

Asynchronous Network-on-Chip Communication Architecture Performance Analysis

Bas Bijlsma and René van Leuken

Faculty of EEMCS/ME/CAS, Delft University of Technology

Mekelweg 4, 2628 CD Delft, the Netherlands

Abstract—Network-on-Chip (NoC) designs attempt to solve the performance bottleneck of traditional shared bus designs. Current NoC solutions have drawbacks in area requirements or power consumption. A new architecture is presented to address these problems. Three models of the architecture are implemented in SystemC: a Globally Asynchronous Locally Synchronous (GALS) NoC model, a synchronous NoC model and a shared bus model. The performance of these three models are compared using Poisson distributed traffic with data-rates up to 4 Gbyte/s. Results show a strong case in favor of NoC design over the standard shared bus in large or high-bandwidth systems. For NoC systems, the performance break-even size (number of nodes) compared to a shared bus can be as low as 5 masters, while at a size of 15 nodes the NoC is already about 300% faster than the shared bus. The GALS NoC shows a 24-27% improvement in the delay time compared to the synchronous NoC. The additional design effort and the high bit-error sensitivity in the GALS case need to be addressed before the GALS NoC can be considered as candidate for future on-chip communication.

Keywords—Asynchronous Logic Circuit Testing, Network-on-Chip, Performance Analysis, SystemC Model

I. INTRODUCTION

With the increasing complexity of chip design, due to increasing transistor densities and Deep Sub-Micron (DSM) problems, the pressure on the on-chip communications backbone is becoming too big to handle by conventional structures. Conventional structures are typically busses in all kinds and shapes, but they share a single treat; multiple masters sharing a single channel. This severely limits the scalability of the bus, because with an increasing number of masters the bus will inevitably slow down the traffic.

A proposed solution is called Network-on-Chip, changing communications from a single line to a grid of lines, and changing the arbitration problem to a routing problem. The communicating end-systems are usually called functional units.

Existing NoC solutions, like the Nostrum [1] and the Æthereal [2], [3], have drawbacks in size and power-consumption respectively. To remove these draw-

backs, a new architecture is proposed, smaller in size and including a Globally Asynchronous, Locally Synchronous (GALS) communication model to lower power consumption. The GALS model is simulated together with a shared synchronous bus structure and a synchronous model of the same NoC architecture.

According to simulation results of Zeferino [4], the break-even point between the shared bus and a NoC is around 24 master functional units. Since we employ a different NoC concept, we expect to find a lower break-even point. The goal of this paper is to answer the question whether the GALS NoC has a performance gain compared to the synchronous NoC model, and if that gain is sufficient to justify increased design time.

II. BACKGROUND

This section contains background information about two NoC designs, the Nostrum and the Æthereal. The drawbacks of these designs lead a formulation of goals of a new NoC architecture. At the end of this section design decisions are discussed.

A. Existing NoC structures

The Nostrum NoC is developed by the Royal Institute of Technology (KTH) in Sweden. It is based on a 2D-mesh structure, with routing locally in the nodes. Routing is based on a deflective or 'hot-potato' style, and packages therefore are datagrams or one Space-Time-Unit (STU) large. To lower the relative addressing overhead, channels are 128-bit wide in each direction. With differential coded lines and extra shielding, this comes down to about 1200 lines between each node.

The nodes are synchronous and Time Division Multiple Access (TDMA) techniques are used to create Temporally Disjoint Networks (TDN). These TDN hold multiple non-overlapping Virtual Channels (VC) at the same point in time. To increase the number of TDN, a number of buffer stages are introduced. But this buffering destroys the greatest benefit of deflective routing, which can work without buffers. Without

this benefit, a disadvantage of the deflective routing comes back into view; the deflection cost. If a package is deflected at the destination, it has to follow a minimum route before it returns to the destination. This deflection cost has a negative effect on average delay time.

The most important disadvantages of the Nostrum NoC are the large size, the inability to work with burst traffic and the inefficient use of deflective routing. The basic ideas behind the Nostrum are sound.

