Development of multiprocessor system-on-chip based on soft processor cores schoolMIPS

A E Ryazanova1, A A Amerikanov1, E V Lezhnev1,2,3

1 National Research University Higher School of Economics, 20 Myasnitskaya Ulitsa, 101000, Moscow, Russia
2 e-mail: elezhnev@hse.ru
3 Author to whom any correspondence should be addressed

Abstract. This work includes a review of MIPS architecture processor cores and a review of network topology consisting of routers. It was demonstrated by realization of 2 multiprocessor systems developed on the basis of mesh topology using modified schoolMIPS soft-processor cores, in which architecture additional blocks and instructions were added, and routers with XY routing. As a result, the obtained NoC performance is up to 1.87 Gbit/s (4 processor cores), and up to 1.54 Gbit/s (10 processor cores). The extended processor core schoolMIPS consumes 452 ALMs and 1692 bits of memory; NoC of 4 processor cores takes 2223 ALMs and 9136 bits of memory; NoC of 10 processor cores – 5696 ALMs and 22840 bits of memory. The obtained results suggest that there is a possibility of NoC development with the number of nodes up to 200 nodes on Stratix IV GX EP4SGX230 (DE4).

1. Introduction
An important task is to investigate the possibilities of using different architecture processor cores. Currently, the MIPS architecture is becoming more widespread through freely distributable cores like MIPSfpga [1] or schoolMIPS [2], which are used in many colleges and universities for teaching students computer architecture, digital design and hardware-software codesign. MIPSfpga soft-processor [1] combines properties of commercial and academic cores, since it is built on the basis of the microAptiv UP core (used in the Microchip PIC32MZ microcontroller), and is also provided with extensive educational materials.

At the same time, in the development of large-scale multiprocessor systems, it is necessary to observe a balance between data transfer rate, amount of resources expended to implement the system, and ability to scale the system without cardinal changes. The network-on-chip architecture influences on all of the above. The architecture provides simultaneous interaction of multiple cores, with data packets transferred from one processor core to another processor core without interfering with each other. Applying this approach to the network of MIPS cores is a promising solution, but it is required to develop a methodology and tools to design new solutions in the field of networks-on-chip.

2. SchoolMIPS soft-processor core
The schoolMIPS processor core [2] is a simplified version of the MIPS processor developed by Sarah Harris [3] and is provided as a set of Verilog files. The basic processor implementation is a single-cycle microarchitecture with a minimal set of instructions, instruction memory, general-purpose registers, but without data memory (Figure 1). With the core source, scripts for project build, for
simulation in ModelSim, and for project loading on development board are provided. Despite the fact that initially schoolMIPS is a simplified processor core, its architecture can be improved to achieve the functionality that is required in a multiprocessor system. Such opportunity makes it a convenient tool for testing various architectural solutions of the network-on-chip communication subsystem.

![Figure 1. SchoolMIPS soft-processor core microarchitecture [3].](image1)

3. **Network-on-chip architecture based on routers**

Currently, there is a tendency to use systems-on-chip (SoC), which include a large number of processors, architectural blocks and peripheral nodes. In such systems, crossbar switch or bus topology as a way of interaction between processors is not effective, resulting in a new architecture called network-on-chip. In this architecture, each core or processor unit is connected to a router, through which it interacts with other units or processors. Routers are united into a network and data packets are sent from one unit or processor to another (Figure 2).

![Figure 2. The interaction of processor cores in network-on-chip with XY routing [4].](image2)

In this type of communication subsystem, multiple units are able to simultaneously transmit data without interfering with each other, which greatly simplifies the topology of the chip and removes scaling limitations. In addition, power consumption is reduced and the speed of interaction increases,
and this network-on-chip architecture allows to combine heterogeneous cores and blocks on a single chip. The network-on-chip is the most rational method of interaction, combining fast data exchange, resource consumption and the ability to add new processor units, without the need to drastically change the topology of the system.

