A Modelling Framework Enabling Cost Analysis of Communication System Hardware in Multicomputers

Abstract

The development of parallel computer systems has been aimed at the high-end user — those whose main aim is to achieve maximum performance. However, if the market in parallel computers is to expand, developers need to look at the medium-end users — those that are looking for the best performance within a given cost. As such, it is not enough to consider the performance of a network but also consideration should be given to the cost of the implementation.

Parallel computer systems are emerging fast and as suggested by Spertus [Sea93], the time has come to shift from the 'big new ideas' to careful quantitative analysis. There has been much research in the area of performance analysis of interconnection networks and recently increasing consideration has been given to the bottleneck at the software level of the communication system. Here the focus is still performance and many of the proposed solutions lead to an increase in complexity at the network level giving rise to an increase in costs.

In this work, a flexible framework for communication system hardware modelling is presented which enables cost analysis of both existing and proposed communication systems at the hardware level of a multicomputer.

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 is assumed to be a direct-network where a fixed interconnection pattern, the topology, exists in the network. An interface — a router, is required to control the flow of information between each processor and the network.

For a given network, design of the router module involves decisions regarding many interdependent design issues, selection and design of the required components, design of their interconnection and finally the choice of an implementation technology. The topology is not only reflected in the interconnection of the routers but in the design of the routers themselves.

2 Background and Motivation

Traditional performance models have not, in general, focused on the cost issue. A number of researchers have looked at the technology constraints with respect to pin-out limitation and wire density [AP92, Aga91, DS87, LOKM96, MOK94]. 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 [Chi93], 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. His aim being to make the supporters of adaptive routers substantiate their claims. That is to consider the performance benefits in the light of the costs involved.

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.

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 [VMM96] 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 a 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 technologies 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.

This article focuses on the cost issue and further information regarding the performance issue may be found in [Had97, HN97].

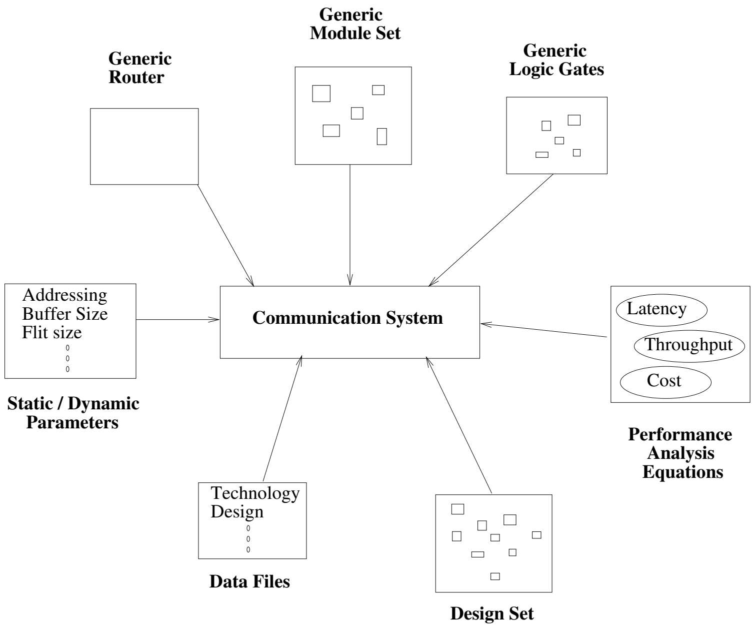

Figure 1: The Network Framework

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 a communication system 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.

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. Column two of table 1, expresses those design issues which effect the performance equations. It should be noted that communication protocol includes both the header size and the packet length used in performance analysis. Also, that the design issues degree and dimension are used to parameterise different network topologies.

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

Issue	Perf.	Const.	Code
Flit Size	x	x	
Comm. Protocol	x		x
Addressing			x
Switching Technique	x		x
Buffer Size		x	
Routing Algorithm			x
Arbitration Method			x
Request Coding			x
Control			x
Number of VCs		x	x
Degree		x	x
Partitioning			x
Dimension	x		

Table 1: Design Issues as Framework Parameters (Perf. = performance, Const. = HDL Constant, Code = Code Change)

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, recompilation and logic synthesis is used to create a new design instance.

The last two columns of table 1 identify those design issues which are used as parameters in the HDL code and those that effect a more major design change. It should be noted that in some cases a design issue may effect both features in different modules e.g. degree.

6 Experimentation

The following examples are chosen to illustrate the key features applicable to cost analysis offered in the framework. That is ease of modelling widely different routers and studying the effect of small design changes. The routers given are described in PDL along with their respective interconnection networks. Design instances for the modules were synthesised using the Aldec's Xilinx Foundation Series [XIL] for XC4000 technology using a XC4036 device. Mapping onto this device produced the cost metric in terms of CLBs. Total cost is generated within the PDL model for the router.

