A Packet-switched Interconnect for Many-core Systems with BE and RT Service

Runan Ma
School of Microelectronics
Fudan University
ShangHai 200433, China
Wuxi Institute of Fudan University
WuXi 214131, China
mrndreamaker@gmail.com

Zhida Hui
Wuxi Institute of Fudan University
WuXi 214131, China
zhidahui@gmail.com

Axel Jantsch
Institute of Computer Technology
Vienna University of Technology
Vienna 1040, Austria
Wuxi Institute of Fudan University
WuXi 214131, China
axel.jantsch@tuwien.ac.at

Abstract—A packet-switched interconnect design which supports real-time and best-effort services is proposed. This interconnect is different from traditional NoCs in that we use direction channels to replace the large input buffers and use less resource to realize the network transfer. The connection between our interconnect design and IP core is an on-chip memory management block named DME. The real-time service implies preferential transfer channel allocation, maximum delay bound and time stamping of every real-time packet. The solution is geared towards many-core systems, such as complex industrial control systems and communication devices, which require these features to facilitate efficient SW and application development.

Keywords—NoC router; packet-switched; real-time; best-effort;

I. INTRODUCTION

The rapid development of integrated circuit technology enables more and more computing and storage resources to be integrated on a single chip. So in the embedded applications multi-core or many-core SoC have become reality and drive the change from bus-based to network-based interconnects, also known as Networks on Chip (NoC). Traditional buses have two main deficiencies which are poor scalability and low quality of service (QoS), while NoC has several advantages which are good scalability, no global wiring, global asynchronous and local synchronous communication, inherently tolerant against faults, natural support for QoS differentiation, etc. [1][2]. However, NoC also brings new challenges, e.g. good support for real-time transfer between many-core processors.

Many traditional NoC routers use virtual channels which leads to a large buffer sizes [4-7]. Besides, the arbitration mechanism cannot guarantee sufficient parallel transfers and provide real-time service. In contrast,, our design handles those inadequacies well. We use direction channels to replace the large virtual channels and can arbitrate the transfers in parallel so that it can realize group arbitration.

Our proposed design connects IP cores through a data management engine (DME) [3] which performs on-chip memory management and supports foreground/background memory access and time stamp adding for real-time services.

The packet-switched interconnect and the DME have been implemented on the Zynq platform for applications which contain large number of sub-systems with high communication demands. Also, they offer APIs that facilitate the SW integration and support the handling of real-time requirements by the control SW. The time stamping feature enables the SW to check the validity of data by inspecting the time stamps.

II. RELATED WORK

A growing number of research institutions and chip makers are aware of the potential of NoC and have invested in it. Peh and Dally had made a delay model for the pipeline of the router and put forward the use of a speculative crossbar distributor to reduce a pipeline stage [4]. Mullins and others put forward the use of speculative crossbar transmission further to reduce a pipeline stage on the basis of Peh and Dally’s work [5]. Kumar et al designed a router structure with single time cycle and they made the crossbar distributor and the virtual channel merge into single pipeline stage [6]. Kim proposed a low cost and low delay routing structure for dimension order routing algorithm based on priority arbitration [7]. Similarly, Dimitrakopoulos and others presented a low delay priority crossbar switch distributor based on permutation network [8].

Based on the works they have done, we put forward a switch which reduces input buffer size and supports real-time service.

III. TOPOLOGICAL STRUCTURE

Topology has significant influence on the performance and cost of entire network. It determines average delay of the network hop and the length of links from one node to the next. Topology also determines the number of paths between different nodes and the available bandwidth of the whole network. In order to make place and route easy, a simple topology is commonly used. We also use a 2-D mesh topology as Fig. 1 shows, where we can easily realize a deadlock-free routing algorithm. But when the system scale increases further to some size, the middle section of the whole network will be quickly saturated so that the performance of the whole system is reduced.
IV. ROUTING ALGORITHM

A suitable routing algorithm is crucial for the interconnect design and the performance of the network. It determines the transfer path for every message. We choose dimension order, deterministic routing as routing algorithm, which is deadlock free.

