science from Jadavpur University, West Bengal. Dr. Sarkar is serving as a Faculty in the Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi (India). His current research areas include Parallel Computing, Machine Learning, Big Data Analytics and Security. Dr. Sarkar has published several research papers in reputed international journals and conference proceedings. He is the author of two books of Computer Science.

How to cite this paper: Ashish Gupta, Bikash Kanti Sarkar,"Parallel Prefix Sum Algorithm on Optoelectronic Biswapped Network Hyper Hexa-cell", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.8, pp.27-35, 2018.DOI: 10.5815/ijcnis.2018.08.03