A Prototype Framework for Modelling Communication Hardware in Multicomputers

Pauline C. Haddow
Department of Computer and Information Science
The Norwegian University of Science and Technology
Trondheim, Norway

Abstract ***This paper presents a new approach to modelling and performance analysis of communication hardware in parallel computers. That is to move away from the single problem, single solution concept to a more general problem but not necessarily general solution. A feature not generally taken into account in performance analysis of communication hardware is the cost of the network routers. That is, what effect do performance improvements have on their complexity and thus cost. The prototype presented provides a common platform for the design and evaluation, in terms of both performance and cost, of current, proposed and future design ideas and technologies within communication hardware.***

Keywords: Cost, Performance, Communication Hardware, Multicomputers, Modelling

1 Introduction

This work focuses on message-passing multicomputers where each node contains an autonomous processor and local memory. Interprocessor communication occurs by routing messages explicitly through an interconnection network (network). The network, the communication system hardware, is assumed to be a direct-network where a fixed interconnection pattern connects the network routers.

A modelling framework is presented which enables not only performance but cost analysis of both existing and proposed communication hardware in multicomputer. This paper presents the key features of the framework and the current prototype.

2 Background and Motivation

There has been a lot of research within the field of performance analysis of communication systems in parallel computers, especially at the hardware level. Using traditional modelling techniques, assumptions are made to limit the complexity of the model to that which meets the model goals. With so many features that may be traded off against each other, a similar range of assumptions is found. Their use does not lessen the validity of the results obtained, but makes comparison of results a somewhat difficult task. It also prevents reuse of existing models for purposes other than those for which the model was designed. As such, a common evaluation platform is required i.e. a modelling framework.

It is acknowledged that software is a major bottleneck in communication systems. Especially for small messages, as is the case in fine-grained processing, the software overhead constitutes a large percentage of message latency [1]. One field of thought is to move some of the software functionality into hardware e.g. compressionless routing [2]. Such solutions lead to an increase in hardware complexity and, as such, give rise to an increase in latency at the hardware level.

To fully understand the effect of these and other design changes at the hardware level, it is not sufficient to consider performance alone: cost analysis is required. However, how should cost be treated? In performance analysis, the goal is to obtain better performance. However, in cost analysis, the question is not so much what is the cheapest cost but more is the performance benefit worth the cost?

A number of researchers have looked at the technology constraints with respect to pin-out limitations and wire density [3, 4, 5]. Pin-out limitations trade-off the number of network channels against the width of these channels for a given sized chip whereas the wire density describes the feasibility of implementing a given wiring structure in a limited area with respect to a given technology. These analyses can be said to be undertaken at the system or register-transfer-level.

The work of Chien [6] identified the importance of cost analyses in assessing adaptive routers. His work introduced a parameterised cost model for a generic router architecture for comparison of routing algorithms. The choice of routing algorithm is one of many interdependent design issues for an interconnection network designer. To be able to assess a wide range of communication system designs, it is important to be able to compare a number of design issues on the same modelling platform or framework. To achieve this aim in sufficient detail to enable cost analysis requires gate-level modelling within the framework. An additional effect of modelling design complexity is improved precision in performance data.

3 Modelling Tools

This work attacks the problem of parallel computer communication architecture modelling on two main fronts. The first is to provide a framework to analyse a wide range of both existing or proposed networks and their respective routers. As such, generalisation is the key. The Performance Description Language PDL [7] is designed specifically for generic performance modelling enabling design instances to be compiled and analysed and is, therefore, chosen for the framework.

The second is to provide cost and performance analysis where performance data is achievable at the gate-level. With this in mind, the performance model needs to be nearer to the hardware implementation and a chosen technology. As such specialisation or in other words instantiation is the key. HDL languages e.g. VHDL and Verilog, offer a flexible medium in which to describe hardware enabling gate-level designs to be realised through the process of logic synthesis.

The combination of PDL's ability to express generality in design and HDLs' ability to realise gate-level, technology specific designs allow the project goals to be met, offering a wide range of design choices in sufficient detail to enable cost and performance analysis to be undertaken.

