Leveraging Application-Level Requirements in the Design of a NoC for a 4G SoC - a Case Study

Rudy Beraha[1], Isask'har Walter[2], Israel Cidon[2], Avinoam Kolodny[2]
[1]Qualcomm Corp. Research and Development, San Diego, California 92121, USA
[2]Electrical Engineering Department, Technion – Israel Institute of Technology, Haifa 32000, Israel
rberaha@qualcomm.com, {zigi@tx, cidon@ee, kolodny@ee}.technion.ac.il

Abstract—**In this paper, we examine the design process of a Network on-Chip (NoC) for a high-end commercial System on-Chip (SoC) application. We present several design choices and focus on the power optimization of the NoC while achieving the required performance. Our design steps include module mapping and allocation of customized capacities to links. Unlike previous studies, in which point-to-point, per-flow timing constraints were used, we demonstrate the importance of using the application end-to-end traversal latency requirements during the optimization process. In order to evaluate the different alternatives, we report the synthesis results of a design that meets the actual throughput and timing requirements of the commercial SoC. According to our findings, the proposed technique offers up to 40% savings in the total router area and a reduction of up to 49% in the inter-router wiring area.**

System on-chip, Network on-chip, Optimization

I. Introduction

Application-specific systems on-chip (SoC) make extensive use of busses as the interconnect infrastructure. These busses are typically enhanced along product generations to match the increasing needs of the application. Such enhancements include increasing the bus frequency and width as well as enriching the bus semantics and transfer modes. By avoiding fundamental changes, the SoC architects can leverage their past experience in designing shared busses and successfully overcome the growing complexity of the design. However, in recent years research has shown that Network on-Chip (NoC) is likely to replace busses in future SoCs, due to superior performance, power and area tradeoffs it offers as the number of modules increases [1][2][3][4]. This is attributed to the spatial parallelism of networks, to their short, unidirectional point-to-point wires and to their scalable architecture [5].

In this work, we discuss the design process of a NoC for a state-of-the-art SoC. Specifically, we describe our experience in designing a cost optimized NoC for a high-performance, power constrained 4G wireless modem application. As the design process has many degrees of freedom creating a very large design space, finding the absolute optimal solution is an extremely difficult problem. Instead, we focus on several important choices made by the system architect while selecting some well-accepted, practical solutions to other questions.

Previous work that has dealt with the design process of the NoC often attempted to minimize power consumption and/or maximize network performance. When real applications are considered, minimizing the power consumption alone (e.g., by module mapping) is impossible, as performance constraints for each given application are to be met. Similarly, maximizing performance alone is inefficient, as excessive power might be used for improving performance beyond the needs of the application. Therefore, we look for a tradeoff between the power and performance of the NoC that is characterized by a minimal power consumption that still meets the demands of the targeted application. Moreover, in studies where network latency was used as a performance goal (either as a cost function or as a constraint), the average delay of all packets over all communicating pairs was typically considered. However, in a practical SoC, different streams of communication may require different delays and therefore the overall average latency is an inappropriate measure.

In this paper, we go further to suggest an improved approach: given the application that is to be used in the SoC, we utilize its *functional timing requirements*, which are defined by the application latency constraints. Each of those end-to-end traversal delay requirements is composed of the cumulative requirement of a sequence (or a "chain") of flows. For example, the application may require that a block of data which is generated by module A is sent to module B in order to be processed. Then, the processed data is to be sent by module B to module C for some additional processing, forming a pipeline of modules. By observing that the performance of the application is subject to the total time it would take the data to get from module A to module C, we can use this delay as the targeted performance measure, rather than specifying two separate latency constraints (for the flow from module A to module B and from module B to module C). Since pair-wise delays may be traded, the timing constraints are relaxed and the optimization tool has more freedom in its operation.

This approach is similar to re-timing of logic paths used in traditional logic synthesis tools which may "borrow time" from one pipeline stage to another to balance the timing paths and achieve high frequency of operation, as long as total latency is not violated. Instead of moving logic from one unit to another, the proposed technique modifies the regular NoC design flow to generate a more efficient implementation. To the best of our knowledge, this paper is the first to discuss and quantify the benefits of specifying the end-to-end traversal requirements during the mapping of the NoC. As the main data path in SoCs is typically composed of such processing pipes, the proposed scheme is not limited to any particalur application.

