Interconnect IP for Gigascale System-on-Chip

Ilkka Saastamoinen, [*] Teemu Suutari,** Jouni Isoaho,*** and Jari Nurmi ****

Abstract – **Today's electronic systems merge multimedia, data and signal processing, and digital communication functions on single die. Thus these systems are becoming true SoC (System-on-Chip) designs. This development is making traditional design and implementation methods limited and insufficient. Implementing an interconnect structure of a complex SoC is one of the issues which cannot be solved without new methods and solutions. In this paper a flexible structure for on-chip communication between IP (Intellectual Property) blocks in gigatransistor SoCs is introduced. This approach uses communication scheme based on packet transfer and error correction. Data transfers between communicating components are implemented using high-speed serial communication lines and asynchronous block interfaces, which themselves form IP for flexible interconnect implementation.**

1 Introduction

The future SoC implementations in year 2005 and beyond will be made in technologies with minimum feature size in the range of 0.05 – 0.10 µm. The application specific integrated circuits will be up to a billion transistor systems operating at 1 GHz in very demanding self-induced noise conditions. The design is no more a block level problem, but a global communication issue. The key elements in achieving functional silicon with high operating frequency and low power consumption will be the on-chip communications and interconnects.

2 System-on-Chip Communication Architectures

Due to noise constraints on the large SoCs, the communication channels will not form wide buses as currently implemented on-chip and on PCBs. Thus, performance optimization will be very different and the overall communication will resemble more computer networking than traditional bus based design. In fact, the System-on-Chip will resemble a computer network also since it will contain several processor cores with shared/distributed memory. The use of processors and configurable communications will be the key to flexibility. Reconfigurable processors and application specific processors as a part of the integration platforms can further add flexibility to the system. One of the central ideas is to implement the custom logic blocks as peripherals or (application specific or reconfigurable) processors. The system would thus ideally consist of processor-based subsystems.

The functionality is assumed to be available as IP blocks, but the communication architecture needs to be established between the blocks. Such architecture can be, e.g., a ring, a mesh, a collection of trees or point-to-point links, or any combination of these.

3 Interconnect IP

The realization of the communication architecture consists of interconnect IP which needs to be represented and abstracted for the system-level design. Due to the changing characteristics of the communication in SoC, alternate realizations of system level synchronization to the subsystems is essential. For the synchronization, the necessary condition for flexible communication architectures is abandoning synchronization with respect to fixed global clock signal. More flexible and protocol based synchronization and arbitration schemes must thus be developed, such as GALS (Globally Asynchronous, Locally Synchronous). The existing IP needs to be encapsulated for use in the applied communication scheme.

In order to solve the obvious design bottleneck, especially in back-end verification, the global interconnects need to be treated as similar IP blocks as processor cores or embedded memories. This will result in significant decrease of the verification and validation times as well as improve testability.

As an alternative to traditional bus based on-chip communication, a switching network is a solution proposed to combat the complexities of next

[*] Tampere University of Technology, Institute of Digital and Computer Systems, P.O.Box 553, FIN-33101 Tampere, Finland. E-mail: ilkka.saastamoinen@tut.fi, Tel: +358-3-3654335, Fax +358-3-3653095.
** University of Turku, Laboratory of Electronics and Information Technology, Lemminkäisenkatu 14-18 C, FIN-20014 Turku University, Finland. E-mail: teemu.suutari@utu.fi, Tel: +358-2-3336939, Fax +358-2-3336950.
*** University of Turku, Laboratory of Electronics and Information Technology, Lemminkäisenkatu 14-18 C, FIN-20014 Turku University, Finland. E-mail: jouni.isoaho@utu.fi, Tel: +358-2-3336947, Fax +358-2-3336950.
**** Tampere University of Technology, Institute of Digital and Computer Systems, P.O.Box 553, FIN-33101 Tampere, Finland. E-mail: jari.nurmi@tut.fi, Tel: +358-3-3653884, Fax: +358-3-3653095.

