Synthesis of On-Chip Interconnection Structures: From Point-To-Point Links to Networks-on-Chip

Alessandro Pinto
Luca Carloni
Alberto L. Sangiovanni-Vincentelli

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-147
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-147.html

November 14, 2006

Synthesis of On-Chip Interconnection Structures: From Point-To-Point Links to Networks-on-Chip

ABSTRACT

Packet-switched networks-on-chip (NOC) have been advocated as the solution to the challenge of organizing efficient and reliable communication structures among the components of a system-on-chip (SOC). A critical issue in designing a NOC is to determine its topology given the set of point-to-point communication requirements among these components. We present a novel approach to on-chip communication synthesis that is based on the iterative combination of two efficient computational steps: (1) an application of the k-Median algorithm to coarsely determine the global communication structure (which may turned out not be a network after all), and a (2) a variation of the shortest-path algorithm in order to finely tune the data flows on the communication channels. The application of our method to case studies taken from the literature shows that we can automatically synthesize optimal NOC topologies for multi-core on-chip processors and it offers new insights on why NOC are not necessarily a value proposition for some classes of application-specific SOCs.

1. INTRODUCTION

Several researchers have advocated the design of packet-switched NOC to organize efficiently and reliably the global communication on chips designed with nanometer technologies (see Section 2). Designers in the academia have already fabricated chips with examples of NOCs. The MIT Raw chip contains a direct on-chip network implementation to connect a fully programmable SOC consisting of an array of identical computational tiles with local storage. Full programmability means that the compiler can program both the function of each tile and the interconnections among them [29]. Also, two distinct chips that were built at KAIST present NOCs that are based respectively on a one-level flat star topology [22] and two-level hierarchical star topology [21]. On the other hand, the fact that to date there are not many fabricated industrial chips featuring a packet-switched NOC is perhaps a sign that in terms of design application and technology processes, we may have not reached yet the point where such concept is a value proposition. In our intent, the present paper offers a contribution to understand better why this may be the case.

It is a fact that sometime in the NOC literature papers the characteristic of on-chip communication are not taken properly in the account while researchers focus on optimizing quality-of-service (QoS) parameters for on-chip (micro-)networks with methods that are typical of traditional computer (macro-)networks. Instead, it is important to consider the following facts:

- with nanometer technologies corner-to-corner on-chip transmission requires multiple clock cycles of latency [7, 14]; in fact, for a given technology process and a chosen transmission frequency, on-chip communication is limited by the *critical sequential distance* of the metal layer, i.e. the maximum distance that a signal can travel in an interconnect that has been optimally sized and optimally buffered uniformly within a single clock period [27];

- with nanometer technologies bandwidth is not an issue for on-chip communication: the availability of tens of thousands of track per chip edge guarantees a bandwidth per signal in the order of gigabytes and a global on-chip bandwidth in the order of terabytes [7];

- data buffering and routing is more expensive than for macro-networks both in terms of area and power (in fact, the area of a NOC router is dominated by buffer space [8]), a fact that should be carefully considered while deciding to build a packet-switched network instead of relying on more traditional communication schemes based on point-to-point channels.

Fortunately, some of these characteristics match well the fact that SOC design is mainly an average-case, throughput-driven design, and not a worst-case latency-driven design [23].

In the present work we rely on careful physical models of the network components (wire, repeaters, switches) recently proposed in the literature (Section 3) and we use them to derive an abstraction of the NOC cost metrics that we use to define formally the on-chip communication synthesis problem (Section 4). This is a combinatorial optimization problem where the inputs are a set of point-to-point communication requirements between SOC components. Each requirements is expressed in terms of distance (based on the relative position of the components ports) and data bandwidth. The optimality criterion for the problem is defined as a cost function that combines the energy spent in communicating and buffering/routing data. The solution of this computationally-hard problem is obtained through an heuristic that combines two efficient computational steps: (1) an application of the k-Median algorithm to coarsely determine the global communication structure (which may turned out not be a network after all), and a (2) a variation of the shortest-path algorithm in order to finely tune the data flows on the communication channels. The final result is an optimal communication scheme for the given requirements that may or may not end up being a NOC. Notice that we focus on the critical step of synthesizing the topology of the communication network, because we think that the choice of the packet-routing protocol is somewhat secondary (in any case for NOC most researchers argue for wormhole or flit-reservation flow-control techniques).

Many researchers have been arguing that NOC are likely to replace bus-based communication, which does not scale well, because they have higher bandwidth and support multiple concurrent communications. A question that has remained largely unaddressed is whether NOC will replace traditional point-to-point communication lines. In Section 6 we show that for application-specific designs this seems unlikely, while NOC may represent an interesting solution for multi-core general-purpose processors. However, the choice of the NOC topology in this case is strongly dependent on the characteristics of the chosen technology process and a delicate balance of power/performance tradeoffs. These results may help explain the relatively small number of commercial chips that have embraced the NOC paradigm to date.