The choice of using behavioural HDL in design descriptions leaves the implementation choice to the logic synthesis tool used. As such, an exact gate-level description of an existing design is not possible unless the original design used the same HDL source code and was synthesised using the same logic synthesis tool. However, the aim of this work is not to represent an exact gate-level replica of a given design but a sufficiently accurate copy that cost and performance analysis may be undertaken at a lower more detailed level than in a more typical parallel computer performance modelling environment. With this aim in mind, it is believed that assumptions may be made regarding implementation details where the behavioural characteristics and principal design decisions made are not effected. As such, with the limited low level information available for existing routers it is still believed that a reasonable more detailed performance model is realisable.

To compare different designs, not only a common modelling platform is required but the same technology base. Different technolo-

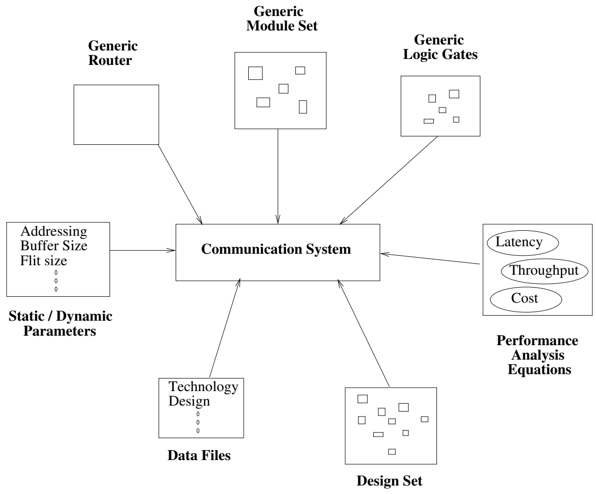

Figure 1: The Network Framework

gies exhibit different performance characteristics thus effecting the analysis of design differences. FPGA technology has been chosen as the implementation technology for the modelling examples presented in this work. The choice of FPGA technology is to provide a flexible implementation medium in keeping with the goals of the framework. Therefore, it is not intended to compare the results obtained against their original implementations as these are based on different technologies. The goal being to investigate the cost of differing design decisions.

4 The Framework

Figure 1 illustrates the key parts of the framework. This section provides a clockwise description of the different parts starting at the generic router.

The main component of the interconnection network of communication hardware is a router. Assuming the case of a distributed router, a router exists at each node of the interconnection network. Therefore, the interconnection network is expressed as an interconnection of routers. Different direct network topologies may be implemented by varying the interconnection of the routers.

The router itself may, in turn, be said to be an interconnection of components where the term component refers to standard components found in the digital libraries of computer aided design tools. It is chosen to model a generic router as an interconnection of a set of modules, where a module is not necessarily a single router component but may be made up of a number of components. However, not all routers require the same modules. As such, a set of modules is provided, a subset of which may be used to describe a given router. Varying the modules or their interconnection allows different router models to be developed. These modules may, therefore, be thought of as building blocks in router design. Selection of the modules for the module set was derived from an analysis of a range of existing and proposed routers to represent both deterministic and adaptive, synchronous and asynchronous designs. A description of these modules may be found in [8].

To take the design process one step further, each module is expressed as an interconnection of logic gates. The framework includes a set of generic logic gate modules, a subset of which may be used to describe a module design. This description takes the form of a net-list.

Performance analysis equations consist of expressions for the three main performance metrics in communication system design — latency, throughput and cost.

To move away from generality and to specialisation, detailed design descriptions are required to create a sufficiently detailed model to enable cost analysis to be undertaken. These are provided in the set of designs. For each module in the generic module set, one or more design instances are available in the design set. Each design is a net-list description of a design implementation for the respective module. These designs are technology specific, generated from logic synthesis of HDL descriptions.

In the case of ASIC technology, the number of logic gates is the cost metric used. As the lowest level in the framework is the logic-gate level then the number of gates for a given module is obtained within the framework from the net-list of a given design. However, in the case of FPGA technology, cost is expressed as the number of combinational logic blocks (CLBs). To obtain the number of CLBs for a given module, the design is first synthesised and then mapped onto the device of choice using a CAD tool. This design data is stored in a design data file. This data is absorbed by the generic modules through attributes in the PDL descriptions. The total cost of the design is expressed in the performance equations described and is equivalent to the total number of logic gates or CLBs, as the case may be.

The final and an important feature of the framework structure is the existence of design parameters. As stated, the goal is to provide a flexible framework such that a wide range of communication systems at the hardware level can be modelled. This wide range of designs is provided, in part, by the generality of design given by the generic modules and also in the addition of parameters to the framework.