The design process is composed of several steps: first, using simulated annealing optimization, we search for a minimal power mapping of modules, taking into account the application latency and throughput requirements. Then, uniform link capacities are defined to meet these performance constraints.

978-3-9810801-6-2/DATE10 © 2010 EDAA

Finally, the resulting uniform NoC is optimized by tuning the capacity of selected links.

The rest of this paper is organized as follows: in Section II, related work is discussed. In Section III, we describe the characteristics of the application of the designed SoC. In Section IV, we discuss the design and optimization process of the NoC and in Section V we report and analyze its cost. In Section VI, we summarize the paper.

II. Related Work

NoC design was the subject of many papers in recent years. In particular, the problem of mapping the communicating cores onto the die has received considerable attention, due to the power and performance implications it has. In [6], the authors propose a branch-and-bound mapping algorithm to minimize the communication energy in the system, but the resulting communication delay is not considered. In [7], a heuristic algorithm is used to minimize the average delay experienced by packets traversing the network. By allowing the splitting of traffic, an efficient implementation is found. In [8], the authors use the message dependencies of the application in addition to its bandwidth requirements to find a mapping that reduces the power consumption and the application execution time. The authors of [9] use a multi-objective genetic algorithm to explore the mapping space so that a good tradeoff between power consumption and application execution time is found. While these papers use unique mapping schemes, they all use packet delay or application execution time as a quality measure rather than as an input to the mapping phase. Moreover, the metrics used does not consider the individual requirements of each pair of communicating cores, only reflecting the overall average delay or performance.

The earliest published work to consider energy efficient mapping of a bandwidth and latency constrained NoC is [10], in which the authors specify a an automated design process providing quality-of-service guarantees. Another mapping scheme that uses delay constraints as an input is described in [11]. There, a low complexity heuristic algorithm is used to map the cores onto the chip and then routing is determined so that all constraints are met. Similarly, the mapping schemes used in [12][13][14] all use the per-flow, source-destination latency requirements of the application as input to the design process and find a cost effective mapping of the cores onto the chip, satisfying the timing demands.

In this work, we motivate a third approach: rather than optimizing the NoC for power only and evaluating the resulting delays; or using the per-flow delay requirements as constraints during the mapping process, we use the application-level requirements which dictate end-to-end processing latencies. Wherever applicable, we replace "a chain" of point-to-point delay constraints with a single, unified constraint, describing the overall latency requirement of the application, measured from the time the first module in the chain generates the data until the last module receives the data, as explained above. A good example of the benefit that lies in leveraging the application end-to-end temporal requirements for NoC design is given in [15], where the application data-flow is analyzed to facilitate the sizing the NoC buffers.

III. The Application

The NoC to be designed is for a 34 nodes ASIC that supports all major 2G, 3G and 4G wireless standards for use in base stations and femto-cells (Cell Site Modem – CSM). The existing, bus-based implementation of this application is depicted in Fig. 1. The CSM is designed to support any of the CDMA or UMTS standards, because different markets around the world are at different points in their adoption of wireless standards.

This CSM is comprised of several subsystems that fall into three basic categories: (1) Generic Element. These are the processor and DSP modules on chip. They are programmable and can be used for a variety of different functions; (2) Dedicated hardware. These blocks are designed to optimize the operations/milliwatt metric. They perform a single (or a small set) of operations extremely efficiently and off-load the work from the generic elements (which typically could perform the same operation but with a significant power penalty); and (3) Memory/IO. As with most SoCs, there are memory elements and I/O modules used for information storage and communication with the outside. For the purposes of this paper, these elements are grouped together.

In the bus-based implementation, the SoC uses a 64-bit wide, 166MHz AXI bus at the top-level. Due to design considerations such as place&route and timing closure, the interconnect fabric is segmented into two separate busses and a bridge, with approximately half the nodes on each bus.

