

NoC Traffic Monitoring for Billion Cycle Application Performance Debug Based on FPGA Platform

Xinyu Li and Omar Hammami

ENSTA PARISTECH

32 Bvd Victor 75739 Paris Cedex

FRANCE

hammami@ensta-paristech.fr

Abstract—In this paper, a NoC traffic monitoring method is proposed for billion cycle application debug which help designer to improve system performance by the analysis of traffic distribution and balance through the network on chip. The hardware monitoring network consists of traffic collectors, which is reconfigurable to collect different traffic information such as packet latency and throughput. Designer can use this global traffic information from NoC traffic monitoring to tune the application task and data allocation and improve their application performance. A matrix multiplication test case illustrates the feasibility and potential performance improvement.

Keywords— emulation, FPGA, NOC, Multicore.

I. INTRODUCTION

As the design complexity of system on Chip keep increasing, the software design effort need has exceeded that assigned to the hardware. High level abstraction based simulation is used to accelerate the validation of software, while the trade-off of accuracy for speed still cannot solve the hardware/software integration challenges. We use another alternative to prototype SoC design and debug application by emulation on FPGA platform, which has an accurate representation of the design and rapid execution of software application. The NoC is proposed as a paradigm for communications within large scale multiprocessor on a single chip (MPSoC). We found that the prototype and debug of billion cycle application on FPGA platform for MPSoC is still tough as the number of IPs is getting larger. In this paper, we present a NoC traffic monitoring method for billion cycle application debug which help designer to improve system performance by the analysis of traffic distribution and balance through the network on chip. The hardware monitoring network on multiprocessor with NoC consists of our traffic collectors, which is reconfigurable to collect different traffic information such as packet latency and throughput. The collectors in the monitoring network collect the statistic traffic information during the sampling time, and send them back through monitoring network to the off-chip memory for further analysis. The statistic information represents the distribution and load balance of application data through the communication network. Traffic jam in the NoC can come from the bad task and data allocation on processors and memories. Designer can use this global traffic information from NoC traffic monitoring to tune the allocation and improve their application performance.

In this paper, an 8-core multiprocessor is implemented on FPGA platform with our hardware traffic monitoring network. Four SRAM memories of 16MB are shared by processors and they are connected by a 2x2 cluster mesh NoC. Parallel matrix multiplication with two different data allocations (block and interleave) is used to illustrate the usage of our hardware NoC monitoring for billion cycle application performance debug.

II. HARDWARE MONITORING NETWORK

Our 8-core multiprocessor is organized as a 2x2 mesh of clusters. Each cluster includes 2 processor elements and a local memories as shown in Figure 1.

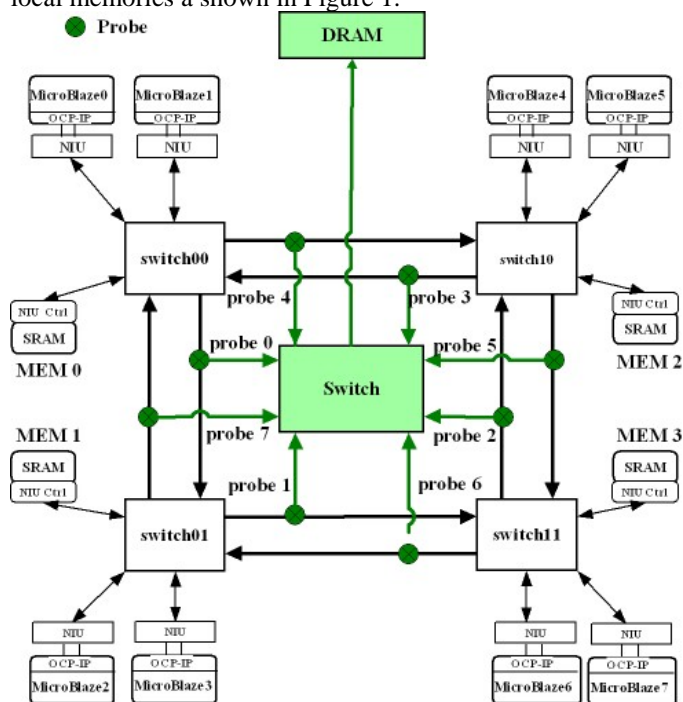


Figure 1. Hardware monitoring network on 8-core multiprocessor

There are 8 probes connected to the links between switches in the NoC. NoC traffic is monitored through hardware performance monitoring unit. Statistics collectors are added in the probes and the results are collected and sent through a switch to off-chip memory as shown in the figure. From the monitoring results of the 8 probes, the data traffic distribution and balance on the NoC can be analyzed.

III. PERFORMANCE EVALUATION

Parallel matrix multiplication with two different data allocations (block and interleave) is used to illustrate the usage of our hardware NoC monitoring for billion cycle application performance debug.