The Torus Routing Chip (TRC) [DS86] was designed for the two-dimensional torus. The TRC is an asynchronous deterministic router that is well referenced in the literature and, as such, may be regarded as one of the better known designs. It also offers a modelling challenge in that not only the communication channels are asynchronously controlled but also the router itself. To implement a synchronous router at this level of detail is a much simpler task and, as such, it was important to show that the framework could handle asynchronous designs.

Figure 2 illustrates a physical path through the TRC. That is a path from input to output port. A similar path exists for each dimension of the router and the processor path. The TRC also

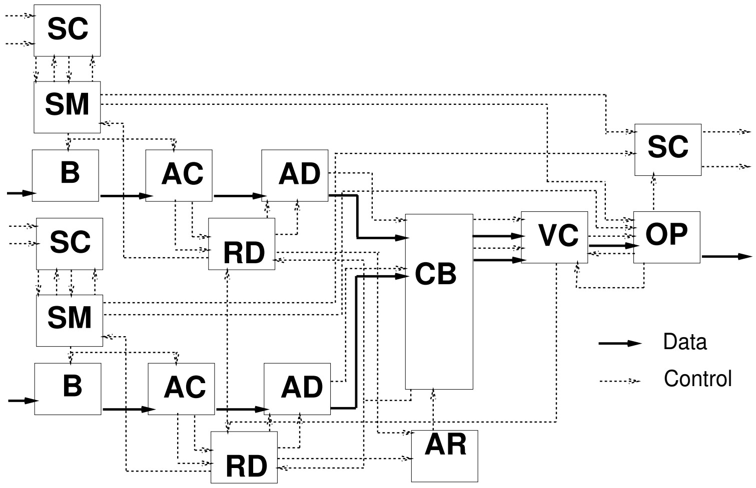

Figure 2: Torus Routing Chip Path

introduced the concept of virtual channels [Dea87]. In this implementation, two virtual channels exist on each physical path as shown by the duplicated logic at the inputs to the router — prior to the crossbar module (CB).

Each block represents a module from the module set and their interconnection is specified in the PDL design file. The modules given represent a signal converter (SC), a signal manager (SM), a buffer (B), an address comparator (AC), a routing decision module (RD), an address decrementer (AD), a crossbar (CB), an arbiter (AR), a virtual channel controller (VC) and an output module (OP). It should be noted that a centralised crossbar and arbiter are implemented in the router and as such only a single crossbar and arbiter module are required to handle all the paths. Also, that the virtual channel controller includes queueing, multiplexing and multiplexor control for the virtual channels of a physical channel.

This modular router description enables the structure of the router to be modelled. That is it offers a building block method to put together a router description. However, each of these modules differ in not only their functionality but also their complexity. As such, it is not sufficient for cost analysis to assume a fixed cost per module but a step nearer to a hardware implementation is needed i.e. selection of designs for each of the modules. The required designs are selected from the design set based on the design decisions made in the TRC i.e. design parameters. If the required designs is not available then a new design may be generated, as described above.

Design variations may also be obtained by varying the design parameters. The original TRC design used an 8 bit flit. Cost Analysis has been undertaken for the TRC using the original 8 bit flit and varying the flit size to both 16 and 32 bit flits. If pinout of the router chip allows an increase in flit size then the increased data path may be expected to increase performance. However what is the cost of this increase?

As the flit size effects the width of the data bus through the router then it effects the number of inputs and outputs of each of the modules on the data path and, as such, increases their complexity. An increase in complexity results in an increase in costs, as shown in table 2. From figure 2 and table 2 it may be seen that it is only the modules on the data path that increase in cost as the flit size increases.

Symbol	8 bit (# CLBs)	16 bit (# CLBs)	32 bit (# CLBs)
AC	5	9	19
AD	4	8	16
AR	34	34	34
B	4	8	16
CB	40	71	135
OP	2	2	2
RD	55	55	55
SCo	12	12	12
SCi	18	18	18
SM	8	8	8
VC			
— Fifo	62	100	194
— Mx	5	9	17
— Ctrl	6	6	6
Total Cost	862	1117	1711

Table 2: Torus Routing Chip Cost

Symbol	75 bit (# CLBs)
AC	15
AR	238
B	38
CB	2371
FU	7
MB	377
MS	46
OP	38
SY	
RD	
VC	
— Fifo	1994
— Mx	21
— Ctrl	60
Total Cost	

Table 3: Reliable Router Cost

Issues	**Components**							
	AC	AD	AR	B	CB	FC	OP	VC
Degree	x	x		x		x	x	x
Number of VCs	(x)	x		x		(x)		
Partitioning			x		x			

