An Asynchronous On-Chip Network Router with Quality-of-Service (QoS) Support

Tomaz Felicijan and Steve B. Furber
The University of Manchester
Manchester, UK

Abstract

This paper presents an asynchronous on-chip network router with Quality-of-Service (QoS) support. The router uses a virtual channel architecture with a priority-based scheduler to differentiate between multiple connections with various QoS requirements sharing the same physical channel. A gate-level prototype of the router has been built and its functionality and performance evaluated. The simulations show that the router is capable of offering a high-level of QoS within the capacity limitations of the network.

Introduction

Networks-on-Chip (NoCs) are emerging as a new design paradigm to tackle the challenge of managing the complexity of designing chips containing billions of transistors. The idea is to divide a chip into several independent components or IP (intellectual property) blocks connected by a communication architecture. Each component performs a well-defined function and can be designed independently using standard design tools. To reduce time-to-market, existing IP blocks can be reused or bought from different IP vendors.

A modern System-on-a-Chip (SoC) design represents a heterogeneous environment with various components interacting in many different ways (event-driven, data streaming, message passing, shared memory, etc.) [1]. Some of these have strict traffic requirements and constraints, and require guaranteed services, such as minimum throughput and bounded communication latency. It is therefore essential for an interconnect to provide QoS capabilities in order to accommodate different components in the same network [2].

However, the adoption of NoCs as the solution for global interconnect still raises the question of which clocking strategy to use. While local wires scale in length with a technology, global wires spanning an entire chip do not - exactly the situation that leads to clock skew problems.

One way to eliminate clock-skew is to use asynchronous logic for a NoC. This leaves only the issue of connecting synchronous IP blocks to an asynchronous network. Interfacing clocked and self-timed circuits is a well-understood discipline for which standard solutions exist [3]. Furthermore, properties such as low power, improved electro-magnetic compatibility (EMC) and robustness, offer additional benefits from the use of self-timed logic for on-chip interconnect.

The work presented in this paper introduces a prototype architecture of an asynchronous on-chip network router with QoS support.

Quality-of-Service (QoS)

In essence, providing QoS requires reserving a certain proportion of network resource for a particular connection. Those resources consist of buffer space and bandwidth.

Reserving bandwidth in synchronous networks is usually done by *time division multiplexing* (TDM) where the time axis is partitioned into time-slots each of which represents a unit of time when a single connection can transmit data over a physical channel. The bandwidth is reserved by dedicating a proportion of time-slots to a particular connection.

In asynchronous networks the TDM is not applicable because it requires global synchronization between network elements. Another way to reserve bandwidth is to employ a scheduling algorithm that will prioritize input requests according to the level of QoS required.

While the scheduling policy plays a major role in the QoS provided by the network, it is only effective if there is sufficient memory space available to store incoming packets. When the amount of incoming traffic exceeds the bandwidth of an output channel it is inevitable that some inputs are served before the others. In this case the pending packets have to be temporarily stored in input buffers until they are forwarded to the next node.

Buffer management has to solve two problems to effectively support QoS: (1) it has to provide enough memory space to accommodate any excess of input traffic with guaranteed services, and (2) it needs to ensure that high-priority packets do not get stuck behind the blocked low-priority ones, a situation called head-of-line (HOL) blocking.

A. Guaranteed Services and Best-effort Services

If the reservation of network resources has been made a service is guaranteed (GS), otherwise it is a best-effort (BE) service. In practice a GS constrains all packets of a flow to follow the same route. As a consequence, a virtual path between a source and a destination has to be established, and all the packets that belong to the flow must follow it.

The downside of using GS is that it requires resource

0-7803-8445-8/04/$20.00 ©2004 IEEE

reservation for worst-case traffic scenarios. This leads to inefficient utilization of the network resources because on average, the amount of traffic is lower than in the worst-case. BE services do not reserve resources and hence can have a better average resource utilization, at the expense of unpredictable worst-case behavior. To improve the link efficiency the residue of the physical bandwidth which is not used by the GS traffic has to be available for the BE traffic.