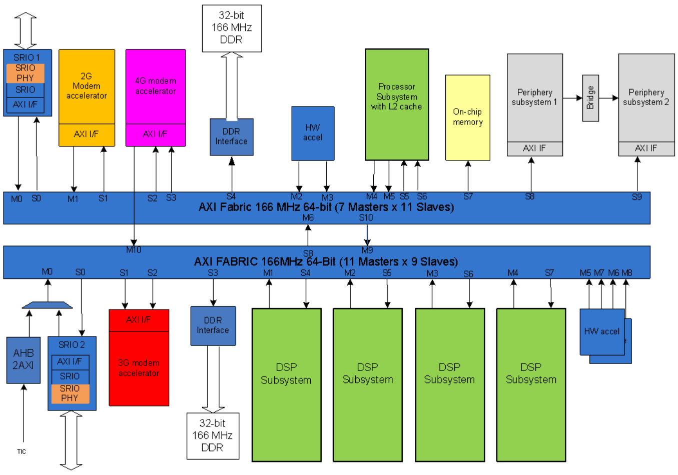
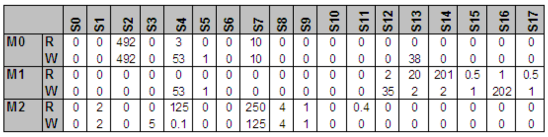


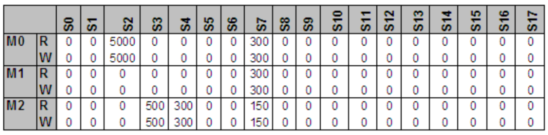
Figure 1: Bus-based system architecture

Table 1



		S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17
M0	R	0	0	492	0	3	0	0	10	0	0	0	0	0	0	0	0	0	0
	W	0	0	492	0	53	1	0	10	0	0	0	0	0	38	0	0	0	0
M1	R	0	0	0	0	0	0	0	0	0	0	0	0	2	20	201	0.5	1	0.5
	W	0	0	0	0	53	1	0	0	0	0	0	0	35	2	2	1	202	1
M2	R	0	2	0	0	125	0	0	250	4	1	0	0.4	0	0	0	0	0	0
	W	0	2	0	5	0.1	0	0	125	4	1	0	0	0	0	0	0	0	0

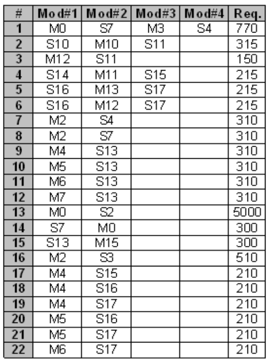
Table 2



		S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17
M0	R	0	0	5000	0	0	0	0	300	0	0	0	0	0	0	0	0	0	0
	W	0	0	5000	0	0	0	0	300	0	0	0	0	0	0	0	0	0	0
M1	R	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0	0	0	0
	W	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0	0	0	0
M2	R	0	0	0	500	300	0	0	150	0	0	0	0	0	0	0	0	0	0
	W	0	0	0	500	300	0	0	150	0	0	0	0	0	0	0	0	0	0

Sample of point-to-point master-slave flow characteristics: (1) Bandwidth demands [Mb/s] and (2) Point-to-point timing requirements [ns]. 'R' is for read operations; 'W' is for write operations. The complete tables are 32 lines long (2 for each of the 16 masters in the system)

Table 3



#	Mod#1	Mod#2	Mod#3	Mod#4	Req.
1	M0	S7	M3	S4	770
2	S10	M10	S11		315
3	M12	S11			150
4	S14	M11	S15		215
5	S16	M13	S17		215
6	S16	M12	S17		215
7	M2	S4			310
8	M2	S7			310
9	M4	S13			310
10	M5	S13			310
11	M6	S13			310
12	M7	S13			310
13	M0	S2			5000
14	S7	M0			300
15	S13	M15			300
16	M2	S3			510
17	M4	S15			210
18	M4	S16			210
19	M4	S17			210
20	M5	S16			210
21	M5	S17			210
22	M6	S17			210

Timing requirements extracted from the application [ns].

