A scalable end effective routing algorithm for multi-FPGA based large scale NoC
Zhiwei Ge, Junyan Tan, Virginie Fresse, Frederic Rousseau, Sunying Yao

To cite this version:
Zhiwei Ge, Junyan Tan, Virginie Fresse, Frederic Rousseau, Sunying Yao. A scalable end effective routing algorithm for multi-FPGA based large scale NoC. COLLOQUE NATIONAL DU GDR SOC-SIP, Jun 2011, Lyon, France. 2 p. ujm-00601664
A Scalable and Effective Routing Algorithm for Multi-FPGA Based Large Scale NoC

Zhiwei GE1, Junyan TAN2, Virginie FRESSE2, Frédéric ROUSSEAU2, Suying YAO1

1ASIC Design Center
NO. 92, Weijin Road
Tianjin University
Tianjin, China
E-mail: {gezhiwei},{syyao}@tju.edu.cn, {junyan.tan},{virginie.fresse}@univ-st-etienne.fr, frédéric.rousseau@imag.fr

Abstract — In the context of the FPGA resource limitation for large scale NoC, multi-FPGA based solutions are proposed. Due to the scalable routing scheme and packet format, the proposed routing algorithm can be used with both intra-FPGA and inter-FPGA communications. The effectiveness of the scheme is that it can be used for any multi-FPGA based NoC with small amount extra resource consumption.

Index Terms— routing algorithm, multi-FPGA, NoC, emulation

I. INTRODUCTION

Network-on-Chip (NoC) has been proposed as a solution to implement future on-chip interconnections [1]. NoCs can provide higher overall bandwidth and more scalability with less power consumption [2].

NoC emulation using FPGA is widely used for evaluating the performances of the communication [3]. This experimental approach suffers from scalability issue, mainly due to the resource limitation of FPGA. Multi-FPGA based emulation systems are proposed to solve this problem [4][5]. No fully scalable NoC architectures for multi-FPGA emulation purpose are proposed in literature.

The contribution of this work is to propose a scalable NoC on multi-FPGA platform from existing mesh NoC using the packet-switching technique. We propose a routing algorithm considering inter-FPGA and intra-FPGA communications based on existing 2D routing algorithms.

II. THE HERMES NOC

In our work, we use Hermes NoC to illustrate the proposed routing algorithm. Any NoC with mesh topology using packet-switching technique can replace the Hermes NoC. The Hermes NoC was designed by Fernando MORAES from FACIN-PUCRS [6], and is depicted in Fig. 1.

The Hermes NoC uses the 2D mesh topology in which router has a set of bi-directional ports EAST, WEST, NORTH, SOUTH and LOCAL linked to other routers and PEs. The XY routing is often used due to its determinism and simplicity. The ATLAS tool can generate synthesizable VHDL NoCs with different flit sizes, buffer depths and routing algorithm. This tool also proposes SystemC simulation models with simulation VHDL models of traffic generators and traffic receptors.

III. THE ROUTING ALGORITHM FOR MULTI-FPGA BASED NOC

A. The proposed routing algorithm

We consider a multi-FPGA platform with M FPGAs (the index of the FPGA that node N belongs to is denoted by \( z_N \)). We denote the source node and destination node of a packet with S and D, respectively, and the current node with C. There is also one identical multi-dimension node \( N_{MD} \) in each FPGA. The \( N_{MD} \) node is the node (2, 2) in Fig. 2 in the 3X3 NoC implemented on 2 FPGAs. In addition to the traditional ports, the switch of \( N_{MD} \) has two extra ports UP and DOWN. The multi-dimension router can transfer packets to the adjacent FPGAs with these two extra ports.

The basic idea of the multi-FPGA based routing scheme is to transfer the packet to the destination FPGA \( z_D \) firstly (if the packet is not in FPGA \( z_D \)). The architecture of the router using the proposed multi-FPGA based routing algorithm is shown in Fig. 3. Three modules implement the router: input buffers, switch control logic and crossbar interconnections. When an input buffer receives the first 2 flits of a packet, it sends the flits to the switch control logic. The switch control logic can establish interconnections between the input ports and the correct outputs using the proposed routing algorithm. The algorithm compares the \( z_C \) with \( z_D \) firstly. Flits must be routed to \( D \) when \( z_C \) equals to \( z_D \). If this is not the case, flits will be routed to the UP port of \( N_{MD} \) when \( z_C \) is greater than \( z_D \) and to the DOWN port of \( N_{MD} \) when \( z_C \) is smaller than \( z_D \).

Fig. 2 The multi-FPGA based NoC and \( N_{MD} \) node

Fig. 3 The architecture of router dedicated to multi-FPGA.

B. The packet format

In order to support and evaluate the described multi-FPGA based routing, we require a five-flit header. The format of the header will be adapted to multi-FPGA based NoC communication and is depicted in Fig. 4. The first and the second flit of the header represent the address of the
destination FPGA $z_D$ and the 2D co-ordinate information of the destination router. The number of flits in the packet payload is stored in the third flit. To evaluate the latency performance of the proposed algorithm, the total latency between the source and destination router and the inter-FPGA link latency are included in the header.

**IV. EXPERIMENTS AND RESULTS**

Implementations and emulations are done on a multi-FPGA platform using four ML506 development board with Xilinx ISE 10.1. The ML506 board contains a Virtex5 FPGA (32640 registers and 32640 LUTs) with high speed serial designs using the RocketIO™ GTP transceivers [7].

**A. Resource utilization**

Resource utilization experiments are done firstly for routers (32-bit flit, 16-flit buffer), as depicted in Fig. 5.

![Fig. 5 Resource utilization results for routers](image)

The extra resources are about 50 registers and 300 LUTs between RPN_2D and RPN_3D (i.e. the same router using respectively the XY and the proposed routing algorithm). The resources used for $N_{MD}$ node are higher than any other nodes. The $N_{MD}$ node uses respectively about 160 more registers and 700 more LUTs than 2D nodes and 110 more registers and 400 more LUTs than 3D nodes. We also compare the resource utilization for different size of NoCs (32-bit flit, 16-flit buffer), as shown in Fig. 6.

![Fig. 6 Resource utilization results for NoCs](image)

**B. Average latency with different links**

The average latency of multi-FPGA based NoC depends on both the offered load and the inter-FPGA link. The minimal latency in clock cycles using the proposed algorithm to transfer a packet is given by:

$$\text{latency} = (\sum_{i=1}^{n} R_i) + P \times 2 + T_{\text{LINK}}$$

(1)

Where $n$ is the number of switches in the communication path (source and target included), $R_i$ is the number of clock cycles for routing data through one router, $P$ is the packet size and $T_{\text{LINK}}$ is the inter-FPGA link latency (flit 4 in the packet from Fig. 4). $R_i$ depends on the routing algorithm. $R_i$ is 7 clock cycles for 2D XY routing and 11 clock cycles for the proposed routing algorithm.

Fig. 7 shows the average latencies with different links for the 3X3X2 NoC to transfer 50 packets with 9-flit length using the NoC and the scenario depicted in 2.