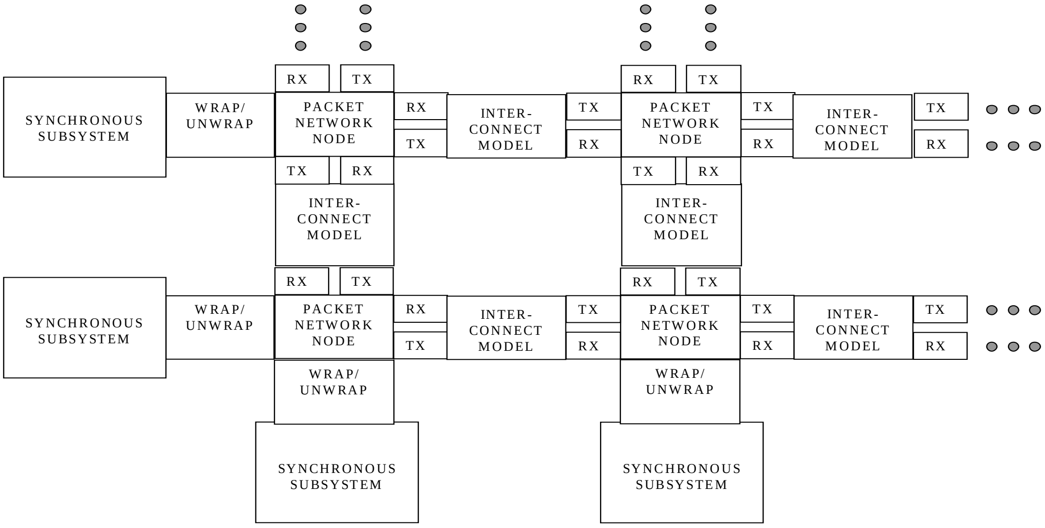

Figure 1: Packet switched network application of the described interconnect IP.

generation SoCs [1, 2, 3]. There are two main switching methods for network communication: circuit switching and packet switching. The former involves establishing a virtual circuit between the source and destination communicating devices, thereby reserving a physical channel between them during the data transfer. This scheme supports a steady flow of bits and has a small number of routing decisions per connection. However, this scheme tends to be inefficient both in time used to form each virtual channel and in total system bandwidth. Alternatively, a packet switching network breaks each message into sequences of small packets with network address and control information. At every node of the network, a routing function decides the correct output port to route the packet.

The selected approach in this paper is the packet switching network, which is more flexible and adaptable than circuit switching method. Thus it is more suitable for a system where requirements may vary remarkably. The general structure of the network is presented in Fig. 1. The basic element of the network is a switching node, which handles the routing of packets. Each node is connected to other nodes through receiving and transmitting serial high-speed interconnects with error correction. The subsystems are the sources and destinations of the communication. They are connected to the communication network through adaptive interfaces, wrappers.

We must remember, however, that the transfer time of packets will vary with time. Therefore, packet switched approach is not the best solution for any applications with real-time data transfers.

4 Implementations of Interconnect IP

The implemented interconnect IP should have certain properties to be useful.

1. The system must be flexible and adaptable. This means that the internal structure must be generic and configurable to support a wide and varying range of communicating blocks with different performance requirements and communication protocols.
2. Integration of different subsystems with different interfaces should be as simple as possible.
3. The system must be able to tolerate increasing on-chip noise.
4. The complexity of the circuit – and thus area and costs - must be minimized.
5. Globally synchronous communication must be avoided for power consumption, noise and performance reasons.

4. 1 Estimation and Modeling

The behavior of a data transfer from a source to a destination depends heavily not only on the characteristics of that transfer but also on the whole network load at that moment. Therefore, modeling of certain network structure characteristics must use as realistic models of data transfers as possible. Because the system is very complex, the estimation cannot be done using very precise models, but it must be run at a high abstraction level.

The system has generic transfer models for the subsystems and more precise behavioral models for the network nodes and the TX/RX components. The nature of data activity in each subsystem is defined using the parameters of the data transfer model. Another part of the estimation system tracks traffic in the network IPs. Using these collected traffic statistics the congestion problems in the network can be detected and the routing scheme and the network structure can be modified according to that.

