Interfacing SoCLib CABA models with NoCBench for NoC perfomance evaluation

Muhammad Moazam Azeem, Alexandre Briere, Manuel Bouyer, Julien Denoulet, Francois Pecheux, Andrea Pinna, Bertrand Granado

Interfacing SoCLib CABA models with NoCBench for NoC perfomance evaluation

Muhammad Moazam AZEEM, Alexandre BRIERE, Manuel BOUYER,
Julien DENOULET, Francois PECHEUX, Andrea PINNA Bertrand GRANADO
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005 Paris, France,
Email: alexandre.briere@lip6.fr

***Abstract*—The number of cores in a single chip increases with the increase of communication needs. *Network on Chips* (NoCs) are developed to handle these needs. We present our work to evaluate models of NoC present in the SoCLib library with the NoCBench platform. NoCBench contains performance evaluation tools for NoC such as a Transaction Generator and an Execution Monitor. SoClib is a library of IP cores, including NoC, written in SystemC, we retain the WiNoCoD RF-NoC as a use case. Various benchmarks can be ported to our NoC to evaluate its performance and it can be compared with existing NoCs which used the same performance evaluation tools under NoCBench.**

I. Introduction

One of the major issues for Chip Multi Processor (CMP) architectures is the ability to make all the components communicate together efficiently. To overcome the latency shortcomings of classic wired and router based Networks on Chips (NoC), and meet with the growing demand for bandwidth and speed, the recent NoCs approaches use new technologies, such as 3D [1] to reduce distances, and optical [2] or Radio Frequency (RF) [3] to increase information propagation speed.

In this context, SoCLib [4] and NoCBench [5] have been proposed to evaluate the contributions of various proposed solutions. SoCLib is an open platform for virtual prototyping of System on Chip (SoC). The core of the platform is a library of simulated component models (IP cores), such as interconnects, processors, memories, etc. These models are written in SystemC with two levels of abstraction, Cycle Accurate Bit Accurate (CABA) and Transaction Level Modeling (TLM). NoCBench [6] is a freely available CMP evaluation platform written in SystemC. It includes a Transaction Generator (TG) that generates traffic on the network and an Execution Monitor that is a Graphical User Interface (GUI) to present various performance statistics of NoC.

In this paper we present our work to interface NocBench with NoCs of SoCLib TLM and SoCLib CABA by developing wrappers specially for CABA models of WiNoCod [7], a dynamically reconfigurable RF-NoC. Section II presents the performance evaluation platform. Section III details the WiNoCoD architecture. Section IV concludes the paper.

II. Platform description

A. Transaction Generator

The Transaction Generator [6] generates traffic to NoC according to the file description written in eXtensible Markup Language (XML) format and it measures many performance metrics during the simulations. The connection of a SoCLib NoC and TG is shown in Fig. 1. Like all SoCLib components, NoC are written in SystemC. If the model is written in SystemC TLM, it can be directly interfaced with TG using Open Core Protocol (OCP). If it is written in SystemC CABA, we need to use a CABA/TLM wrapper to interface it with the TG. We interfaced SoCLib TLM with NoCBench and we also wrote a SoCLib CABA $\Leftrightarrow$ TLM wrapper to interface CABA models with TG. We have run various benchmark applications at NoC using TG which are task graph based and described in XML file. When a data token is received, the task is activated and it executes to perform certain computations. This task then initializes the transfer of data to other tasks. These transfers are mostly write operations and the simple read operations are supported using memories. Transaction Generator collects the statistics including PE utilization, task execution counts, latency and the amount of transferred data automatically. We can further modify the application and the mapping of tasks can be done to desired PEs in the XML file.

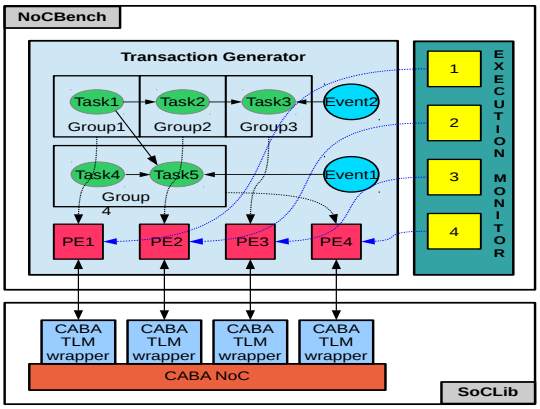

Fig. 1. NoCBench WiNoCod architecture

The computation normally contains the set of tasks, one or multiple tasks are then assigned to unique processing elements (PEs). The processing time for a given task can be measured in clock cycles. The dependencies also exist, for instance, the transmissions from multiple tasks might be required in order to start computation on present PE. The architecture of application in XML file defines resources including PEs and memories for example, task mapping then defines the PEs where the tasks are executed. Moreover, there is also possibility of defining the frequency of each PE in the architecture.

The transferred bytes are also estimated which can be derived from bandwidth by dividing it with the transfer rate.

B. Benchmark Results

We have tested various traffic models with our NoC. In this article we present the Telecommunication benchmark that is Ericsson Radio Systems with 4488 MB/s [5]. This application may contain several parallel tasks and need heavy computation. For this heavy communication system various tasks are defined in XML file but the exact functionality of this Radio system is confidential. The 16 PE's are defined in the file and all tasks are assigned to various PE's. The task graph can be generated from XML file. We normally assigned one task per processing elements. The statistics for this Ericsson Radio System model are shown using graphical user interface called Execution Monitor [6] (that is a Java tool for visualizing statistics of any SoC) and described here in TABLE I. It shows many performance measures including task execution cycles, PE utilization and total bytes transferred.