Table 4: Design Issues Effecting the Number of Units Required of each Module

It is important not only to consider the increase in cost of the modules lying on the data path but also the overall cost increase in the router. In table 4, the design issues which effect the number of modules required in a design are identified. The modules given are a selection of the modules from the module set. The degree of the network indicates the number of physical paths through the network and those modules marked with an 'x' require at least one unit of the module for every physical path. The 'number of VC's' refers to the number of virtual channels. For those modules where an 'x' is given require a unit of the module for every virtual channel. That is that the number of modules required is equivalent to the number of virtual channels multiplied by the number of physical channels. Where '(x)' is given, either a unit may be placed on the physical channel or the module may be distributed over the virtual channels at the designers discretion. Partitioning refers to the case where a crossbar is partitioned instead of a single central crossbar. The number of units of the crossbar required depends on the method of partitioning chosen by the designer. As an arbiter is the crossbar controller, it is generally partitioned in line with the crossbar and therefore the number of units is equivalent to the number of crossbar modules.

Considering the design of the Torus Routing Chip with two virtual channels, degree of two and a non-partitioned crossbar where all input modules are duplicated on the virtual channels, this gives rise to a total cost of 862 CLBs as given in table 2. It is important to consider these total cost figures before comparison against performance improvements such that a fair comparison may be made. In the results shown, using the crossbar as an example, cost has increased by a factor of around 3.5 in the 32 bit case as compared to the 8 bit case. However, total cost has only increased by a factor of 2. However, this total cost increase is seen at every router in the interconnection network and therefore this factor will be seen at every node in the network and therefore the change in total cost for the network will depend on the network size. These results, if required, may be generated within the framework.

The Reliable Router (RR) [DD94] is chosen to illustrate modelling of a more recent router. The Reliable Router is an adaptive router designed for a 2-dimensional mesh. A single path description of the RR is presented in figure 3. The modules not seen in the Torus Routing Chip include message build (MB), synchroniser (SY), Message Split (MS) and Flow Control (FC).

Unlike the Torus Routing Chip the RR is fully synchronous. Synchronous design is, in general, less complex than an asynchronous design as shown in table 5. Here, a number of modules for the TRC have been adapted for synchronous control. However, the cost analysis for the RR, presented in table 3[1] shows a significant increase in cost over that of the Torus Routing Chip. This is due to many newer features, not seen in earlier routers, such as adaptive routing and fault tolerance. The adaptive routing implementations chosen and the fault tolerance give rise to an increase in the number of virtual channels i.e. duplicated logic, increased queueing and a more complex routing module. Another performance improvement is the introduction of bidirectional simultaneous signalling. A previous solution used to avoid the arbitration delay for a single bidirectional channel was the use of two unidirectional channels. This solution, however, increased the pinout problem, reducing the possible width of the data channels. However, bidirectional simultaneous signalling enables an efficient single bidirectional channel, thus allowing larger data paths and increased complexity within the router.

Symbol	**Async.** (# CLBs)	**Sync.** (# CLBs)
AC	5	3
L	4	4
CB	87	43
Fifo	88	54

Table 5: Asynchronous Vs. Synchronous Control

7 Conclusions

An important feature of this type of analysis is looking into the future. It is not sufficient to design a modelling tool for todays technology. Technology is changing fast and therefore it is important that cost analysis may be obtained in today's and tomorrow's technology. In the framework presented, the use of HDL languages to create module design instances allows these designs to be instantiated into different technologies, within the limits of the technologies offered by the HDL tools.

[1] The results missing were unavailable at the time of submission but will be added to the final version of the paper.