2. RELATED WORK

Hemani *et al.* were among the first to propose the idea of a packet-switched NOC as part of a design methodology aimed to cope with the complexity of the billion-transistor SOCs of the future [11]. At DAC'01 Dally and Towles argued the various benefits of designing an on-chip packet-switched network as a shared resource for global communication instead of using conventional circuits connected via the dedicated wires that are automatically routed by CAD tools [8]. In particular, they advocated that such

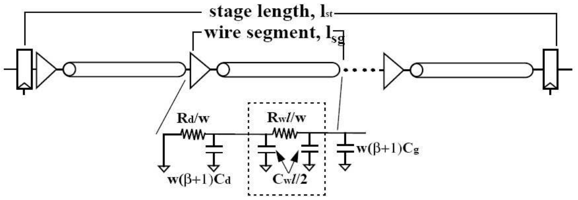

Figure 1: First-order RC model of a wire [13].

micro-networks should be built with the top-level metal layers and carefully tuning their electrical parameters. In turn this would enable the use of high-performance circuits (e.g., based on pulsed on-chip current-mode signalling [18]) that result in significantly lower power dissipation, higher propagation velocity, and higher bandwidth than what traditional interconnect can achieve. In a IEEE Computer paper, Benini and De Micheli postulated that "the layered design of reconfigurable micronetworks, which exploits the methods and tools used for general networks, can best achieve efficient communication on SOCs" [3]. They clarified that, differently from macroscopic networks that emphasize general-purpose communication and modularity, SOC developers can design the network fabric on silicon from scratch, thereby tailoring the NOC to to the target class of applications (*design-time specialization*) [3].

Over the last four years several research groups have proposed various NOC design methodologies [9, 12, 17, 32]. In particular, NetChip [4] is a complete design flow that starts from a high-level application specification, derives an optimized NOC configuration with respect to different design objectives, and instantiates the selected *application-specific* on-chip micro-network. The synthesis is partitioned into major steps (topology mapping, selection, and generation) and relies on a library of pre-designed topologies (mesh, torus, hypercube, butterfly,...). The design flow combines CAD tools (SUNMAP [24], ×pipesCompiler [16]) leveraging the flexibility of ×pipes [6], a library of reusable and scalable components (interfaces, switches, links) that are design-time tunable.

Instead of seeking an application-specific NOC, other researchers have focused on mapping the application on a programmable multi-core architecture. Hu and Marculescu [15] assumed to have given (1) a 2-D mesh NOC and (2) an application characterization graph (APCG) representing the interaction of the functional blocks for the desired application and focused on the solution of two problems (mapping and routing) to implement the APCG on the NOC with optimal energy and performance. Lahiri *et al.* also addressed the problem of mapping a system's communication requirements to a regular communication architecture, but they specialized on network topologies consisting of collections of channels (point-to-point links or shared busses) interconnected by bridges [20].

In [25] Pinto *et al.* proposed an algorithm for the automatic synthesis of a communication architecture among a set of computational blocks once their relative positions and required pairwise communication bandwidth is provided. However, their results are limited to the synthesis of bus-based topologies. Further, since the original algorithm is computationally very expensive, they proposed an heuristic that splits the optimization problem into two steps: clustering of constraints and cluster implementation [26]. Srinivasan *et al.*, proposed an LP-based approach for the synthesis of application-specific NOCs that does not assume an existing interconnection scheme [28]. However, this approach considers only the router architecture and, as many others before, neglects the advantages that traditional point-to-point connections implemented by VLSI wires offer in terms of bandwidth and power.

As far as we know the present paper is the first one to present a method for the hierarchical synthesis of NOC topologies that (1) does not rely on mapping onto pre-designed regular communication fabrics (but it synthesizes them!) and (2) accounts for the peculiar technology characteristics of on-chip communication.

3. MODELING ON-CHIP COMMUNICATION

We present here the physical model for wires and routers, i.e. the "building blocks" of our on-chip communication structures.

Wires. It is well-known that, while the on-chip propagation delay increases proportionally to the square of the interconnection length because both capacitance and resistance increase linearly with length, designers can insert an optimal number of optimally sized repeaters in order to divide the interconnect wire into smaller subsections, thereby making the delay linear with its length [2]. Figure 1 shows the classic first-order RC model of a repeated wire [2, 14, 13] where R_d is the driving repeater resistance, w is the width of the repeater NMOS transistor normalized by the minimum technology width, β is the PMOS-to-NMOS sizing ratio, C_d and C_g are diffusion and gate capacitance per unit width, R_w and C_w are wire resistance and capacitance per unit length, and l_{sg} is the length of the repeated segment. This first order model leads to a delay of the wire segment l_{sg} equal to

$$d = 0.7\Big[\frac{R_d}{w}\cdot\big[w(\beta{+}1)(C_d{+}C_g){+}l_{sg}\cdot C_w\big]{+}l_{sg}^2\frac{R_w+C_w}{2}{+}l_{sg}\cdot R_w\cdot w(\beta{+}1)C_g\Big]$$