TABLE I
NoC performance measures for Ericsson Radio System

Results by Transaction Generator using Execution Monitor						
Process name_a	Id	PE	Thread	Exec. Cycles	Total Bytes	Rsp. time
Reply_b_and_send_c_12	12	cpu0	Thread_0	100	32000000	80.001
Reply_b_and_send_c_13	13	cpu1	Thread_1	100	32000000	80.001
Reply_b_and_send_c_14	14	cpu2	Thread_2	100	32000000	80.001
Reply_d_and_send_e	11	cpu7	Thread_7	50	16000000	80.001
Reply_f_and_send_g	3	cpu15	Thread_15	8	4000000	125.001
Reply_i_and_send_c	6	cpu10	Thread_10	100	32000000	80.001
Send_a_and_h	4	cpu8	Thread_8	200	1536000000	1920
Send_d	15	cpu3	Thread_3	100	128000000	320.001
Sink7	7	cpu11	Thread_11	0	0	0

III. Use case : the WiNoCoD architecture

In this section we describe the architecture of one of the NoCs present in SoCLiB. We proposed a RF NoC that is able to perform dynamic frequency allocation on demand, Wired RF Network on Chip reconfigurable on Demand (WiNoCoD) [7]. RF waves travel close to the speed of light and can be transmitted through a transmission line or an antenna. The use of RF in NoC seems to be the best alternative, since 3D NoC suffers from heat dissipation problems while optical NoC requires an external light source. Also, the simultaneous execution of different and/or irregular applications within a CMP introduces a spatial and temporal heterogeneity of communications, that can have a drastic impact on the architecture performances. The bandwidth is distributed among several channels that are dynamically allocated to each node according to the needs. Fig. 2 illustrates the hierarchical architecture of the CMP using WiNoCoD. It features three levels of hierarchy: tiles, clusters and the entire CMP. Each level is associated with a specific level of the interconnect: crossbar, grid and RF. The tile is the lowest hierarchical level. It contains a local RAM, a DMA and P processors with a data and instruction cache. These components are connected by a local crossbar, which is linked to a router to access the interconnect second level (the grid). Clusters are grids of $M \times M$ tiles where each tile router is connected to the four nearest routers. The size of a cluster is limited by the grid

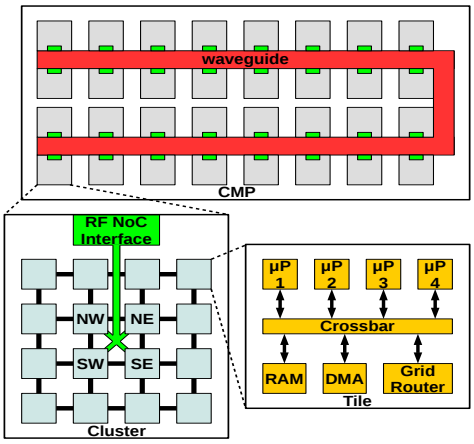

Fig. 2. CMP hierarchical RF NoC architecture

that suffers from latency, which increases along with the grid size. RF NoC is then used as a third level interconnect when its latency becomes lower than the grid latency. It is thus possible to integrate more cores on a single chip while controlling the increase of latency. To allow inter cluster communications, each cluster includes a RF Interface providing access to the RF-NoC through the transmission line.

IV. Demonstrator

In this article, we presented the interfacing of SoCLib NoC with the NoCBench performance evaluation platform. NoCBench consists of Transaction Generator (TG) that collects various statistics of benchmark applications for any specific NoC including the utilization of PE's, total execution cycles, latency and the total bytes transferred. We have done the interfacing of TG with SoCLib CABA model of WiNoCoD interconnect by writing TLM/CABA wrappers and tested various benchmark applications of NoCBench. We have provided the results of application Ericsson Radio System in the article. We will further use this tool to evaluate the performance of WiNoCod Architecture, a RF NoC with improved latency as compared to conventional packet routing NoCs.

References

[1] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3d chip multiprocessors using network-in-memory," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2, pp. 130–141, 2006.

[2] I. O'Connor, "Optical solutions for system-level interconnect," in *Proceedings of the 2004 international workshop on System level interconnect prediction*. ACM, 2004, pp. 79–88.

[3] M. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S. Tam, "Cmp network-on-chip overlaid with multi-band rf-interconnect," in *High Performance Computer Architecture, HPCA 2008. IEEE 14th International Symposium on*, pp. 191–202.

[4] SoCLib Consortium, "The soclib project: An integrated system-on-chip modelling and simulation platform," CNRS, Tech. Rep., 2003. [Online]. Available: http://www.soclib.fr

[5] E. Pekkarinen, L. Lehtonen, E. Salminen, and T. Hamalainen, "A set of traffic models for network-on-chip benchmarking," in *System on Chip (SoC)*. international Symposium, 2011, pp. 78–81.

[6] [Online]. Available: http://www.tkt.cs.tut.fi/research/nocbench/

[7] A. Brière, J. Denoulet, A. Pinna, B. Granado, F. Pêcheux, E. Unlu, Y. Louët, and C. Moy, "A dynamically reconfigurable rf noc for many-core," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*. ACM, 2015, pp. 139–144.