The CSM chosen for this study supports multiple modes of operation, each identified by its own bandwidth and latency requirements. In particular, it can operate in a 2G mode, in a 3G mode, in a 4G mode, and in a combination of modes for simultaneous voice and data transmissions. To find a low-cost 2D mesh topology for the NoC, an artificial set of bandwidth requirements is generated [16] [17]: for each pair of nodes, the maximum bandwidth requirement it has in any of the modes of operations is selected. Similarly, we combined all the latency requirements in one table. This scenario represents the worst-case requirements in any of the modes ("synthetic worst-case" [16], "design envelope" [17]). Designing the NoC according to this scenario is likely to make it easier to meet requirements of all modes of operation in the following phases of the design, while other approaches are left for future work.

For the purpose of this paper, the 95 point-to-point (P2P) flows in the system running between the 16 masters and 18 slaves were described using two tables: one table specifies the bandwidth of each flow and the other specifies its timing requirements. Sample of these tables are shown in Table 1 and Table 2 (due to space limitation, those tables only contain partial information, while all relevant data is made publicly available in [18] to be used by the NoC community as a benchmark for future research). A third table specifies the application's end-to-end traversal delay requirements, derived from the application characteristics (Table 3). The tables reveal that there is a wide variability in the requirements, at both the bandwidths and the delays. For example, Master0 sends Slave2 492Mb per second, with a latency constraint of 5000ns, while Master4 sends Slave16 only 10 Mb per second, but with a much tighter delay requirement of 200ns. This variability, which is very common in modern SoCs, makes the problem of designing an efficient NoC more challenging.

IV. NoC Design and Optimization

The design process of the NoC is composed of four phases: mapping the communicating modules (e.g. [6]-[14]); trimming and adjusting the network resources to meet the application requirements [19]; synthesizing the network; and placing and routing of the NoC. The initial topology chosen for the NoC is a widely used 2D regular mesh grid that mitigates the concern of deadlocks and also simplifies the routing algorithm. However, the proposed design technique is applicable to other topologies too. In order to simplify the mapping process, all modules are considered to be of the same size during this step of the optimization, leaving it to the place&route tool to account for the actual placement of the chip. A more complex approach is left for future work. In order to minimize the buffering cost and allow fast delivery of data, wormhole switching is used.

A. Cost Optimized Mapping

In order to find the best 2D mesh topology, we explore three possible optimization goals: (1) Power-only: in this mapping, only the bandwidth requirements of the application are considered, while meeting the timing requirements is left for the following stages of the design process; (2) (Power+P2P)-based mapping: Here, point-to-point latency requirements are introduced as constraints in the mapping phase; (3) (Power+E2E)-based mapping: instead of specifying latency requirements for each source-destination pair, the end-to-end (E2E) traversal latency constraint of the stream of information in the application is used. For example, if data is sent from node-X to node-Y and then from node-Y to node-Z, the E2E latency is measured between node-X and node-Z. These constraints are extracted from the application's characteristics and replace some of the P2P requirements (a P2P requirement that is not a part of a longer chain cannot be replaced), creating a more relaxed set of constraints.

In order to find an optimal mapping for the SoC, we define a cost function which is used to compare different mappings. The cost function is defined as:

$$Cost = \alpha\, AREA_{router} + \beta \sum_{l \in links} BW_l \qquad (1)$$

where $AREA_{router}$ is an estimate for the total resources required to implement the router logic (accounting for each individual router number of ports and the hardware needed for the capacity it provides, which change from one mapping to another), and BW_l is the bandwidth delivered over a link l. While $AREA_{router}$ models the area and static power used by the NoC resources, the second term is commonly used to capture the dynamic power consumed by the communication (e.g. [20][21]).

In order to search for an optimal mapping, a topology optimization tool that uses a simulated annealing (SA) algorithm was developed. The tool, which is capable of evaluating different MxN configurations for the 2D mesh, takes as input a spreadsheet listing connectivity and bandwidth requirements between nodes. In addition, it can read a spreadsheet with latency requirements which are specified in one of two ways: (1) a list of the maximum latency allowed between any two nodes on the network; (2) list of the E2E streams and their allowed latency (including P2P requirements that could not be replaced), i.e., the nodes a particular operation must traverse and the total latency allowed for that set of flows.

The SA algorithm starts with a random mapping of all nodes on a 2D mesh and calculates the cost (Eq. 1) for this initial state. It then proceeds to try and swap nodes in order to find a lower cost solution. The bandwidth spreadsheet will drive the selection of a topology as this is directly included in the cost. However, for each solution that the SA algorithm generates, the tool uses the latency spreadsheet to check if the latency requirements are met. When the requirements are not met, the solution is rejected regardless of its cost.

