A Dynamic NoC Approach for Communication in Reconfigurable Devices*

Christophe Bobda, Mateusz Majer, Dirk Koch, Ali Ahmadinia, and Jürgen Teich

Department of Computer Science, University of Erlangen-Nuremberg
Am Weichselgarten 3, D-91058 Erlangen, Germany
{bobda, mateusz, dirk.koch, ahmadinia, teich}@cs.fau.de
www12.informatik.uni-erlangen.de

Abstract. A concept for solving the communication problem among modules dynamically placed on a reconfigurable device is presented. Based on a dynamic network-on-chip (DyNoC) communication infrastructure, components placed at run-time on a device can mutually communicate. A 4x4 dynamic network-on-chip communication infrastructure prototype, implemented in an FPGA occupies only 7% of the device area and can be clocked at 391 MHz.

1 Introduction

On-line algorithms [2] have been developed in the past for temporal placement on reconfigurable device. Almost all those algorithms consider the modules to be rectangular boxes without communication among each other. In [1] a new on-line placement strategy which takes into account the communication information is presented. However, this method helps only to reduce the communication cost among the modules by placing connected modules near to each other on the chip. It does not determine how the communication will be realized. The dynamic placement of components on a reconfigurable device requires a viable communication infrastructure to support the dynamic communication requirements. This paper presents a new network-on-chip-based concept to dynamically handle the communication between modules on a reconfigurable device. A case study is provided as an FPGA implementation of a dynamical network-on-chip arranged in a 4x4 mesh.
The rest of the paper is organized as follow: in Section 2 we present some previous approaches for handling dynamic on-chip communication. The requirements on the architecture are presented in Section 3. Section 4 deals with the dynamic connection of components on the network. In section 5, an FPGA-implementation of a network infrastructure is shown. Section 6 concludes the work.

* Supported in part by the German Research Foundation (DFG), SPP 1148 (Rekonfigurierbarer Rechensysteme)

J. Becker, M. Platzner, S. Vernalde (Eds.): FPL 2004, LNCS 3203, pp. 1032–1036, 2004.

2 Related Work

Recently, Network-on-Chip (NoC) have been shown to be a good solution to support communication on System-on-Chip [3]. Dally et al [3] have proven that NoCs encounter many advantages (performance, structure and modularity) toward global signal wiring. A chip employing a NoC is composed of a set of network clients like DSP, memory, peripheral controller, custom logic, etc. Instead of connecting the modules using dedicated routing wires, they are connected to a network that route packets among them. Maresceaux et al [4] proposed the use of a NoC communication infrastructure as a component of an operating system for reconfigurable systems. However, their NoC approache supports only fixed processing modules defined as tile on the chip at compile time. We seek an approach which allows modules being placed on a reconfigurable device at run-time to communicate with other on-chip modules as well as off-chip modules without restriction on the placement.

3 Communication Infrastructure

To be able to dynamically establish communication between newly placed components, two approaches can be followed: A packet-based communication or an on-line signal routing. On-line signal routing can be used to establish a set of dedicated point to point communications. However the routing of signals is a computational intensive task which can only be efficiently computed off-line. Moreover, if signals are routed on a given area and a new component is placed in that area, then it will affect the signals currently routed. The packet-based approach has the advantage that the alteration of the network will not hinder the communication, since packets can always find their way in a strongly connected network. The packet-based approach can be realized using a *dynamic NoC approach*. In a static NoC, the clients are placed in rectangular tiles on the chip and communicate with other clients via a fixed network. It has been shown in [3] that the area occupied by the network logic is small (about 6 % of the tile's area). This ratio is expected to drastically decrease with the rapid growth in size of reconfigurable devices. In order to have a better network logic/PE ratio and make an optimal use of the resource, we impose the following requirements to the architecture of the communication infrastructure:

- The PEs should be flexible, but coarse grained computing elements. With fined grained PEs, the rapport network logic/PEs becomes big thus wasting the device area.
- Each PE should have access to the network. This condition is very important since it allows any module being placed on the reconfigurable device to always access the network via one of its surrounding PEs.
- PEs should directly communicate with their neighbors. This is helpful because it allows wiring to be done locally within a module boundary.
- The network logic should be flexible enough to be used within the module to which it belongs. Whenever a component is placed on a given region of the

device, the network modules deep inside the module area cannot be used for network operations. Therefore, they should be used as additional resource for the module they belong to. A component will use the resource of its network elements for the time it is running.

Figure 1 shows a communication infrastructure as described below on a reconfigurable device.

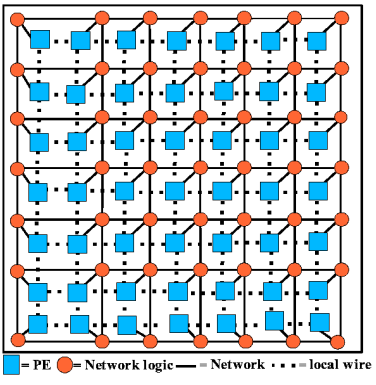

Fig. 1. The dynamic NoC approach for reconfigurable communication

Having defined the communication infrastructure of the device, the main questions are to know how to develop components which can be dynamically connected to the network at run-time. We provide some answers to those questions in the next sections.

