Design and Evaluation of a Complexity Effective Network-on-Chip Architecture on FPGA

Kenichi Mori, Abderazek Ben Abdallah, Kenichi Kuroda
The University of Aizu,
School of Computer Science and Engineering,
Aizu-Wakamatsu 965-8580, Fukushima, Japan

Abstract: Current Systems-On-Chip (SoCs) execute applications which demand extensive parallel processing. Networks-On-Chip (NoCs) provide a good way of realizing efficient interconnections, and largely alleviate the limitations of bus-based solutions. In this paper, we present architecture and preliminary design result of a complexity effective on-chip interconnection network, named OASIS, on FPGA. The NoC was designed and prototyped with respect to simplicity and a small hardware footprint.

1 Introduction

Deep sub-micron processing technologies have enabled the implementation of new application-specific embedded architectures that integrate multiple software programmable processors and dedicated hardware components together onto a single chip. Recently, these application-specific architectures are emerging as a key design solution for today’s nonoelectronics design problems, which are being driven by emerging applications in the areas of: (1) wireless communication, (2) broadband/distributed networking, (3) distributed computing, and (4) multimedia computing. Despite these new opportunities, designers of these systems are currently confronted with the enormously difficult task of designing these complex heterogeneous multicore architectures.

Network-on-Chip NoC [1, 7, 5] is becoming an attractive option for solving bus based problems. It is a scalable architectural platform with huge potential to handle growing complexity and can provide easy re-configurability. The basic idea is that processors are connected via a packet switched communication network on a single chip similar to the way computers are connected to Internet. A chip based on NoC interconnection consists of several network clients (e.g. processors, memories, and custom logic) which are connected to a network that routes packets between them.

Packet switching supports asynchronous transfer of information. It provides extremely high bandwidth by distributing the propagation delay across multiple switches, thus pipelining the signal transmission. In addition, the NoC offers several promising features. First, it transmits packets instead of words. Dedicated address line like in bus systems are not necessary since the destination address of a packet is part of the packet. Second, transmission can be conducted in parallel if the network provides more than one transmission channel between a sender and a receiver. Thus, unlike bus-based system on chip, NoC presents theoretical infinite scalability, facile IP core reusing, and higher parallelism [4].

This paper presents architectural design and preliminary evaluation of a complexity effective on-chip interconnection network, named OASIS. We present detailed description of the major hardware modules of OASIS system.

Our system was described in Verilog HDL language and synthesized with CAD tools.

![Fig. 1: Circuit and Packet Switching NoC Routing.](image)

2 On-chip Interconnection Overview

On-chip interconnection networks use layered approaches, which are well suited to describe protocol functions that operate on data units at different levels of abstraction (in the form of streams, packets, bits or analog waveforms) and that are subject to various time granularity constraints [3]. Each layer may include one or more closely related protocol functions, such as data fragmentation, encoding and synchronization. Similar to interconnection networks for conventional parallel computers, the NoC interconnection paradigm is also characterized by its topology, protocol, and flow control. There are several decisions that should be made to design such system. These decisions should be made on communication protocol, switching style, network topology, clock synchronization method, signaling scheme, etc. We show in Figure 1 the major components that make a typical system based on NoC paradigm. Various type of interconnect architectures for MCSoC architectures have been proposed so far. Most of them borrowed ideas from the area of parallel computing with the consideration of different set of constraints, such as power, complexity, etc. The common desires are low latency and high throughput. The latter depends on the flow control mechanism,
which deals with the allocation of channel and buffer resources to packets as they traverse paths [7]. There is a large protocol space to select from for NoCs. Circuit switching, packet switching, and wormhole switching are possible choices for NoC protocols. These schemes can be mainly distinguished by their flow control methodologies. When these switching techniques are implemented in on-chip networks, they will have different performance along with different requirements on hardware resources.

