Topological Channel Routing Using Constraint Logic Programming

by

Andrew Hughes, Anthony J. Kusalik, Carl D. McCrosky

Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada

Abstract

Channel routing is an essential part of electronic design automation known to be a "hard" problem. Currently-employed channel routers typically restrict the geometry of the channel layout to efficiently complete a routing. Features of modern VLSI technologies, such as a large number of available layers, are poorly utilized by such approaches.

An innovative approach to solving the channel routing problem which addresses these shortcomings has been developed. The overall problem is broken into two components: topological definition followed by physical placement. Topological definition is further decomposed into the canonical problems of topological sort and graph colouring. The resulting computational tasks are naturally modelled as a constraint-satisfaction problem and solved using constraint logic programming. An experimental channel router conforming to this design has been implemented and subjected to testing. It exhibits many superior qualities, including the ability to exploit multiple routing layers to reduce channel height.

This paper focuses on the algorithm used to solve the topological definition subproblem, the presence and nature of constraints in that subproblem, and use of constraint techniques in the solution. Other aspects of the research are more briefly discussed.

1. Introduction

Channel routing is the process of creating a layout in a rectangular region of a VLSI chip to realize an interconnection network. It is known to be a hard problem, with many simplified graph representations shown to be NP-complete. Approximate solutions are employed on a routine basis as part of electronic design automation. CLP (Constraint Logic Programming) is an effective tool for dealing with computationally difficult problems [Cox+92, JaMa94] where approximate or heuristic solutions are otherwise used.

While previous work [ChSa93, LiTh93, Suz+95, Zhou96] has addressed the difficult search problems associated with channel routing using techniques derived from artificial intelligence, this has usually been in a conventional reserved layer channel routing framework with a set number of routing layers. Advances in VLSI technology are providing more and more layers for routing. Typical channel-routing algorithms, designed for a fixed number of layers, cannot take advantage of additional layers when present. When migrating a design to a new technology with additional layers, the unused layers cannot be utilized to reduce the channel height and the overall size of the circuit. In any case, there is a need for a channel router designed to exploit an arbitrary number of layers.

This paper concerns a new channel routing approach which explicitly separates the channel-routing problem into two relatively independent portions: topological definition and physical placement. The latter subproblem has been adequately addressed in the literature, albeit in the guise of channel compaction. The focus in this paper is on solving the topological definition subproblem. An approach is used which is well-suited to being modelled and implemented using CLP. This paper presents the design of the new topological channel-routing algorithm, and outlines a successful implementation. A fuller discussion of the work — with more detail on the implementation, a description of a complete router, and an extensive description of testing — is presented elsewhere [Hugh98].

The paper is organized as follows. The remainder of Section 1 provides background on VLSI design, the channel routing problem, existing channel routing techniques, and related work involving constraints applied to VLSI design. Section 2 describes our new approach. Section 2.1 introduces the topological model which is used to derive the topological definition subproblem described in Section 2.2. Constraint-based solution to the resulting subproblems is the focus of Section 3. Section 4 briefly surveys other work performed as part of this research. The last section summarizes results and proposes a direction for future effort. The paper assumes a familiarity with CSPs and CLP.

1.1 VLSI Concepts

This section briefly presents relevant background material on VLSI design technology and terminology. More information can be found in sources such as Wolf's standard text [Wolf94].

A VLSI circuit is fabricated as a set of alternating layers of conductive and insulating material deposited on a silicon substrate (Fig. 1). Each layer is etched to form a pattern, and the combination of these layers and patterns forms an electronic circuit. The VLSI characteristic of greatest interest in this work is the number of layers of conductor deposited. The conductive layers are called the *routing layers*. The pattern etched into these layers is called the *routing layout* and it determines if two locations are electrically connected. An electrical connection between two adjacent layers is accomplished using a *via* — a gap in the insulation between two layers, filled with a plug of conductor. Figure 1 is a cross-sectional view of the routing layers.