For a given technology process and a chosen metal layer, there exists a minimum length l_{sg} (*critical repeater length*) beyond which inserting an optimal-sized repeater makes the interconnect delay smaller than that of the corresponding un-repeated wire [27]. The delay d of a critical repeater length is sometime referred as *critical delay*. Figure 1 shows also the minimum-sized flip-flops that are used to drive and sample the signal on the wire [1]. These flip-flops are separated by a stage length l_{st}. Therefore, one can define the *critical sequential length* as the maximum distance that a signal can travel in an interconnect that has been optimally sized and optimally buffered uniformly within a single clock period T, whose duration is determined by the particular technology process [27]. In fact, the technology process sets a lower bound on T, while designers can trade-off the duration of T with the length of l_{st}, based on the following relation:

$$l_{st} = \frac{T}{d}\cdot l_{sg} \tag{1}$$

Saxena *et al.* have shown how, under the assumption of classic 0.7× scaling theory, both the critical repeater and sequential lengths decrease with process scaling (at approximated rates of 0.57× and 0.43× per generation respectively) and that the percentage of wires requiring clocked repeaters continues to grow. As predicted by several other researchers, *wire pipelining* will become pervasive in nanometer technologies making on-chip global communication require an increasing number of clock cycles.

For instances, Heo and Asanovic calculated that using a clock period of $500ps$ with a $70nm$ technology the critical sequential length l_{st} varies between $2.10mm$ and $8.88mm$ depending on the wire metal layer [13]. Similarly, Kumar *et al.* considered a $65nm$ process with 4 metal layers in the 1X plane and 2 layers in each of the 2X, 4X, and 8X planes, and, for transmission clock frequency of $2.5Ghz$ at $1.1V$, reported the following values of l_{st}: $1.5mm$ for 1X, $3.0mm$ for 2X, $5.0mm$ for 4X, and $8.0mm$ for 8X [19]. Notice that these clock frequencies are not very aggressive for such processes leaving room for designers to increase them as long as they are willing to tolerate smaller value of l_{st} and, consequently, higher point-to-point communication latencies in terms of clock cycles. In fact, this is trade-off is likely as designers are typically forced to trade-off latency for bandwidth.

The power dissipated by a line of length L composed of n stages l_{st}, running at frequency $\frac{1}{T}$ with an activity factor ρ is [13]:

$$P = \frac{1}{2}\cdot\frac{\rho}{T}\cdot V_{dd}^2\Big[(1+\frac{3}{2}q)C_w L+n\cdot C_{ff}\Big]+V_{dd}^{1+\gamma}\Big[k_{rep}(\frac{3}{2}q)C_w L+n\cdot k_{ff}C_{ff}\Big]$$

[1] In this paper, we assume traditional on-chip signalling while we leave the extension of our approach to more advance circuit techniques [18] for future work.

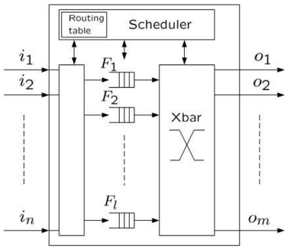

Figure 2: Block diagram of an input-queued router.

where q is the ratio of the repeater gate cap and wire cap within a wire segment, C_{ff} is the flip-flop capacitance and k_{rep} and k_{ff} are leakage power coefficients for repeaters and flip-flops respectively. The first term of the equation is the switching power component and the latter is the leakage power component. Leakage power scales super-linearly in nanometer technologies ($1 < 1 + \gamma < 2$).

Routers. The task of a router (Figure 2) in a packet-switched network is to receive data streams, or *commodities*, form input ports $i_1, ..., i_n$, and switch them to the output ports $o_1, ..., o_m$. A packet arriving at an input port is temporarily stored in a local memory. The local storage is represented by the set of queues F_i in Figure 2. The head of queue packets are sent to the output ports depending on the content of a routing table. If more that one packet must be sent to the same output port, a scheduler decides which one goes first depending on their priority.

The area of a NOC router is dominated by the buffer space [8] but the total area overhead of the NOC with respect to the whole SOC is estimated to be minor (about 6.6% in [8]). Assuming that buffering queues, cross-bar switch, and scheduler run at the same clock frequency, the power dissipated by a router is [15, 31]:

$$P_{router} = P_{buffer} + P_{switch} + P_{scheduler}$$

To model the router power dissipation we rely on the results of the Orion project [10], which demonstrated that this is linear in the packet injection rate, i.e. in the channel communication bandwidth [31]. The reason is that the contribution of the input queues and crossbar switches are dominant.

Network Cost Metrics. Our virtual communication library is composed of two elements: communication links and routers/repeaters. The modeling of their cost and performance is the result of a careful abstraction of their implementations on silicon and, particularly, accounts for parameters that are becoming increasingly important with nanometer technologies like the critical sequential length and leakage power.

Let $G(V, E, \omega, \pi)$ be a graph denoting a network. The set of vertices is V is partitioned in the set of sources S, destinations D and routers R. For each pair $(s, d) \in S \times D$, let $b(s, d)$ be the amount of (s, d)-commodity that must be routed in the network G. The function $\omega : E \times S \times D \rightarrow \mathbb{R}_+$ associates to each edge $e \in E$ and each (s, d)-commodity, a positive value representing the amount of that commodity that flows through e. The function $\pi : V \rightarrow \mathbb{R}^2$ associates to each node a position in the Euclidean plane. Also, let $w : E \rightarrow \mathbb{R}_+$ be the aggregate data flow function defined by $\forall e \in E,\ w(e) = \sum_{(s,d)\in S\times D} \omega(e, (s, d))$.