V. DATA TRANSFER MODE

Our switch design supports BE (Best-Effort) and RT (Real-Time) services. RT transfer has higher priority than BE transfer. BE does not provide any timing or QoS (quality of service) guarantees. In a BE network all users obtain unspecified transfer rate and transfer time, depending on the current traffic load. An RT service offers a preferential allocation of resources leading to a minimum guaranteed bandwidth and shorter average delay. Together with a design method that limits the amount of RT traffic, it guarantees maximum delay bounds independent of the dynamic traffic load at any particular time. It is compulsory for RT applications which are required to produce results within a predefined time interval. In addition, the RT communication adds a time stamp to data packets which informs the receiver when data has been generated. This allows the receiver to distinguish between invalid (old) and valid data. In some applications this is an important feature that facilitates discarding of invalid data.

VI. FLOW CONTROL

Store-and-forward, virtual cut-through and wormhole switching are three main flow controls of message transfer. We choose wormhole switching as our flow control mechanism which transfers, stores and allocates flits during data transfer. As long as the next switch or router has at least one available buffer, the flit can be transferred. When there is no network congestion, wormhole switching is the same as virtual cut-through. Compared with other flow control mechanisms, wormhole switching has an advantage that it reduces the demand of buffer size. So we use wormhole switching as our flow control mechanism.

VII. MICRO ARCHITECTURE

The micro architecture of the packet-switched interconnect is showed in Fig. 2 below.

Our interconnect design is called “Switch” and consists of two module types “Input module” and “Arbiter module”. Input module can dispatch packets to different direction channels according to routing analysis result. The Arbiter module provides fair arbitration similar to a conventional matrix arbiter. This architecture can realize parallel transfer and can also support grouping arbitration, as Fig. 3 shows. The input north and south just have two direction channels because of the XY dimension routing algorithm that once the X-axis transfer has been done, the packet just will be transferred north or south or local. This interconnect method can avoid large resource consumption and a complicated design.

When we transfer packets from several input ports to the same output port, such as input ports W, E, L transfer packets to output port N in the same cycle, the west arbiter module will give one of them the priority to occupy the output channel. Without mutual interference, input ports N and S also can transfer packets to the other output port L, the east arbiter will decide which one can occupy the output buffer. This kind of grouping arbitration improves the network throughput.

A. Input Module

The input module consists of direction channel allocator (DC Allocator), synchronous FIFOs and some control circuits. Every flit of packet is stored in the first buffer and then will be
stored in one of the direction FIFOs based on the transfer
destination and the state of FIFO that as long as the FIFO has
one vacancy, and there is no need to wait for whole FIFO
empty. In our design and test, we choose the FIFO depth four,
which also means the maximum length of packet. Traditional
NoC router has a large input buffer size because every input
port has several virtual channels. However, XY dimension
routing algorithm leads to a result that X-axis transfers are
commonly busier than Y-axis transfers. So there is no need to
put extra virtual channels in north and south input ports.
Furthermore, traditional NoC router uses virtual channel
allocator, switch allocator, arbiter and a crossbar to calculate
and allocate flits to the relevant output. However, in our
design, we just use direction channel allocator and three
separated arbiter to realize same function.

B. Arbiter module

This module judges, determines and grants one of the
inputs the priority to occupy the output channel when several
transfer requests are coming. It has two different kinds of
units which are two inputs arbiter and four inputs arbiter. Both
of them always grant real-time packets higher priority over
best-effort packets and employ round-robin arbitration
between packets of the same traffic priority class. However,
rigid round-robin is modified by giving the most recently
served port the lowest priority, which increases the fairness of
the arbitration as illustrated in Fig. 4.

