Parallel Matrix Multiplication on Centralized Diamond Architecture

Masumeh Damrudi*, Kamal Jadidy Aval
Department of Computer Science, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran.

* Corresponding author. Email: m.damrudi@gmail.com
Manuscript submitted August 5, 2014; accepted June 12, 2015.
doi: 10.17706/ijcce.2016.5.2.151-157

Abstract: In this paper, we introduce a matrix multiplication algorithm on the new born parallel architecture named Centralized Diamond architecture. Matrix multiplication is a fundamental mathematical operation which has numerous applications in most of scientific fields. There are some sorting and searching algorithms on Centralized Diamond architecture which have appropriate time complexity. Once different kinds of parallel algorithms are applicable to a parallel architecture, the architecture is considered to be more functional and can be employed for different purposes and applications. The theoretical analysis proves that the matrix multiplication operation on Centralized Diamond architecture runs in a competitive number of steps compared to existing solutions.

Key words: Centralized diamond architecture, EREW, matrix multiplication, parallel processing.

1. Introduction

One of the most exciting research areas in the computer science is parallel processing. Parallel solutions have gained lots of interests in the past decades. Parallel algorithms are required for parallel execution. The aim of parallel programs is to solve bigger problems faster by running different parts of the solution on a number of processor elements [1], [2]. Parallelism is applied in various applications including sorting, searching, time consuming cryptographies [3]-[6], and in every real time world application that needs to be fast. There are different architectures that are employed for various parallel algorithms but researchers always attempt to find new and optimal ones. The centralized diamond inherits its main specifications from Diamond architecture that was first presented in [7]. Later on, the architecture was developed in [8] by adding one processor element in the center to gain more useful interconnections among first level processor elements. Different sorting and searching algorithms are proposed based on this architecture [9]-[12]. There are different parallel algorithms in the parallel world. It is important for an architecture to be capable of being employed for different purposes and applications. According to its intrinsic features coming from connections, Centralized Diamond architecture has this capability.

Matrix multiplication is one of the most fundamental operations in mathematics. There are numerous applications in different scientific fields including physics, statistics, computer graphics, applied mathematics, etc. depending on matrix multiplication. Various researchers have consumed a lot of time and effort to achieve faster serial and parallel algorithms to perform matrix multiplication [13].

In this paper, we present the matrix multiplication on the Centralized Diamond architecture which is a Single Instruction Multiple Data (SIMD) architecture with a time complexity of $O(m)$ on EREW PRAM (Exclusive Read Exclusive Write Parallel Random Access Machine).

2. Matrix Multiplication on Centralized Diamond Architecture

The interconnections between PEs (processor elements) of Centralized Diamond architecture are presented in Fig. 1 for N=29 for processor elements. Each Centralized Diamond architecture has four levels. The relation between data elements and PEs in this architecture is $N = 7/4\,n + 1$. The details of the Centralized Diamond architecture are presented in [11].

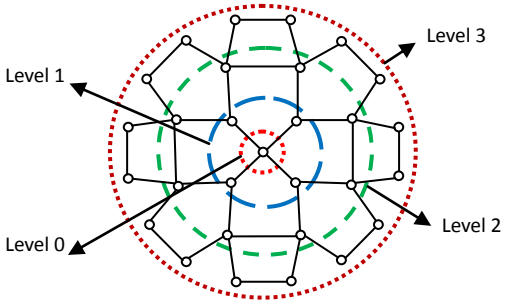

Fig. 1. Centralized diamond architecture.

The matrix multiplication on Centralized Diamond architecture is more like the multiplication on tree [14]. The architecture itself is similar to combination of trees. Fig. 2 shows an example where the Centralized Diamond architecture is formed by interconnection of four trees.

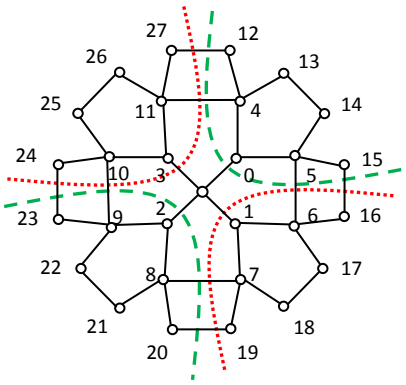

Fig. 2. Combination of trees in centralized diamond architecture.