In circuit switching, a physical path from the source to the destination is reserved prior to the transmission of data. Once a transmission starts, the transmission is not corrupted by other transmission since packets are not stored in buffer as in packet switching (discussed later). The advantage of circuit switching approach is that the network bandwidth is statically reserved for the whole duration of the data. Moreover, because circuit switching does not need packet buffers, area and power consumption can be reduced. The overhead of this approach is that the setting up of an end-to-end path causes unnecessary delay. In summary, circuit switching can provide high performance but little flexibility. Another alternative to circuit switching is so named packet switching scheme. Packet based communication has been brought to NoCs from the Internet world but loses the original advantage of reliability in the absence of dynamic routing. Currently, most of the proposals for routing in NoCs are based upon static routing mechanisms - XY-coordinate discipline. In packet based communication, data is divided into fixed length packets and whenever the source has a packet to be sent, it transmits the data. Every packet is composed of a control part, the header, and a data part (also named payload). Network switches inspect the headers of incoming packets to switch the packet to the appropriate output port. In this scheme, the need for sorting entire packets in a switch makes the buffer requirement very high. The last technique is the so called wormhole packets-switching, where each packet is further divided into flits (flow control unit) and the input and output buffers are expected to store only a few flits. Therefore, the buffer space requirement in the switches can be small and compact compared to the packet switching scheme. The header flit reserves the routing channel of each switch, the biddy flits will then follow the reserved channel, and the tail flit will later release the channel reservation. The advantage of the wormhole routing is that it does not require the complete packet to be stored in the switch • buffer while waiting for the header flit to route to the next stages. Thus, it requires much less buffer spaces. One packet may occupy several intermediate switches at the same time. Therefore, because of these advantages, wormhole routing is an ideal candidate switching technique for on-chip multiprocessor interconnects networks. Fig. 2 shows a router with virtual channel.

3 OASIS NoC Architecture

OASIS NoC is a 4x4 mesh network and adopts wormhole switching. Data path of one single switch is shown in Fig. 3. Input signal is 76 bit flit and stop signal at one input port, the maximum number of ports is five. OASIS is structured by three main modules. In input port, flits data is fetched and decoded to determine new next port direction which is needed next switch. Next port information is transmitted from input port to switch allocator, and decode to decide packet whose output direction. Crossbar send packet to next appropriate switch input.

3.1 Switching

OASIS NoC employs wormhole routing. The size of one flit is 76 bit, and it has information of destination address, next port direction information and payload. Flits are transmitted one by one as shown in Fig.4. If flit has no meaningful data, no need to buffer it into FIFO. Flits are buffered in input port FIFO as shown in Fig. 7. Each input port has FIFO which is 76 and depth is 4. Depth of the FIFO vastly concern area of switch, but performance may be increase.

3.2 Topology

OASIS NoC is a 4x4 mesh as shown in Fig.5. Each switch has X-Y coordinates which is called address, and some input port whose numbers are different by X-Y coordinates, for instance, switch at (0, 0) has 3 input ports which are
3.3 Routing

The next port direction in next address is decided in input port. To compare next address and destination address, next port in next address is decided. If Y of destination address is larger than Y of next address, next port is SOUTH in a contrasting situation, next port is NORTH. If X of destination address is larger than X of next address, next port is EAST in a contrasting situation, next port is WEST. When next address and destination address is equal, next port is SELF. This information is sent into switch allocator. Direction of flit in input port whose output port and when it should be output is decided in switch allocator. If some flits which in another input port request same output port, transmitting may be block. To solve this problem, function of arbiter is used. Arbiter determines high priority flit in input port, OASIS arbiter schedules in round-robin scheme, this scheme serves each input port in fair.

Fig. 8 indicates switch allocator. Fig. 9 shows crossbar used in the switch. This module sort which data send into next port, SOUTH, NORTH, WEST, EAST and LOCAL by using control signal from switch allocator. Fig. 6 shows the Stall-Go mechanism.

4 Preliminary Evaluation

We have designed OASIS NoC in Verilog HDL and synthesized with Altera CAD tools. We studied its complexity of two systems: 1x1 and 2x2 mesh based architectures. Area, speed and power over different optimization are investigated. Tables 1 shows the hardware design results. Fig. 10 shows 2x2 OASIS chip floorplan.

---

1 Stratix III indicate area as ALUTs
5 Conclusion

In this paper, we presented architecture, hardware design and preliminary evaluation results of a complexity effective network on chip architecture, named OASIS. The system was designed using Verilog HDL and was correctly synthesized with Altera CAD tools. OASIS system uses wormhole routing and only input ports have buffers.

We analyzed the complexity of OASIS in terms of speed and area over various optimization. From the preliminary evaluation, we conclude that OASIS can be efficiently used for the design of systems consisting of several cores. Thanks to the adopted complexity effective techniques. Thus, scalability, performance of the target system are guaranteed.

In our future work, we plan to simulate the system with real workloads so that we can evaluate its real performance. Several optimizations will be also investigated.

Table 1: OASIS Synthesis Result and Complexity Evaluation.

<table>
<thead>
<tr>
<th>System</th>
<th>Area</th>
<th>Power (W)</th>
<th>Speed (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 1</td>
<td>221</td>
<td>0.616</td>
<td>138.7</td>
</tr>
<tr>
<td>2 x 2</td>
<td>3.017</td>
<td>0.632</td>
<td>211.7</td>
</tr>
</tbody>
</table>

References


