Physical-aware link allocation and route assignment for chip multiprocessing

Nikita Nikitin	Satrajit Chatterjee	Jordi Cortadella	Mike Kishinevsky	Umit Ogras
Univ. Politècnica	Strategic CAD Lab,	Univ. Politècnica	Strategic CAD Lab,	Strategic CAD Lab,
de Catalunya	Intel Corp.	de Catalunya	Intel Corp.	Intel Corp.
Barcelona, Spain	Hillsboro, OR, USA	Barcelona, Spain	Hillsboro, OR, USA	Hillsboro, OR, USA

Abstract—The architecture definition, design, and validation of the interconnect networks is a key step in the design of modern on-chip systems. This paper proposes a mathematical formulation of the problem of simultaneously defining the topology of the network and the message routes for the traffic among the processing elements of the system. The solution of the problem meets the physical and performance constraints defined by the designer. The method guarantees that the generated solution is deadlock free. It is also capable of automatically discovering topologies that have been previously used in industrial systems. The applicability of the method has been validated by solving realistic size interconnect networks modeling the typical multiprocessor systems.

I. INTRODUCTION

The constantly increasing complexity of chip multiprocessing (CMP) systems requires scalable and efficient communication topologies. Network-on-Chip (NoC) [1], [2] has become the dominant interconnection paradigm for the design of CMPs. The modern CMPs and NoCs require a thorough elaboration process that involves a variety of design problems: topology selection and mapping, physical planning, routing and switching schemes, and other optimization tasks. The large number of options and constraints makes it impossible to fully explore the solution space. On the other hand, dividing the design problem into smaller subproblems and doing a myopic optimization for each one of them may result in largely suboptimal solutions.

Let us consider a design example. A CMP system is specified by a set of processing elements (*PE*), routers and communication requirements between *PE*s. Let us assume that the underlying system topology has been selected and the assignment of the *PE*s to the routers has been performed. Some of the design problems we would like to solve are as follows:

- Find a subset of links satisfying the communication requirements of the system and minimizing the design cost.
- Define the routing paths for each pair of communicating *PE*s that satisfy the performance requirements.
- Guarantee that the selected routes to communicate the *PE*s are deadlock-free.

The first problem can be referred to as the *link allocation* problem. The other two problems are related to the efficient *route assignment* with *deadlock avoidance*. These problems have different optimization criteria. By solving one of them optimally and independently from the others, no acceptable solution might be found when solving the subsequent problems. It is therefore necessary to devise non-myopic strategies to explore the solution space in a way that all design constraints are met and the implementation cost is minimized.

This paper presents a mathematical formulation that combines the three previous problems in one model. Various constraints and cost functions are combined to do *link allocation* and *route assignment* simultaneously while optimizing the cost and performance of the system. The model also guarantees that the derived solution is *deadlock-free*. The model can be defined as a conjunction of linear inequalities and a linear (or quadratic) cost function with Boolean and integer variables, thus enabling the use of integer programming (IP) solvers.

The paper is organized as follows. Next section summarizes the related work. Section III presents an overview of the paper and illustrates the basic contributions with an example. The problem description, formulation and solution are presented in Sect. IV. The results obtained from various experiments are discussed in Sect. V. Finally, Sect. VI concludes the paper.

II. RELATED WORK

The application of linear programming techniques to the design of on-chip networks has already been proposed in [3]. The authors presented mixed integer linear programming formulations for the problems of floorplanning, topology and route generation for NoCs. However, the model introduced in this paper differs by ensuring deadlock freedom of the routing solution. It also explicitly states the link allocation problem for mesh networks and discusses a number of specific cost functions and constraints.

Many approaches have been suggested for the on-chip routing problem. Several schemes have been proposed to guarantee deadlock and livelock-free properties of the communication algorithm, such as odd-even routing [4] or turn prohibition [5]. Numerous works on the on-chip routing enhancements include combination of the deterministic and adaptive schemes for performance improvement [6], incorporation of the application-specific topology, traffic and bandwidth information for congestion avoidance and performance increase [7], [8].

Though the link allocation problem for CMP has not been emphasized so far, irregular meshes [9] have been recently
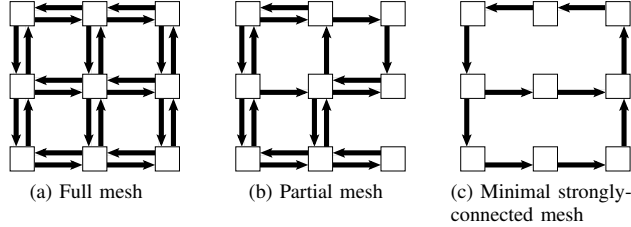
Fig. 1: Different link allocation solutions for a 3x3 mesh.

Fig. 2: (a) Communication graph; (b) and (c) two different link allocation and route assignment solutions.

found to provide a simple and flexible extension of the regular mesh topologies in order to support *PE*s of different sizes. The interest for irregular mesh structures is sustained by the works that have appeared lately using this type of topology. An algorithm for the deadlock-free communication in irregular meshes, extending the deterministic XY-routing with hard-coded routing tables, was presented in [10]. A variety of adaptive strategies is also available in the literature. The work in [11] proposes an adaptive routing for irregular mesh-based NoC topologies, supported by a floorplanning method to generate the layouts suitable for this algorithm. Application-specific information is used to perform deadlock-free adaptive routing for irregular meshes with regions in [12]. Another algorithm for traffic balancing is presented in [13].

This paper proposes a novel approach to simultaneously define the network topology and the message routes for communicating processing elements. Various optimization targets and constraints are incorporated into the design space exploration algorithm. The deadlock freedom of the routing configuration is guaranteed by the turn prohibition technique [5]. The quality of the solutions in terms of area and performance is demonstrated with the comparison with the XY and odd-even [4] routing implemented in a full-mesh topology.

III. OVERVIEW

This section gives an overview of the contributions of the paper by using a simple example. In this work, we consider communication topologies that can be mapped onto a two-dimensional grid. Every link can connect two adjacent routers and the transit time through each link is constant and known a priori. We also assume that the system has already been floorplanned and each *PE* is connected to a router.