4 Network Access

Each task is implemented as a component, represented by a rectangular box and stored in a database. A box encapsulates a circuit implemented with the resource in a given area. Therefore, a component can access the network using one of the network elements on its boundary. Without loss of generality, we select the network element on which the upper right PE of the component is attached. After the placement of a new component on the device, the placer will set the coordinates of the feasible network element as the module address for communication with the module. When placed on the device, components hide part of the network which is restored when they complete their execution. This makes the network dynamic. We call such a network a ***dynamic network-on-chip (DyNoC)***.
During the temporal placement, modules will always be placed such as to main-

tain a strongly connected[1] network. As shown in figure , each placed component is always surrounded by the network elements.

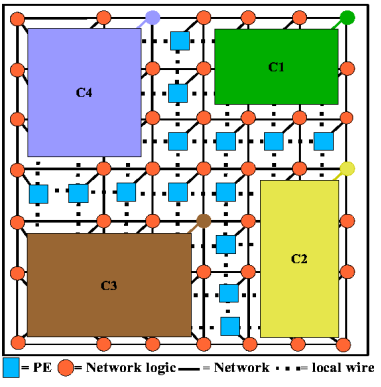

Fig. 2. A temporal placement on the DyNoC

While in a static NoC, each router always has four active neighbor-routers[2], this is not always the case in the DyNoC presented here. Whenever a component is placed on the device, it covers the routers in its area. Since those routers cannot be used, they will be deactivated by setting the corresponding control signals until the component completes its execution. Upon completing its execution, the deactivated routers are set to their default state. With this, a slightly modification of the routing algorithm is required. Before sending the packet in a given direction, a router must check if the router in that direction is activated. If so, then the packet will be routed perpendicular to the direction previously chosen.

5 Case Study

To investigate the feasability of the concept presented here, we have implemented a DyNoC similar to that described in Section 3 on an FPGA Virtex II 6000. The resulting circuit is presented in Figure 3. The complete communication infrastructure is made upon sixteen routers interconnected in a mesh network. The routers are connected by a 32-bit wide bus and 4 control lines and contain six 32-bit wide FIFO buffers with a depth of 4. The complete design occupies 7 % of the device area. If large components are placed on the device, they will cover a large set of routers, thus reducing the total area used by the routers. The router has also been implemented as described in the previous Section. Each

[1] A network is strongly connected, if for each pair of network elements a path exists which connects the two elements

[2] The neighbor-routers of the routers around the chips are assumed to be the package pins through which external modules can access the network

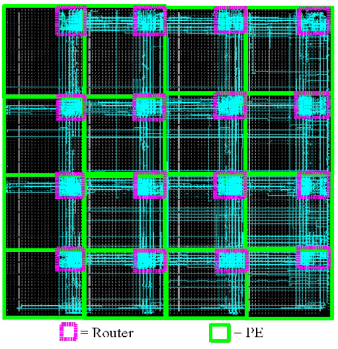

Fig. 3. FPGA implementation of a 4x4 DyNoC

router occupies less than 0.5 % of the device area and has a latency of 2.553 ns corresponding to a frequency of 391 MHz. Because the maximum number of routers to traverse is 6, two components running on the device with a frequency of 50 MHz can send and receive the packets without delay in their execution, if the network is free.

6 Conclusion

In this paper, we have presented a concept for handling the problem of dynamic communication among modules dynamically placed on a reconfigurable device. We have shown how a dynamically changing network-on-chip can be used as a viable communication infrastructure. A prototype of a 4x4 dynamic network-on-chip implemented in an FPGA has been built. The complete network infrastructure occupies only 7% of the device and the network elements can be clocked at 391 MHz.

References

1. Ali Ahmadinia, Christophe Bobda, Marcus Bednara, and Jürgen Teich. A new approach for on-line placement on reconfigurable devices. In *Proc. of IPDPS-2004, Reconfigurable Architectures Workshop (RAW-2004), Santa Fe NM, USA, April 26-27*, 2004.
2. K. Bazargan, R. Kastner, and M. Sarrafzadeh. Fast template placement for reconfigurable computing systems. *In IEEE Design and Test - Special Issue on Reconfigurable Computing*, January-March:68–83, 2000.
3. William J. Dally and Brian Towles. Route packets, not wires: on-chip interconnection networks. In *Proceedings of the Design Automation Conference*, pages 684–689, Las Vegas, NV, June 2001.
4. T. Marescaux, J-Y. Mignolet, A. Bartic, W. Moffat, D. Verkest, S. Vernalde, and R. Lauwereins. Networks on Chip as Hardware Components of an OS for Reconfigurable Systems. In *Proceedings of 13th International Conference on Field Programmable Logic and Applications*, Lisbon, Portugal, September 2003.