The generalized form of multiplication on Centralized Diamond architecture for:

$$N = 29,$$

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{17} & a_{18} & a_{19} & a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{30} & a_{31} & a_{32} \\ a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} & a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} \end{bmatrix},$$

$$U = [u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} u_{11} u_{12} u_{13} u_{14} u_{15} u_{16}]$$

are presented in Fig. 3. From 29 PEs, just the PEs on the leaves are multipliers (16 PEs) and the rest of PEs are adders (13 PEs). The relation among number of data elements of matrix U and each row of matrix A and number of PEs in this architecture is $N = 7/4n + 1$. For the presented figures $N = 29$ and $n = 16$, the flow of operations is shown in this figure with arrows. The multiplication and addition operations are shown in the PEs which indicate the type of operations. The vector U is placed in the leaves and each row of the matrix A are entered in each step. The multiplication will be performed in the leaves and the results

of the multiplications are sent to upper PEs for additions. While the addition of multiplication results for the first row of A is running in the second level, the multiplication of second row of A with U is being performed in the third level. Also, whereas the third row of A is multiplying with U, the addition of multiplication results of second row for the A is being performed in the second level and the remaining addition of multiplication results for the first row of A is being carried out in the first level. All levels are working in parallel which leads to a faster matrix multiplication. Finally the results are generated in the centralized processor elements, which is in the middle of architecture in the level 0, as shown in Fig. 3 step 4.