Figure 1 shows three different topologies based on an underlying 3x3 mesh. Figure 1a depicts a fully-connected mesh in which all possible links have been laid out. This topology can provide a very high performance. Every router can reach any other router in no more than four hops. However, this topology requires a costly implementation in wiring and router area. Every router implements a crossbar that has a quadratic cost on the number of links of the router.

On the other hand, Fig. 1c shows a mesh with the minimum number of links to preserve strongly connectedness, which results in a very area-efficient implementation. However, the diameter of the network doubles since some routes may require eight hops to communicate one *PE* with another.
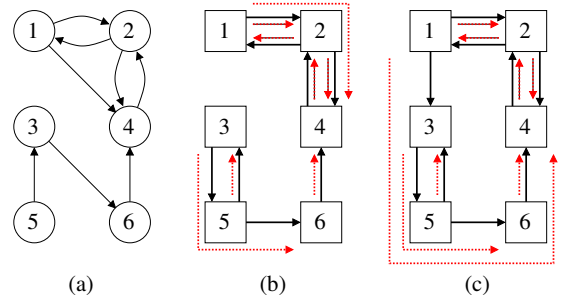
Additionally, some links may become over-congested if they have to be shared among different routes that carry dense traffic, thus incurring in a significant throughput penalty due to the contention in the network.

The designer will probably want to find an intermediate topology that satisfies certain throughput constraints with a reduced implementation cost. Figure 1b depicts one of these solutions. The cost has been reduced by removing some of the links of the full mesh. However the diameter has already been increased (some routes require five hops). This may also involve extra congestion in some specific links.

The solution space of this type of problems is huge. It becomes even larger when we consider the route assignment problem. Figure 2a depicts the communication graph for six *PE*s that need to exchange information. In this particular example, every *PE* is assumed to be attached to a router. Figure 2b shows a solution in which the links have been allocated to provide a minimum hop-count for every communication edge. The routes are represented by the dotted lines. The longest routes are for the pairs $(1, 4)$, with the route $1 \rightarrow 2 \rightarrow 4$, and $(3, 6)$, with the route $3 \rightarrow 5 \rightarrow 6$.

Let us assume that the traffic through the links $1 \rightarrow 2$ and $2 \rightarrow 4$ is very congested due to the intensive communication requirements of the *PE*s attached to those routers. Let us also assume that edge $1 \rightarrow 4$ is not critical and has a low priority. Hence, the designer might consider to deviate the traffic $1 \rightarrow 4$ through another route, as shown in Fig. 2c. This solution implies an extra link in the network, but may contribute to meet the throughput requirements of the system. Note that the route $1 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 4$ is not using the shortest path. The model presented in this paper allows the exploration of non-optimal paths to alleviate the traffic in congested links.

Even though this work assumes a 2D grid as the underlying structure, the topology of the network is not limited to regular meshes. Irregular meshes with *PE*s exceeding the size of one tile can also be considered, thus providing an extra flexibility in the exploration of solutions. Furthermore, the links are not constrained to have equal length. This makes the design flow even more flexible in terms of floorplanning and placement.

A. Sketch of the model

To explore the space of solutions, the mathematical model presented in this paper introduces a set of constraints for link allocation, route assignment and deadlock avoidance. The set of constraints can be extended to cover other design criteria.

One of the most practically important constraints is the limitation of the number of ports in every router. As it will be discussed further, port limitation highly reduces the complexity of the router, thus saving area and power resources of the system. The traditional link capacity constraint is also supported. These two types of constraints are associated to *physical* requirements of the design.

To guarantee a sufficient *performance* in the system, a set of delay constraints are defined. These constraints are modeled as a maximum hop-count (structural latency) for each net[1]. The *communication* demands are defined by the bandwidth requirement between each pair of *PE*s.

An essential property of route assignment is *deadlock and livelock freedom*. The incorporation of turn prohibition constraints guarantees the absence of any deadlock and livelock in the explored solutions.

The mathematical model includes four optimization objectives in the cost function: the minimization of the number of links in the grid, the maximum hop-count over all nets, the total net delay (sum of all net delays) and the uniform traffic distribution. The first objective is related to the area resource optimization, whereas the other three objectives guide the search toward increasing the performance of the system.

The constraints of the model can be represented as linear inequalities with integer variables. In this way, the problem can be specified with an integer linear programming (ILP) or integer quadratic programming (IQP) model, depending on the cost function. Even though these are NP-complete problems, the experimental results show that optimal solutions can often be found with moderate computational cost. Section V will present results obtained from different benchmarks.

IV. THE INTEGER PROGRAMMING MODEL

This section presents the integer programming (IP) model for link allocation and route assignment problem. Table I summarizes the input parameters for a quick reference. We introduce several types of variables and set notations to formulate the problem. The summary of the IP notations can be found in Table II.

A. Parameters and variables of the problem

The problem consists of defining a set of routes in a 2-dimensional grid structure that satisfies a set of physical and performance constraints. The routes must support the communication among the *PE*s of the system.

The grid structure with size (x, y) is represented as a directed graph $G(\mathcal{R}, \mathcal{L})$. The vertices of the grid define a set of routers $\mathcal{R} = \{R_0, .., R_{r-1}\}$, where the total number

[1]We refer to a net as a logical connection between two *PE*s, represented as an edge in the communication graph.

TABLE I: Input parameters of the problem.

Input	Description
(x, y)	Grid size
n	Number of nets
B_k	Required bandwidth for net N_k
D_k	Maximum hop-count for net N_k
C_j	Capacity of link L_j
$P_{i,in}$	Maximum number of input ports for router R_i
$P_{i,out}$	Maximum number of output ports for router R_i

TABLE II: Notation for the IP problem.

Notation	Type	Description
L_j		Link presence in the solution
L_j^k	Binary variable	Link usage by net N_k
T_p		Prohibition of turn T_p
D_{max}	Real variable	Maximum net delay
$\mathcal{I}(R_i)$	Set	Incoming links of router R_i
$\mathcal{O}(R_i)$		Outgoing links of router R_i

of routers is $r = x \cdot y$. The edges of the grid define a set of uni-directional links $\mathcal{L} = \{L_0, .., L_{l-1}\}$, where the total number of links for a grid with size (x, y) is calculated as $l = 2\left(x(y - 1) + (x - 1)y\right)$.