It should be noted that flexibility within the framework allows for both the expansion of the module set and the design set. However, since the generic modules each offer frames for a range of module designs, creation of a new module is only required if a new function is added to a router design which is not allowed for within the generic modules provided. An example, is where a proposed solution to the software bottleneck problem involves the movement of a function currently handled at the software level to the hardware level of the communication system.

5 Design Parameters

A set of design parameters has been developed from a study of the key issues in router design. In the PDL descriptions, static parameters provide information regarding design choices made and thus allow these choices to be fed into a model and their effects evaluated. Dynamic parameters may replace static parameters, at the users choice, such that a range of results are obtained for a range of inputs within one run of the evaluator. These parameters may, for example, effect the size of the interconnection network and as such the total cost of the network.

At the HDL level, each module is characterised with key parameters that effect the design and thus complexity of a module. In some cases the effect of a design change may be reflected by a simple change to a constant in the HDL code for a given module. However, it is often the case that a design change effects more than one module. As such, it was decided to put these design parameters into a package available from an HDL library included in all the module descriptions. Where possible, all references to design decisions within the modules refer to the constants and variables of the package. As such, in these cases, module refinement occurs through changes to variables or constants within the VHDL code, re-compilation and logic synthesis to produce a new design instance.

Other parameter changes require recoding of at least part of a given module. As such, it is not possible to automatically update these modules but the presence of an existing version allows reuse of parts of the existing code. Again, re-compilation and logic synthesis is used to create a new design instance.

6 Creating a Model

At the network level, a model contains a number of generic router blocks connected together in the desired topological structure. The process of creating a router instance for this network is illustrated in figure 2. In general, the design parameters — key features of the design, may be relatively easily identified from the literature. However, identifying the generic modules can be a somewhat more complicated task dependent on the amount and type of information available.

Design information may be available in the form of circuit diagrams, written descriptions or missing. A different naming scheme may have been used to describe a given function

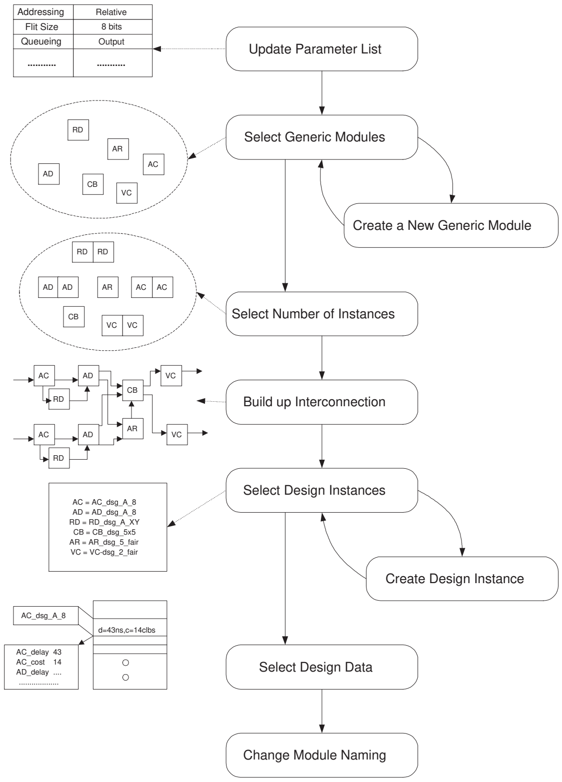

Figure 2: Choosing a Router Configuration

	TRC		RR	
	Cost clbs	**Delay** ns	**Cost** clbs	**Delay** ns
AC	5	7.086	16	9.925
AD	4	6.536	—	—
AR	43	12.954	15	10.88
B	4	5.448	2	3.624
	—	—	48	3.624
CB	71	10.12	1306	12.98
CC	88	19.55	38	3.624
FC	—	—	10	15.733
FU	—	—	37	7.038
MB	—	—	91	4.023
MS	—	—	42	3.624
OP	5	4.847	64	13.257
RD	43	33.932	34	22.494
SCi	9	7.756	—	—
SCo	13	9.415	—	—
SM	5	13.695	—	—
SY	—	—	38	1 CC
VC				
FIFO	88	19.55	1782	17.866
Mx	5	4.898	25	10.457
Ctrl	1	4.919	15	8.04

Table 1: The Torus Routing Chip and Reliable Router Design Data

and different functions may have been lumped together, thus adding to the complexity of this task. It helps to look away from the names used in the design specification and instead consider the functionality of each generic module and match these to the features of the router to be modelled. The framework provides the building blocks to piece together a given design, similar to a jigsaw puzzle in that once a number of blocks are identified, the missing blocks become easier to identify.

