A Dynamically Reconfigurable Packet-Switched Network-on-Chip

Thilo Pionteck, Carsten Albrecht, Roman Koch
University of Lübeck
Institute of Computer Engineering
D-23538 Lübeck, Germany
{pionteck, albrecht, koch}@iti.uni-luebeck.de

Abstract

This paper presents the design of an adaptable NoC for FPGA based dynamically reconfigurable SoCs. At runtime, switches can be added or removed from the network, allowing to adapt the NoC to the number, size and location of currently configured hardware modules. By using dynamic routing tables, reconfiguration can be done without stopping or stalling the NoC. The proposed architecture avoids the limitations of bus-based interconnection schemes which are often applied in partially dynamically reconfigurable FPGA designs.

1. Introduction

In recent years, dynamic reconfiguration and packet-based Network-on-Chips (NoCs) have raised an increasing interest by researchers and in industry, especially with regard to FPGA designs. Dynamic reconfiguration is considered to be a solution to adapt systems to changing operating conditions and to overcome a limited amount of hardware resources. Network-on-Chips provide better utilization of routing resources and avoid the problems of traditional interconnection architectures such as wire lengths, driver strengths and limited bandwidth. Both design paradigms are often used for System-on-Chips (SoCs) where the design is partitioned into larger blocks. Yet, most often these design paradigms are not used in combination, since the fixed locations of the connection ports of the NoC hamper efficient placement of dynamically exchanged hardware modules of different sizes. Therefore, bus-based interconnection schemes are often used in partially dynamically reconfigurable FPGA designs. Some work has been done on adding dynamic elements to the structure of NoCs, yet still many limitations apply to these approaches. In [2], the problem of fixed locations of the interconnection ports is alleviated by a huge number of switches. At runtime, these switches can be deactivated and their hardware resources can be reused for dynamically inserted hardware modules.

The strongly connected NoC and a special routing algorithm guarantee that packets can be routed through the NoC at any time. Nevertheless, the problem of fragmentation still exists and the huge number of switches contribute to a significant overhead in area and transport delay.

A way to overcome these problems is to apply dynamic reconfiguration not only to the hardware modules, but also to the structure of the NoC. The NoC contains almost only those switches which are currently needed for operation. When hardware modules are inserted or removed by dynamic reconfiguration the number of switches and their locations have to be adapted by the same mechanism as well. To guarantee full connectivity, dynamic routing tables are used. Routing tables are provided by a separate control unit and are communicated over the NoC. The control unit is also responsible for the process of dynamic reconfiguration. When changing the structure of the NoC special precautions have to be taken in order not to isolate parts of the NoC. The underlying scenarios for these precautions are described in Section 2. Section 3 gives a short overview of the hardware structure and presents an application scenario using a Xilinx Virtex-II Pro FPGA as a hardware platform. Ongoing work is described in Section 4.

2. Scenarios

When changing the structure of a NoC special care has to be taken not to isolate parts of the NoC and to make sure that a new switch can be accessed by the neighbouring switches. A typical scenario is depicted in figure 1. Here, a hardware module (M2) is replaced by two smaller hardware modules (M7 and M8) which fit in the area of M2. As for the new configuration a connection port for M7 to the NoC is missing, a new switch has to be inserted between switch one and two. Since the dynamic reconfiguration should not affect other hardware modules and their communication, it has to be guaranteed that no packets are sent directly from switch one to switch two or vice versa. Therefore, routing tables of switch one to switch four have to be adapted (see figure 1b). In addition, the routing table of switch one should
hold information to access the new switch. These updates are done by means of internal NoC-packets containing the routing tables of the affected switches. These packets are sent by the global control unit of the system. The switches are addressed by a physical address in the header of the packets, while for the hardware modules logical addresses are provided. The internal NoC-packets are marked highest priority, ensuring that they are processed immediately by the packet-switched NoC. During dynamic reconfiguration it is assumed that the connection between switch three and switch four is not affected. This can be realized with a Xilinx Virtex-II Pro FPGA, even with its limitation to reconfigure only complete columns of the device, as it provides the capability of glitchless reconfiguration. When packet traffic is rerouted, switch five is added to the system as well as the new hardware modules. Now the first step is to initialize the routing table of switch five. The corresponding internal NoC-packet can be forwarded by switch one, as it obtained knowledge of the new switch and its physical address during the last update of its routing table. The final step consists of updating the routing tables of all surrounding switches, thereby reenabling the connection between switch one, five and two. The procedure to remove a switch from the NoC corresponds to the first step of inserting one.

3. Architecture

The implementation of the NoC is tailored to the structure of a Virtex-II Pro. For the specification of the header and the maximum packet size, the requirements of the later application area, a dynamically reconfigurable network coprocessor [1] is considered. This leads to a maximum packet size of 1036 bytes and a header of 12 bytes, divided into a data-link layer, network layer and application layer header. Inside the switches only the data-link layer is processed. It consists of physical source and destination address, packet size, priority, packet type and an error control field. For each 16 bit wide input port of the switches, a Virtex-II Pro Block-RAM is used as a FIFO allowing storage of a complete packet. To support wormhole routing, the header is immediately analyzed and passed to an arbiter. The arbiter coordinates access to the routing table according to the priority of potentially concurrently incoming packets from different ports, and is realized as a LUT. The same applies to the routing table, so that after five clock cycles from receiving the first 16 bit of a packet, the data can be forwarded to the corresponding output port of the switch. In order to prevent input FIFOs from overflowing, switches can send NoC internal flow control messages. Internal NoC-messages are also used to distribute updates of the routing tables sent by a packet dispatcher in conjunction with the reconfiguration control unit. The dispatcher analyses incoming packets to the system, determines which hardware modules can perform the required task and adds a header to the packet. If there is no matching hardware assist in the system, the reconfiguration control unit is triggered to dynamically insert the required unit.

4. Outlook

At present, the different components are combined into one system. Extensive simulations are in progress as well as efforts to reduce the hardware requirements of each component. With regard to algorithms, special focus is set to the improvement of the functionality of the dispatcher and the reconfiguration control unit.

References