Also the physical properties of the interconnection links need to be modeled for system-level design and verification. This has been described, e.g., in [4],

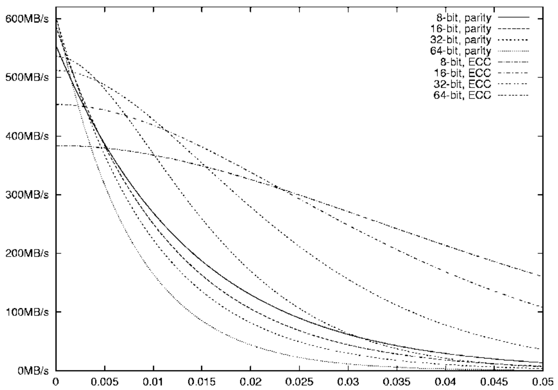

Figure 2: Throughput as a function of bit error rate with parity and ECC-protected data. Cases of multiple re-sends are not considered in calculations.

where delay analysis for different topologies and technologies has been carried out.

The use or Error Correction Coding (ECC) on top of the interconnects has a dramatic effect, as shown in Fig.2 [5]. Next, a communication scheme employing ECC embedded in a serial communication link is described.

4.2 High-speed Serial Communication Scheme

The TX/RX components shown in Fig. 3 [5] hide many details from the upper level. The clocks of different subsystems need not to be synchronized, error correction is performed on the fly, and only termination needs to be set according to the wire length.

The scheme used consists of an asynchronous 32-bit interface, low voltage differential signaling, and embedded error correction circuitry. The TX/RX components only implement the lowest level of communication protocol. As the basic component is only a one-way transmission device, it requires a two-way communication protocol on top of it, capable of re-transmission of missed or corrupted data.

Most of the TX/RX circuitry area is consumed by a large cascade and the termination. As a single transmitter or receiver is a relatively small device concerning the area consumed and has low static power consumption, these components are ideal for use with large SoCs. As a coupled serial line requires a fraction of the chip area compared to parallel wiring, the IC can probably be smaller, and serial lines are also easier to route.

The on-chip communication using this approach is very robust as seen from the eye diagrams in Fig. 4. These TX/RX blocks can be used as IP in building the communication network in the selected approach.

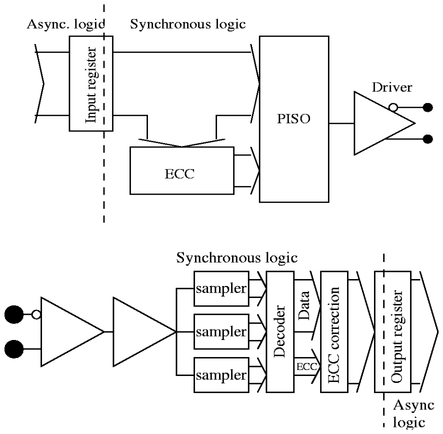

Figure 3: The transmitter and the receiver. The impedance matching circuits are not shown.

Furthermore, the blocks are adaptable for off-chip communications, enabling scalability of the system even beyond the future single-chip integration capabilities.

The transmission protocol is divided in four layers, of which TX/RX components implement two. The layers are: physical, transmission protocol and ECC, re-transmission protocol (trusted transfer), and data transmission protocol. The protocol layers are illustrated in Fig. 5.

4.3 Subsystem Interface

Each subsystem is connected to the network via an interface unit, built into the node. This interface provides the necessary transformation between the subsystem and the network space. In VSIA documents [6] a virtual interface for IP blocks is presented. In [7] a VSIA compliant GBI (Generic Bus Interface) unit is implemented. It connects IP block with standard interface to several commercial on-chip buses. The reverse of this approach will be used in the wrappers of communication IP. Wrappers must be designed in a way that the standard interface of the switching network can be connected to subsystems which have non-standard interfaces. The wrappers must be tailored to the needs of the subsystems, instead of the other way around. Functions of the wrapper are:

- Provide protocol translation.
- Convert data to packet format with header, data, and error correction fields and vice versa.
- Buffer incoming data flow.
- Emulate hardware signaling.
- Provide address translation between subsystem address space and network node space.