An edge $e = (u, v)$ in G is the abstraction of an on-chip communication link (which ultimately corresponds to a bundle of wires). The cost of such a link depends on the distance that it spans and on the number of data bits flowing through it in the unit time. In particular, we consider two contributions to the total edge cost: a dynamic cost and a static cost. For the dynamic cost, we consider as a metric the power delay product, that is proportional to the aggregate flow times the length squared. For the static cost, we consider as a metric the leakage power that is proportional to the edge length. The total cost of an edge $e = (u, v)$ is $C_e(u, v) = C_{e,d}(u, v) + C_{e,s}(u, v)$ where $C_{e,d} \propto w(e)||\pi(u) - \pi(v)||_2^2$ and $C_{e,s} \propto ||\pi(u) - \pi(v)||_2^2$. Notice that the static cost should be considered only if an edge is installed.

A node $r \in R$ in G is the abstraction of an on-chip router/repeater. Its cost is dominated by the storage cost of packets into queues that is proportional to the total input flow: $C_v(r) \propto \sum_{(u,r)\in E} w(u, r)$.

The total cost of a given network, is the sum of the cost of edges plus routers:

$$C(G) = \sum_{(u,v)\in E} (aw(e) + b)\, ||\pi(u) - \pi(v)||_2^2 + c \sum_{r\in R} \sum_{(u,r)\in E} w(u, r) \qquad (2)$$

where a is the cost of carrying a unit of data through a wire of unit length and c is the cost of storing a unit of data in a router/repeater. Normalizing both terms in this equation by a we obtain the objective function for the optimization problem formulated in the next section:

$$C(G) = \sum_{(u,v)\in E} (w(e) + \alpha)\, ||\pi(u) - \pi(v)||_2^2 + \lambda \sum_{r\in R} \sum_{(u,r)\in E} w(u, r) \qquad (3)$$

where $\alpha = b/a$ and $\lambda = c/a$. These two parameters have a straightforward interpretation and play an important role in trading-off different network implementation: λ is the relative cost of storing one bit of data with respect to moving it from one point to another of the chip while α is the relative cost of leakage versus dynamic power for a unit wire.

4. ON-CHIP COMMUNICATION SYNTHESIS

Networking is about sharing media. Consider an ideal case where installing wires does not cost anything and, moreover, the cost of a wire is linear in the throughput. For this special case, a shared wire has the same cost as two wires carrying half its throughput. Consequently, in the case of linear cost, the optimal solution is always a collection of point-to-point channels. Sharing wires is advantageous only if their cost is a *concave* function of the throughput which also subsume the case of installation cost (fixed-charge networks).

The on-chip communication synthesis problem can be formulated as a mixed-integer linear programming (MILP) problem. We introduce the auxiliary binary variables z_e such that $z_e = 1$ if $w(e) > 0$ and $z_e = 0$ if $w(e) = 0$; also, for an edge $e = (u, v)$ let $\delta(e) = ||\pi(u) - \pi(v)||_2^2$:

$$\begin{aligned}
&\min_{\omega,\pi} \sum_{e=(u,v)\in E} w(e)\delta(e) + \alpha \sum_{e=(u,v)\in E'} z_e\delta(e) + \lambda \sum_{r\in R} \sum_{(u,r)\in E} w(u, r) \\
&s.t. \\
&\forall s \in S, (s, d) \in S \times D \quad \sum_{(s,j)\in E} \omega((s, j), (s, d)) = b(s, d) && (4) \\
&\forall d \in D, (s, d) \in S \times D \quad \sum_{(i,d)\in E} \omega((i, d), (s, d)) = b(s, d) \\
&\forall r \in D, (s, d) \in S \times D \quad \sum_{(i,r)\in E} \omega((i, r), (s, d)) = \sum_{(r,j)\in E} \omega((r, j), (s, d)) \\
&\forall e \in E \quad w(e) \le U z_e && (5) \\
&\forall e \in E,\ w(e) \ge 0 && (6) \\
&\forall e \in e \quad z_e \in \{0, 1\} && (7)
\end{aligned}$$

Where $U = \sum_{(s,d)\in S\times D} b(s, d)$. The uncapacitated fixed-charge network problem, that is NP-Hard, can be reduced to our problem that is, therefore, at least as hard. In Section 5 we explain why using an MILP solver is not a viable solution. The cost function is the one discussed above. Equation (4) captures the flow conservation constraints while Equation (5) forces the flows to be different from zero only if an edge is installed.

5. TWO-STEP HEURISTIC ALGORITHM

Consider a chip whose vertical and horizontal dimensions are h and w respectively. First, we define a graph G on which we perform an optimization in terms of node positions π and data flow routing ω. Nodes in such graph correspond to either switching points or clocked repeaters that are used to pipeline a signal travelling a distance larger than l_{st}. We build the graph G by sampling the two chip dimensions h and w with a spacing σ. The total number of routers or clocked repeaters is $|R| = \lceil h/\sigma \rceil \lceil w/\sigma \rceil$.