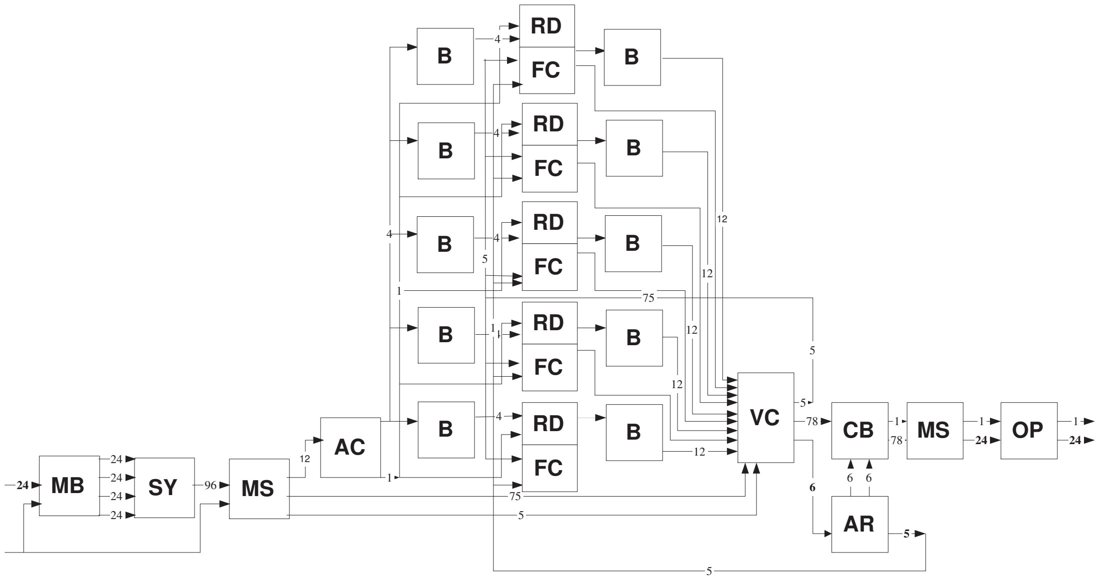

Figure 3: Reliable Router Path

A standardisation effort is in progress to standardise logic synthesis methods. That is that logic synthesis tools following the standard will generate similar net-lists for a given design. However, lack of completion of this standard means that each logic synthesis tool may give rise to different net-lists. As such, it is advisable that all modules in a design instance originate from the same logic synthesis tool or at least the effect of this feature be taken into account when results are analysed.

This paper has presented a framework, focusing on the cost analysis feature and the types of results obtainable. Experiments have been undertaken to study the effect of variation in design ideas. Small design changes are shown to effect the complexity of the design and thus the cost of it's implementation. Newer design ideas which achieve increased performance are shown to increase cost. However, it is not the aim of this work to indicate that increased cost is necessarily a negative factor, but that cost should be taken into account for performance improvement justification. On the other hand, it is as important to remember the performance achievable for these costs as it is to assess the cost of increased performance. That is to say that these this type of analysis should go hand in hand in modelling communication system architecture. An interesting effect of the more detailed modelling approach required to enable cost analysis is that the more complex model may also be used to obtain more detailed information about performance and thus enhance performance analysis.

References

[Aga91] A. Agarwal. Limits on Interconnection Network Performance. *IEEE Trans. on Parallel and Distributed Systems*, 2(4), Oct. 1991.

[AP92] S. Abraham and K. Padmanabhan. Performance of Multicomputer Networks under Pin-Out Constraints. *Journal of Parallel and Distrbuted Systems*, pp 237–248, Aug

1992.

[Chi93] A. A. Chien. A Cost and Speed Model for K-Ary N-Cube Wormhole Routers. In *Proc. of Hot Interconnects '93*, Aug. 1993.

[DD94] W. J. Dally and L. R. et al Dennison. Architecture and Implementation of the Reliable Router. In *Proceedings of the Hot Interconnects 11*, Aug. 1994.

[Dea87] W.J. Dally et al. Deadlock-Free Message Routing in Multicomputer Interconnection Networks. *IEEE Trans. Comp.*, C36(5), 1987.

[DS86] W. J. Dally and C. L. Seitz. The Torus Routing Chip. *Distributed Computing*, 1 pp 187–196, Jan 1986.

[DS87] W. J. Dally and P. Song. Design of a Self-Timed VLSI Multicomputer Communications Controller. *Proceedings of the Intern. Conf. on Computer Design*, (IEEE CS Press, 2473) pp 230–234, 1987.

[Had97] P.C. Haddow. A Generalisaton of Router Chip Design. In *5th Euromicro Workshop on Parallel and Distributed Processing*, pp 307–313, Jan. 1997.

[HN97] P.C. Haddow and L. Natvig. A Case Study in Multilevel Architecture Modelling using HDLs. In *The 4th Asia-Pacific Conference on Hardware Description Languages (APCHDL'97)*, 1997.

[LOKM96] 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*, pp 595–602, Oct 1996.

[MOK94] R Sotudeh M Ould-Khoua, L M Mackenzie. Comparitive Evaluation of Wormhole-Routed Hypergraph and Multi-Stage Interconnection Networks. *The Computer Journal*, 39(3) pp 232–240, 1994.

[Sea93] E. Spertus et al. Evaluation of Mechanisms for Fine Grain Parallel Programs in the J-Machine and the CM-5. Technical report, AI, Lab MIT and EECS, University of California, 1993.

[VMM96] B. Vemuri, R. Mandayam, and V. Meduri. Performance Modelling Using PDL. *IEEE Computer*, pp 44–53, April 1996.

[XIL] Xilinx Founation Series. *http://www.aldec.com/*.