Fig. 4.  The grouping arbitration

Viewed from Fig. 4, it becomes obvious that simple round
robin priority arbitration has an unavoidable problem when all
IP/cores send packets to the same destination and keep these
transfer requests in a long period, the IP/core “A” will be
always chosen. On the contrary, the modified rotary priority
arbitration does not run into this problem. But if not all input
ports send requests to the same output port, the arbitration
unfairness is also inevitable.

VIII. HARDWARE COST

We use Zynq7000 FPGA to synthesize the circuit and
analyze the performance. The resource cost and speed of
different logic synthesis are listed in TABLE 1.

<table>
<thead>
<tr>
<th>Synthesis</th>
<th>Slice Register</th>
<th>Slice LUTs</th>
<th>Maximum Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing First</td>
<td>2476 (2%)</td>
<td>2393 (4%)</td>
<td>336.912MHz</td>
</tr>
<tr>
<td>Area First</td>
<td>313 (0%)</td>
<td>1163 (2%)</td>
<td>267.666MHz</td>
</tr>
</tbody>
</table>

IX. EVALUATION

We promote a simple method that it can make the
evaluation more sufficient with high coverage so that we can
find the advantage of using our design which sets two kinds of
packet priority: real-time and best-effort. Our analysis bases
on the nine nodes 2-D mesh structure which is showed in
Fig. 5. According to the connection, there are three kinds of
node: Circular is node A, triangle is node B and rectangle is
node C in gray. Each kind of node has its own transfer paths.
If the network has no any transfer conflict that these three
types of nodes will show the different average delay for one
packet.

Fig. 5.  The switch-based interconnect: 9 nodes 2-D mesh.

A. No-load Average Delay

We separately transferred 100 best-effort packets that each
of them contains four flits from node A to different
destinations with no conflicts. So does the node C and node C.
TABLE II shows and compares the number of cycles for these
three kinds of nodes.

<table>
<thead>
<tr>
<th>Node Kind</th>
<th>Node A</th>
<th>Node B</th>
<th>Node C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay Cycles</td>
<td>1278</td>
<td>1050</td>
<td>1161</td>
</tr>
<tr>
<td>Average Delay</td>
<td>12.78</td>
<td>10.50</td>
<td>11.61</td>
</tr>
</tbody>
</table>

As we can see, node locates at centre network will have
small average delay. So, if the network is not loaded, the
average delay is 11.63 cycles.

B. No-Priority Network Performance

The switch enables multiple cores to run programs in
parallel but in a synchronized manner. We generate a random
injection start time for each node and then inject best-effort
packets from different node at different start time. We test six
different injection rates that we inject one packet every several
cycles for each node from every 10 cycles to every cycle.
Based on the test, when the injection rate is less than 30, the
curve is similar to linear line. With the injection rate
increasing, the network congestion is getting aggravated...
which leads to no input vacant buffers for new injection packets. It is obvious for us to know this variation from Fig. 6.

![Fig. 6. The average delay of no-priority transmission](image1)

C. Priority Network Performance

The switch based interconnect can support time sensitive real-time communication. We inject real-time packets into one of the nodes and best-effort packets into the rest of nodes. The node which injects real-time packets sends packets to the other eight nodes. When several packets try to jump into the same node, the arbiter will give the real-time packet higher priority to occupy the data path. That is to say, if there is only one node injected real-time packets and other nodes injected best-effort packets, the real-time packets are unobstructed in network leading to no violations of time constraints. The average delay for the real-time packet is a constant no matter how much best-effort load is on the system. In contrast, the delay for the best-effort packets will be aggravated because of the existing of RT packets. The results are showed in Fig. 7.

![Fig. 7. The average delay of no-priority transmission](image2)

X. CONCLUSION

The switch based interconnect avoids large input buffers and supports grouping arbitration. This design scales well and can guarantee quality of service in real-time transfers.

XI. ACKNOWLEDGMENT

This paper is partially funded by the 1000 Talents Plan Project. Project Name: Recruitment Program of Foreign Experts (1000 Talent Plan), shorted as RPFE. Project No. WQ20123200043.

REFERENCES