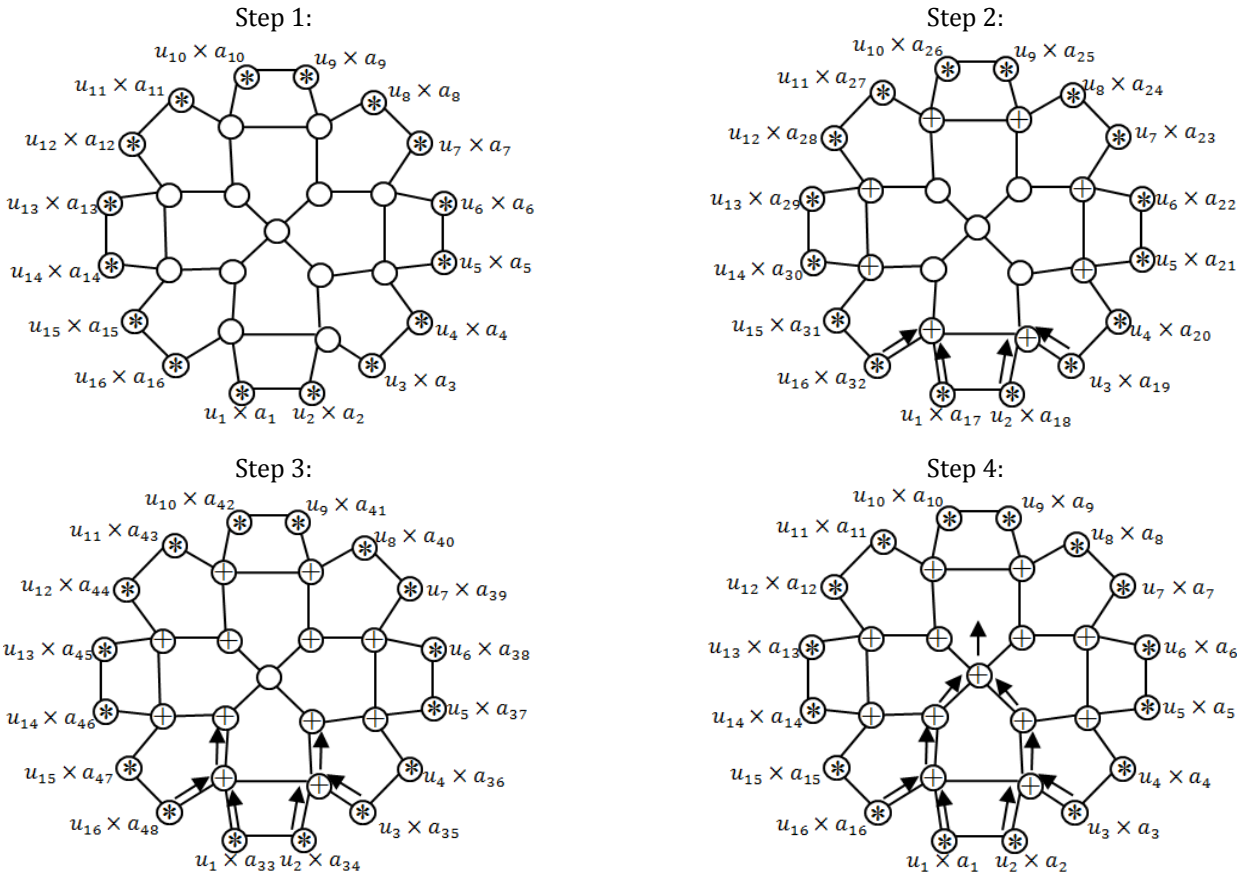

Fig. 3. Multiplication and addition operations in centralized diamond architecture.

The algorithm of multiplication on centralized diamond architecture is presented as following.

Multiplication Algorithm

for $i = 3n/4$ *to* $N-2$ *(in parallel)*
  for $j = m+1$ *to* $m+n$
    compute $u_i \times a_j$
    send result to parent
  $m = m + 16$
for $i = n/4$ *to* $\frac{3n}{4} - 1$ *(in parallel)*
  compute the sum of the two inputs
  send result to parent
for $i = 0$ *to* $\frac{n}{4} - 1$ *(in parallel)*
  compute the sum of the two inputs
  send result to output

3. Time Complexity, Cost and Simulation Results

In the matrix multiplication algorithm on Centralized Diamond architecture, as it is seen in the algorithm, each level of the architecture operates in parallel which time complexity is $O(1)$. The number of levels is four which is fixed in all sizes of the proposed architecture. The total time complexity is based on the number of rows in matrix A. If the number of rows is considered to be m, the total time complexity is $O(m)+O(1)+O(1)+O(1)$ which is $O(m)$. There is a matrix multiplication proposed on tree architecture [14] which needs m-1+log$n$ number of steps for this operation for a $m\times n$ matrix. The theoretical analysis shows that the time complexity of the proposed algorithm on the Centralized Diamond is better than the mentioned algorithm.

We have conducted simulation for the proposed solution. The SIMULINK of MATLAB is an environment which leads to a simpler and faster simulation. Fig. 4 is drawn using the SIMULINK environment of MATLAB SIMULINK R2011a. In this simulation, the PEs and their connections are defined as a centralized diamond with 15 processor elements where 8 PEs are multipliers and 7 PEs are adders. In the given example to evaluate the proposed algorithm, Matrix U and A are defined as

$$U = [1\ 3\ 7\ 20\ 15\ 8\ 11\ 3]$$

$$A = \begin{bmatrix} 6 & 7 & 13 & 8 & 10 & 3 & 5 & 2 \\ 4 & 6 & 8 & 6 & 3 & 7 & 7 & 5 \\ 10 & 1 & 5 & 2 & 7 & 4 & 2 & 3 \end{bmatrix}.$$

The simulation is performed exploiting the function blocks. The multiplication results of U and A is

$$\begin{bmatrix} 6 & 7 & 13 & 8 & 10 & 3 & 5 & 2 \\ 4 & 6 & 8 & 6 & 3 & 7 & 7 & 5 \\ 10 & 1 & 5 & 2 & 7 & 4 & 2 & 3 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \\ 7 \\ 20 \\ 15 \\ 8 \\ 11 \\ 3 \end{bmatrix} = \begin{bmatrix} 513 \\ 391 \\ 256 \end{bmatrix}.$$

The result of multiplication is shown in the central PE in Fig. 4 using Display block.

Step by step procedure for the proposed matrix multiplication algorithm on centralized diamond architecture is presented as a flowchart in Fig. 5. This diagrammatic representation illustrates the solution model.

4. Conclusion

We proposed matrix multiplication on Centralized Diamond architecture. This architecture is a new interconnection network where different sorting and searching algorithms are proposed based on it. To have a functional architecture, it would be better to be employed for various algorithms.

Our algorithm has a suitable order of time complexity using an acceptable hardware cost where the total time complexity is $O(m)$. Although the architecture is heterogeneous, but there is no delay for this architecture because all the processors in the leaves are the same and all PEs in other levels are adders. The only limitation of this matrix multiplication on Centralized Diamond architecture is on the number of each row. The number of each row must be even. On the other hand, this architecture needs less time complexity than tree for matrix multiplication due to the fixed number of levels which is four.

