Vlsi Implementation Of Qos Supported Efficient Adaptive Network On Chip Architecture

P.Krithiga[1] C.Sujatha[2] Dr.R.Ganesan[3]

Abstract-**In the next decade, it will be possible to integrate hundreds of billions of transistors on a single chip, which will allow for the integration of hundreds or even thousands of processor cores on a single die along with the interconnect infrastructure and memory. Network-on-chip (NoCs) have emerged as a promising on-chip interconnect for future multi/many-core architectures as NoCs are able to scale communication links with the growing number of cores. In the previous NoC architectures they are using static routing algorithms and this is not effective to predict the system behavior during run time. In this paper we address these problems with a runtime adaptive network on chip architecture. In this architecture an runtime adaptive algorithm is used to improve the QoS level. Here one on demand buffer assignment scheme is used which reassigns buffer blocks on demand. The area overhead is also reduced due to the on-demand buffer assignment at each output port. The on-demand buffer assignment scheme decreases the overall buffer use when compared to a fixed buffer.**

Keywords- **adaptable architectures,multi core/single-chip multiprocessors,on-chip interconnection networks, network-on-chip (NoC), quality-of-service (QoS).**

I INTRODUCTION

Advances in semiconductor technology have enabled very complex large scale systems on a chip (SoCs) designs. Each new SoC generation integrates more processing elements (IPs) and offers increased functionality. As the number of IPs increases, traditional interconnects, such as busses, become a bottle neck. Networks -on-chip (NoCs) are a modular, scalable interconnect solution . Currently, they tend to become the preferred interconnect solution for large scale inherently multiprocessor SoCs. However, NoCs require sophisticated tools to aid in design-time decisions. Furthermore, with increasing complexity there is also a strong need for run-time NoC monitoring which must be accounted for in the design phase. This is in turn driven by debugging and performance monitoring Quality of Service (QoS) . With the introduction of NoCs the on-chip communication becomes more sophisticated relying on run-time programmable solutions. In centralized bus-based systems a single bus monitor is enough to be able to track the whole history of the system. In NoC-based SoCs, due to the inherent parallel behaviour of communications, where multiple pipelined parallel communications may exist between IPs, multiple monitors have to be employed. The problem of how many such monitors are needed, their automatic placement in the NoC-based SoC by means of a monitoring aware NoC design flow and the associated area cost implications have not been previously investigated. Monitors and the traffic they generate are traditionally added non-intrusively into the SoC by using a separate monitoring NoC. The cost of such a solution is high however, and a more efficient solution is use the same NoC for both monitor data and user data, as suggested in .When monitoring traffic uses an interconnect of its own, it can be dimensioned after the user data NoC is designed. This merely adds an extra step in the design flow. However ,when monitor and user data must share the same NoC, the overall design flow must be revised .NoC design flows for ASIC type designs are normally split in several steps as topology selection, mapping, path selection and slot allocation . Some design flows may omit or combine various steps. Each step adheres to the decisions taken in the previous steps. As prerequisites for NoC design, communication requirements must be derived, and the set of IPs to be connected to the NoC must be specified. In the topology selection step, the router network together with the bordering NIs are generated, based on the previously derived communication requirements. We have two interdependent problems: the one of functional dimensioning of the NoC and mapping of cores

while accounting for their communication requirements, and the other of monitor placement and monitoring bandwidth specification. If these two problems are solved sequentially, the monitoring communication requirements can be pre computed. However by increasing the topology, the number of NoC routers increases. In turn, the mapping, path selection and allocation of resources may change and the number of required monitoring probes may increase as well (e.g. if probing all routers is required) and their communication requirements may change. In the mentioned cases the monitoring problem (whether driven by debugging or by run-time performance analysis) must be solved within or at least tightly coupled with the NoC design process. The task of placing the monitors must therefore be automated and integrated in the NoC design flow. Contribution. We propose a monitoring-aware NoC design flow able to take into account the monitoring requirements at all steps in the NoC design flow. We illustrate this with a debug driven monitoring case study. Simple, area-efficient transaction monitors, attached to NoC routers, are used to enable debugging of the NoC-based SoC at transaction level. This is one of the most difficult cases, where the monitoring requirements are only known after the path selection step.