Using an MILP solver to implement the optimal on-chip network is not a viable solution. The reason is that the number of variables of the optimization problem is $|E|(1+|S \times D|)$ where $|E|$ is of the order of $(|S| + |D| + |R|)^2$. For $h = w = 20mm$, $\sigma = l_{st} = 2.5mm$ and $|S| = |D| = 16$ the number of variables exceeds 2 millions, a complexity that no solver can handle.... Hence, we solve the problem by dividing the synthesis flow in two steps: optimal assignment of source and destination nodes to extra nodes (e.g. routers) and optimal routing. Both sub-problems are NP-Hard: the optimal assignment is the *k-Median problem* [30], while the optimal routing in presence of installation cost is the *fixed-charge uncapacitated multicommodity flow network optimization problem.* Algorithm 1 shows the pseudocode of our synthesis algorithm.

Algorithm 1

```
Require: $p \leftarrow (h, w, \sigma, l_{st}, \lambda, \alpha)$
  network_synthesis($S, D, \pi, b, k, p$)
  $R \leftarrow$ grid($p.h, p.w, p.\sigma$)
  $C \leftarrow S \cup D$ ; $F \leftarrow R$
  $(\Phi, I) \leftarrow$ k_median( $k, F, C, \pi$)
  $G' \leftarrow$ routing_graph( $F, C, \Phi, I$)
  $routes \leftarrow$ opt_routing($G', S, D, b, \pi, p.l_{st}, \lambda, \alpha$)
```

Let $p = (h, w, \sigma, l_{st}, \lambda, \alpha)$ be the tuple containing the parameters defined above. The first step is to determine the set of routers R by superimposing a grid on the chip area whose granularity is determined by the spacing parameter σ. Then, an instance of the k-median problem is solved where the cities are the set of sources and destinations and the facilities are the set of routers. Notice that the function π is used by the algorithm to determine connection costs. The k-median problem can be stated as follows. Let G be a bipartite graph with bipartition (F, C) of its vertices, where F is a set of facilities and C a set of cities and let k be the maximum number of facilities that are allowed to be opened. Let c_{ij} be the cost of connecting city j to (opened) facility i. The problem is to find a subset of facilities $I \subseteq F$, $|I| \leq k$ and an assignment $\Phi : C \rightarrow I$ of cities to facilities in I such that the total connection cost is minimized. We use the approximation algorithm given in [30]. The running time of such algorithm is $O(m \log m(L + log(n)))$ where $n = |F| + |C|$, $m = |F||C|$ and $L = \log(\max(c_{i,j})/\min(c_{i,j}))$ (where in our case $m = |S \times D|hw/\sigma^2$ approximately).

Our second step is the optimal routing between sources and destinations. Algorithm 2 shows the pseudocode that performs such step. Notice that each source destination constraint must be routed on a simple path because a cycle would obviously increase the network cost and flow splitting would also increase the cost since the edge cost function is concave in the bandwidth. In other words, given a path that carries a single commodity $b(s, d)$, the ratio between dynamic and static power is $b(s, d)/\alpha$, which means that the installation cost is worthier for higher throughput constraints. For this reason we sort all (s, d)-commodities in decreasing order. For each commodity in this order, we route it using a modified version of shortest path where the edge relaxation is done in the following way. Given an edge $e = (u, v)$ and the current commodity $b'(s, d)$, if $w(e) > 0$ then $d(u, v) = d(u) + b'(s, d)(\delta(u, v) + \lambda)$ otherwise $d(u, v) = d(u) + b'(s, d)(\delta(u, v) + \lambda) + \alpha\delta(u, v)$. If $d(v) > d(u) + d(u, v)$ then set $d(v) = d(u) + d(u, v)$ and let u be the predecessor of v. Edges are selected only if their length is smaller that l_{st}. The complexity of **opt_routing** is $O\big(|S \times D|(|S| + |D| + |R|)^2\big)$.

Algorithm 2

```
  opt_routing($G', S, D, b, \pi, p.l_{st}, \lambda, \alpha$)
  $b' \leftarrow$ sort($b$)
  $w(e) \leftarrow 0, \ \forall e \in E'$
  for all $(s, d)$ commodity in the order $b'$ do
    $routes(s, d) \leftarrow$ shortest_path($G', s, d, b'(s, d), \pi, p.l_{st}, \lambda, \alpha, w$)
    $w \leftarrow$ flow_update($routes(s, d), b'$)
  end for
```

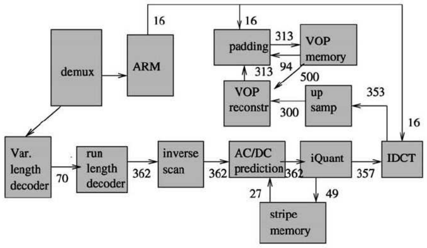

Figure 3: Block diagram of the VOPD [4].

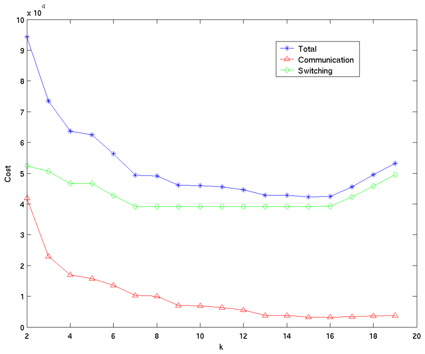

Figure 4: Network implementation costs as function of k parameter for VOPD ($l_{st} = 2.5mm$).