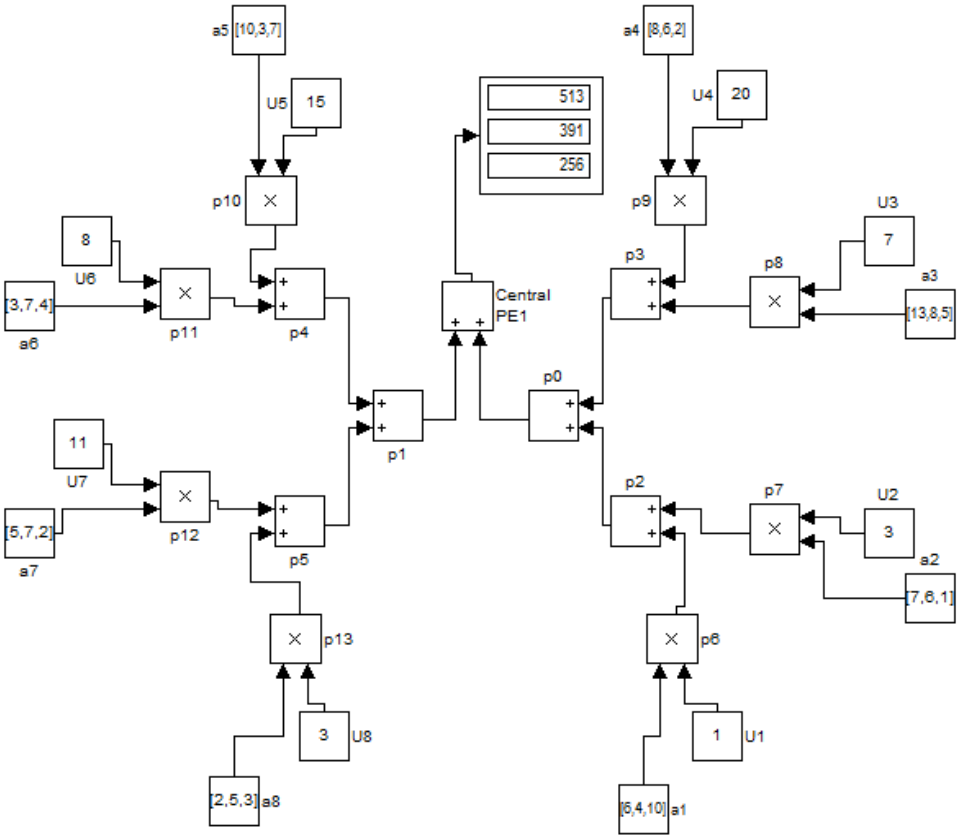

Fig. 4. Simulation of matrix multiplication on centralized diamond architecture.

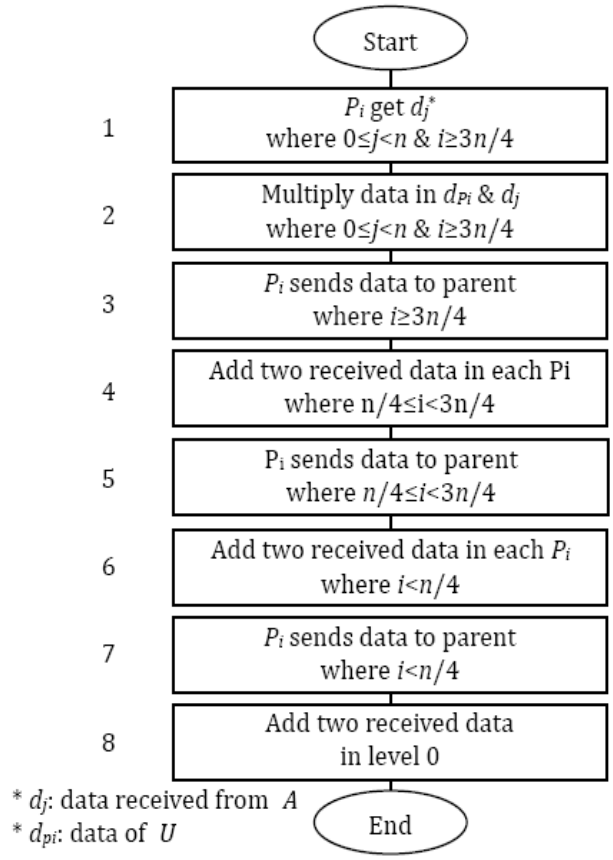

Fig. 5. Simulation of matrix multiplication on centralized diamond architecture.

