Switched Interconnect for System-on-a-Chip Designs

Daniel Wiklund and Dake Liu
Dept. of Physics and Measurement Technology
Linköping University
S-581 83 Linköping
{danwi,dake}@ifm.liu.se

Abstract

With the increased use of IP cores in chip designs, an increasing amount of time is spent on design and verification of glue logic. To solve this problem together with the bottleneck problem of arbitration based buses, a novel approach in system-on-a-chip interconnect has been investigated. The approach is based on a switched interconnect structure, with small crossbar switches connected in a mesh for intercore communications with low latency in system-on-chip solutions.

The interfaces between the interconnect network and the cores are handled by configurable wrappers that adapt the port parameters from core to network format.

The core functionality of the interconnect network can be fully verified with a fairly low work effort even when configurable, so the main problem for cutting verification time is the quite complex wrappers. The concept is to make the wrappers highly configurable yet needing short verification time in an application by making a fairly complete verification of the wrappers for all configurations. How this can be achieved is under investigation.

The approach described in this paper is mainly aimed for use in communication equipment where high bandwidth and low latency is essential.

1. Introduction

With the recent advances in communication systems, the classical way of building electronics is becoming infeasible due to the inherent problems with speed, power consumption, physical size, and so forth. These problems have forced the industry to use higher and higher levels of integration to keep up with competition and customer demands. Nowadays, many products are implemented using only a few or even a single chip. The trend towards higher integration is associated with new problems such as design complexity, higher economic risk, heavy verification work, as well as physical problems like power and clock distribution. Since many chip designs use intellectual property cores, the main problem beside verification of the IP cores is the complexity of glue logic and interconnects. Something has to be done to cut the design and verification complexity of the glue logic and interconnects in order to allow even larger designs to be accomplished successfully on time.

The trend towards more and more battery powered designs where power efficiency is a main objective implies that the clock rates in the system should not be unnecessarily high. This together with the problem of clock distribution will lead to designs that may use several clock domains. Thus a good way to bridge between clock domains is needed.

The current bus solutions for system-on-a-chip designs are almost all based on a shared synchronous media and uses arbitration to allow several initiators to use the bus. This is the classical way of implementing interconnects that has been used for many years. This bus type

is rapidly becoming a major bottleneck for system performance in system-on-a-chip solutions as an increasing amount of data is transferred between different parts of a chip. Especially, it is impossible to use a classical bus for central services in communication, such as a radio base station or a gateway for voice over IP. In order to increase the performance of interconnections, buses based on arbitration must be exchanged for novel architectures that allow higher compound bandwidth as well as greater simultaneous connectivity.

The concept of networks in system-on-chip design has a lot of similarities with networks for parallel computers, although many requirements differ between the two areas. The single largest difference is that while computers emphasize on compound bandwidth, latencies are often the most critical in communications. There is also a big difference in that the jobs are better known at the design stage in applications in the communications field than for computers. The routing algorithms for parallel computer networks also require high tolerance against dynamic faults such as broken cables and nodes. This is a fairly small problem in on-chip networks since the system is less susceptible to failures.