The *Æthereal* is the NoC approach of Philips. It has no fixed layout, and can be used in a 2D-mesh structure. It uses sequentialization to reduce the number of lines and supports burst traffic. In a synthesized example, bus lines were 32-bit wide and packets 8 STU long.

The *Æthereal* is designed with Guaranteed Bandwidth (GB) traffic in mind, using TDMA techniques to reserve slots for priority traffic. The remaining slots are used for non-priority traffic and is handled with Best Effort (BE). Packaging and de-packaging is included in the Network Interfaces (NI), which requires a parallel-to-serial conversion and vice-versa. Each node has a FIFO buffer, 8 STU deep.

As a consequence of buffering and the TDMA techniques, delay times go up. To reduce this, higher clock frequencies could be used, but the price is a vastly higher power consumption, since dynamic power consumption is related to the square of the frequency.

We have implemented two NoC models based on a the basic principles of both Nostrum and *Æthereal*, but simplified to avoid the associated problems. We have implemented one additional model based on a shared bus to compare the three simulation results.

B. Design Decisions

The conclusions that can be drawn from the Nostrum and *Æthereal* NoC are the use of small channels, e.g. 32-bit width, with burst packages of a limited length. Instead of using a deflective routing algorithm, a fixed shortest-path algorithm is used. To avoid high power consumption, buffering is limited and asynchronous communication is used. According to Oberg [5] the clock lines can use up to 50% of the total power budget. Asynchronous designing brings a variety of different challenges, which has a negative influence on design time. Since delay insensitive design is used, the handshake between two nodes has to be 4-phase at least, because the second part of the handshake cycle is needed to reset the validation signals.

To compare the GALS NoC design with a synchronous model, the latter is also created. The synchronous model is based on the same architecture in order to determine the synchrony impact. But since a synchronous design is not confined to using 4-phase handshakes, a 2-phase synchronous handshake is used. This increases throughput while reducing dynamic power consumption.

A choice has to be made between different layout possibilities; the 2D-mesh, the 2D-torus and the binary fat-tree. While the torus has the best theoretical specifications for small to medium size (number of nodes is around 64) the 2D-mesh is still chosen. This is mostly because the torus is difficult to build in CMOS technology. Node interleaving is required for all the connecting lines to be of equal length. And the torus only performs better when long paths are used. In general layout algorithms restrict the locality.

III. MODELS

In total three models are created. One shared bus model, one GALS NoC model and one synchronous NoC model. In a simulation environment these three are connected to data-generating (master) and absorbing (slave) functional units. The interface of these units is based on the Wishbone interface, as described in [6]. The shared bus model is a Wishbone bus, and combines a synchronous handshake with an asynchronous setup. It supports burst-traffic and allows throttle control by both master and slave.

A. Shared Bus Model

The shared bus model is based on the Wishbone bus specification [6]. The Wishbone bus is a multiplexed bus, using a multiplexer between the master output and the slave input, and a multiplexer between slave output and master input. A simple random arbitration is created using a mutual exclusion (MUTEX) element.

The asynchronous part of the wishbone handshake gives rise to a timing constraint. One full loop from master-slave-master must be achievable within one clock cycle.

B. NoC Models

Both GALS NoC and synchronous NoC have the same architecture. Each NoC consists of Nodes and Channels. A Channel has of a request, acknowledge and data part in one direction. The Node connects channels from 5 directions, called after the four cardinal directions and a destination direction as shown in

Figure 1. The functional units of the proposed NoC are implemented with Wishbone compliant interfaces, and Network Interfaces (NI) are used to translate from Wishbone protocol to the NoC protocol. This demonstrates the general nature of NoC. Each direction has

