Implementation Of Dynamically Reconfigurable NoC Using Flexible Dimension Order Routing (FDOR) Algorithm on FPGA

[1]Divyashree K and [2]Shwethaa A R
[1,2]Department of Electronics & Communication Engineering, AMC Engineering College, Bangalore, India.
[1]divyashri.013@gmail.com, [2]shwethaa2588@gmail.com

Article Info
Jenitta J and Swetha Rani L (eds.), *International Conference on VLSI, Communications and Computer Communication*, Advances in Intelligent Systems and Technologies,
Doi: https://doi.org/10.53759/aist/978-9914-9946-1-2_25

Abstract: NoC-Based Dynamic Reconfigurable Systems (DRSs) implemented over FPGA devices change their configuration at the run time by re-positioning or replacing the existing processing modules into the network. Several Dynamically Reconfigurable NoCs (DRNoCs) in the literature, propose adaptive routing algorithms in order to handle the network structure alteration. Nevertheless, their implementation cost is severe in terms of chip area and time required to reconfigure the routing scheme, which results in non-well scalable solutions for DRSs. In this work, we propose an alternative DRNoC approach, based on a traditional 2-D mesh, using a logic-based implementation of the Flexible Direction Order Routing (FDOR) algorithm, thus inheriting its simplicity and deadlock-freedom. Several scenarios have been considered in order to prove the applicability of the FDOR algorithm in the context of a DRNoC accompanied by performance and synthesis results. In conclusion, we demonstrate that FDOR is a suitable solution for DRNoCs. This project was aimed for implementing low-cost optimized FPGA architecture. Model sim 6.5f is used for simulating the NOC and synthesized using Xilinx ISE 14.7. Then the implementation is done in Spartan3 FPGA Kit.

Keywords: NOC, FPGA, DRS

I. INTRODUCTION

Dynamically Reconfigurable Systems (DRS) are characterized by their high level of modularity and flexibility, which enables the implementation of a higher number of processing modules over a limited chip area, the Network on-Chip (NoC), capable of dynamically change its communication structure during the allocation and placement of processing modules over the changing their configuration according to the computation requirements.

One particular case of DRSs, is known as Dynamically Reconfigurable Network on-Chip (DRNoC), which is an extension of the Network on-Chip (NoC), capable of dynamically change its communication structure during the allocation and placement of processing modules over the network structure. [8] DRNoCs have the same properties of traditional NoCs, with the difference that they must be re-adapted each time a dynamic reconfiguration occurs. For example, the routing algorithm must be reconfigured to the irregular network topologies obtained from the positioning of new processing modules over the network structure after the dynamic reconfiguration processes.

The routing algorithm plays an important role in DRNoCs, since it must guarantee safer communications for all the different contexts, even in the presence of obstacles generated by the processing modules placement. [9] Ensuring this condition is not so simple because the transition between two routing algorithms induces extra dependencies that may lead to deadlock and/or live lock.

Some related works about DRNoCs [1][2][3][4] have been proposed in the literature, presenting different implementations of routing algorithms that can accomplish a safe dead lock free communication. However, their complexity comes with a high chip area and, therefore, higher power consumption. Also, some of these alternatives require a high time to reconfigure their routing algorithm, which limits their relevancy for DRNoCs.

The Flexible Dimension Order Routing (FDOR) algorithm represents an interesting alternative since it is a deadlock free routing solution with a significantly lower complexity in comparison with other routing algorithms. The FDOR switching scheme do not require additional Virtual Channels (VCs), but guarantees connectivity and deadlock freedom in irregular shaped topologies obtained from the positioning of processing modules over the network.

In this work, we propose a DRNoC with a Reconfiguration System based on pure logic gate implementation of the FDOR algorithm for a 2-D Mesh, to automatically adapt the routing algorithm according to the current network topology obtained from the processing modules over the network structure. [10] It has been named as DyAFNoC (for Dynamic Automatic FDOR-based NoC), whose architecture was described using VHDL and simulated by using the Dynamic Circuit Switching (DCS) Technique. A performance analysis was also made in order to characterize the traffic in different FDOR topologies that can help the designer to choose the adequate module positioning for obtaining the best traffic performance in the different scenarios that conform DRS.

NoC-Based Dynamic Reconfigurable Systems (DRSs) implemented over FPGA devices change their configuration at the run time by repositioning or replacing the existing processing modules over the network structure being known as Dynamic Reconfigurable NoCs (DRNoCs).

In this work, we propose an alternative DRNoC approach, based on a traditional 2-Dmesh using a logic base implementation of the Flexible Direction Order Routing (FDOR) algorithm. Characterized by its simplicity low complexity and deadlock-freeness.