B. QoS Architecture

Fig. 1 shows a QoS architecture using virtual channels [4] to differentiate between individual connections sharing the same link. Instead of implementing a conventional input buffer organization where each input is associated with a FIFO queue, an input channel is associated with several lanes of small FIFO buffers in parallel (virtual channels). The buffers in each lane can be allocated independently of the buffers in any other lane. A blocked packet holds only a single lane idle and can be passed using any other lanes.

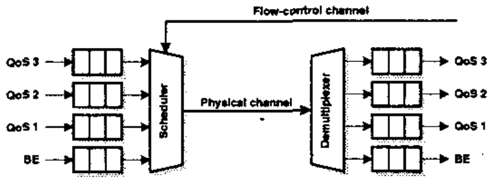

Fig.1. QoS architecture using virtual channels.

Each QoS connection is assigned to an individual virtual channel and best-effort traffic shares a single virtual channel. The network is therefore able to support N-1 QoS connections where N equals the number of virtual channels.

When a new packet arrives at the router it is assigned a virtual channel at the appropriate output port according to the information saved in the header of the packet. This assignment persists until the last flit of the packet leaves the network node. If the particular virtual channel is already engaged the packet is blocked until the channel is released. The packet traverses the network following the same procedure at every node on its path until it reaches its destination node.

This buffer organization provides means to establish a virtual path from a source node to a destination node and, consequently, to allocate buffer space for a particular connection using the path. As long as the network prevents the rest of the traffic from maliciously using a particular virtual channel, the connection will have the buffer resources available at any given time.

The scheduler allocates bandwidth on a per-flit basis according to the priority level of a virtual channel. This implies that a higher priority packet preempts transmission of a current packet with a lower priority. The flow control ensures that only virtual channels with free buffer space at the receiver compete for the physical channel.

Router Architecture

The router presented here has four bi-directional network ports to form a two-dimensional mesh network and a bi-directional service port to enable clients to inject and eject packets to/from the network.

A top-level schematic of the implementation is shown in Fig. 2. The router consists of four main components: an input port controller (IPC), an output port controller (OPC), a switch and a route management unit (RMU). For clarity, the figure shows only one IPC and one OPC; there are actually 5 instances of each controller implemented in the design.

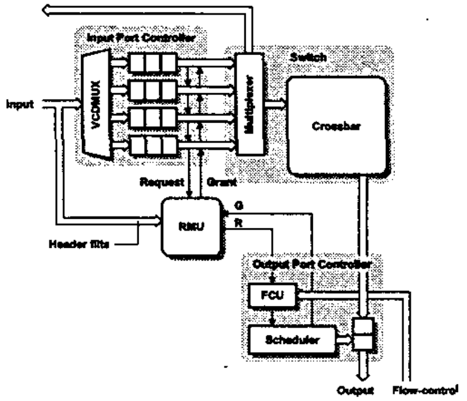

Fig. 2. Top-level schematic of the router.

A. Switching

Switching is closely related to the internal flow-control of a network and has a great impact on the amount of buffering required in individual network nodes. Buffer space in a NoC directly impacts the silicon area overhead of the network and must therefore be kept to a minimum.

Wormhole switching [5] reduces the theoretical minimum buffer space to one flit per virtual channel, however in this case the input buffers are able to store up to three flits for two reasons: (1) to decouple the input link from the switch thus introducing more concurrency into the design, and (2) to close down the flow-control loop between two neighboring network nodes in order for a single connection to be able to use 100% of the available physical bandwidth.

B. Packet organization

Determining the right packet size is crucial to make optimum use of the network resources. The optimum size depends highly on the characteristics of the application. If a message has to be split into too many packets the overhead of disassembling and reassembling them might be too high. On the other hand, if the packet is too large it might block other traffic affecting the performance of the system.

A variable-packet-length organization is proposed in this work in order to improve the flexibility of the network. This way, it can be decided dynamically how to split a message into packets in order to achieve the best performance.

C. Flow-control

The router employs a credit based flow-control mechanism (FCU in Fig.3) to prevent data being sent to a full buffer. Each virtual channel has a separate credit based counter which is decremented when a request is forwarded to the scheduler. If the counter is zero the request is blocked until new credits are received from the receiving node.