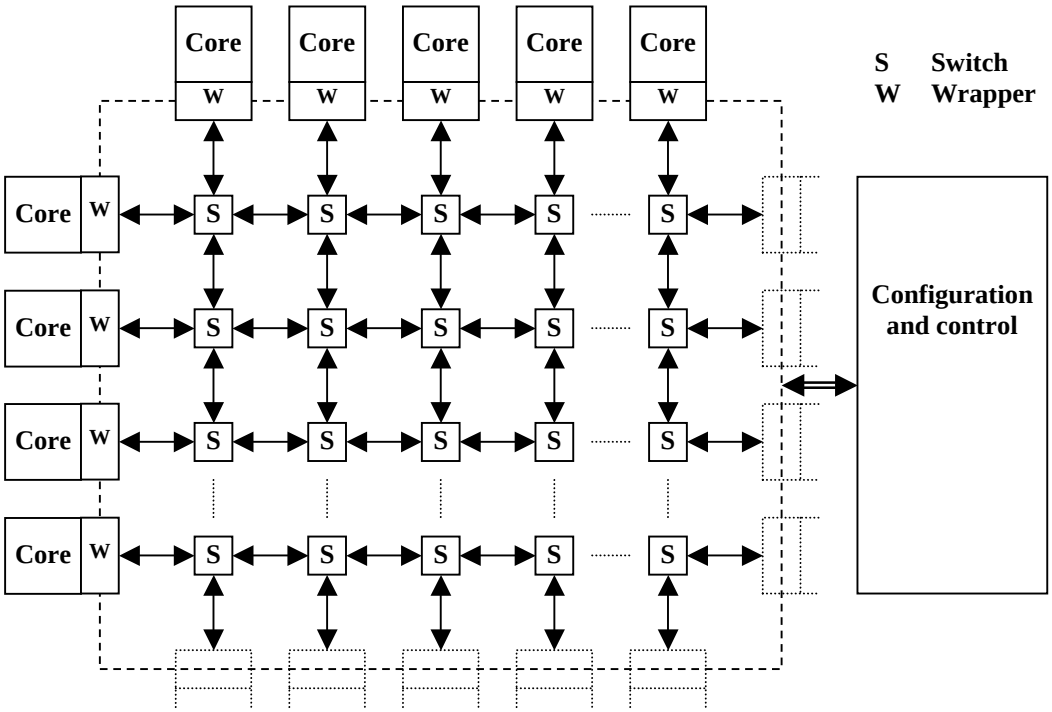

Figure 1. Interconnect system overview

2. Bus System Overview

The bus system considered consists of four major parts. These are the sender and receiver wrappers, the network and the switching nodes as found in the overview of the system in figure 1. The purpose of each of these parts will be explained later in this paper. Wherever it is needed, a core may have several wrappers connected, both senders and receivers, in order to allow the communication that is necessary for the core.

The dataflow in the system is that when a core needs to send data over the bus system, a route is first set up to the receiver. This route is then locked during the entire time that a transfer is in progress. After that, the actual payload will be delivered through the interconnection network and finally the route is cancelled and the resources that were in use will be available

again. Since the system should have as low latency as possible, a circuit switched technique is used. Packet switched systems tend to have too much overhead in packet header processing and buffering. A circuit switched system can also guarantee the latency and available bandwidth for one transmission since no other transmissions can use the route concurrently.

The tasks of the wrappers are to adapt the port and transfer parameters between the cores and the interconnect network. The main parameters that need to be handled are data width, data rate, clocking and synchronization. The configuration and control block performs general control tasks such as supervising the network, doing initial setup and run-time configuration. This block also monitors the network for faults.

3. Network Architectures

There are several network topologies that are worth considering for the bus system. The most commonly seen in parallel computers are fat trees, meshes and tori [2]. A fat tree is the best choice from a connectivity point of view, but the fat trees are complex when seen from a wiring perspective and will thus consume more silicon area.

The network architecture that most closely resembles the layout of a chip is a mesh. The dimension of the mesh is limited to a fairly low number in order to fit on a chip without problems. Higher dimensions lead to better performance but also lead to more complex switching nodes. The simplest solution here is to use a two-dimensional mesh with common 4-input / 4-output crossbar switches and links that allow duplex DMA connections. The links may be either true full duplex, but may just as well use time division duplex or some other technique to allow for full duplex at the DMA level.

For a small system (up to some 8 or 16 nodes), it is possible to use a fat tree as network topology. For larger systems, a 2-D mesh is more suitable since it has simple wiring while using a single switch type. This circumvents the problem with extensive and complicated wiring to achieve good connectivity since this problem does not occur in a low-dimensional mesh.

4. Switching Node

The actual routing in the network is done in the switching nodes. The nodes consist of a crossbar switch, a routing controller and some buffers (if needed). The crossbar does the actual switching while the routing controller decides what way to route incoming data through the crossbar. The crossbar is able to route incoming traffic to any output except the port that leads back to the previous node. More than one route can go through the crossbar as long as each route uses input and output ports that are not occupied by another route.

Buffers in the switches should be avoided, since buffering inevitably will increase latencies in the network. Buffering is almost always needed in a packet switched network because the switches must buffer the incoming packet whenever there is congestion or if the routing decision takes time. In order to keep a low latency and high bandwidth in the network, it is better to use a circuit switching strategy that sets up an entire route and then sends the data as quickly as possible along the route. The route is then released immediately when the transfer has ended so that the resources become free as soon as possible.

5. Routing Algorithms

Routing algorithms for interconnect networks come in many flavors. One major feature that distinguishes different algorithms is whether the algorithm uses deterministic or adaptive (dynamic) routing. Deterministic algorithms always use the same route between two ports, while a dynamic algorithm can use different routing each time. Other important issues in routing are deadlock and indefinite postponement. Since the algorithm should be robust considering these issues, the algorithm must be both fair and free from deadlocks.

Simple deterministic routing algorithms, such as the west-first algorithm are easy to show if they are fair and deadlock free or not. The west-first algorithm (also known as X-Y routing) [2] has both of these qualities, but it is too simple to be usable in the system outlined in this paper. The major flaws are that it has no built-in fault tolerance and that it is very inflexible.