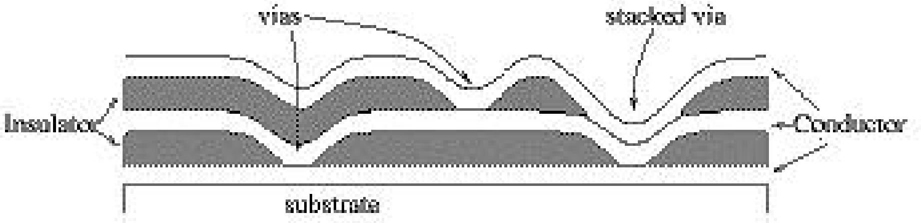


Figure 1: VLSI Layers

Most VLSI circuits (chips) are assembled using *modules* , predesigned circuit elements with known dimensions. Modules have *terminals* defined on their boundaries. These terminals are electrically connected together to complete the circuit design. A group of terminals and the routing layout which connects them is termed a *net*. The modules are placed on the chip and the regions between them are used for the routing layout. This process forms *channels* , rectangular regions bounded on facing sides by modules. The distance between modules is the *channel height*. The *channel congestion* at a given horizontal location in the channel is the number of nets which have terminals on both the right and left of this location. Figure 2 shows an example channel. Terminals which must be interconnected are labelled with the same (net) number in the figure.

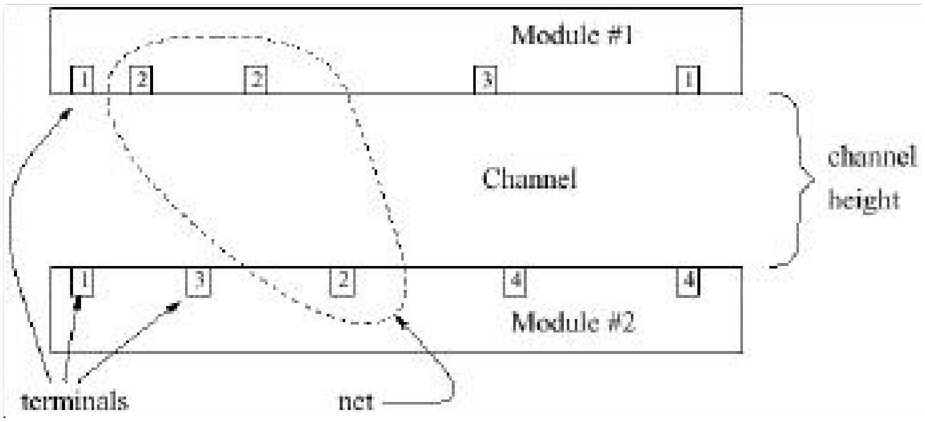


Figure 2: Channel Definition

Most VLSI routing is done using rectangles of conductor called *doglegs* and *trunks*. The major axis of a trunk is parallel to the axis of the channel, while for a dogleg it is perpendicular (Fig. 4). The *width* of a dogleg or trunk is its extent along the dimension of its minor axis. Nets are realized using sets of wires. A *wire* is a sequence of contiguous, alternating doglegs and trunks all on the same layer. It is terminated on both ends with either a terminal or a via. In Fig. 4 the net is composed of three wires. Since the amount of current a net must carry is known and the thickness of the conductor is fixed, the minimum width of the wires composing the net can be determined.