II.EXISTING APPROACH

The runtime adaptively in the Ad NoC architecture is employed both at the system-level as well as at the architecture-level. At the architecture-level, several novel methodologies to adapt the underlying interconnections are employed on-demand in response to changing communication requirements imposed by an application, i.e., through a runtime application mapping request due to reliability issues or user behavior. To provide on-demand interconnections, a novel adaptive routing algorithm wXY-routing algorithm that meets quality-of-service (QoS) requirements presented. The routing algorithm makes decisions locally at each router depending on the available bandwidth in each direction to the neighboring router. Dynamic connections are realized by reassigning a certain number of buffer blocks to different output ports of a router on-demand. On-demand buffer assignment also increases the resource (buffer) utilization. To achieve successful adaptation at the architecture-level of the AdNoC, the communication architecture needs to be observed. Therefore, to provide runtime observability , a novel low-cost runtime observability infrastructure is used.

A. Weighted XY Routing Algorithm

To provide bandwidth guarantees in AdNoC, the underlying communication infrastructure needs to provide an adaptive route allocation scheme motivated from the adaptive routing schemes for large scale networks. In a physically static NoC, the routing decision can be distributed or a source-based deterministic routing scheme may be employed. In a distributed deterministic routing scheme, the routing decision is determined locally at each router using predefined rules, e.g., XY-routing algorithm in the QNoC architecture. The source-based deterministic routing scheme keeps the complete route in the header of transaction packets and needs the global view of the whole chip before execution or even at design time. That is why both schemes are not suitable for the AdNoC architecture where the subset of tasks and their mapping may change during runtime. Therefore, finding a route for a given NoC and physical mapping of the application is a major challenge. The weighted XY-routing (wXY-routing) algorithm presented assigns each output port a weight based on available bandwidth and , the coordinate (columns) distance or , the coordinate (rows) distance between the current and the destination node.