A global assumption about the grid is that every pair of neighboring routers may have up to two uni-directional links to send data in both directions. Each link L_j has a maximum capacity parameter C_j ($flits/cycle$). It limits the amount of data that can be transmitted over the link in one cycle.

Another input of the problem is the underlying communication graph $GC(PE, \mathcal{N})$ that represents the logical connectivity of the network. Every vertex represents a processing element and every edge represents a logical connection between a pair of processing elements. Every edge in the set $\mathcal{N} = \{N_0, .., N_{n-1}\}$ is a net of the system. Each net N_k has two associated parameters: the required bandwidth B_k (flits/cycle) and a maximum delay constraint D_k (hops) for the packet transmission from source to destination.

The additional parameters $P_{i,in}$ and $P_{i,out}$ specify constraints on the number of input and output ports for router R_i, respectively.

B. Path selection constraints

We focus on the deterministic routing path selection, without considering path diversity mechanisms. The latter would allow multiple paths for a pair of communicating processors. On the contrary, we assume there is only one path to send data packets for each communicating pair. Path diversity is an option for the routing path selection task. Our assumption for considering only one path eliminates the need to perform packet ordering at the destination router.

We start the constraint set description with introducing the basic mechanism for path selection and link representation in the model. Any configuration for link allocation contains a subset of links from the full grid. The presence of each link

in a configuration is represented by the set of variables L_j[2], i.e., $L_j = 1$ iff link L_j is present in the configuration.

The routing paths for every net are represented by another set of binary variables L_j^k, specifying the fact net N_k uses link L_j in its routing path. As we show below, this provides high flexibility for the solution space as every net may be routed through any arbitrary subset of links.

To reduce the potential large number of L_j^k variables ($n \cdot l$), it is possible to introduce rules that bound a routing region for a net. For example, we may not be interested in having long routing paths for the short nets with source and destination routers located at the neighboring grid vertices. In this case we may limit search region to be within few hops. An example is presented in Fig. 3. Net N_k is connecting two neighboring nodes, located in the corner of the grid. By limiting the maximum path length to 5 hops, only 10 links are eligible for selection in the route (marked with dashed lines). The max-hop constraints contribute to significantly reduce the number of link variables.

The sets of variables L_j and L_j^k are both related to the selection of link L_j. The L_j^k variable defines the relationship between a link and *a particular* net N_k, while L_j defines whether there is *at least one* net $N_{k'} \in \mathcal{N}$ such that $L_j^{k'} = 1$. In other words, $\forall k : L_j^k \Rightarrow L_j$. Assuming both sets L_j and L_j^k consist of binary variables and n is the total number of nets, we can write the following relations for each L_j:

$$L_j \leq \sum_{\mathcal{N}} L_j^k,$$
$$\sum_{\mathcal{N}} L_j^k \leq n \cdot L_j. \tag{1}$$

These two constraints guarantee the consistency of the variables from both sets in the IP model.

We now formulate the set of routing constraints that allow one and only one path selection for each net. Let us denote the set of incoming links to the router R_i as $\mathcal{I}(R_i)$ and the set of outgoing links as $\mathcal{O}(R_i)$. For each net N_k with source at router R_s, destination at router R_d and any intermediate router $R_i \in \mathcal{R} \backslash \{R_s, R_d\}$, the following constraints are defined:

$$\text{for } R_s \ : \sum_{\mathcal{I}(R_s)} L_j^k = 0, \ \sum_{\mathcal{O}(R_s)} L_j^k = 1$$
$$\text{for } R_d \ : \sum_{\mathcal{I}(R_d)} L_j^k = 1, \ \sum_{\mathcal{O}(R_d)} L_j^k = 0 \tag{2}$$
$$\text{for } R_i \ : \sum_{\mathcal{I}(R_i)} L_j^k = \sum_{\mathcal{O}(R_i)} L_j^k.$$

The first two equations in (2) represent the boundary conditions on the path for the source and destination routers, while the last one can be treated as a path maintenance constraint for the intermediate routers. Indeed, the source router R_s is the one that injects the N_k packets into the network, thus there

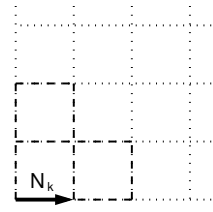[2]For the sake of notation simplicity we use L_j to denote both the variable and the link.

Fig. 3: Path region limitation for N_k with 5 hops.

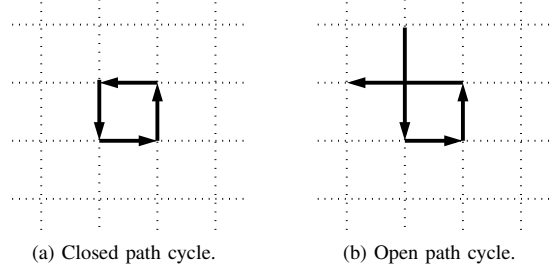(a) Closed path cycle.　　　(b) Open path cycle.

Fig. 4: Path cycles introducing redundant links.

should be no input links to this router. The number of output links, carrying the N_k packets from R_s should be equal to one, as we allow only one path for each net. The inverse situation is observed at the destination router R_d, that consumes N_k packets: there is one input link that delivers the N_k packets to R_d, while the number of output links is zero. The last equation in (2) guarantees that if an intermediate router R_i has an input link for N_k, then it will have an output link for this net. This condition assures that the path will be constructed correctly from source to destination node.

Note that the path constraints in (2) do not prevent the configuration from having cycles like the ones depicted in Fig. 4. Generally, a closed path cycle (Fig. 4a) may occur without breaking any constraint in (2), but allocating extra links. An open path cycle (Fig. 4b) may also occur, replacing one of the path turns with the sequence of three complementary turns and occupying redundant links. In the IP model, these cycles can never appear due to the turn prohibition mechanism to avoid deadlocks and the cost function of the problem, that tends to minimize the number of links in the network.

For efficiency reasons, we have found interesting to add explicit constraints on the number of input (output) path links for each router:

$$\text{for } R_i \ : \sum_{\mathcal{I}(R_i)} L_j^k \leq 1. \tag{3}$$

Even though the overall number of constraints in the model increases, our experiments show that the problem is solved faster due to the limitation of the solution space. The constraints (3) avoid the exploration of solutions with redundant links that will never be optimal.