4. The choice of topology for multiprocessor system

Compared to other topologies, the mesh topology [5] provides sufficient network performance and evenly distributes the load on the nodes, thus avoiding blocking, and also relatively simple scaling. Mesh topology provides sufficient network performance even with a large amount of transmitted data and an average network load. In a network with a mesh topology, a router can redirect a transmitted packet over several alternative free connections.

In this research, two networks-on-chip of 4 and of 10 processors were developed. The architecture of these networks based on the mesh topology is shown below (Figure 3-4).

![Figure 3. Network-on-chip with a mesh topology of 4 processor cores.](image)

![Figure 4. Network-on-chip with a mesh topology of 10 processor cores.](image)

5. SchoolMIPS

According to the proposed method the rising complexity of the processor core for the needs of a multiprocessor system, the simplest version of the core has been selected (00_simple) [2], after which it was modified as follows [6]:

1. A block of local RAM memory was added to the microarchitecture for data storage (Figure 5).

![Figure 5. SchoolMIPS microarchitecture with RAM memory block.](image)

2. For interaction of the core with the router required the presence of a network interface – an architectural unit by which the router's data is sent/received. The network interface is implemented as two separate modules for input to the processor and to output from it. The inputCPU module, by which the GET command receives data from the router, and the
outputCPU module, which implements transmission from the processor core, by which the SEND command transmits data to the router.

3. Support of GET and SEND instructions for receiving and sending data has been added to the microarchitecture of the processor core. The presence of these instructions is not described in the technical documentation for the schoolMIPS processor [7], and is not provided by the standard MIPS instruction set. Therefore, a 32-bit format has been independently determined for them (Figure 6), unused operation/functional codes have been allocated, registers and signals have been set, on which these instructions will be executed.

![Figure 6. 32-bit format of the SEND and GET instructions.](image)

Format of new instructions in MIPS assembler language is as follows:
- `send $reg_data, $reg_addr`
- `get $reg_data`

Here `reg_data` is the register with the data to be sent ([25:21] command bits; rd1 – input to the schoolMIPS), and `reg_addr` – register with the destination processor address ([20:16] command bits; rd2 – input to the schoolMIPS).

6. Structure of the router for NoC
In this research, a simple NoCSimp router [8] is used, which implements the separation of the switching part into input and output blocks connected by a FIFO buffer memory. This architecture involves the elimination of unnecessary resource costs, and also ensures the continuous use of buffer memory (Figure 7).

![Figure 7. The structure of the router [9].](image)

Arbiter module through respective enable signals and acknowledgment/transmission controls the interaction with this router connected to the processor core as well as with the neighboring routers. The destination address is assigned by the source that forms the message. All messages arriving at the router are divided into packets from 1 to 8 flits in length. Each flit contains 37 bits: 32 data bits, 1 end bit, one value of which indicates the last flit in the package, and 4 bits for the destination router address, which corresponds to the destination processor (Figure 8). Thus, the network can consist of a maximum of 16 routers. The router performs packet switching based on the classic deterministic XY routing algorithm, that is, first the packets transmitted over the network in a horizontal direction, and then in the vertical direction.

![Figure 8. The structure of the flit [9].](image)

7. The development of networks-on-chip
As mentioned before, in the network-on-chip architecture each IP-core or processor unit is connected to a router, and the routers themselves are connected with each other, thereby forming a network. In
this research, mesh topology is used, in which only interaction with neighboring routers is assumed. Based on this, the network has been designed for 4 and 10 routers.

![Figure 9. The core schoolMIPS with added architectural solutions.](image)

7.1. Network-on-chip of 4 processors
In this network the routers interact in accordance with the incidence matrix (Table 1). Each router has been attached processor core so that the core network interface outputs connects to inputs of the router, and data or signals from the router output (Figure 10) come to the interface inputs.

| Router Number | 1   | 2   | 3   | 4   |
|---------------|-----|-----|-----|-----|
| 1             | E   |     | S   |     |
| 2             | W   |     |     | S   |
| 3             | N   |     |     | E   |
| 4             |     | N   | W   |     |

For example, data output from the dataOutL[36:0] router is connected to the input of the dataFromRouterToCPU[36:0] network interface, and the input of the dataInL[36:0] router is connected to the data output of the dataOutCPU[36:0] processor. The same implemented for read/write request/acknowledge signals.