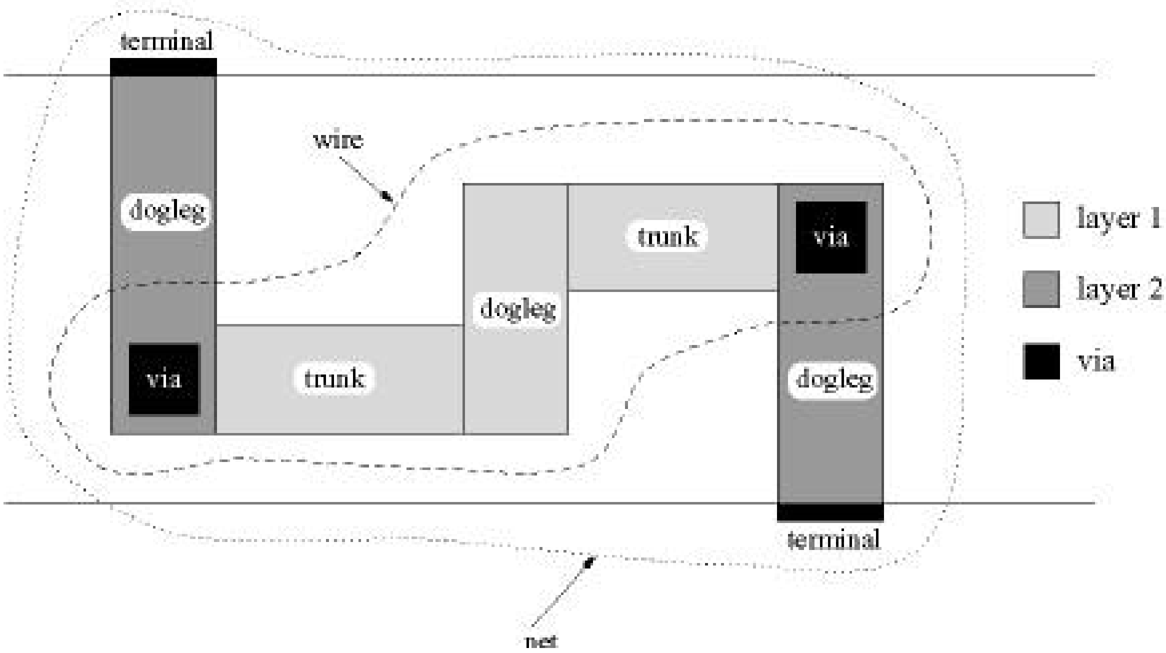


Figure 4: Channel Routing Definitions

1.2 The Channel Routing Problem

The *channel routing problem* is the determination of a layout in the channel such that the terminals are correctly connected. Additionally, channel height and via count should be minimized. Often a significant amount of chip area is required for routing. Minimizing the channel height reduces this area. Delay in transmitting signals degrades the performance of a circuit. Increasing the number of vias increases the delay and also decreases the chance that a circuit will be correctly fabricated [Wolf94].

This research tackles a subset of the general channel routing problem, the *two-terminal channel-crossing problem.* In this restriction, it is assumed that all nets have only two terminals, one on either side of the channel, and that there are no through-channel nets (nets which have terminals outside of the channel). These restrictions do not detract from the value of the approach as they allow the majority of channels to be considered (i.e. most channels can be simulated as a two-terminal channel-crossing case) and can be relaxed at a later date.

1.3 Design of Typical Channel Routers

Channel routers are classified by the number of layers they are designed to exploit and how they route wires within these layers. The latter determines if a router is *grid-based* and if it uses a *reserved layer* model [Sher93]. Grid-based routers place a regular grid — to which doglegs and trunks musst conform — over the channel. The wires are all the same width, the width of the largest component, usually a via. With *gridless* routers doglegs and trunks can occupy any location in the channel and wires can differ in width according to their electrical requirements. This increased flexibility comes at a computational cost, however. Reserved layer routers require that wires consist of a single dogleg or trunk, with doglegs and trunks assigned to different layers as specified by the router. *Unreserved layer routers* have no such restrictions; a dogleg or trunk can be on any layer. This allows greater flexibility in routing wires and can significantly decrease final channel height. It also makes the problem significantly more difficult. For these reasons many channel routers use a grid-based, reserved-layer model. In contrast, the technique proposed in this work is gridless and unreserved-layer.

1.4 Related Work

Topological models have been used before in solving the channel routing problem. For example, Haruyama [Haru92] uses a topological model which represents terminals, vias, and crossings of nets in a topological graph. The method minimizes both wiring area and via count. However, it only considers the two-layer channel routing situation, and physical locations for nets are determined directly from the topological graph. Constraints inherent in the topological model are not explicitly represented or manipulated.

Constraint satisfaction and constraint logic programming (CLP) have been often used for problems in VLSI. For example, they have been used for simulating VLSI circuits and for diagnosing faults in circuits [Hei+92]. CLP has also been used for the channel routing problem [Simo90, Wilk96, Zhou96]. However, these other works have typically tackled the problem in a more conventional manner — following a reserved layer approach — rather than decomposing the problem as done here. Thus they tend to suffer from the same disadvantages and restrictions as reserved layer channel routers implemented using procedural programming techniques.