C. Deadlock avoidance

Deadlocks and livelocks may occur in the wormhole routing networks due to the limited capacity of the router input
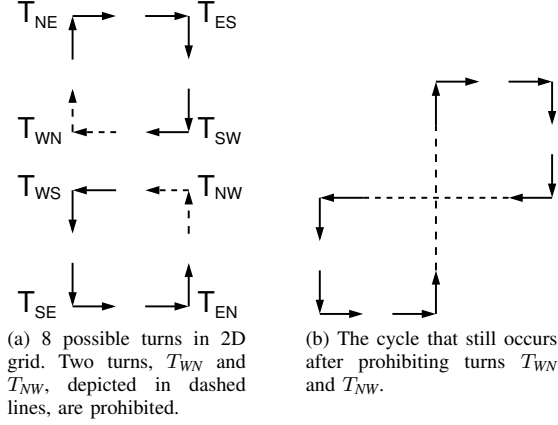
(a) 8 possible turns in 2D grid. Two turns, T_{WN} and T_{NW}, depicted in dashed lines, are prohibited.

(b) The cycle that still occurs after prohibiting turns T_{WN} and T_{NW}.

Fig. 5: Turns in a 2D grid.

(a) No turn in case of intersection.

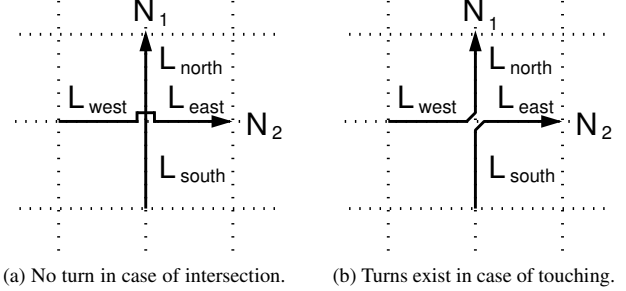(b) Turns exist in case of touching.

Fig. 6: Turn existence in dependence of nets positioning.

$$
\begin{aligned}
T_{WN} + T_{NW} &\leq 1, \\
T_{NE} + T_{EN} &\leq 1, \\
T_{ES} + T_{SE} &\leq 1, \\
T_{SW} + T_{WS} &\leq 1.
\end{aligned}
\tag{5}
$$

buffers [14]. An important property of the routing algorithm is deadlock freedom. In deterministic routing, the propagation paths for every net are defined statically. Thus, deadlock freedom can be guaranteed by incorporating certain restrictions into the path selection procedure.

One of the approaches for deadlock and livelock avoidance is *turn prohibition* [5]. There are eight possible turns a packet may follow in a 2D grid (Fig. 5a). We refer to a turn according to the directions of the input and output links of the turn, namely: west-north (WN), north-east (NE), east-south (ES), south-west (SW) in the clockwise direction and west-south (WS), south-east (SE), east-north (EN), north-west (NW) in the counter-clockwise direction. In order to guarantee that deadlocks never occur, certain turns should be prohibited in both cycles (clockwise and counter-clockwise). Specifically, prohibition of one turn from each cycle is enough to assure that the cycles will not occur. However prohibition of some turn pairs will still allow deadlocks resulting from the complex cycles depicted in Fig. 5b. Luckily, these are just four pairs and they are easy to identify [5]. Thus, when prohibiting two turns, one from each cycle, we should check that they do not belong to the same pair.

Finally, we want to apply the turn prohibition mechanism to guarantee deadlock freedom in the IP model. We introduce a set of binary turn variables to represent each one of the 8 possible turns: $\{T_{WN}, T_{NE}, T_{ES}, T_{SW}, T_{WS}, T_{SE}, T_{EN}, T_{NW}\}$. For example, the west-north turn will be prohibited in the final solution if and only if $T_{WN} = 1$. We formulate three sets of the turn constraints, based on the considerations above.

First, we have to guarantee that one turn is removed from each of the two potential cycles (Fig. 5a), that is

$$
\begin{aligned}
T_{WN} + T_{NE} + T_{ES} + T_{SW} &= 1, \\
T_{WS} + T_{SE} + T_{EN} + T_{NW} &= 1.
\end{aligned}
\tag{4}
$$

Second, the excluded turns should not belong to the same pair that still allows complex cycles (Fig. 5b):

To ensure that none of the selected paths incorporates a prohibited turn, we should guarantee that from each pair of links, that contribute to the turn, at most one link can be selected for the net path. We need to formulate these constraints with the net-related variables L_j^k.

Two neighboring links may exist in the solution independently, but the turn will occur only when there is a net that traverses these links in sequence. This idea is illustrated with the examples in Fig. 6. On the left example, two intersecting nets are depicted: N_1 propagating from south to north and N_2 from west to east. All four links exist in the routing solution:

$$
L_{north} = L_{east} = L_{south} = L_{west} = 1.
$$

However the solution does not contain any turn. This fact is reflected by the net-related variables that take the following values:

$$
\begin{aligned}
L_{north}^1 = 1, \ L_{east}^1 = 0, \ L_{south}^1 = 1, \ L_{west}^1 = 0, \\
L_{north}^2 = 0, \ L_{east}^2 = 1, \ L_{south}^2 = 0, \ L_{west}^2 = 1.
\end{aligned}
$$

None of the pairs $\{L_{west}^k, \ L_{north}^k\}$ or $\{L_{south}^k, \ L_{east}^k\}$ has both variables set to 1 (that would describe a turn condition). On the right example, two touching nets are shown: N_1 propagating from west to north and N_2 from south to east. All the four links are still present, but the net-related variables have different values now:

$$
\begin{aligned}
L_{north}^1 = 1, \ L_{east}^1 = 0, \ L_{south}^1 = 0, \ L_{west}^1 = 1, \\
L_{north}^2 = 0, \ L_{east}^2 = 1, \ L_{south}^2 = 1, \ L_{west}^2 = 0.
\end{aligned}
$$