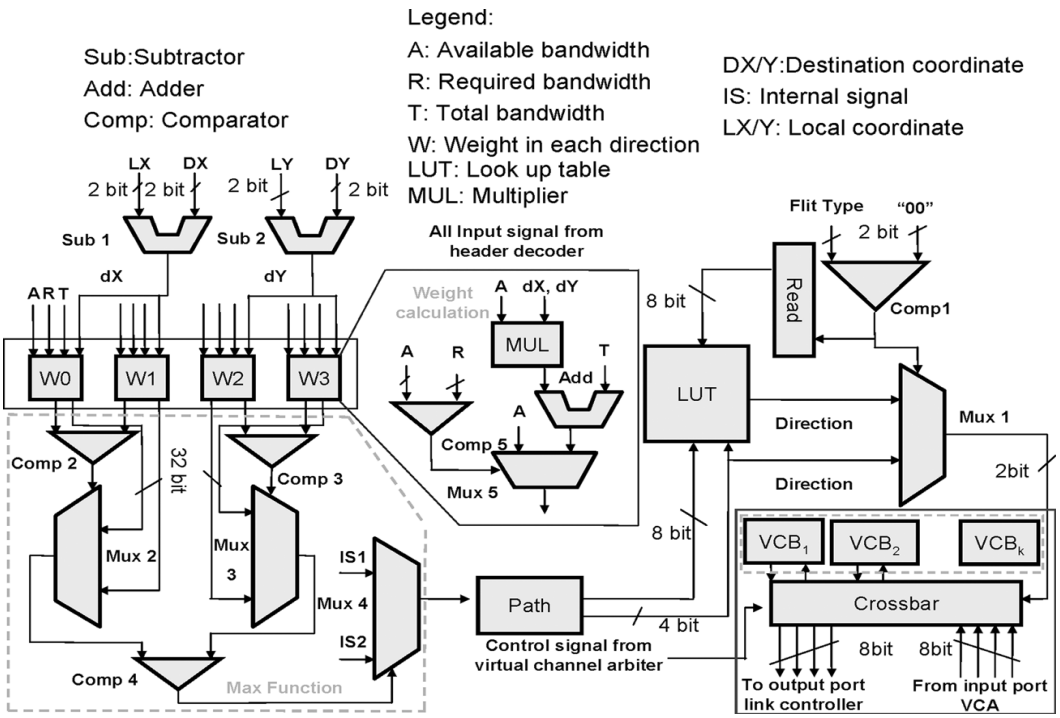

Fig.1 Micro architecture of wXY routing

This ideally gives the packet a maximum number of sensible routing choices along its route as it allow the packet to be routed toward its destination in both the directions. The weight is also proportional to the available bandwidth. If the output port is chosen with the highest associated available bandwidth, the used bandwidth is distributed as evenly as possible among the output ports.

B.On-Demand Buffer Assignment

When transmitting data over a packet-switched network, it is necessary to store parts of the data at each intermediate hop. In a wormhole router, this requires VCBs for each router through the complete route of each transaction. Until now, the number of VCs, implemented by VCBs in an output port, has always been fixed at design time. With on-demand assignment, the VCBs are not tied to ports, but only to the router itself. The router may distribute the VCBs to any route as needed by assigning them to that output port. Physically, it is realized by using a pool of FIFOs connected to each output port through a crossbar matrix. Pointers need to be saved for each output port to remember the current state of the VCB assignment. The benefits of such on- demand assignment is evident: through on-demand assignment, buffers are only assigned when needed meaning that VCBs can be reused by different ports. The obvious drawback of this method is that additional cycles are needed to retrieve the buffer pointers and the buffer contents.

A. Monitoring Events

The event aggregation and processing schemes explained in Fig. 2. The aggregation is done through the NI by sending messages to the source of the transaction. The processing is done partially in the NI and in the cluster agent. The NI takes care of retransmission while the cluster agent is invoked if a mapping is needed. The monitoring component is situated partially between the router and the NI. It is therefore able to interact with the NI to send its own packets over the regular communication network. This means, however, that the monitoring traffic must compete with regular transactions for network resources.