The modules of the growing model are then used to build up a path through the router, as shown in figure 3 for the case of the Reliable Router [9] — see [10] for more information. In this router a single physical path consists of six virtual paths. Design parameters identify the number of units of each module required for the complete router design and, following the identified path structure, the complete register transfer level design is put together.

To obtain a gate-level model, the next stage involves selection of the appropriate design instances and the relevant data for inclusion in the model. These design instances are chosen, based on the design parameters, from the design library available. It is a requirement of the PDL language, that each copy of a a given design instance within a model be given a unique name. These names are automatically generated within the framework.

7 Router and Network Evaluation

After modelling specific designs to the gate level the next stage is to be able to evaluate the results. The basis of these results are the performance data from the instantiated module designs. Table 1 provides the performance data for models based on the Torus Routing Chip [11] and the Reliable Router. The Reliable Router offers a number of newer router features in addition to having very wide data

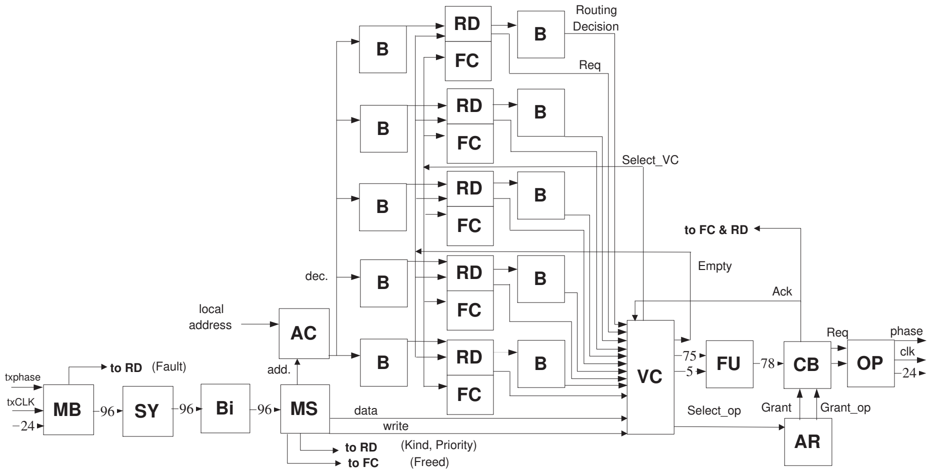

Figure 3: Reliable Router Path

paths. The Torus Routing Chip, on the other hand, is an older router offering less functionality but posing an extra challenge to the modeller to model an asynchronous design.

This performance data represents the cost and delay of single modules within the respective router. Cost and delay of the router itself may be obtained from the model. Cost is dependent on the total number of units and the cost of each module and delay is equivalent to the longest path delay through the router.

The cost of the complete TRC design is 943 CLBs. The Reliable Router, on the other hand, requires 4541 CLBs plus 6 FIFO banks each of which costs 8910CLBs. Although, as stated, this reflects a modelled prototype and not an actual implementation — which may be achieved much more efficiently, it does indicate that the feasibility of a similar design may lie in the implementation of the FIFO banks. This is a problem that was addressed in the actual implementation of the Reliable Router, thus enabling increased throughput to be achieved through these large FIFO banks. The asynchronous TRC model results in a router delay of 149.4ns. The synchronous Reliable Router requires a maximum of 8 clock cycles for it's longest path. The slowest module in the Reliable Router model is the routing decision module (RD) with a minimum clock period of 22.494ns (44.5Mhz) gives a longest path delay of 179.95ns. The Reliable Router design is designed for high throughput and fault tolerance, requiring a much more complicated routing decision module than that which was implemented in the Torus Routing Chip.