7.2. Network-on-chip of 10 routers
In a network of 10 routers, the interaction is carried out in accordance with the incidence matrix (Table 2). In order to facilitate the process of designing a network of 10 processor cores, the router and its local core were combined into a single unit, thereby forming a network cell. This module combines the inputs and outputs of the processor core and router.

| Router Number | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1             | E   |     | S   |     |     |     |     |     |     |     |
| 2             | W   | E   |     | S   |     |     |     |     |     |     |
| 3             | W   | E   |     | S   |     |     |     |     |     |     |
| 4             | W   |     | E   |     |     | S   |     |     |     |     |
| 5             |     | W   |     |     | E   |     |     | S   |     |     |
| 6             | N   |     |     |     |     |     |     |     |     |     |
| 7             |     | N   | W   |     |     | E   |     |     |     |     |
| 8             |     |     | N   | W   |     |     | E   |     |     |     |
| 9             |     |     |     | N   | W   |     | E   |     |     |     |
| 10            |     |     |     |     | N   | W   |     |     |     |     |
In a network of 10 nodes with a selected topology, the transfer of packets between nodes that are far away from each other will be quite slow. A packet will have to go through 5-6 routers, each of which will first check for place in the FIFO, then exchange request/permission signals to send data to neighboring routers. In addition, central routers can be busy processing a large number of incoming packets, which significantly affects the delay of transmission. The solution of these problems can be achieved by applying instead of the mesh other, more optimal topologies, such as, for example: quasioptimal topologies [10] or circulant topologies [11].

7.3. Network-on-chip simulation
The simulation made it possible to verify the performance of the NoCSimp router, as well as each developed network-on-chip and the correctness of packet transmission over the network. The simulation also confirmed the correctness of the interaction of the modified schoolMIPS processor cores with the NoC communication subsystem. Simulations were performed for single packet transactions over the network in order to track their path through the network and verify that the routing algorithm works correctly, as well as for multiple simultaneous transactions, to ensure that the network is resilient to loads with multiple packets.

7.4. Analysis of performance and hardware costs
NoCSimp router was synthesized in Quartus Prime. Its implementation requires 142 ALMs and 592 bits of memory, which is less than 1% of the resources of the De1-SoC board. The maximum frequency of the router is 200 MHz, and the bandwidth reaches 2.13 Gbit/s. Ideally, such bandwidth should be stored in each network-on-chip, however, taking into account the network load and delays, the network of 4 routers provides a speed of 1.87 Gbit/s, and a network of 10 routers – 1.54 Gbit/s.

The original schoolMIPS processor core (Figure 1) occupies 209 ALMs and 768 bits of memory. The extended core consumes 415 ALMs (98% more) and 1,692 bits of memory (120% more), which is nevertheless still comparable to the resources consumed even by the simple NoCSimp router [8] (Table 3). Naturally, the schoolMIPS functionality is incomparably less, but for the tasks of NoC architectures prototyping and networks with a large number of cores modeling on chips of the lower and middle price range, and at the same time at high frequencies, this core fits perfectly. As the project under development expands, the amount of resources expended increases linearly, suggesting that the Terasic De1-SoC [12] board can accommodate NoC sizes up to 50 nodes and up to 200 on Stratix DE4 [13].

| Table 3. Project resources consuming at each stage of development |
|---------------------------------------------------------------|
| Stage             | ALMs | Memory bits |
| schoolMIPS core   | 209  | 768         |
| NoCSimp router    | 142  | 592         |
| NoC of 4 routers  | 415 (1%) | 2,368 (<1%) |
| NoC of 10 routers | 1,176 (4%) | 5,920 (<1%) |
| Modified schoolMIPS | 452 (1.5%) | 1,692 (<1%) |
| NoC of 4 routers with connected cores | 2,223 (7%) | 9,136 (<1%) |
| NoC of 10 routers with connected cores | 5,696 (18%) | 22,840 (<1%) |

8. Conclusion
As a result of this research, networks-on-chip of 4 and 10 NoCSimp routers were developed, after which the schoolMIPS soft-processor cores were attached to each router. Mesh topology was chosen for modeling purposes.