Since run-time, dynamic effects for congestion are hard to predict during the mapping phase, hop-count is often used instead (e.g., [11]). Therefore, the check reflects the length of the path traversed by the packet and the pipeline delay of the routers along that path. This approximation is accurate enough to be used by the mapping algorithm as NoCs are typically designed to operate in light loads such that congestion effects are not dominant. However, other, more elaborate analytic delay models can be equally used to account for source queuing, VC multiplexing and contention [19], packetization/reassembly delay, processing time within modules, etc. Specifically, we assume a 3 cycle router pipeline delay, operating at 200MHz, and account for contention in subsequent stages of the design. We use the SA tool to generate mappings using the Power-only, Power+P2P, and Power+E2E schemes, resulting in three topologies to compare. For the purpose of this paper, we use α=10, β=1 and relative empirical weights for routers with different numbers of ports, as generated by synthesis tools. Fig. 2 shows the mappings generated by the three schemes.

B. Setting Link Capacities

As a significant portion of the NoC area and power consumption is due to the network links, minimizing the resources used by the links has a considerable impact on the design process. In this phase, we find the required link capacity or each of the mappings generated in the mapping step.

We define the total capacity of the NoC as

$$NoC\_Capacity = \sum_{l \in links} C_l \qquad (2)$$

where C_l is the capacity assigned to link l, and attempt to find the minimal total capacity that would still meet all the latency constraints (same ones that were used in the mapping phase). As the mapping tool doesn't consider the dynamic contention within the network, this phase of the optimization process should account for all run-time effects, so that the network can deliver the required performance.

In this paper, we consider two possible schemes: a uniform allocation, in which all links have the same capacity, and a heterogeneous allocation where different links may have different capacities. Uniform link capacity is commonly used in wormhole networks. In such cases, the process of finding the minimal capacity that meets the latency requirements using simulations is rather simple, as a single parameter (the identical capacity of all network links) is optimized. However, due to the variety of timing requirements presented by the application, this allocation causes some links to be over-provisioned. In order to reduce the cost of the links, we differentiate between two types of links: the first type of links is links that are used to route at least one flow which has timing requirement. The second type of links is those that deliver flows with no such requirements. Intuitively, it is possible to scale down links of the latter type more aggressively than those of the former type. However, it is important to note that scaling down the capacity of links that have no flows with timing requirement may hinder the delivery of flows that have latency constraints but do not traverse these links. This is due to the backpressure mechanism of wormhole switching: when a flow is slowed down in a certain router on its path, it occupies resources in other routers on its path for a longer time. Consequently, the delay of flows that share these other routers and which may have latency constraints increases.

In this work we generate the custom, tuned allocation by scaling down the capacity found in the uniform assignment scheme: the capacity of links that are used only by flows with no timing requirements is re-assigned according to a selected utilization factor. The capacity of links which have at least one flow with latency constraint is reduced proportionally to the slack time of the flow with the lowest slack, so that reducing the capacity any further would definitely violate the timing constraint of that flow. Simulation is then used to verify that all latency constraints are met. If not, capacity is increased by a small factor and performance is verified again. In both the uniform and custom tuning schemes, links that are not used by any flow in any of the modes are completely removed.

Using an OPNET-based simulator [22] that models a detailed wormhole network (accounting for the finite router