In this case one turn is introduced by each net: an east-north turn T_{EN} by N_1 and a north-east turn T_{NE} by N_2. In the IP model, this fact is observed by obtaining two pairs of non-zero variables: $\{L_{west}^1 = 1, \ L_{north}^1 = 1\}$ and $\{L_{south}^2 = 1, \ L_{east}^2 = 1\}$. Therefore, in order to exclude a turn from the solution, we must prevent all contributing link pairs from having both variables set to 1. More formally, if turn T_p is prohibited, then for any net N_k and any pair of links L_j and L_j' that contribute to the turn, the following implication is required: $T_p \Rightarrow \neg(L_j^k \wedge L_{j'}^k)$, that is equivalent
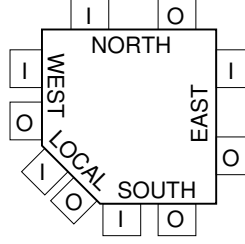
129

Fig. 7: NoC router with I/O ports in 5 directions.

to $\neg(T_p \wedge L_j^k \wedge L_{j'}^k)$. This represents the third set of the turn prohibition constraints for the problem, that we can formulate in the following manner. For every net N_k, turn T_p and all pairs of links L_j and L_j', that form T_p:

$$T_p + L_j^k + L_{j'}^k \leq 2. \tag{6}$$

The constraints (4), (5) and (6) are sufficient for the IP model to ensure a deadlock and livelock-free solution.

D. Port limitation constraints

A new design option introduced in this paper is the constraint on *port limitation*. A typical router for a 2D grid network has input (I) and output (O) ports in 5 directions, i.e. 10 ports in total (Fig. 7). However the router complexity highly depends on the number of ports. For instance, the size of the internal crossbar grows quadratically with the number of ports, contributing to the overall area and power consumption of the network. Also by constraining the number of ports, the physical design of the router and the routing of the wide links become easier. Finally, few-ported routers can often be implemented with single cycle latencies, low area and short cycle time. Many-ported routers often require a trade-off between latency, cycle time and area. By considering the use of few-ported routers, it is possible to have a global view of the optimization problem since we are not a priori restricted to the larger areas and latencies inherent in many-ported routers. Thus, it is useful for the designer to have the capability of limiting the number of ports for each particular router.

The IP model can be easily extended with port limitation constraints. These limitations can be reduced to limitations on the number of links each router is connected, since each link is connected to a port of the router. The constraints may also distinguish between input and output ports. Let the limits for the number of input and output ports of the router R_i be $P_{i,in}$ and $P_{i,out}$, respectively. Let us also introduce an indicator function $PE(R_i)$, that is equal to one if router R_i has a processing element connected to it, or zero otherwise. If $PE(R_i) = 1$, then a local port connection exists and the port limitation should be decreased by one. We have the following set of constraints for each network router R_i:

$$\sum_{\mathcal{I}(R_i)} L_j \leq P_{i,in} - PE(R_i),$$

$$\sum_{\mathcal{O}(R_i)} L_j \leq P_{i,out} - PE(R_i). \tag{7}$$

E. Link capacity constraints

Another set of constraints in our model refers to the link *capacity*. A link L_j can support a bandwidth up to C_j flits per cycle. The bandwidth of each link is one of the input parameters of the problem.

As one physical link may be used by several nets, the total traffic in the link will be defined by the sum of the bandwidths of all nets that are routed through the link. The net-related variables L_j^k can be used to define whether the path of net N_k uses the link. The following constraint guarantees that the total traffic in the link does not exceed the link capacity:

$$\sum_{\mathcal{N}} L_j^k \cdot B_k \leq C_j. \tag{8}$$

F. Net delay constraints

The proposed model offers a high flexibility in the selection of the routing path for any net as there are no limitations on the path shape or length. However, designers might be interested in limiting the path hop-count. This may be especially important for time-critical nets or some short nets that we want to prevent from having very long paths. In other words, we want to introduce performance constraints that approximate the *net delay* by the hop-count metric. This simple metric enables us to use the IP formulation, and yet at the same time accurately captures latency for low traffic loads. The hop-count of a net can be calculated as the sum over all net-related link variables, since only the links with $L_j^k = 1$ contribute to the path. Given a limit D_k for the hop-count of net N_k, we obtain the following constraint for the delay of net N_k:

$$\sum_{\mathcal{L}} L_j^k \leq D_k. \tag{9}$$

G. Cost functions

A variety of cost functions are introduced to find solutions with different optimization criteria. These cost functions are further discussed in the experimental section. The first three cost functions are linear, so the obtained problem is classified as an Integer Linear Programming (ILP) problem, while the last cost function is quadratic, resulting into an Integer Quadratic Programming (IQP) problem.

The cost functions do not need to be used in isolation. Linear combinations with weighted coefficients can be used for a multiple cost optimization.

1) Number of links: the total number of links is obtained by summing variables L_j over the set $\mathcal{L}$:

$$\min \sum_{\mathcal{L}} L_j. \tag{10}$$

Solving the IP problem with the cost function in the form (10) tends to find a feasible routing solution with minimized area and power consumption.

2) Maximum net delay: we may want to minimize the maximum net delay over all nets in the network in order to find a feasible routing solution with the highest performance (net delay constraints for particular nets may still be incorporated). The introduction of a new variable to represent the maximum delay, D_{max}, is required:

$$\sum_{\mathcal{L}} L_j^k \leq D_{max}. \tag{11}$$

and the cost function is simply:

$$\min D_{max}. \tag{12}$$

3) Total net delay: the minimization of the total delay over all nets (that is equivalent to minimizing the average net delay) can be regarded as another performance metric. The cost function in this case is obtained by summing all the net-related variables L_j^k for all nets from $\mathcal{N}$:

$$\min \sum_{\mathcal{N}} \sum_{\mathcal{L}} L_j^k. \tag{13}$$

4) Uniform traffic distribution: this cost function aims at assigning a uniformly distributed traffic over all the links of the network. The distribution tends to decrease the contention delays in the network and, hence, increase the overall network performance by improving the throughput and the average packet delay. This cost function introduces quadratic terms, so the problem becomes an IQP problem. A more uniform distribution is obtained by minimizing the sum of the squares of the link traffics:

$$\min \sum_{\mathcal{L}} \left(\sum_{\mathcal{N}} B_k \cdot L_j^k \right)^2. \tag{14}$$

H. Problem formulation

Having discussed the set of the constraints and the cost functions, we are now ready to present the formulation of the IP problem for link allocation and route assignment:

Find

the optimal value for a cost function obtained as a linear combination of (10), (12), (13) and (14)

subject to

  physical constraints (7),
  application-specific communication constraints (8),
  performance constraints (9),
  deadlock-avoidance constraints (4), (5), (6)
  and additional model constraints (1), (2), (3), (11).