A. Billion cycle emulation on FPGA platform

Our 8-core multiprocessor is used in this study case. It is implemented on the ZeBu FPGA platform.



Benchmarks with different traffic sizes are executed and presented in the table I. Billion cycle applications can be emulated with 7 minutes on the FPGA platform with a emulation clock of 10MHz.

TABLE I. EMULATION TIME WITH DIFFERENT TRAFFIC SIZES

Packets Size	Simulation cycles of application	Emulation time with 10MHz clock (s)
1 M packets	89,719,056	39
10 M packets	897,072,450	361
100 M packets	8,970,802,754	3591

B. Matrix Multiplication with different data allocations

The 2D parallel matrix multiplication application is used for the illustration of NoC traffic monitoring in this study case. The primary partitioning of data is by block rows or block columns. Assuming we have two matrix A and B of dimension n, the result of matrix multiplication $C=A \times B$. Two different data allocations: by block and by interleave as shown in figure 2, are used to show the impact of data allocation on traffic distribution.

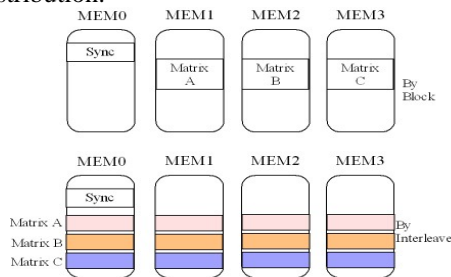


Figure 2. Data allocation of matrix multiplication by block and interleave

There are four shared memories in our 8-core multiprocessor as shown in figure 2. Each matrix is located on one separated memory in block allocation mode. The synchronization variables are in the memory 'MEM0'. In interleave allocation mode, all the matrix are distributed by column onto the four memories.

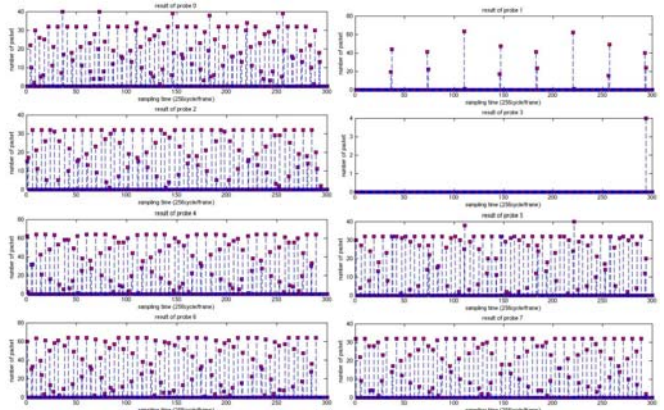


Figure 3. Traffic monitoring results of 8 probes in block mode.

The throughput monitoring results of block mode are presented in figure 3. Seen from the results, the data traffic is not well balanced, as the memory 'MEM0' is only used for synchronization. Designer should find new data allocation to full use the network capacity, interleave mode in this test case.

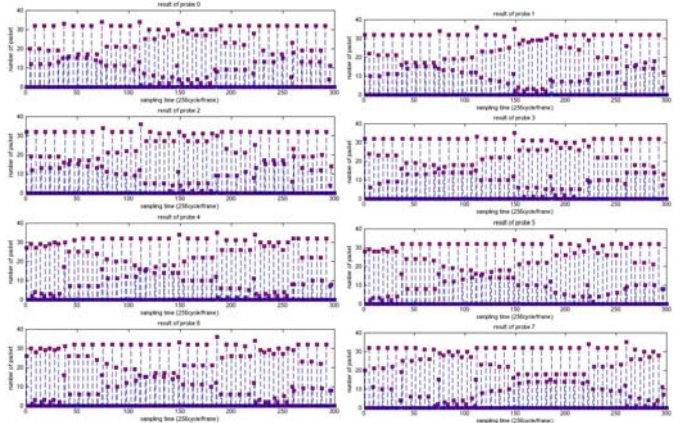


Figure 4. Packets Traffic monitoring results of 8 probes in interleave mode.

The monitoring results of interleave mode are presented in figure 4. We can see that the traffic is better balanced on the NoC. And the system performance is improved. The execution time is improved from 14,370,025 cycles to 8,420,067 cycles for 256x256 parallel matrix multiplication. The gain is 41,4%.

IV. CONCLUSIONS

In this paper, we present a NoC traffic monitoring method for billion cycle application debug which help designer to improve system performance by the analysis of traffic distribution and balance through the network on chip. The matrix multiplication test case illustrates the feasibility and potential performance improvement. No online monitoring is done in this work but it will be the scope of future studies.

REFERENCES

[1] Xinyu Li and Omar Hammami; "Multi-FPGA emulation of a 48-cores multiprocessor with NOC," Design and Test Workshop, 2008. IDT 2008. 3rd International, pp.205-208, 20-22 Dec. 2008
 [2] EVE Zebu Emulation platform. www.eve-team.com