Dynamic algorithms generally have better performance under heavy load than the simpler deterministic ones. The trade-off here is that a dynamic algorithm calls for a more complicated and possibly slower routing controller in the switching nodes. Another drawback is that a dynamic algorithm is much more complicated than a deterministic algorithm when it comes to deadlock issues. The concept of proving deadlock freedom is much harder with a dynamic algorithm than for a deterministic one.

A suitable routing algorithm is a variation of the pipelined circuit-switching algorithm found in the work by Gaughan and Yalamanchili [3]. If a deterministic algorithm is to be used, the routing algorithm developed by Badr and Podar [4] is suitable. Since both of these algorithms are developed for parallel computers, some modifications will probably be needed to adapt them to the bus system. Both algorithms have some built-in fault tolerance that makes the system more robust against failures in the network.

6. Wrappers for Network-Core Connections

In order to allow IP cores from different vendors and with different interfaces to coexist on the bus, the wrappers need to be very flexible. This flexibility is achieved by making the wrappers highly configurable. This in turn leads to big problems at the verification stage where the wrappers must be proven correct for every possible configuration. If this is not done, the wrappers and the bus system must be verified in every application it is used in.

The wrappers should take care of such tasks as synchronization, clocking and adoption of data parameters. All of these can be of very various types. Synchronization can for example be achieved by using one or more extra signals (e.g. strobe or mask), self clocked data or through an asynchronous handshaking protocol. Clocking can also be done in many ways, such as positive or negative slope, dual edge and self clocked data. All this lead to a very complicated wrapper that needs to be simplified at the design-time configuration in order to save die area and power. Since all the different configurations will lead to different simplifications at the ASIC design phase, it is very hard to prove that the system is correct in all cases. Verification of these kind of configurable systems is an area where much more research needs to be done. The current standardization effort by the VSI Alliance is interesting since it may simplify the wrappers considerably if most IP providers will follow the standards.

7. Configuration

Configuration of the bus system must be done at several levels. First, at the ASIC design stage where the ports of each wrapper are configured in accordance with the ports of the corresponding core. This may include such things as port widths, clocking and synchronization. The second level of configuration is done during the software design stage, where the system is directed towards a specific application with e.g. fixed routes. Finally, the system is configured while the system is running. This includes things like on-the-fly routing.

Many parts of the configuration could be done at more than one level. What to configure at each level is determined by trade-off between flexibility and complexity. The later the configuration is done the more flexible system you get, but at the expense of possibly larger silicon area and higher power consumption.

8. Related Work

The concept of switched networks for on-chip solutions has surfaced quite recently. Because of this there is not very much work available on these kinds of systems. One of the few is the work by Guerrier and Greiner [1]. They have reached a high bandwidth, strong connectivity, and reasonable silicon area. The drawback is that they use a packet switched fat tree topology that has the inherent drawbacks of a quite high average latency and a probability to get prohibitively high latencies for some packets. The maximum usage is in practice also limited to approximately 40%-50%. This means that their solution is not very suitable for chips in communication systems.

9. Conclusions

For an efficient interconnect network to be designed, a lot of issues need to be considered. The most promising design consists of a 2-D mesh with crossbar switches that uses adaptive routing. This leads to simple physical routing of wires on the chip combined with good performance and fault tolerance. For smaller system, a fat tree can be considered instead of the 2-D mesh, but with considerable differences in the routing compared to the mesh.

In order to allow different IP cores to connect to the bus system, wrappers need to be designed that are highly configurable in respect of data parameters, clocking, synchronization and so forth. The verification of these wrappers is a major obstacle since they have to be fully verified in order to allow easy integration with the other parts of a chip.

In most respects, the configurability must be high for the bus system to be usable in as many applications as possible. While this work primarily emphasizes communications, computers can also be considered a candidate for this system.

10. References

[1] Pierre Guerrier and Alain Greiner. A Generic Architecture for On-Chip Packet-Switched Interconnections. *Proceedings of the Design, Automation and Test in Europe Conference 2000*. IEEE, 2000.

[2] David E. Culler, Jaswinder Pal Singh and Anoop Gupta. *Parallel Computer Architecture, A hardware/software approach*. Morgan Kaufmann Publishers Inc., 1999. ISBN 1-55860-343-3.

[3] Patrick T. Gaughan and Sudhakar Yalamanchili. Pipelined Circuit-Switching: A Fault-tolerant Variant of Wormhole Routing, *Proceedings of the fourth IEEE Int'l Symposium on Parallel and Distributed Processing*. IEEE, 1992.
[4] Hussein G. Badr and Sunil Podar. An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies, *IEEE transactions on computers, October 1989*. IEEE, 1989.