II. LITERATURE-WORK

DRNoCs have been proposed with different routing algorithms. DyNoC based on a 2-D mesh structure composed of 5-port routers, use a modified Dimension-Order-Routing or DOR algorithm (e.g., XY and YX algorithms) known as Surrounding XY (S-XY) that surrounds the obstacle modules until reach its destination. Other proposal is CuNoC based on a 2-D mesh structure composed of 4-port routers known as Communication Units (CU), also use the S-XY algorithm. Each router has a local routing logic that determines the routing path under the presence of obstacle modules. For both cases, deadlock-freedom cannot be guaranteed for all irregular network topologies, although the probability of deadlock is low when using virtual channel-based approaches.

QNoC is an extension of the CuNoC architecture that adopts a TBR algorithm which updates all the routing tables for the dynamic reconfiguration process. TBR is not a well solution for DRS since each routing table is composed of three layers of 32-bits, requiring a high cost of hardware resources for its implementation with a high reconfiguration time for updating the routing tables. Other routing algorithms have been proposed for the implementation of fault-tolerant NoCs in which the routing schemes change in case of appearance of faulty regions in the network structure, since faulty routers imply network paths to be cut. It is based on a LBDR implementation for each channel, requiring average of 18 configuration bits per router in order to reconfigure the network.

The proposed implementation guarantees deadlock-freedom and requires les hardware resources than TBR, because according to the FDOR algorithm requires just one configuration bit per router. In this work, we will propose the DyAFNoC system using the FDOR as routing algorithm for the reason that it has a low complexity, requiring a lower implementation cost in comparison with the related works. The contribution of these works is the implementation of a Mesh Reconfiguration Control System (MRCS) that automatically calculates the routing scheme based on the topology information provided by the positioning of a new processing module over the network mesh structure.

III. SYSTEM ARCHITECTURE

The DyAFNoC architecture was developed starting from the baseline HERMES network architecture [5]. Its structure is a 2-D Mesh network composed of 5-port routers (North, South, East, West and Local Port). Ports are bidirectional and contain two registers for storing incoming two flits of a packet respectively in order to be used in the Wormhole Switching mode. Each router is connected to a Processing Element (PE) by the local port. Processing Modules (PMs) will be allocated and positioned over the network forming different contexts during the run time of DyAFNoC.

These processes are executed by the Mesh Reconfiguration Control System (MRCS) and the Partial Dynamic Reconfiguration Control System (PDRCS), respectively.

Mesh Reconfigurable Control System (Mrcs) Architecture

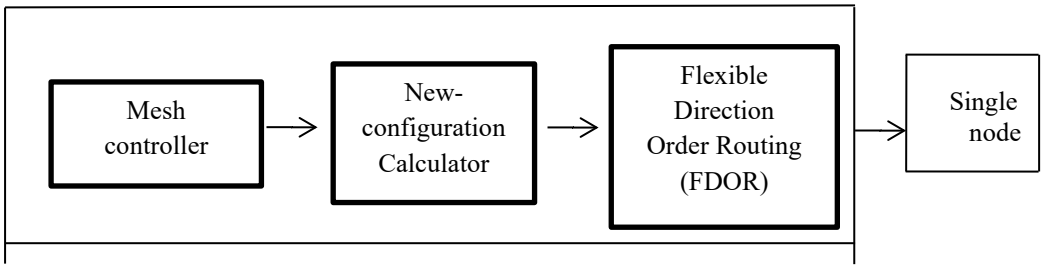

Fig 1. Mesh Reconfiguration Control System Architecture.

The MRCS Fig 1, receives the configuration map of the network (configuration map bit matrix). This map is used to calculate the DOR routing (XY or YX). The New configuration map is used to calculate the DOR routing (XY or YX). For each router of the network through a block known as New FDOR).

This block identifies which routers belong to the core and flanks respectively, and according to this, automatically determine the DOR algorithm that they will compute. [6**]** When this calculation is done, the 2-D Mesh controller starts the network drain operation. Each router may send to the MRC Sits" free" condition before proceeding with the routing algorithm update, each switch states its free condition if all the buffers of the router are empty. When all routers are free, the MRCS updates the current configuration matrix setting up the apply configuration signal to configure all the routers.

This functioning exploits the Typically Static Reconfiguration (TSR) approach Fig 2, by which deadlock freedom is ensured while reconfiguration is applying, avoiding the presence of two different algorithms at the same time that could lead extra dependency and consequently to deadlock. The AFDOR logic is a pure asynchronous logic that determines in few cycles the complete set of FDOR configuration bits. It is composed of a matrix of NxN logic blocks, each one related to router. Each block has I/O bits communicating with the adjacent blocks in order to propagate neighboring signals, which indicate the FDOR sub-region (core or flank) [7] to whom the activated routers belong to.