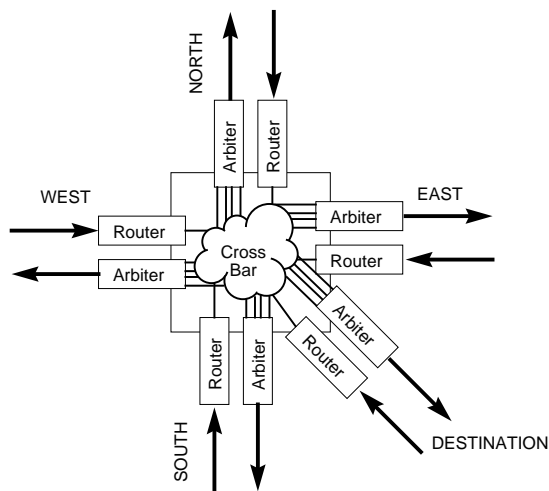


Fig. 1. Global Node Layout

a Router-input and an Arbiter-output. The Router makes the routing decision and changes the address accordingly. A simple ΔXY routing scheme is used, with the destination reached when ΔXY is zero¹. The Arbiter decides which channel gains access to the output. Arbitration (or contention) is on a random basis, using a MUTEX.

For the asynchronous or GALS case, both Router and Arbiter are based on a micro-pipeline structure by Sutherland [7], using Muller C-elements for the handshake. Buffering is provided with hold-pass latches, and internally validation signals govern the processing order. This is according to the rules of delay-insensitive design. For the GALS NoC the Router is moderately complex, while the Arbiter is simple. However, it took a substantial design effort to achieve this.

The synchronous NoC design is based on the GALS design, but uses flipflops in combination with XOR-elements to achieve 2-phase handshaking. One minor disadvantage of this simplistic approach is a passing burst must leave the node in the same state it was entered. This means a burst has to be of even length (header plus an odd number of data STU). Design time was low, but while the Router complexity was reduced compared to the GALS Router, the Arbiter complexity increased somewhat.

¹Actually using a XY-routing scheme would simplify the Router even more.

The two designs of the NoC are comparable in complexity and area estimation², but the asynchronous design took more time to create. The use of delay-insensitive design strategies make the asynchronous design sensitive to bit-errors.

C. Simulation environment

The performance analysis is based on a statistical analysis with masters generating and slaves receiving data. At the generation time a timestamp is stored, and when the total data is received another timestamp is recorded. The difference between these two is referred to as delay time. Data is generated with a Poisson distribution, which has an exponential distributed inter-generation time. The distribution is used because it has proved in the field of telecommunications to be a good model for many data-transfer situations. The rate in which data is generated can be adjusted by a parameter λ (λ is taken as the number of packets generated each $100\mu s$).

In the shared bus simulation, a fixed number of masters is set, and an equal number of slaves. Measurements are averaged over all master-slave pairs. In the NoC simulations, only one measuring master-slave pair was used, randomly placed in the network but with a fixed distance. The rest of the nodes are connected to background masters that generate traffic with a random destination throughout the NoC. The component (logical) delay times are based on a Xilinx Virtex-4 FPGA in 90 nm copper CMOS technology, running on a clock of maximum 500 MHz. Where the Virtex-4 data-sheets were not applicable, estimations were used. The line delay is calculated using a simple lumped T-network RC model, and values calculated with ADS Momentum. Because some values in this model had to be estimated using technology data-sheets, a best and a worst case scenario is taken, keeping in mind the best case is more likely because these channels are probably optimally designed. Based on a line of 2mm length, copied from a synthesized Nostrum example, the total line delay time³ is 10 ps in the best case and 500 ps in worst case.

IV. RESULTS

A set of simulations was run with varying values for λ and size (number of masters). For small sizes and/or low data-rates, the bus outperforms both NoC designs. Figures 2 and 3 show delay versus the number

²Due to the limitations of SystemC, no synthesis could be performed.