Note that the user is free to select a subset of constraints if certain features are not required for the design. The cost function can also be extended with small effort due to the flexibility of the model.

V. EXPERIMENTAL RESULTS

This section presents the experimental results to demonstrate the functionality and the quality of the proposed IP model. We define the trade-offs and discuss the results for several design problems with various constraint sets and optimization objectives.

We use CPLEX [15] to solve the IP model and an accurate flit-level C++ simulator with a variety of routing schemes to obtain the network parameters. Different testcases are used throughout the experiments: we start with artificial configurations and we next consider a testcase with typical server workloads from the SPEC2006 benchmarks.

Three types of the experiments are introduced. Section V-A analyzes the area-performance trade-off. In Sect. V-B, the application of the model for performance optimization by means of the routing paths redistribution is analyzed. Finally, the use of port limitation constraints for design exploration and tuning is presented in Sect. V-C.

A. Area-performance trade-off

One of the optimization tasks in the design of multiprocessor interconnection network is the minimization of the number of links. Given a set of constraints, the goal is to find the minimal number of links that satisfy the constraints, determine the link allocation and assign the traffic routes. This optimization contributes to decrease area and leakage power. However, the average hop-count delay may increase as the number of the links decreases and the packets have to follow longer roundabout paths. For this reason, the dynamic power may also increase. Note, that the variation in power will be defined by the relation between leakage and dynamic power.

This set of experiments demonstrates the ability of the model to explore the area-performance trade-offs by link reallocation. Given the communication graph, we first search for the minimal number of links to enable the connectedness of all routers and estimate the maximum net delay value. Additionally, we investigate minimal link solutions subject to the limitation on the maximum net delay. In order to find the minimal link allocation, we solve the problem with the cost function in form (10). We apply the constraints (9) to limit the net delay and (4)-(6) to guarantee the deadlock freedom.

We show the area-performance trade-off for a 4x3 network with a complete communication graph, i.e. with net between every pair of processing elements. The minimal number of links required to connect all routers is 12 and forms the uni-directional ring topology, that is depicted in Figure 8a. For this allocation the packet delivery will take up to 11 hops for some nets. However, the diameter of the network can be reduced to 5 hops. The solution obtained with this delay limitation is presented in Fig. 8b. It incorporates 20 links, but the delay for any net is now guaranteed not to exceed 5 hops. Obviously, there is a trade-off between these two cases. We explore it by varying the delay constraint value in the specified range. The set of solution points is displayed in Fig. 10 ("Minimal"). Based on this dependency, one can determine the minimal
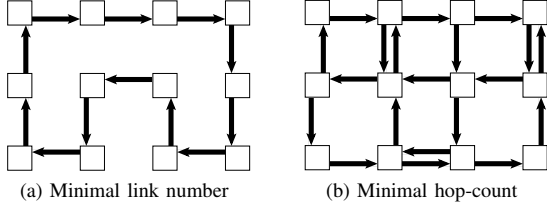
(a) Minimal link number   (b) Minimal hop-count

Fig. 8: Link allocation solutions for a 4x3 network.

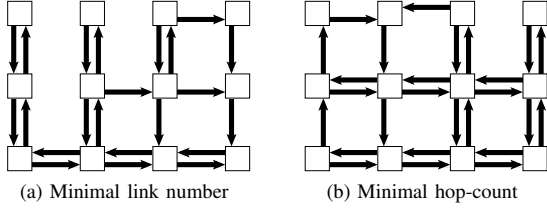(a) Minimal link number   (b) Minimal hop-count

Fig. 9: Deadlock-free link allocation for a 4x3 network.

number of links to guarantee a particular network diameter (for example, 14 links are required for the 8-hop network).

Another important property is deadlock freedom. We provide a similar function after incorporating the turn prohibition constraints into the problem. Due to the extra limitations in routing paths, deadlock-free solutions tend to include more links for the same hop-count limit. Thus, the minimal number of links to provide full connectivity is 22 (Fig. 9a) and the solution with minimal delay of 5 hops has 26 links (Fig. 9b). This trade-off is also depicted in Fig. 10 ("Deadlock-free").

Finally, we note that even the "Minimal" solutions can be designed to be deadlock-free by choosing a suitable architecture. For example, the solution shown in Fig. 8a can be realized in practice by using a token-ring architecture [16] or virtual channels and dateline scheme [14]. Our model is also capable of discovering well-known structures, such as the bi-directional ring, that is seen in the variety of cell processors [17]. This proves that the class of the generated solutions is actually used in practice.

The area-performance trade-offs discussed in this section demonstrate the suitability of the model for design exploration and optimization. By incorporating additional application constraints, the user is allowed to perform more accurate, application-specific optimizations.

B. Performance optimization by route reassignment

Another application of the model is related to the optimization of the network delay by routing path redistribution. In this experiment we assume the communication requirements of the network, including the nets and their bandwidths, are specified. The objective is to minimize the average packet delay of the network. The average optimal hop-count delay can be obtained with the cost function (13), but the contention affects the delay value significantly once the network enters the saturation region. However, the contention delays can be alleviated by distributing the traffic uniformly over the network. For this purpose, we are using the cost function (14) to select routing paths that distribute the traffic more uniformly. The quality
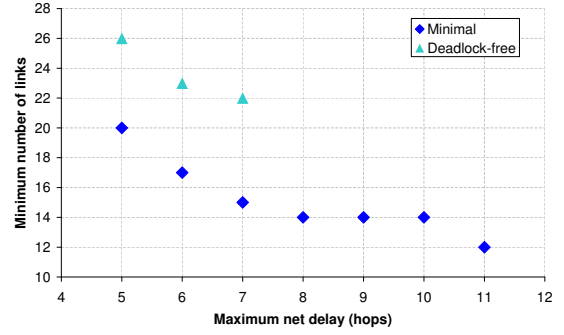

Fig. 10: Area-performance trade-off points in terms of the link number and net delay for a 4x3 network.

of the solution is estimated by simulation and compared to that of the XY and odd-even routing algorithms. Using the example of a typical server workload, we demonstrate that the obtained solutions improve the network delay as compared to the classical routing algorithms for a wide range of injection rates. Furthermore, the throughput increases as the saturation occurs at higher injection rates.