6. EXPERIMENTAL RESULTS

We present the application of our synthesis flow to two paradigmatic case studies of SOC design: (1) a chip-multiprocessor (CMP) with 16 cores similar to the ones discussed in [13, 15, 19] and (2) the *Video Object Plane Decoder (VOPD)*, an application-specific SOC for which various possible NOCs are analyzed in [4]. For both cases, we assume to implement the design using a 65nm technology process, similar to those discussed in [13, 19, 27].

Determining the critical sequential length. According to Equation 1, the choice of the transmission clock frequency determines the value of the critical sequential length l_{st}. This choice is an input parameter in our CAD tool because, as discussed in in Section 3, designers may want to trade-off latency for bandwidth. In fact, as done for low-power circuit design [5], additional pipelining can be applied to a wire beyond what is necessary to span a certain distance L in order to recover extra times slack that can be used to scaled down (statically) the supply voltage. This leads to obtain a quadratic reduction in active power and also a super-linear reduction in leakage power (as leakage current has a strong dependency on drain voltage in nanometer processes) [13]. For the examples discussed next we have chosen two values of l_{st} equal respectively to 2.5mm and 5mm. Notice that in building the on-chip communication structure we limit ourselves to consider only intermediate metal layers sharing the same value of l_{st} [2].

Results for the VOPD. Figure 3 shows the block diagram of

[2] This assumption is not very restrictive considering that typically (1) intermediate layers have the same, or very similar, physical structure and (2) the lower metal layers are used to interconnect the logic on the functional blocks while the top metal layers are dedicate to distribute clock signals, supply voltage and ground planes. Still, in future work we plan to extend our approach to multiple heterogeneous metal layers that are likely become common in more aggressive technology processes.

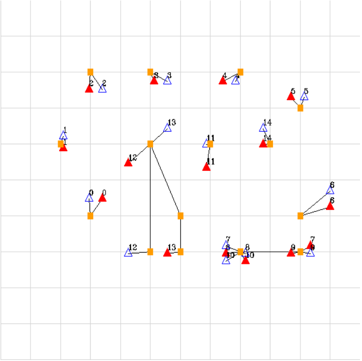

Figure 5: NOC for VOPD ($k = 14$, $l_{st} = 2.5mm$).

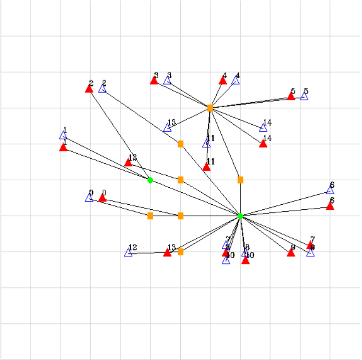

Figure 6: NOC for VOPD ($k = 2$, $l_{st} = 2.5mm$).

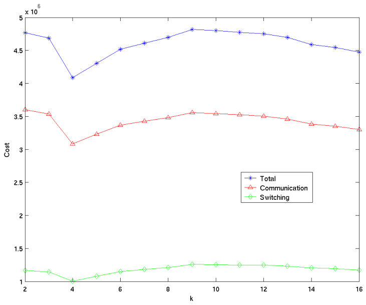

Figure 7: Network implementation costs as function of k parameter for 16-core processor ($l_{st} = 2.5mm$).

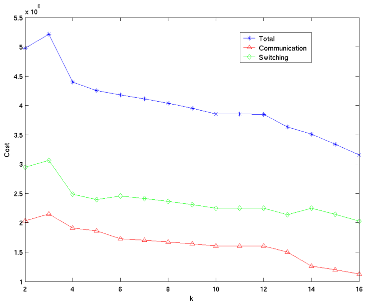

Figure 8: Network implementation costs as function of k parameter for 16-core processor ($l_{st} = 5mm$).

the VOPD from [4] with the channel communication bandwidth annotated in MB/s. We used PARQUET [1] to derive a floorplanning for the VOPD design with our $65nm$ technology, and we obtained a chip whose dimensions are $7.5 \times 5mm$. Figure 4 reports the total network costs values of Equation 3 and the two values of its two terms (communication and switching cost) as we sweep k in the range $[2, 19]$ with $l_{st} = 2.5mm$. The optimal solution is obtained for $k = 14$ and the corresponding final communication scheme is reported in Figure 6. A suboptimal solution is shown in Figure 6 ($k = 4$). In these diagrams, the triangles correspond to source or destination ports, the square to clocked repeaters (flip-flops) and the circle to routers. Notice that the optimal solution corresponds to a collection of point-to-point channels, while the suboptimal solution contains a couple of packet-switching routers. In fact, for both $l_{st} = 2.5mm$ and $l_{st} = 5mm$ we have that the optimal schemes rely on point-to-point communication, suggesting that with such technology processes a NOC is not a value proposition for application-specific SOCs like the VOPD that present a fairly linear block-level diagram where most blocks communicate only with their neighbors (Fig. 3).