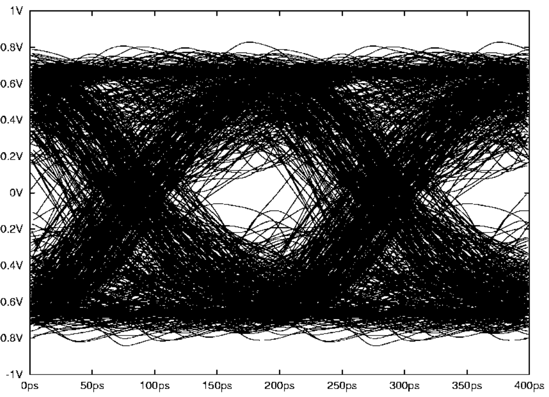

Figure 4: Eye diagram of the signal at the end of the line (on-chip wire 2.5 mm).

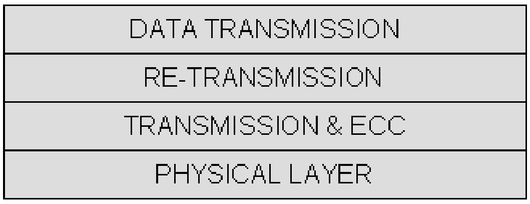

Figure 5: Network protocol stack representation.

4.4 Network Nodes

The routing algorithm and the choice of the node architecture is one of the key factors influencing system performance and cost. Buffering delays at each node represent increasing transfer latency. Instead of buffering the entire packet, an alternative strategy can be adopted. There the packet is routed before the tail is received. In this method the header is inspected and a routing decision made as soon as the suitable output port is available.

Addressing becomes an important issue since we have two domains of address space. Each subsystem connected to the network will have its own address space. It should operate independently of network node addressing since blocks are meant to be integrated into the network system with no interface modification required.

The complexity of the packet network and the number of different routing possibilities is so large, that only small systems can be constructed by hand. For real gigascale systems an automatic generation environment is needed. However, it is good to remember that also the generation of an efficient on-chip network is a very complex task, which demands a lot of future research. Also, reducing the complexity overhead of switching networks offers challenges. This remains as future work.

5 Conclusions

The concept of a generic packet switching network approach for complex system interconnections was presented. The network can be implemented out of interconnection IP blocks. This embedded structure offers improved flexibility in many areas including scalability, addressing methods and general IP component interfacing. The main strength of the proposed packet switched structure is flexible connectivity of subsystems. The interface wrappers allow integration of subsystems with minimal modification required.

The used asynchronous serial communication medium achieves relatively high performance. The effects of noise and crosstalk are reduced by using error correction and differential signaling.

Acknowledgement

This work was supported at TUT by the Academy of Finland under project #47587.

The team wishes to thank Chris Kearney for his contribution to the project.

References

[1] J. A. J. Leijten et al, “Stream Communication between Real-Time Tasks in a High-Performance Multiprocessor,” Proc. DATE 1998, Paris, France, February 1998, pp. 125-131.

[2] P. Guerrier and A. Greiner, “A Generic Architecture for On-Chip Packet-Switched Interconnections, “ Proc. DATE 2000, Paris, France, March 2000, pp. 250-256.

[3] A. Hemani et al, “Network on a Chip: An Architecture for Billion Transistor Era,” Proc. NORCHIP 2000, Turku, Finland, November 2000, pp. 166-173.

[4] T. Nurmi, S. Virtanen, J. Isoaho and H. Tenhunen, ”Physical Modeling and System Level Performance Characterization of a Protocol Processing Architecture,” Proc. NORCHIP 2000, Turku, Finland, November 2000, pp. 294-301

[5] T. Suutari, J. Isoaho and H. Tenhunen, ”High-speed Serial Communication With Error Correction Using 0.25 µm CMOS Technology,” Proc. ISCAS 2001, Sydney, Australia, May 2001.

[6] Virtual Socket Interface Alliance, “Virtual Component Interface Specification OCB 2 1.0,” September 1999.

[7] J. Lähteenmäki, M. Kuulusa, I. Saastamoinen, and J. Nurmi, ”General Bus Interface for IP Design,” Proc. International Workshop on IP Based Synthesis and System Design, Grenoble, France, December 1999.