³Defined as the 10-90% signal transition time.

of active masters, the sharp rising line represents the bus, while the top horizontal line represents the synchronous NoC. But as Figure 2 shows, for a medium

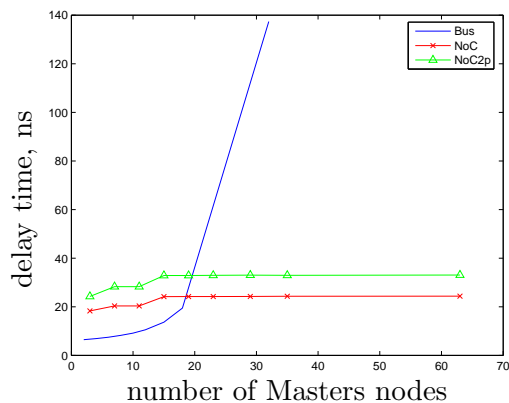


Fig. 2. Comparing Delay time for $\lambda = 10$

data-rate ($\lambda = 10$ compares with 120 Mb/s) the NoC has a lower average delay time than the bus if there are more than 20 masters. While for a high data-rate ($\lambda = 100$) the break-even size becomes 5 or 6 masters. The difference between the GALS NoC and the synchronous NoC, in both cases is about 25% of the total delay time, while the difference between the NoC models and the shared bus model is about 300% for a size of 15 masters. Similar comparisons, but with the

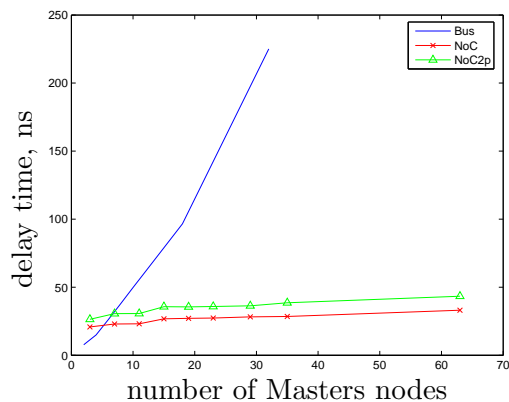


Fig. 3. Comparing Delay time for $\lambda = 100$

larger line delay, show the NoC structures to be almost unaffected, while the bus clock frequency needs to be scaled down to retain the timing specification of Wishbone.

In terms of throughput and saturation behavior, the NoC models are far superior to the shared bus, but equally matched with each other.

V. CONCLUSIONS

The simulations show the superior performance of a NoC circuit for large and/or high bandwidth situations compared to the shared bus. The break-even point which lay around 24 according to Zeferino [4] is found to be strongly dependent on the used bandwidth. Under the right conditions the break-even point can drop to 5 or 6 masters. For a size of 15 masters and a high data-rate, the NoC performs roughly 300% better than the shared bus. The GALS NoC has a 24-27% better performance in terms of delay time compared to the synchronous NoC.

Both NoCs compare very well to each other in terms of throughput, the synchronous NoC actually performs slightly better, but not significantly. Both NoC models perform about 500% better than the shared bus in throughput at high data-rate and a size of 15 masters. In terms of area cost, both NoCs are evenly matched, although they were not actually synthesized. But in design effort cost, the GALS NoC took far more time to create. The GALS NoC have a higher susceptibility for errors on the communication lines, due to the delay-insensitive design where signals can wait indefinitely. These situations of deadlock need to be detected and solved. Only then is the GALS NoC a good candidate for future on-chip communication.

REFERENCES

- [1] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip," in *Design, Automation and Test in Europe Conference Proceedings*, IEEE, February 2004.
- [2] A. Rădulescu, J. Dielissen, K. Goossens, and E. Rijpkema, "An efficient on-chip network interface offering guaranteed services, shared-memory abstraction, and flexible network configuration," in *Design, Automation and Test in Europe Conference Proceedings*, IEEE, February 2004.
- [3] E. Rijpkema and et al., "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," in *Design, Automation and Test in Europe Conference Proceedings*, IEEE, September 2003.
- [4] C. A. Zeferino, M. E. Kreutz, L. Carro, and A. A. Susin, "A study on communication issues for systems-on-chip," in *Symposium on Integrated Circuits and Systems Design Proceedings*, IEEE, 2002.
- [5] J. Oberg, "Clocking strategies for networks on chip," in *Networks on Chip*, Kluwer Academic Publishers, 2003.
- [6] OpenCores, *WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores*, September 2002. revision B.3.
- [7] I. E. Sutherland, "Micropipelines," *Communications of the ACM*, June 1989.