Results for the On-Chip Multiprocessor. For the chip-multiprocessor (CMP) with 16 homogeneous cores we assumed: a placement based on a regular 4×4 grid topology, a chip size of $20 \times 20mm$, and that each processor communicates with every other processor at the compound rate of $2GB/s$. Figures 7 and 8 illustrate the results in terms of total network costs values as we sweep k in the range $[2, 16]$ with $l_{st} = 5mm$ and $l_{st} = 2.5mm$ respectively. It is clear that to halve l_{st} (i.e. we double the amount of wire pipelining) decreases the cost contribution due to line communication (on average by 50%) while it increases the one due to switching (on average by 50%). However, the saving in power consumption gained by the wires dominates the more expensive switching activity leading to a net saving in the total cost.

These facts translate into the optimal solution of Fig 9 (a pipelined ring) for $k = 4$ when $l_{st} = 5mm$ and the one of Fig. 12 (a regular mesh) for $k = 16$ when $l_{st} = 2.5mm$. For comparison consider the alternative suboptimal solution of Fig. 10 for $k = 16$ when $l_{st} = 5mm$ and Fig. 11 for $k = 4$ when $l_{st} = 2.5mm$.

We emphasize that all these network topologies are generated automatically by our tool and that, differently from other approaches, we don't complete a mapping onto a library of pre-designed topologies (meshes, tori, butterflies,...).

7. CONCLUDING REMARKS

We presented a tool for the automatic synthesis of on-chip micro-network topologies. Given the complexity of the problem that is in general NP-Hard, we separate it in optimal network access allocation (e.g. k-median) and optimal routing. We selected efficient algorithms to solve both steps.

Our optimization takes into account parameters that are directly connected to the physical implementation of on-chip networks. In particular we consider both dynamic and static power consumption of wires, trade-off between communication and switching power, as well as the role played by the critical sequential length.

We evaluated our algorithms on two examples: the VOPD decoder and a 16 cores On-Chip Multiprocessor. In the first case, we identified as best solution a point-to-point implementation. In the second case the optimal topology depends on the value of the critical sequential length and varies from a ring to a mesh.

8. REFERENCES

[1] S. N. Adya and I. L. Markov. Fixed-outline floorplanning: Enabling hierarchical design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(6):1120–1135, Dec. 2003.

[2] H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, Reading,MA, 1990.

[3] L. Benini and G. D. Micheli. Networks on chip: A new SoC paradigm. *IEEE Computer*, 2002.

[4] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. D. Micheli. NoC synthesis flow for customized domain specific multiprocessor systems-on-chip. *IEEE Transactions on Parallel and Distributed Systems*, 16(2):113–129, Feb. 2005.

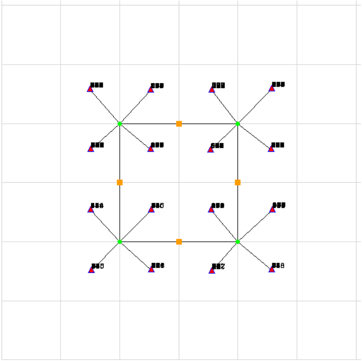

Figure 9: NOC for 16-core processor ($k = 4$, $l_{st} = 5mm$).

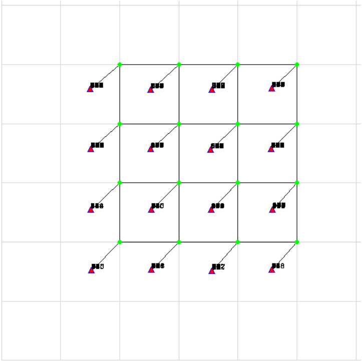

Figure 10: NOC for 16-core processor ($k = 16$, $l_{st} = 5mm$).

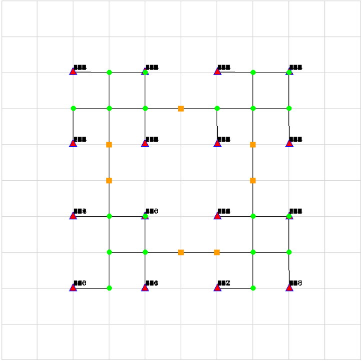

Figure 11: NOC for 16-core processor ($k = 4$, $l_{st} = 2.5mm$).

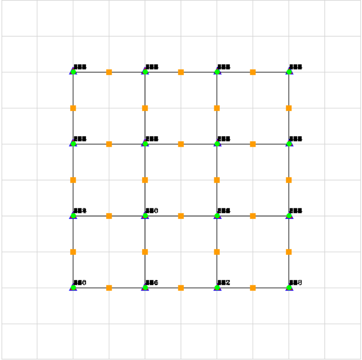

Figure 12: NOC for 16-core processor ($k = 16$, $l_{st} = 2.5mm$).

[5] A. Chandrakasan, S. Sheng, and R. Brodersen. Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4):473–484, Apr. 1992.

[6] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini. ×*pipes*: a latency insensitive parameterized network-on-chip architecture for multi-processor SoCs. In *Proc. Intl. Conf. on Computer Design*, pages 536–541, Oct. 2003.