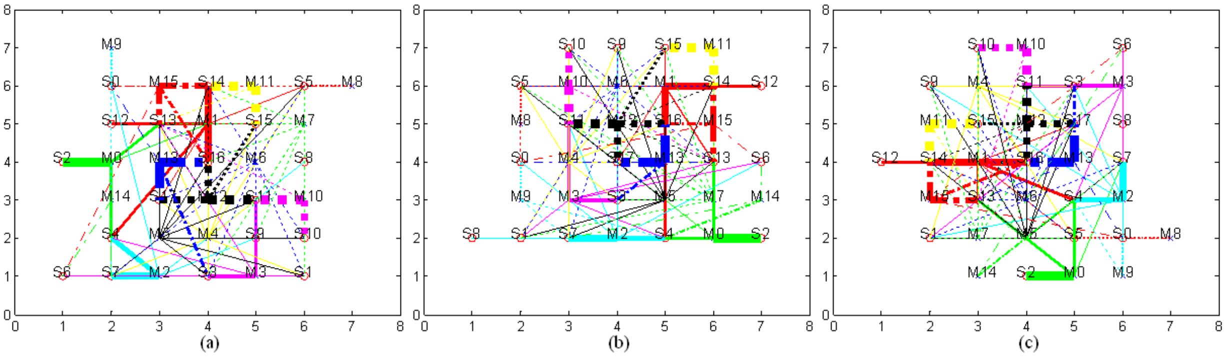


Figure 2: Mapping results

(a) Power optimized; (b) power optimized+P2P timing constraints; (c) power optimized+E2E traversal timing constraints. Line widths represent the relative volume of traffic.

queues, backpressure mechanism, virtual channel assignment, link capacities, network contention, etc.), the basic three topologies (generated by the Power-only, Power+P2P and Power+E2E optimizations) were evaluated, using one and two virtual channels (VCs). For each case, we find the optimal network bandwidth for both the uniform and the tuned links capacity cases. This phase results in 12 generated networks (3 basic mappings * 2 VC configurations * 2 capacity schemes). At the end of this phase, all timing requirements are met (P2P constraints in the Power-only and Power+P2P mappings; and E2E-traversal constraints in the Power+ETE generated mappings). Fig. 3 summarizes the results, presenting the total capacity required in each of the 12 configurations.

It should be noted that tuning the capacities of links in this phase may result in arbitrary capacity values. However, the implemented hardware can support only a finite, discrete set of capacities. While setting unique link frequencies is possible, in this work, customized capacities are achieved by means of different flit sizes (32, 64 and 128 bits width). The cost of the hardware required for the translation (rate matching blocks) is accounted for in the following section.

V. Synthesis Results

The optimization of the 1 VC network versus the 2 VC network results in different capacity requirements for both the uniform and tuned cases. For some links, the 2 VC approach resulted in a lower link capacity because of the improved link utilization offered by the additional VC. However, the area impact of a two VC router must also be taken into account when choosing the best topology. Another factor to consider in the design of the network is the supported flit width. While the network bandwidth allocation algorithm allowed for any speed, the implementation of the NoC on the ASIC is limited to the clock frequencies and flit widths available in the design. For this reason, we bin the resulting router configurations into discrete categories supported on chip. We applied this binning strategy to all topologies and synthesized the network for each. Fig. 4 and Fig. 5 show the results reported by the TSMC 65nm process technology synthesis tool, separately listing the cell area and routing area. The cell area includes the area taken up by the rate matching blocks needed for translating one flit width to another in the network. It also accounts for the trimming of the routers, achieved by the removal of unused ports.

Analysis of the results shows that using the network capacity allocation scheme reduced the capacity of the over-provisioned links thus saving area and power. The results also indicate that the Power+E2E latency approach provides a considerable better solution. To understand this, we must go back to mapping phase (Section IV.A). In the Power-only case, the latency requirements are completely ignored, which gives SA algorithm the most flexibility in placing the nodes on the network. When latency is included in the topology planning, the tool will reject any solution that does not meet the latency requirements. This effectively reduces the solution space for the SA algorithm. Because of this, the Power+P2P scheme is the most restrictive, while in Power+E2E scheme the tool has some more flexibility in moving the nodes around as long as the latency is met for the full E2E traversal path.

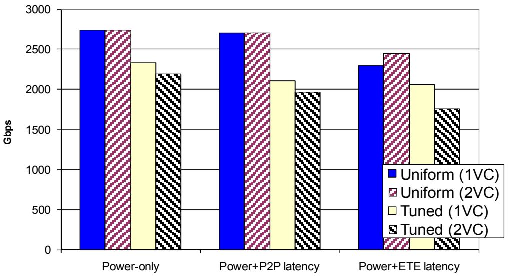


Figure 3: Capacity requirements
The total capacity needed to meet the requirements of the examined configuration, using one and two virtual channels.