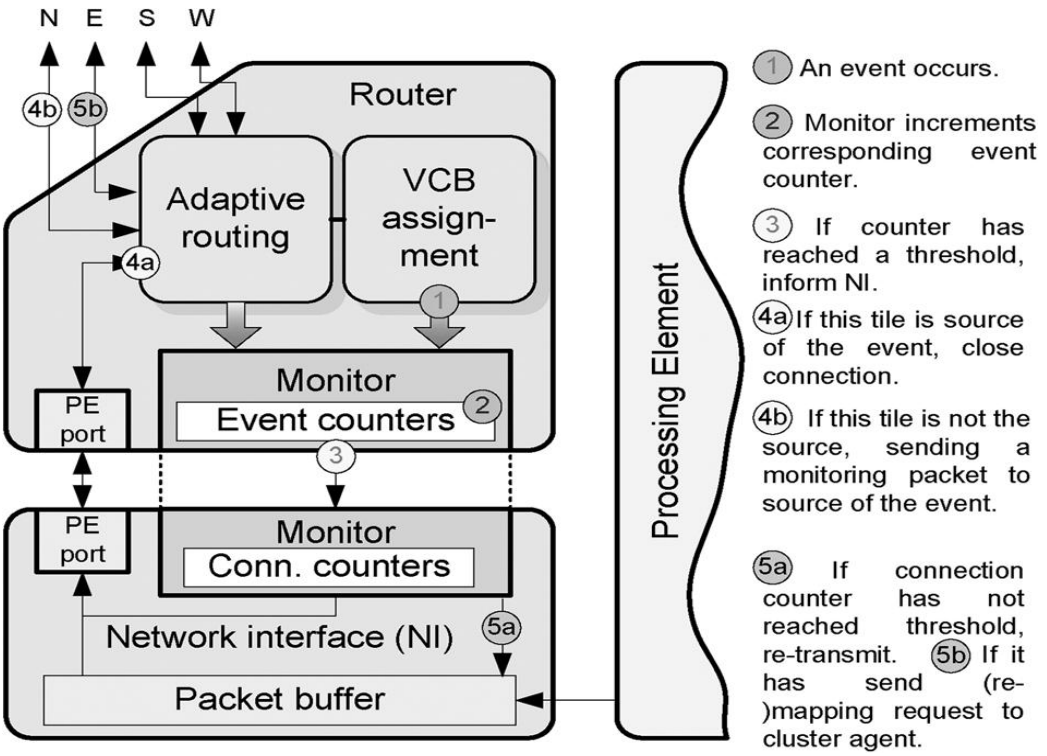

Fig.2. Overview of the monitoring component.

The monitoring packet must be of a higher priority than regular packets and it is never dropped during transmission. .

C. Hardware implementation of the AdNoC

An overview of the AdNoC-specific router micro architectures shown in Fig.3. All flits that traverse a router first enter theinput decoder (ID). Here, packet information such as a unique transaction identifier (transaction ID), the required connection bandwidth, and the destination is extracted from header flits. This information is sent to the wXY-routing component, and the flit is forwarded to the VCA. The VCA is in charge of sending incoming flits to an appropriate VCB. If a VCB is full,it notifies the VCA which then stops adding new incoming flits. A space division multiplexer (SDM) then sends the flits to the corresponding outputs.

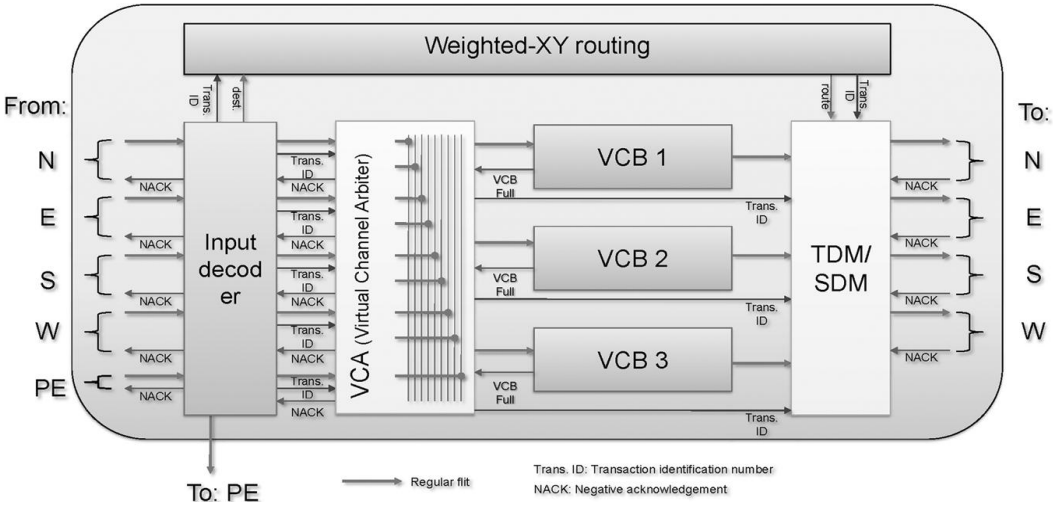

Fig.3 Micro Architecture of the AdNoC-specific router.

TABLE I
Buffer Saving Comparison

DEVICE DETAILS	BUFFER USAGE	BUFFER SAVINGS
Previous methods	58%	42%
In this method	50%	50%

The NI part of the monitoring component, upon receiving an event, first compares the transaction source with its own address. If it is not the sender then, a packet is sent to the remote sender. Otherwise, the transaction send count is examined to find out if there are previous send attempts by comparing the transaction-send counter stored in a register with a given resend threshold. Based upon this, the NI is either told to resend the packet if the threshold has not been met or a (re-)mapping is required. If the packet is resent, the transaction-send counter is incremented. Each router has five input ports resulting in five possible simultaneous transactions to be set up in a monitoring component. Also, using a LUT entails a one-cycle delay per read/write operation in which new transactions/events cannot be processed. To allow each tile to function using only one monitoring component, FIFOs are added to buffer its inputs.