The above analysis considers register-transfer level and gate-level analysis in the network. At the network level, latency is calculated from the longest path delay. A separation is made between the delay seen by the header of a packet that seen by the flits of a packet. Cost of the network is calculated from the total number of routers, their respective cost and the cost assigned to their interconnections. Throughput analysis, although provided for in the framework, is not available in the current prototype.

This network level analysis enables more typical performance analysis to be conducted at the network level whilst retaining the gate-level precision in the cost and latency figures.

The framework also offers a platform to explore the design space of a router design. This may be in the form of altering design parameters to change, for example, the switching technique, arbitration method or the network degree or it may be to explore different implementation technology possibilities. This later is achieved by synthesising the module design instances into the technology of choice, within the limits of the synthesis tools available.

A design change that may be considered, if pin-out is sufficient, is to increase the bus width in the network. If it is assumed that channel width is altered accordingly then the flit size is assumed to equal the channel width. Otherwise extra hardware is required to handle fragmentation and re-assembly of each flit. Both alternatives may be tested out using the Framework. Assuming the former, for the case of the modelled TRC with 8 (original), 16, and 32-bit buses the cost of the router is 943, 1306 and 2022 CLBs respectively. From the 32 bit case, it can be seen that although the bus width has increased by a factor of four, router cost has only increased by a factor of around two. Although this result sounds promising for the performance improvement, it still indicates that the performance improvements do not come for free.

8 Summary and Future Work

A new approach to performance analysis of multicomputer communication hardware modelling has been introduced — a modelling framework. It provides a common platform for the design and analysis of current, proposed and future communication hardware design ideas and technologies. Improved accuracy in performance analysis is achieved due to the inclusion of gate-level performance data and design dependent cost analysis is extended to the gate-level.

Implementation of the prototype has reached the stage where the viability of the approach may be analysed. Further improvements to the prototype are required such as extension to the design library and automisation of design instance selection from the design parameters given. Future work will involve improvements to traffic handling and the introduction of realistic traffic to the modelling process. Use of BSPlab[12], where realistic traces of computation, synchronisation and memory access may be generated for a given parallel architecture, is one area to be investigated.

References

[1] J. Hsu and P. Banarjee. Performance Measurement and Trace Driven Simulation of Parallel CAD and Numeric Applications on a Hypercube Multicomputer. In *Proceedings of the 17th Annual International Symposium on Computer Architecture*, pages 260–269, 1990.

[2] J. H. Kim, Z. Liu, and A. A. Chien. Compressionless Routing. In *Proceedings of ISCA'94*, 1994.

[3] S. Abraham and K. Padmanabhan. Performance of Multicomputer Networks under Pin-Out Constraints. *Journal of Parallel and Distrbuted Systems*, pages 237–248, Aug 1992.

[4] A. Agarwal. Limits on Interconnection Network Performance. *IEEE Trans. on Parallel and Distributed Systems*, 2(4), Oct. 1991.

[5] S. Loucif, M. Ould-Khaoua, and L. M. Mackenzie. The Express Channel Concept in Wormhole-Routed Hypermsh and K-Ary N-Cubes. *Proc. 8th IEEE Symposium on Parallel and Distributd Prcessing, SPDP'96*, pages 595–602, Oct 1996.

[6] A. A. Chien. A Cost and Speed Model for K-Ary N-Cube Wormhole Routers. In *Proc. of Hot Interconnects '93*, 1993.

[7] B. Vemuri, R. Mandayam, and V. Meduri. Performance Modelling Using PDL. *IEEE Computer*, pages 44–53, April 1996.

[8] P.C. Haddow. A Generalisaton of Router Chip Design. In *5th Euromicro Workshop on Parallel and Distributed Processing*, pages 307–313, Jan. 1997.

[9] W. J. Dally and L. R. Dennison. The Reliable Router: A Reliable and High Performance Communication Substrate for Parallel Computers. In *Proceedings of the 1st International Parallel Computer Routing and Communication Workshop*, 1994.

[10] P. Haddow. *A Framework for Modelling Communication Hardware in Multicomputers.* PhD thesis, The Norwegian University of Science and Technology, 1998.

[11] W. J. Dally and C. L. Seitz. The Torus Routing Chip. *Distributed Computing*, 1:187–196, Jan 1986.

[12] L. Natvig. Computational Models for Parallel Computing and BSPlab. Technical Report 2/98, Department of Computer and Information Science, NTNU, 1998.