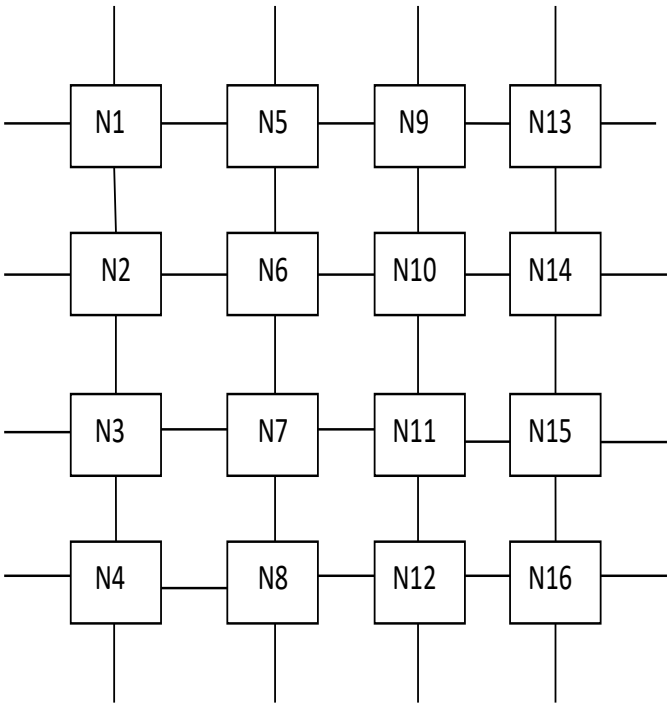

Fig 2. (4x4) Router Mesh

IV. CONCLUSION

Using the FDOR algorithm we can obtain low hardware implementation cost. We can configure at run time by replacing are repositioning the processing element by routers and allocators in the network. Dynamical reconfigurable and flexible Noc. The performance is evaluated based on latency and throughput.

References

[1] C. Bobda and A. Ahmadinia, "Dynamic interconnection of reconfigurable modules on reconfigurable devices," Design Test of Computers, IEEE, vol. 22, no. 5, pp. 443–451, Sept 2005.

[2] S. Jovanovic, C. Tanougast, and S. Weber, "A new high-performance scalable dynamic interconnection for fpga-based reconfigurable systems, "in Application-Specific Systems, Architectures and Processors,2008. ASAP 2008. International Conference on, July 2008, pp. 61–66.

[3] S. Jovanovic, C. Tanougast, C. Bobda, and S.Weber, "Cunoc: A Dynamic scalable communication structure for dynamically reconfigurable FPGAs," Microprocessors and Microsystems, vol. 33, no. 1, pp.24 – 36, 2009, selected Papers from ReCoSoC 2007 (Reconfigurable Communication-centric Systems-on-Chip).

[4] T. Pionteck, C. Albrecht, R. Koch, and E. Maehle, "Adaptive communication architectures for runtime reconfigurable system-on-chips," Parallel Processing Letters, vol. 18, no. 02, pp. 275–289, 2008.

[5] T. Skeie, F. Sem-Jacobsen, S. Rodrigo, J. Flich, D. Bertozzi, and S. Medardoni, "Flexible dor routing for virtualization of multicore chips," in System-on-Chip, 2009. SOC 2009. International Symposium on, Oct 2009, pp. 073–076.

[6] J. Huang, X. Xu, N. Wang, and S. Chen, "Reconfigurable topology synthesis for application-specific NoC on partially dynamically reconfigurable systems," Integration, vol. 65, pp. 331–343, Mar. 2019, doi: 10.1016/j.vlsi.2018.02.012.

[7] S. Abed, "Implementation of Edge Detection Algorithm using FPGA Reconfigurable Hardware," Journal of Engineering Research, vol. 8, no. 1, Mar. 2020, doi: 10.36909/jer.v8i1.7956.

[8] G. P. Ramesh and S. Prabhu, "FPGA Implementation of 3D NOC Using Anti-Hebbian for Multicast Routing Algorithm," Lecture Notes in Networks and Systems, pp. 301–313, 2021, doi: 10.1007/978-981-33-4687-1_29.

[9] M. Rozkovec, "Implementation of Dynamically Reconfigurable Test Architecture for FPGA Circuits," 2008 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, Apr. 2008, doi: 10.1109/ddecs.2008.4538782.

[10] N. Kapre and A. Dehon, "An NoC Traffic Compiler for Efficient FPGA Implementation of Sparse Graph-Oriented Workloads," International Journal of Reconfigurable Computing, vol. 2011, pp. 1–14, 2011, doi: 10.1155/2011/745147.