[1]Liyaqat Nazir,
[2]Mohd Asifuddola,
[3]Mumtaz Ahmed,
[4]Mir Mohsina Rahman

Implementation of Dimension Order Routing Algorithm on Reconfigurable Hardware for NoC applications

Abstract: - Inter-processing element communication encounters significant hurdles arising from constraints in power consumption, physical space, and data transfer speed. Storing data packets temporarily during communication uses a lot of the chip's power. Many contemporary network-on-chip (NoC) architectures are heavily resource-intensive due to the extensive use of router buffers. Eliminating these buffers and virtual channels can streamline router design and reduce power consumption. The routing and arbitration strategies implemented within a NoC router are pivotal for optimizing the overall performance of the NoC mesh. The routing algorithm plays a critical role in NoC systems, as it is responsible for balancing traffic load across network channels, even in scenarios of asymmetric traffic distribution. This paper introduces a model for X-Y routing algorithms, specifically tailored for implementation on FPGA platforms utilizing a flexible state diagram. The distinctive contribution of this research lies in its synthesis and optimized realization of the X-Y routing algorithm on FPGAs, providing Network-on-Chip (NoC) designers with a robust framework for developing efficient routers tailored to their FPGA architectures.

Keywords: Network-on-chip, Dimension order, Buffer.

I. INTRODUCTION

Network-on-Chip (NoC) communication architecture has appeared with promising solutions in communication in the field of integrated circuits. The NoC approach has provided an approved data transfer by delivering improved bandwidth and superior performance for communication between different components on a chip. The paradigm has also proffered scalable design, making it specifically effective when implemented on reconfigurable platforms, yielding for flexibility and adaptability in various applications.[1]. Network-on-chip establishes an effectual communication paradigm that transforms system-on-chip (SoC) design, having significant improvements in efficiency and overall performance [2] as illustrated in figure 1. Several processing elements within a System on a Chip (SoC) are interconnected through Network-on-Chip (NoC) routing nodes. These routing nodes are spatially organized in various configurations, for example mesh, linear, toroidal, and both 2D and 3D topologies, as depicted in Figure 2. To ensure optimal performance, routers must ensure the delivery of high bandwidth and low latency. [3]. The performance of a Network on Chip (NoC) is primarily reviewed through its throughput, a crucial metric which is a function of various factors. While the performance parameters of the NoC are normally perceived by its throughput, which is characterized by various parameters such as routing node throughput, the traffic load within the network, and the network topology [4]. The throughput of the routing node is controlled by the data path unit's most significant critical path in the node and the efficiency of control path units [5-8]. The control paths of on-chip communication routing fabric are chiefly constituted of arbitration and allocating units [9]. Allocators perform a crucial role in efficiently assigning Virtual Channels (VC) and expertly matching resources in each cycle. [10-12]. The data paths of the on-chip router incorporate buffering structures, VCs, and switching fabric. When the flits at the input port arrive for transmission, conflict for access to the fabric arises, with flits competing at both the input and output stages. The router unit enforces particular handshake signals to facilitate the transfer of data or flits. Subsequently, a VC allocating unit manages the allocation among the incoming flits, permitting only one flit competing at the input port to be routed to the designated output port.[13]. To mitigate the issue of blocking, the competing flits are temporarily stored in the VCs or buffering space of the router unit, enabling their processing during the subsequent appropriate clock cycles. [14]. Eliminating input buffers can streamline the architecture by removing the necessity for VCs. However, this approach may lead to increased head-of-line blocking, which can

[1]Electronics and Communication Engineering, Islamic University of Science & Technology Awantipora, India

[2]Department of Electronics Engineering, Z.H.C.E.T, Aligarh Muslim University, Aligarh, India

[3]Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India, (Corresponding Author)

Email: mahmed1@jmi.ac.in

[4]National Institute of Technology, Srinagar J&K, India