FACULDADE DE INFORMÁTICA

PUCRS - Brazil

http://www.inf.pucrs.br

The YeAH NoC Router

Matheus T. Moreira, Leandro S. Heck,
Guilherme Heck, Matheus Gibiluka,
Ney L. V. Calazans, Fernando G. Moraes

TECHNICAL REPORT SERIES

Number 082
December, 2014

The YeAH NoC Router

Matheus T. Moreira, Leandro S. Heck, Guilherme Heck,
Matheus Gibiluka, Ney L. V. Calazans, Fernando G. Moraes
Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Ciências de Computação
Grupo de Apoio ao Projeto de Hardware
matheus.moreira@acad.pucrs.br

December 23, 2014

Abstract

This work presents the network-on-chip router YeAH!. The router was designed as part of the work developed in the *Sistemas Integrados em Chip* class, lectured by Professor Fernando Moraes. The proposed router is based on the Hermes router, allowing the implementation of networks with multiple configurations and characteristics, covering different network topologies, routing algorithms and arbitration policies. This technical report details the YeAH! router, which main difference between the original Hermes router is the distributed architecture that splits the routing algorithm and arbitration policy across the ports of the router.

1 Introduction

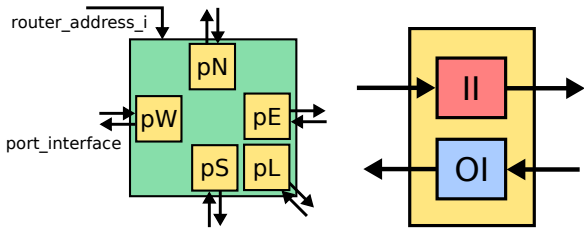
This technical report introduces the NoC router yet another Hermes (YeAH!). The new router proposes a new distributed architecture, which is different from the original Hermes router, where routing and arbitration are centralized. The YeAH! router distributes these tasks across the ports of the router, creating a modularized design targeting applications requiring high-performance interconnect.

The remainder of the document is organized as follows: Section 2 presents the YeAH! router and its building blocks. Simulation environments available on the project are presented in Section 3. Section 4 presents the final remarks about the work.

2 The YeAH! Router

The YeAH! router follows the same top-level architecture of Hermes, providing up to 5 ports: one exclusive for local IP connection (pL), and the remaining

for network communication (pE, pW, pN, pS). Figure 1(a) illustrates the the top level view of a five-port router. Due to the high level of modularity of the design, routers at the corners of the network have unconnected ports removed to improve interconnect area. Each port is composed of an Input and an Output Interface.



(a) YeAH! (b) Porta

Figure 1: Overview of the YeAH! router. (a) shows the router and (b) presents each port.

The input interface, as depicted by Figure 2, contains a buffer (*input_ buffer*) and a control block (*input_ control*), which is responsible for packet routing. The control block receives routing requests from the *input_ buffer* when a new packet header is detected. The packet's destination address, contained in the first flit of the packet header, is sent to the routing unit (internal to *input_ control*), which is responsible for selecting the output port to which the packet must be sent to. The router currently support the XY routing algorithm only. However, other routing algorithms can be easily implemented due to the input interface's modularity.

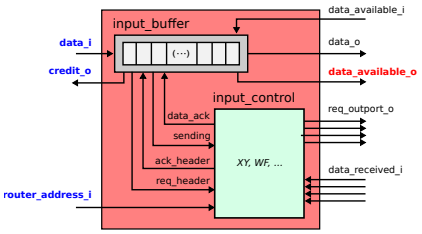


Figure 2: Input interface.

Figure 2 illustrates the output interface, which is composed of a arbitration block (*arbiter*) and a control circuit (*output_ control*). The control module implements the crossbar between each input port and the output interface's output port – which is different from the original Hermes network, where a central crossbar connects all input ports to all output ports. Round robin arbitration is supported with two implementations: one targeting area reduction, and the other optimizing performance.

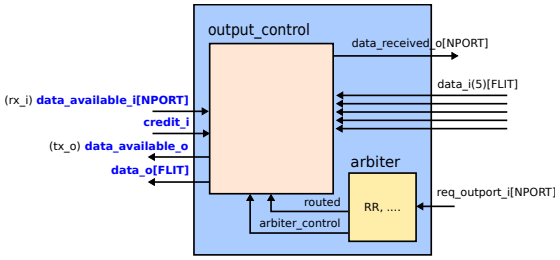


Figure 3: Output interface.

3 Validation

Due to the high complexity of the design, each building block of the router was tested individually during its design, and then, further testing was conducted when blocks were combined to create more complex structures. This reduces verification time, as issues can be detected early during design. To ensure the modularity of the router, a generic entity was created to interconnect the routers and create a NoC.

Three system-level NoC verification enviroments were also developed: (1) a simple VHDL testbench; (2) a modularized SystemC environment, composed of data injectors and monitors; (3) a SystemC environment compatible with the current NoC verification enviroment used at GAPH, Atlas.

4 Conclusion

This technical report explored a new structure for NoC routers, based on distributed routing and arbtration, proposing a new router called YeAH!. Additionally, a new modular design, was created with the objective of easing logic synthesis and ASIC design. Physical and logic synthesis are left as future work, with the objective of comparing area, power consumpotion, performance, and operating frequency in MPSoCs designs.