The upcoming strategy for developing the processor core was proposed, where only the necessary components were added to the initially simple implementation of the schoolMIPS soft-processor, such as core network interface allowing to interact with the network and hardware support needed to interact with the network communication instructions.
The analysis of the consumed resources demonstrates that such an approach allows to significantly reduce the consumption of resources in comparison with the opposite method, when the initially functional processor core is optimized for resources by abandoning part of its functionality. The results suggest the possibility of networks-on-chip development with the number of nodes of schoolMIPS to 50 on Terasic De1-SoC board and up to 200 on DE4 board at operating frequencies up to 200 MHz. This allows modeling and performing hardware prototyping of various architectural solutions for NoCs communication subsystems with a large number of nodes development.

Acknowledgments
This work is an output of a research project implemented as part of the Basic Research Program at the National Research University Higher School of Economics (HSE).

References
[1] Harris S L, Harris D M and Chaver D 2017 MIPSfpga: using a commercial MIPS soft-core in computer architecture education IET Circuits, Devices & Systems Vol. 11(4) pp 1–9.
[2] Zhelnio S, ‘Processor core schoolMIPS. User Guide’, 2018 [Online] Available: https://github.com/MIPSfpga/schoolMIPS (Accessed 29-Sep-2018).
[3] Harris D M, Harris S L 2012 Digital design and computer architecture Morgan Kaufmann Vol. 2 p 712.
[4] Tsai W C, Lan Y C and Hu Y H 2012 Networks on Chips: Structure and Design Methodologies // Journal of Electrical and Computer Engineering Vol. 4. pp 1–15.
[5] Kumar A, Tyagi S and Jha C K 2017 Performance analysis of network-on-chip topologies Journal of Information and Optimization Sciences Vol. 38. – No. 6. pp 989–977.
[6] Ryazanova A ‘Multiprocessor systems based on the schoolMIPS processor core’ [Online] Available: https://github.com/ylianova0704/07_schoolMIPS_NoC (Accessed 30-Sep-2018).
[7] MIPS Technologies, Inc 2013 MIPS® Architecture For Programmers Volume II-A: The MIPS32® Instruction A p 372.
[8] Romanov A Y, Ivannikov A D and Romanova I 2016 Simulation and synthesis of networks-on-chip by using NoCSimp HDL library IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO) (Kiev: IEEE) pp 300–303.
[9] Lysenko A N and Romanov A Y 2011 Resource efficient router for multiprocessor network-on-chip Vestnik NTU "KPI": Collection of scientific papers. Thematic issue: Informatics and modeling №17 (NTU “KPI”, Kharkov) pp 86 – 92.
[10] Romanov A Y and Romanova I I 2015 Use of irregular topologies for the synthesis of networks-on-chip 2015 IEEE 35th International Conference on Electronics and Nanotechnology (ELNANO) (Kiev) pp 445–449.
[11] Romanov A Y, Romanova I I and Glukhikh A Y 2018 Development of a Universal Adaptive Fast Algorithm for the Synthesis of Circular Topologies for Networks-on-Chip Implementations 2018 IEEE 38th International Scientific Conference on Electronics and Nanotechnology (ELNANO) (Kiev) pp 110–115.
[12] Terasic Technologies, Inc. 2015 De1-SoC User Manual p 115.
[13] Terasic Technologies, Inc. 2015 DE4 User Manual p 185.