D. Switch

The router employs a 5-by-5 multiplexed crossbar switch. Based on a restriction that packets are not allowed to be sent back to the source node, the crossbar is only partly connected to minimize the silicon area.

In synchronous network routers there is usually a single control unit which schedules packets through the crossbar. The controller has a global knowledge of all inputs and is thus able to optimize the sequence in which packets traverse the crossbar to achieve optimal throughput and prevent contention between the virtual channels sharing the same input port.

In asynchronous networks this is rather impractical because it would require synchronization between all of the inputs. Therefore, each output of the switch has a separate controller.

E. Scheduling

The scheduler uses a low latency asynchronous arbiter with a fixed priority algorithm [6]. The function of the output buffers in Fig. 3 is to decouple an arbitration cycle from an output transaction cycle. This way the system is capable of pipelined operation performing arbitration for the next flit while transmitting the current flit over a network link. Furthermore, if the arbitration is faster than the output transaction cycle the system can allocate more than 50% of the output bandwidth to a single contender [6].

F. Routing

A dimensional ordered routing algorithm has been implemented in the design because it offers a deadlock-free and livelock-free operation with deterministic behavior and is relatively simple to implement in hardware.

G. Implementation

The router is implemented using a quasi-delay-insensitive (QDI) [7] technology generally employing 1-of-4 data encoding with a return-to-zero signaling protocol [7]. The data path is eight bits wide composed of four 1-of-4 QDI channels with a common acknowledge signal.

Evaluation

To evaluate the performance of the router a small test network was constructed (Fig. 3). Four connections (QoS3, QoS2, QoS1 and BE) were routed through the network in such a way that at least one network link is shared among all the connections. The shared link represents a bottleneck, as all the connections have to compete for the same resource. By measuring throughput, latency and jitter of each connection it is possible to determine the level of QoS the router is capable of providing.

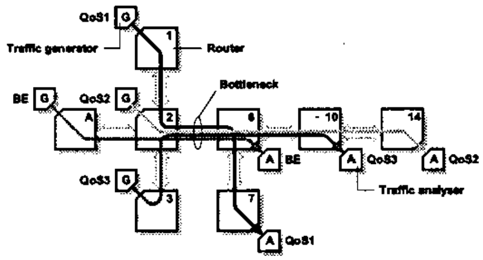

Fig. 3. Test network.

Three connections with different QoS requirements (QoS3, QoS2 and QoS1) were set to generate various types of network traffic typically present in a modern SoC.

The QoS3 connection represents a real-time control channel with low, bounded latency requirements. The traffic source generates short packets (five flits) with a constant packet rate acquiring 5% of the bandwidth. Data is injected into the network with the maximum flit-rate using the highest priority virtual channel (VC3) to achieve low latency.

The QoS2 connection models a constant bit-rate (CBR) data stream similar to the uncompressed output of an audio device. The source generates packets of five flits with a constant flit-rate acquiring 25% of the physical bandwidth using the second priority channel (VC2). The connection requires guaranteed throughput and controlled jitter.

The traffic model of the QoS1 connection is based on MPEG-4 video traces and represents a variable bit-rate (VBR) data stream. The maximum bit-rate generated by the source represents 68% of the available bandwidth, however the average bit-rate uses only 25% of the bandwidth. The length of the packets varies from 134 to 1741 flits. Consequently, the data is injected into the network with a variable flit-rate. The connection requires guaranteed throughput with controlled jitter.

According to the bit-rates given above the maximum aggregate throughput of the QoS traffic requires almost all of the physical link bandwidth (98%). However, the average utilization of the link is merely 55%, while the rest of the bandwidth is still available for the BE traffic.

The following set of simulations was conducted to test whether the network is capable of accommodating multiple connections with mixed traffic characteristics and QoS constraints in parallel with the BE traffic. The sources of QoS3, QoS2 and QoS1 were configured according to the specifications explained above and the BE source was set to inject fixed length packets (10 flits) exponentially distributed across the time axis. The total workload of the network was varied by changing the mean bit-rate of the BE source.