[7] W. J. Dally and S. Lacy. VLSI architecture: Past, present and future. In *Proceedings of the Advanced Research in VLSI Conference*, pages 232–241. IEEE, Mar. 1999.

[8] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proc. of the Design Automation Conf.*, June 2001.

[9] K. Goossens, P. Wielage, A. Peeters, and J. V. Meerbergen. Networks on silicon: Combining best-effort and guaranteed services. In *Proc. of the Conf. on Design, Automation and Test in Europe*, 2002.

[10] L. S. P. H. S. Wang, X. Zhu and S. Malik. Orion: A power-performance simulator for interconnection networks. In *Proceedings of the 35th International Symposium on Microarchitecture (MICRO)*, Nov. 2002.

[11] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Vberg, M. Millberg, and D. Lindqvist. Network on chip: An architecture for billion transistor era. In *Proc. of the IEEE NorChip Conference*, Nov. 2000.

[12] J. Henkel, W. Wolf, and S. Chakradhary. On-chip networks: A scalable, communication-centric embedded system design paradigm. In *Proceedings of the 17th International Conference on VLSI Design (VLSID04)*, 2004.

[13] S. Heo and K. Asanovic. Replacing global wires with an on-chip network: a power analysis. In *Proc. of the Intl. Symposium on Low Power Electronics and Design*, pages 369–374, 2005.

[14] R. Ho, K. W. Mai, and M. A. Horowitz. The future of wires. *Proceedings of the IEEE*, pages 490–504, April 2001.

[15] J. Hu and R. Marculescu. Energy- and performance-aware mapping for regular NoC architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(4):551–562, Nov. 2005.

[16] A. Jalabert, L. Benini, S. Murali, and G. D. Micheli. ×*pipesCompiler*: a tool for instantiating application-specific NoCs. In *Proc. of the Conf. on Design, Automation and Test in Europe*, Feb. 2004.

[17] A. Jantsch and H. Tenhunen, editors. *Networks on chip*. Kluwer Academic Publishers, Hingham, MA, USA, 2003.

[18] A. P. Jose, G. Patounakis, and K. L. Shepard. Near speed-of-light on-chip interconnects using pulsed current-mode signalling. In *Symposium on VLSI Circuits*, June 2005.

[19] R. Kumar, V. Zyuban, and D. M. Tullsen. Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling. In *Proc. Annual International Symposium on Computer Architecture*, pages 408–419, June 2005.

[20] K. Lahiri, A. Raghunathan, and S. Dey. Design space exploration for optimizing on-chip communication architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(6):952–961, Dec. 2004.

[21] K. Lee, S. Lee, S. Kim, H. Choi, D. Kim, S. Kim, M. Lee, and H. Yoo. A 51mW 1.6GHz on-chip network for low-power heterogeneous soc platform. In *ISSCC Digest of Technical Papers*, Feb. 2004.

[22] S. Lee, S. Song, K. Lee, J. Woo, S. Kim, B. Nam, and H. Yoo. An 800Mhz star-connected on-chip network for application to systems on a chip. In *ISSCC Digest of Technical Papers*, Feb. 2003.

[23] T. Lin and L. T. Pileggi. Throughput-driven IC communication fabric synthesis. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 274–279, 2002.

[24] S. Murali and G. D. Micheli. SUNMAP: A tool for automatic topology selection and generation for NOCs. In *Proc. of the Design Automation Conf.*, pages 914–919, June 2004.

[25] A. Pinto, L. P. Carloni, and A. L. Sangiovanni-Vincentelli. Constraint-driven communication synthesis. In *Proc. of the Design Automation Conf.*, pages 783–788, New Orleans, LO, June 2002. IEEE.

[26] A. Pinto, L. P. Carloni, and A. L. Sangiovanni-Vincentelli. Efficient synthesis of networks on chip. In *Proc. Intl. Conf. on Computer Design*, pages 146–151, San Jose, CA, Oct. 2003. IEEE.

[27] P. Saxena, N. Menezes, P. Cocchini, and D. Kirkpatrick. Repeater scaling and its impact on CAD. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):451–462, Apr. 2004.

[28] K. Srinivasan, K. S. Chatha, and G. Konjevod. Linear programming based techniques for synthesis of network-on-chip architectures. In *Proc. Intl. Conf. on Computer Design*, pages 422–429, Oct. 2004.

[29] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffmann, P. Johnson, W. Lee, A. Saraf, N. Shnidman, V. Strumpen, S. Amarasinghe, and A. Agarwal. A 16-issue multiple-program-counter microprocessor with point-to-point scalar operand network. In *ISSCC Digest of Technical Papers*, Feb. 2003.

[30] V. Vazirani. *Approximation Algorithms*. Springer, Berlin, 2003.

[31] H. Wang, L. Peh, and S. Malik. A power model for routers: Modeling alpha 21364 and infiniband routers. *IEEE Micro*, 23(1):26–35, 2003.

[32] P. Wielage and K. Goossens. Networks on silicon: Blessing or nightmare? In *Euromicro Symposium On Digital System Design*, Dortmund, Germany, Sept. 2002. Keynote speech.