The above explanation taken alone would imply that the Power-only case should produce the best results because the topology tool has the highest flexibility. However, Fig. 4 and Fig. 5 show that the Power-only implementation has the largest area, in each VC/allocation scheme. To understand this, one must examine the bandwidth and latency requirements: there are some communication streams that have relatively low bandwidth but still have strict latency requirements. The nature of the topology cost function will place high bandwidth nodes close together in order to minimize the cost. When high bandwidth nodes are put close together, other nodes get pushed further apart. As a result, some low latency nodes will be separated by many hops. As explained above, during the link capacity tuning phase, link bandwidth is set so that all timing requirements are met. The further apart latency-critical nodes are from each other, the higher the link capacity along the path will need to be. This is why the Power-only case results in a very high network capacity, and thus to wider flits and an overall larger area.

The Power-E2E scheme had the most flexibility to map nodes while at the same time making sure that the latency critical signals were relatively close together. Hence, during the network capacity allocation phase, a lower link speed could be used as compared to the Power-only case. This translates into the use of smaller flit widths for the network. Consequently, the ETE-traversal approach reduces the cell area by 25%-40% and wiring resources by 13%-49% compared to the traditional power+P2P mapping scheme.

Interestingly, when we consider the number of VCs, we see that one VC is preferable from an area perspective. Though using VCs reduces some of the link capacities, the savings are more than offset by the increased router sizes. Therefore, the 2 VC approach does not benefit the application.

Finally, we see that the Uniform and Tuned Power+E2E topologies have the same area. The reason for this is the binning strategy: since we are limited to 32/64/128bit flits, our link and router selection is limited set of discrete choices. While it is true that the tuned Power+E2E topology can run some links at a slower speed, the difference from the uniform topology is not significant enough in this case. For example, the tuned topology can reduce the speed of some links down from 15Gbps to 14Gbps, but given the supported flit widths, this does not change the size of the link or router we are able to choose.

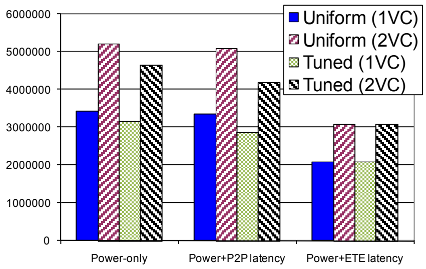


Figure 4: Total router logic area
Total area consumed by routers in each of the three mapping schemes.

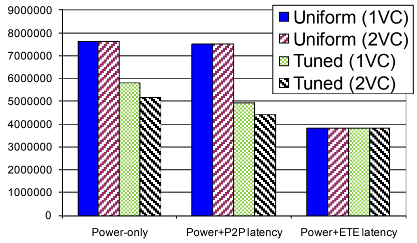


Figure 5: Total wiring area
Total area consumed by inter-router wires in each of the three mapping schemes.

VI. Summary and Conclusions

The increasing communication requirements in system on-chip (SoC) implementations created a need for a new interconnection paradigm. In this paper, we describe our efforts to design a complex SoC around a NoC-based interconnect. In the first phase of the design, we explore three schemes to perform the placing of cores onto the chip: the first scheme only considers the power consumed by the transmission of packets while the second scheme uses the application source-destination latency constraints during the mapping phase. A third technique replaces the pair-wise requirements with application-level end-to-end latency constraints, allowing more freedom in the process of seeking a solution that minimizes power consumption. Next, we trim redundant network resources (links, ports) and tune the bandwidth of links so that the requirements of the application are met. Finally, we synthesize the resulting networks to estimate their cost.

The main contribution of this work is the introduction of the end-to-end traversal delay constraints during the NoC mapping process. By replacing the source-destination requirements with end-to-end requirements wherever possible, we reduce the total area of the routers by 25% to 40% and the link wiring resources by 13%-49%. In addition, we evaluate the potential benefit that lies in the implementation of links with individually assigned capacities. While we focus our analysis on a wireless modem application supporting a plethora of wireless standards and applications, such processing pipes are very typical in SoCs. Therefore, the techniques described in this paper can be used for designing and optimizing NoCs for other high performance, power constrained SoCs. Future work includes placing and routing the NoC and evaluating it against a bus-based system that delivers the same performance.