A. Throughput

Fig. 4 shows the throughput of each individual connection versus the BE traffic demand. The throughput is measured as the number of bytes transmitted between a pair of network nodes per unit of time represented as a normalized value against the physical bandwidth. The results show the router allocates the residue of the bandwidth (not used by the priority packets) to the BE packets without affecting the throughput of the QoS traffic.

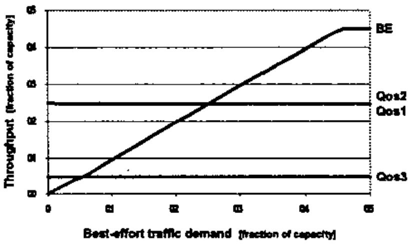

Fig. 4. Throughput versus best-effort (BE) traffic demand.

B. Latency

The measured latency of each packet represents the difference between when the packet was created and when the last flit of the packet has left the network. Consequently, a longer packet with a lower bit-rate would normally exhibit longer end-to-end latency.

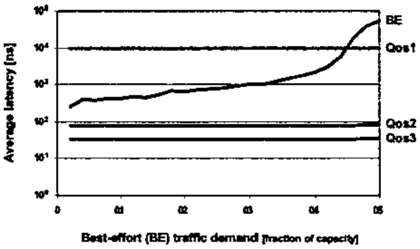

Fig. 5. Average end-to-end latency versus best-effort (BE) traffic demand.

Fig. 5 shows the results of the average end-to-end latency of each connection. Note that the Y-axis represents a logarithmic scale to accommodate a large range of values. Again, the graph shows that the latency of the QoS3, QoS2 and QoS1 connections remains almost constant regardless of the level of the BE traffic injected into the network, while the average latency of the BE packets increases rapidly as the traffic load approaches the physical limits of the network.

C. Jitter

Variation in packet delay or jitter is measured as the difference between the maximum and the minimum latency of the packets tied to a logical flow of data.

The network generates a relatively small amount of jitter which is practically unaffected by the BE traffic, as shown in Fig. 6. For example, the QoS1 connection would require a 10-flit output buffer in order to smooth out the jitter generated by the network. This represents less than 1% of the longest packet generated by the source.

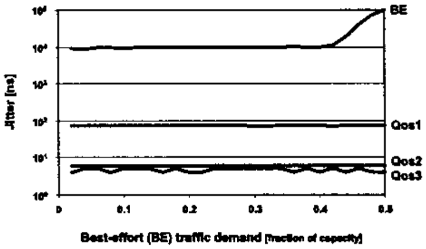

Fig. 6. Jitter versus best-effort (BE) traffic demand.

Conclusions

This paper has presented a prototype of an asynchronous on-chip network router with QoS support. The router employs virtual channels and a priority based algorithm to differentiate between the GS and BE traffic.

The simulation results show that the network is capable of providing guaranteed services, namely minimum throughput and bounded communication latency for individual connections as long as the GS traffic is carefully managed and does not exceed the physical limitations of the network. Furthermore, the router is able to assign unused bandwidth to BE packets without affecting the GS traffic, thus giving better than worst-case utilization of network resources.

References

[1] M. Sgroi, *et al.*, "Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design," *DAC'2001*, pp. 667-672, June 2001.

[2] K. Goossens, *et al.*, "Network on Silicon: Combining Best Effort and Guaranteed Service," *In Proc. of DATE*, pp. 423-425, March 2002.

[3] S. Moore, "Point to Point GALS Interconnects," *In Proceedings of ASYNC'02*, pp. 69-75, April 2002.

[4] W. J. Dally, "Virtual-Channel Flow Control," *IEEE Tran. on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194-204, March 1992.

[5] W. J. Dally and C. L. Seitz, "The Torus Routing Chip," *Distributed Computing*, vol. 1, pp. 187-196, 1986.

[6] T. Felicijan, *et al.*, "An Asynchronous Low Latency Arbiter for QoS Applications," *In Proc. of ICM'03*, pp 123-126, December 2003.

[7] J. Sparsø and S. Furber, Principles of Asynchronous Circuit Design: A System Perspective, Kluwer Academic Publishers, 2001.