In this experiment we are considering the typical server workload traffic pattern collected using the SPEC2006 benchmarks. A 16-core application is assumed to be mapped onto a 4x4 full mesh, with all links present (Fig. 11). The cores connected to the routers 1 and 2 are the memory controllers that receive high traffic from the other cores. The traffic injected by each core is assumed to have Poisson distribution.

In Fig. 12, the comparison for the average delay estimation at different traffic rates is shown. We draw the packet delay as a function of the total injection rate to the network (in packets/cycle), for each of the three mentioned routing algorithms: XY, odd-even and the one using routing tables, based on the IQP solution. The timeout for the IQP solution was set to 1000 seconds. Simulation shows that the average packet delay, obtained with the IQP routing, is better than the delays of XY or OE schemes in the large range of injection rates. The XY-routing is only winning slightly the IQP configuration when the injection rates are small, as contention is low and the XY scheme results into the most uniform solution. However, as soon as contention effects start to contribute significantly to the delay value (injection rates ≥ 0.3 pkt/cycle), the IQP routing improves the average packet delay, as compared to
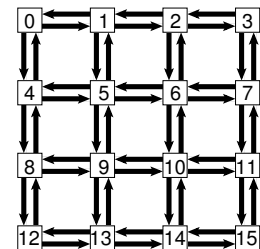

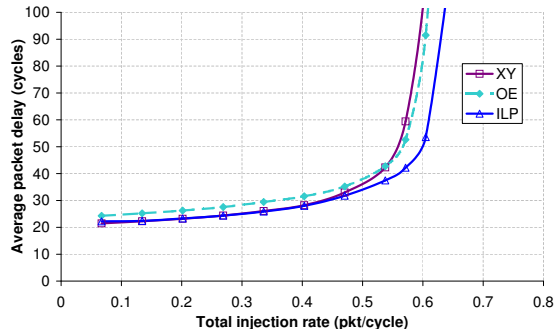Fig. 11: Underlying mesh for a typical server workload test-case.

Fig. 12: Average delay depending on the injection rate.

Fig. 13: Network layout with 4 input and 4 output port limitation. Links connecting routers with the co-located *PE*s are not shown.

both XY and odd-even schemes. It can be also seen from the graph that the saturation occurs at higher rates, hence, the network throughput is increased.

C. Design optimization by port limitation

Port limitation is another feature introduced by the model in order to extend the user design flexibility. The ability to limit the number of ports provides the means to account for the router design complexity at the network planning stage. Typical routers with 5-in and 5-out ports (Fig. 7) have complex designs and are not capable of running the full bandwidth. Hence, a mismatch between the network floorplanning stage and the router functionality appears, resulting in a potential loss of performance. By limiting the maximum number of ports of the routers, the use of complex routers during the network planning is avoided. This limitation also allows the optimization of area and leakage power.

We use a simple intuitive model to measure the area variation of the components in the network. We assume that the major network components are the links and the routers. The total link area is proportional to the number of links that is obtained from the IP solution. The area of the router can be approximated by the complexity of the crossbar and buffer area [18]. The crossbar area has a quadratic dependency on the number of ports, while the buffer area dependency is linear.

We assume that the leakage power is proportional to the network area. In order to estimate the variation of the delay and the dynamic power of the solution, we use a simulator with the incorporated Orion power model [19].

The same typical server workload example of the system, mapped to the 4x4 network, will be used to demonstrate the port limitation functionality. We perform a set of experiments, aimed at finding the optimal route assignment and link allocation, subject to additional limitations on the maximum number of input and output ports of the routers. Further we estimate the network parameters and make comparison to the results obtained for the full mesh solution with XY-routing.

In these experiments, the number of ports includes the local connections to *PE*s (see discussion of (7)). In the full mesh solution, there are no limitations on the number of router ports. The largest routers with 5 input and 5 output ports appear in locations 5, 6, 9 and 10 (Fig. 11). Table III shows the results of
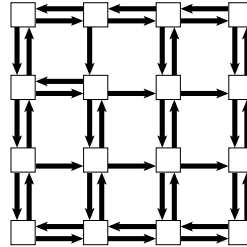
solving the route assignment and link allocation problem with the number of input, output or both types of ports limited to 4. Each row is related to a different experiment with a particular port limitation. The first two columns of the table represent the maximum number of ports that a router may have. The values in the following columns are normalized to those obtained for the full-mesh solution with XY-routing. Thus, the ratio of the total link, crossbar and buffer area is reported in columns from third to fifth. The average packet delay and dynamic power are reported in the last two columns.

An example of the network layout for the experiment with 4-input and 4-output port limitation (4th experiment in Table III) is depicted in Fig. 13. This layout contains 42 links instead of the 48 links in the full-mesh solution. The total area of links, crossbars and buffers in the presented solution has been decreased by 12.5%, 13.7% and 6.4%, respectively. One can observe the average packet delay increase by 9.9% as well as the dynamic power increase by 6.4%. The increase of average delay, unless the contention is high, is caused by the removal of links, since the minimal path for the neighboring routers rises to 3 hops. However, the increasing delay may be an acceptable solution if certain nets are not critical (see discussion of the example in Fig. 2). Otherwise, a designer is allowed to put a limiting hop-count constraint for the critical nets. The variation in power should be calculated together with the leakage power decrease, that is correlated with the network area. The total power estimation is technology dependent, but due to the growing importance of the leakage power resulting from the technology downscale [20], the variation in power may be negligible as compared to the area savings.

This example demonstrates the potential introduced by the port limitation mechanism. Its applicability can be combined with other design constraints. This provides a designer with a vast spectrum of possibilities for exploration and tuning.

TABLE III: Port limitation results for server workload testcase.