Acknowledgment

This research was supported by Islamic Azad University, Firoozkooh Branch. The authors would like to thank reviewers of this paper for their valuable comments and suggestions that improve the presentation of this paper.

References

[1] Timothy, G., & Beverly, A. (2005). *Patterns for Parallel Programming*. USA: Addison-Wesley.

[2] Li, H., & Li, J. Z. (2008). A new compact dual-core architecture for AES encryption and decryption. *Canadian Journal of Electrical and Computer Engineering-Revue Canadienne De Genie Electrique Et Informatique, 33*(*3-4*), 209-213.

[3] Damrudi, M., & Ithnin, N. (2011). State of the art practical parallel cryptographic approaches. *Australian Journal of Basic and Applied Sciences, 5(7)*, 660-677.

[4] Damrudi, M., & Ithnin, N. (2013). Parallel RSA encryption based on tree architecture. *Journal of the Chinese Institute of Engineers, 36(5)*, 658-666.

[5] Damrudi, M., & Ithnin, N. (2012). An optimization of tree topology based parallel cryptography. *Mathematical Problems in Engineering,* 10.

[6] Damrudi, M., & Ithnin, N. (2013). Numerical analysis of parallel modular exponentiation for RSA using interconnection networks. *Scienceasia, 39*, 103-106.

[7] Damrudi, M., & Aval, K. J. (September 20-21, 2009). Diamond architecture with NOD sorting. *Proceedings of IEEE Youth Conference on Information, Computing and Telecommunication* (pp. 431-434). Beijing, China.

[8] Damrudi, M., Aval, K. J., & Asadollahi, H. (2010). Centralized diamond architecture, a parallel approach. *Proceedings of 2nd International Conference on Information and Multimedia Technology* (pp. 197-200). Hong Kong.

[9] Damrudi, M., & Aval, K. J. (2010). A new parallel sorting on diamond architecture. *Proceedings of the 4th Conference on European Computing* (pp. 284-287). University Politechnica of Bucharest, Bucharest, Romania.

[10] Damrudi, M., & Aval, K. J. (2011). Sorting data elements by SOCD using centralized diamond architecture. *Computer Technology and Application, 2(5)*, 374-377.

[11] Damrudi, M., & Aval, K. J. (2012). Generalized approach to SOCD sorting on centralized diamond architecture. *Indian Journal of Science and Technology, 5(9)*, 3288-3291.

[12] Damrudi, M., & Aval, K. J. (2011). Searching data using centralized diamond architecture. *Journal of Communication and Computer, 8(9)*, 807-811.

[13] Bae, S. E., Shinn, T.-W., & Takaoka, T. (2014). A faster parallel algorithm for matrix multiplication on a mesh array. *Procedia Computer Science, 29*, 2230-2240.

[14] Akl, S. G. (1989). *The Design and Analysis of Parallel Algorithms*. USA: Prentice-Hall.

Masumeh Damrudi has received her B.Sc. and M.Sc. degrees in computer science (software engineering) from Islamic Azad University, Tehran South Branch, Tehran, Iran in 2001 and 2004 respectively. She has received her PhD degree from Universiti Teknologi Malaysia (UTM), JB, Malaysia in 2013.

She has been a lecturer in the Islamic Azad University, Firoozkooh Branch since 2006. She has published more than 20 papers in various journals and conferences. Her research interests are parallel processing, cryptography and image processing.

Kamal Jadidy Aval has received his B.Sc. and M.Sc. degrees in computer science (software engineering) from Iran University of Science and Technology and Islamic Azad University, Tehran South branch, Tehran, Iran in 2002 and 2004 respectively. He will receive his PhD degree from Universiti Teknologi Malaysia (UTM), JB, Malaysia in 2015.

He has been a lecturer in the Islamic Azad University, Firoozkooh branch since 2006. He has published more than 20 papers in various journals and conferences. His research interests are image processing, evolutionary algorithms, WSNs, and MANets.