Acknowlegments

This work was partially supported by the Semiconductor Research Corporation (SRC) and Intel Corp.

References

[1] P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnections", Proc. Design, Automation and Test in Europe (DATE) 2000, 250-256

[2] K. Goossens, J. Dielissen, and A. Radulescu, "AEthereal Network on Chip: Concepts, Architectures, and Implementations", IEEE Design and Test of Computers, 2005, 414-421

[3] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS Architecture and Design Process for Network on Chip", Journal of Systems Architecture, Vol. 50, February 2004, 105-128

[4] D. Bertozzi and L. Benini, "Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip", Circuits and Systems Magazine, IEEE Volume 4, Issue 2, 2004, 18-31

[5] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost Considerations in Network on Chip", Integration - the VLSI Journal, Vol.38, 2004, 19-42

[6] J. Hu and R. Marculescu, "Energy-Aware Mapping for Tile-Based NoC Architectures Under Performance Constraints", Proc. Asia South Pacific design automation (ASP-DAC) 2003, pp. 233–239

[7] S. Murali and G. De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures", Proc. Design, Automation and Test in Europe Conference (DATE), 2004, pp. 896–901

[8] C. Marcon, N. Calazans, F. Moraes, A. Susin, I. Reis, and F. Hessel, "Exploring NoC Mapping Strategies: an Energy and Timing Aware Technique", Proc. Design, Automation and Test in Europe Conference (DATE), 2005, pp. 502–507

[9] G. Ascia, V. Catania, and M. Palesi, "Multi-Objective Mapping for Mesh-Based NoC Architectures", Proc. International conference on hardware/software co-design and system synthesis, 2004, pp. 182–187

[10] S. Murali, L. Benini, and G. De Micheli, "Mapping and Physical Planning of Networks-on-Chip Architectures with Quality-of-Service Guarantees", Proc. Asia South Pacific design automation, 2005, pp.27-32

[11] K. Srinivasan, and K.S. Chatha, "A Technique for Low Energy Mapping and Routing in Network-on-Chip Architectures", Proc. Low Power Electronics and Design 2005, pp. 387–392

[12] A. Hansson, K. Goossens, and A. Radulescu, "A Unified Approach to Constrained Mapping and Routing on Network-on-Chip Architectures", Proc. International conference on Hardware/software co-design and system synthesis (CODES ISSS), 2005, pp. 75–80

[13] K. Goossens, J. Dielissen, O. P. Gangwal, S. G. Pestana, A. Radulescu, and E. Rijpkema, "A Design Flow for Application-Specific Networks on Chip with Guaranteed Performance to Accelerate SoC Design and Verification", Proc. Design, Automation and Test in Europe Conference (DATE), 2005, 1182-1187

[14] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli, "Mapping and Configuration Methods for Multi-Use-Case Networks on Chips", Proc. Asia South Pacific design automation, 2006, pp. 146-151

[15] A. Hansson, M. Wiggers, A. Moonen, K. Goossens, and M. Bekooij, "Enabling Application-Level Performance Guarantees in Network-Based Systems on Chip by Applying Dataflow Analysis", in *IET Computers & Digital Techniques*, 2009

[16] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli, "A Methodology for Mapping Multiple use-cases onto Networks on Chips", Proc. Design, Automation and Test in Europe Conference (DATE) 2006, pp. 118-123

[17] R. Gindin, I. Cidon and I. Keidar, "NoC-Based FPGA: Architecture and Routing," First International Symposium on Networks-on-Chip (NOCS), 2007, pp. 253-264

[18] MATRICS website, http://webee.technion.ac.il/matrics/publications.html

[19] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Network Delays and Link Capacities in Application-Specific Wormhole NoCs," VLSI Design, vol. 2007, Article ID 90941, 2007

[20] D. Shin and J. Kim, "Communication Power Optimization for Network-on-Chip Architectures", *Journal of Low Power Electronics*, vol. 2, pp. 165-176, August 2006

[21] R. Tornero, J.M Orduna, M. Palesi, and J. Duato, "A Communication-Aware Topological Mapping Technique for NoCs", Proc. the 14th int. Euro-Par conference on Parallel Processing, 2008

[22] OPNET modeler (www.opnet.com)