Implementation of a behavioral simulator for on-chip switched networks

Daniel Wiklund

Dept. of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden
danwi@isy.liu.se

ABSTRACT

As the degree of integration increases, the on-chip communication is becoming a bottleneck. A solution to this problem is to use an on-chip switched interconnect network. Such a system-on-chip network was proposed in 2000 by the same authors. One of the main aids in design of this system is a behavioral simulator. A general purpose event driven simulator kernel has been implemented as a basis for the behavioral simulator development. This paper presents the issues involved in the design and implementation of the network behavioral simulator in detail.

1. INTRODUCTION

As more and more transistors are available on chip the integration of systems onto chips continue in an ever increasing rate. When multiple processors are integrated the traditional design using time-division multiplexed buses, see fig 1a, will soon be a major bottleneck for on-chip system performance because of the shared media. A better infrastructure for on-chip communication is a switched network similar to those used for parallel computer systems, see fig 1b, that minimizes the problem of shared resources.

We have previously proposed the use of a two-dimensional switched network for system-on-chip communication [1]. Later work by Dally and Towles [2] and Sgroi et al. [3] confirm the general trend towards on-chip networks.

The expected result of this research project is a general platform for system-on-chip integration using the two-dimensional network as the interconnect structure. The platform should be easy to use and ease the burden of verification during the system-on-chip integration phase.

Daniel Wiklund is a Ph.D. student at the division of Computer Engineering, Dept of Electrical Engineering, Linköping University.

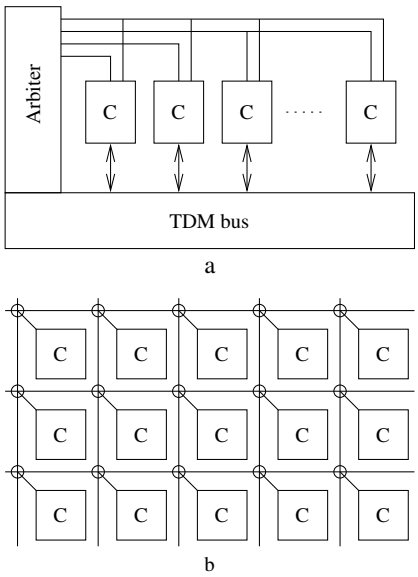

Fig. 1. Comparison of traditional TDM bus and a switched interconnect bus replacement

2. NETWORK COMPONENTS

The network can be seen as a two-dimensional matrix of tiles shown in fig 2. These tiles are made up of three components except for the IP core. These are the switch nodes, the source wrappers, and the drain wrappers. In addition to these tiles there is a simple control circuit with global responsibility.

The wrappers are used to isolate the internal format of the network from the port format of the custom cores. This isolation is achieved by using the highly configurable wrappers to adapt the port parameters to the network. The implementation decisions affecting the wrappers have a very high impact on system integration verification time and should be designed to reduce the verification time.

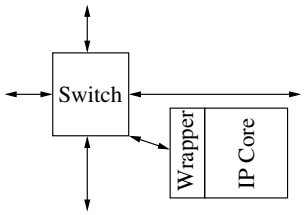

Fig. 2. Network connected processing tile

The switch nodes are responsible for routing the data transport streams between source and drain ports. The choice of implementation of the switch nodes mainly affect the connectivity and routing latency in the network.

3. BEHAVIORAL SIMULATOR

Because of the vast space of different design and implementation decisions involved in such a network, a behavioral simulator has a very high importance in the design flow for system integration using the switched system-on-chip network. The same behavioral simulator can also be used to evaluate and select the different design parameters, e.g. routing algorithms and switching schemes, while designing the actual network.

The input to the behavioral simulator is the code for switches (including the routing code) and the code for source and drain wrappers. The code containing the configuration (network size etc.) is also input to the simulator. There is also a need for an user supplied description of the transmission traffic patterns to get best simulation results although different standard benchmarking patterns can be used (e.g. uniform traffic).

The simulator kernel is then responsible to run the complete network including both control setup and data transport for a period of time and to check the data to ensure functional correctness. The simulator also collects performance measurements such as momentarily and average available bandwidth, maximum and average latency, and contention measures.

At system integration time, the primary output of the simulator is used to evaluate the three main areas of interest in implementation. These are the correctness, the performance of the network, and the cost to implement and verify the network, see table 1. The configuration and verification cost is estimated from the amount of change in the network system needed for the current configuration. During development of the network components this can be used for evaluation of design choices that affect different costs and benchmarks to allow for trade-offs between different design issues such as increased cost of switches compared to increased connectivity and data throughput.

Table 1. Output generated by the behavioral simulator

Benchmark	Connectivity eval
	Data throughput eval
	Latency evaluation
Functionality	Correctness
Cost	Circuit selection cost
	Protocol selection cost
	Control subsystem cost
	Configuration cost
	Verification cost

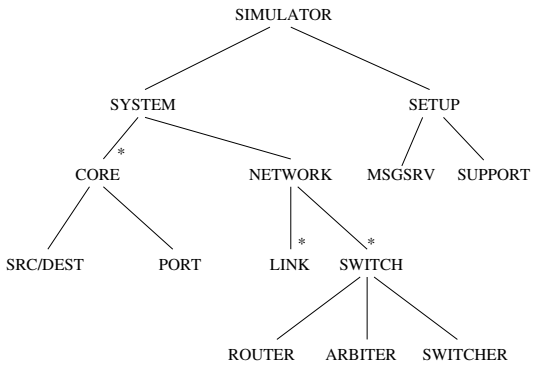

Fig. 3. Object structure of the behavioral simulator network model

4. BEHAVIORAL SIMULATOR IMPLEMENTATION

The behavioral simulator is developed using object oriented programming in C++. The simulator is implemented using a message passing structure that resembles the structure of the real network system. The simulator is event-driven and both bit and cycle true in its execution.

The object structure of the simulator can be found in figure 3. The leaf nodes in the system branch corresponds to the actual network components except for the switch that is subdivided into its subtask components where the arbiter is for local arbiting between the ports of a switch. The setup branch is the message passing support for the simulator. This program structure is very suitable for the simulation of an essentially message-passing hardware like the on-chip

network presented in this paper.

The kernel is implemented using an event mechanism similar to those used in VHDL simulators using event lists and delta cycles to keep track of absolute time ordering of events. This implementation is very simple and efficient while allowing considerable flexibility in the simulation setup. It is also very general allowing the simulator kernel to be used in other simulators if needed.

5. CONCLUSIONS

A simple simulation kernel has been implemented using C++. This kernel is then used to build a behavioral simulator for the evaluation of system-on-chip switched network implementation decisions. This simulator is useful both in network development and in the customer flow to evaluate performance/cost trade-offs in specific implementations.

6. ACKNOWLEDGMENT

This work is being supported by the Swedish Foundation for Strategic Research (SSF) through the Integrated Electronics (Intelect) program and is supervised by prof Dake Liu.

7. REFERENCES

[1] D. Wiklund and D. Liu, "Switched interconnect for system-on-a-chip designs," in *Proc of the IP2000 Europe conference*, 2000.

[2] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc of the DAC*, 2001.

[3] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vincentelli, "Addressing the system-on-chip interconnect woes through communication-based design," in *Proc of the DAC*, 2001.