III.PROPOSED APPROACH:

IV.RESULTS AND DISCUSSION:

In the previous NoC architectures the buffer utilization is large, also the area, power and speed are also large .Here an adaptive route allocation algorithm is used by this we can reduce the area, speed and power .An on demand buffer assignment scheme is used to reduce the buffer utilization .the various output parameters are shown in table II.

TABLE II

Output Parameter table

Power in mW	Area in %	Speed in MHz	Buffer Usage
46.5	65%	278.75	18

TABLE III

Proposed Approach

V.CONCLUSION

In this project we are designing a run time adaptive on chip communication architecture. It uses an runtime adaptive route allocation algorithm for monitoring the system behavior during runtime and for varying workloads. Our proposed on-demand buffer assignment scheme increases the on-chip resource utilization and decreases the overall buffer usage compared to a static approach where a fixed number of buffer blocks are tied to the output port. The monitoring component used in this system is used for runtime observability. Due to this runtime observability the connection success rate is increased.

Due to the on demand buffer assignment scheme the area overhead is reduced. So, speed of the system and the power utilization of the system are reduced compared to the static NoC architectures.

In future one additional algorithm is used to minimize the power consumption and the whole system can be implemented in FPGA.

REFERENCES

[1] F. Angiolini et al., “Networks on chips: From research to products,” inProc. Design Automation Conference, 2010.

[2] L. Benini et al., “Networks on chips: A new SoC paradigm,” Computer,vol. 35, no. 1, 2002.

[3] E. Bolotin et al., “QNoC: QoS architecture and design process for networkon chip,” J. Syst. Architecture, vol. 50, no. 2–32004.

[4] L. P. Carloni et al., "Networks-on-chip in emerging interconnect paradigms: Advantages and challenges," in Proc. Network On Chip, 2009,.

[5] P. Gratz et al., "On-chip interconnection networks of the TRIPS chip,"IEEE Micro, vol. 27, no. 52007.

[6] X.Zhu, S.Malik, "A Hierarchical Modeling Framework for On-Chip Communication Architectures", ICCD 2002, pp. 663-671, Nov 2002.

[7] I.Saastamoinen, D.Siguenza-Tortosa, J. Nurmi, "Interconnect IP node for future system-on-chip designs", Proc. of The First IEEE International Workshop on Electronic Design, Test and Applications, pp. 116-120, Jan. 2002.

[8] M. A. A. Faruque et al., "ADAM: Run-time agent-based distributed application mapping for on-chip communication," in Proc. Design Automation Conference, 2008,

[9] M. A. A. Faruque et al., "QoS-supported on-chip communication for multi-processors," in Proc. International Journal of Parallel Programming, 2008

[10] A. Hansson et al., "A unified approach to constrained mapping and routing on network-on-chip architectures," in Proc. CODES+International Conference on Software co design and System Synthesis,2005.

[11] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch.The Nostrum backbone - a communication protocol stack for networks on chip. In Proc. Int'lConference on VLSI Design, pages 693–696,2004.

[12] S. Murali and G. De Micheli.Bandwidth-constrained mapping of cores onto NoC architectures. In Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE), pages 896–901, 2004.

[13] V. Nollet, T. Marescaux, P. Avasare, D. Verkest, and J.-Y.Mignolet. Operating-system controlled network on chip. In Proc. Design Automation Conference(DAC), pages 256–259, 2004.

[14] J. W. van den Brand. Runtime networks-on-chip performance monitoring.Technical Report 2006/00218, Philips Research, Mar. 2006.

[15] D. Wingard. Socket-based design using decoupled interconnects.In J. Nurmi, H. Tenhunen, J. Isoaho, and A. Jantsch, editors,Interconnect-Centric Design for Advanced SoC and NoC, chapter 15,pages 367–396. Kluwer, 2004.