Port limit		Area			Average	Dynamic
in	out	link	xbar	buffer	delay	power
5	5	1.000	1.000	1.000	1.003	1.001
5	4	0.916	0.928	0.969	1.037	1.027
4	5	0.916	0.928	0.969	1.040	1.037
4	4	0.875	0.863	0.936	1.099	1.064

TABLE IV: CPU time for link minimization.

Number of cores	Network size	Minimal link count (sec)	Deadlock-free (sec)	
			Feasible	Optimal
8	4x2	0.21	0.70	1.98
9	3x3	0.81	5.31	31.74
10	5x2	0.55	4.40	23.60
12	4x3	7.86	49.90	4031.25
16	4x4	147.96	2117.89	Timeout

D. Computational time

The computational complexity of the IP model depends significantly on the number of binary variables of the model that determines the span of the branch-and-bound search. For a square mesh of P processing elements, the number of variables is about $4P^3$. Still, efficient ILP solvers can handle this model for problems with moderate size.

Table IV shows the CPU time for solving the model with the link minimization cost function (10), that is the most time consuming linear cost function. The third column shows the time to find the minimal number of links that guarantee network connectedness. The last two columns report the CPU times for finding deadlock-free solutions, which are larger due to the introduction of the turn prohibition constraints.

When an optimal solution is hard to find, a feasible solution close to the optimal might be also sufficient. The column "Feasible" reports the time required by the solver to find the optimal solution, while the rest of the time was spent to prove the non-existence of a better solution. The solution for the 4x4 network was obtained by defining a CPU timeout of three hours. The reported solution is the last one obtained within the timeout, without knowing whether it was optimal or not. Given the behavior for the other cases, we conjecture that this solution is very close to the optimal.

The results show that optimal solutions can be obtained for moderate size networks. The model is also useful to partially explore the search space with CPU time limits, still obtaining high-quality solutions.

VI. Conclusions

In this paper we have presented a mathematical formulation of the problem for the simultaneous route assignment and topology selection in multiprocessor networks. It provides a solution for a large variety of routing and optimization problems in the design of on-chip networks with already floorplanned processing elements.

The proposed IP model enables the designer to perform network topology exploration and tuning, subject to a large set of user-defined constraints. The port limitation constraint prevents the incorporation of complex routers and enables a large flexibility for exploration and optimization.

The model was validated with a set of testcases, including typical server workloads, demonstrating the capability to explore solutions with different area-performance trade-offs and performing independent optimizations in various domains.

REFERENCES

[1] W. J. Dally and B. Towles, "Route packets, not wires: on-chip intecon-nection networks," in *DAC '01: Proceedings of the 38th conference on Design automation*. New York, NY, USA: ACM, 2001, pp. 684–689.

[2] G. D. Micheli and L. Benini, *Networks on Chips: Technology and Tools (Systems on Silicon)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2006.

[3] K. Srinivasan, K. S. Chatha, and G. Konjevod, "Linear-programming-based techniques for synthesis of network-on-chip architectures," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 4, pp. 407–420, 2006.

[4] G.-M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 7, pp. 729–738, 2000.

[5] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," *SIGARCH Comput. Archit. News*, vol. 20, no. 2, pp. 278–287, 1992.

[6] J. Hu and R. Marculescu, "Dyad: smart routing for networks-on-chip," in *DAC '04: Proceedings of the 41st annual Design Automation Conference*. New York, NY, USA: ACM, 2004, pp. 260–263.

[7] M. Palesi, R. Holsmark, S. Kumar, and V. Catania, "A methodology for design of application specific deadlock-free routing algorithms for noc systems," in *CODES+ISSS '06: Proceedings of the 4th international conference on Hardware/software codesign and system synthesis*. New York, NY, USA: ACM, 2006, pp. 142–147.

[8] M. Palesi, G. Longo, S. Signorino, R. Holsmark, S. Kumar, and V. Catania, "Design of bandwidth aware and congestion avoiding efficient routing algorithms for networks-on-chip platforms," in *NOCS '08: Proceedings of the Second International Symposium on NoCs*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 97–106.

[9] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Qnoc: Qos architecture and design process for network on chip," *J. Syst. Archit.*, vol. 50, no. 2-3, pp. 105–128, 2004.

[10] E. Bolotin, A. Morgenshtein, I. Cidon, R. Ginosar, and A. Kolodny, "Automatic hardware-efficient soc integration by qos network on chip," in *Electronics, Circuits and Systems, 2004. ICECS 2004.*, Dec. 2004, pp. 479–482.

[11] M. K. F. Schafer, T. Hollstein, H. Zimmer, and M. Glesner, "Deadlock-free routing and component placement for irregular mesh-based networks-on-chip," in *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 238–245.

[12] R. Holsmark, M. Palesi, and S. Kumar, "Deadlock free routing algorithms for irregular mesh topology noc systems with rectangular regions," *J. Syst. Archit.*, vol. 54, no. 3-4, pp. 427 – 440, 2008.

[13] S.-Y. Lin, C.-H. Huang, C.-H. Chao, K.-H. Huang, and A.-Y. Wu, "Traffic-balanced routing algorithm for irregular mesh-based on-chip networks," *Computers, IEEE Transactions on*, vol. 57, no. 9, pp. 1156–1168, Sept. 2008.

[14] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.

[15] "CPLEX," http://www.ilog.com/products/cplex.

[16] F. Deslauriers, M. Langevin, G. Bois, Y. Savaria, and P. Paulin, "Roc: A scalable network on chip based on the token ring concept," in *Circuits and Systems, 2006 IEEE North-East Workshop on*, June 2006, p. 157.

[17] M. R. Marty and M. D. Hill, "Coherence ordering for ring-based chip multiprocessors," in *MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 309–320.

[18] R. R. Dobkin, R. Ginosar, and A. Kolodny, "Qnoc asynchronous router," *Integration, the VLSI Journal*, vol. 42, no. 2, pp. 103 – 115, 2009.

[19] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: a power-performance simulator for interconnection networks," in *Microarchitecture, 2002. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, 2002, pp. 294–305.

[20] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in *Design, Automation and Test in Europe Conference and Exhibition, 2009. DATE '09.*, April 2009, pp. 423–428.