2. A New Solution Strategy

Many channel routers are restricted in their ability to exploit multiple layers and minimize via count. These restrictions are typically a result of the reserved-layer model. The routing technique discussed here circumvents these restrictions. It separates channel routing into two subproblems: *topological definition* and *physical placement.* Topological definition specifies the assignment of wires to layers and the mapping from a two-dimensional topological model to a set of intervals. Physical placement is the process of creating a layout from the topological relationships. It defines the sequence of doglegs and trunks composing a wire and specifies the placement of these items within the channel. The focus of this work is on solution of the topological definition subproblem. Physical placement is briefly discussed in Section 4.

The subsequent subsections reference the channel routing problem shown in Fig. 5. The channel is implemented in a two-layer technology with terminals a' , b' and c' located on layer one and terminals a'' , b'' and c'' on layer two. The terminals which form a net have the same letter.

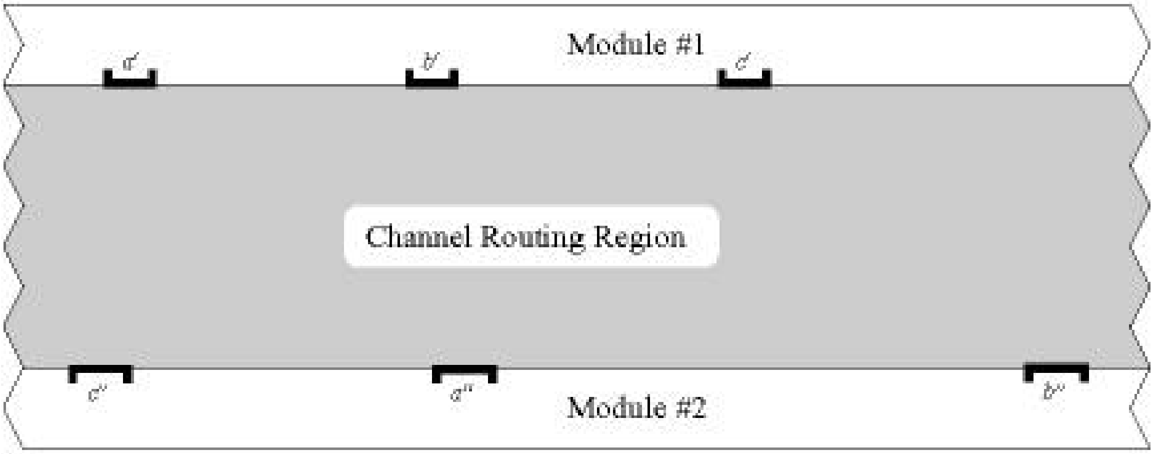


Figure 5: Example Channel

2.1 Topological Model

Solving the topological definition subproblem involves division of the nets into wires based on the interaction of the nets in the channel. To proceed, a topological model of the net is established wherein a net is defined as being topologically equivalent to a line segment between the terminals. The channel is modelled as a set of intersecting line segments with the intersections between the lines defining the division of the nets into wires. There is one wire (per net) for each intersection and one wire for each terminal. This model, as applied to the channel routing problem in Fig. 5, is illustrated in Fig. 6.

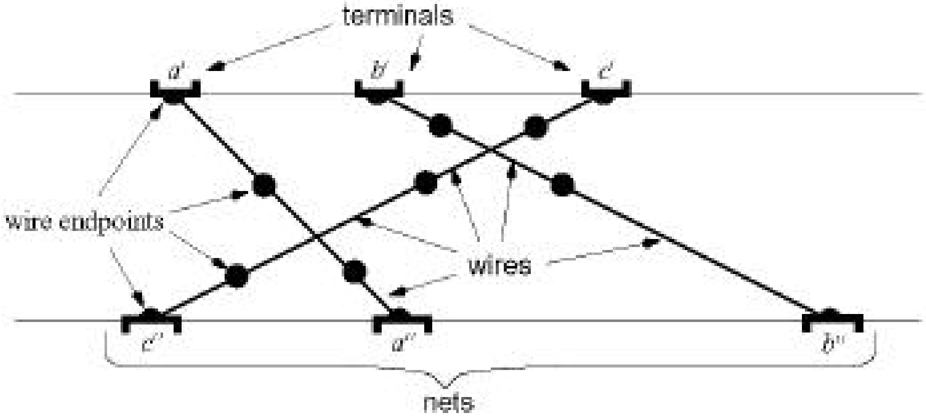
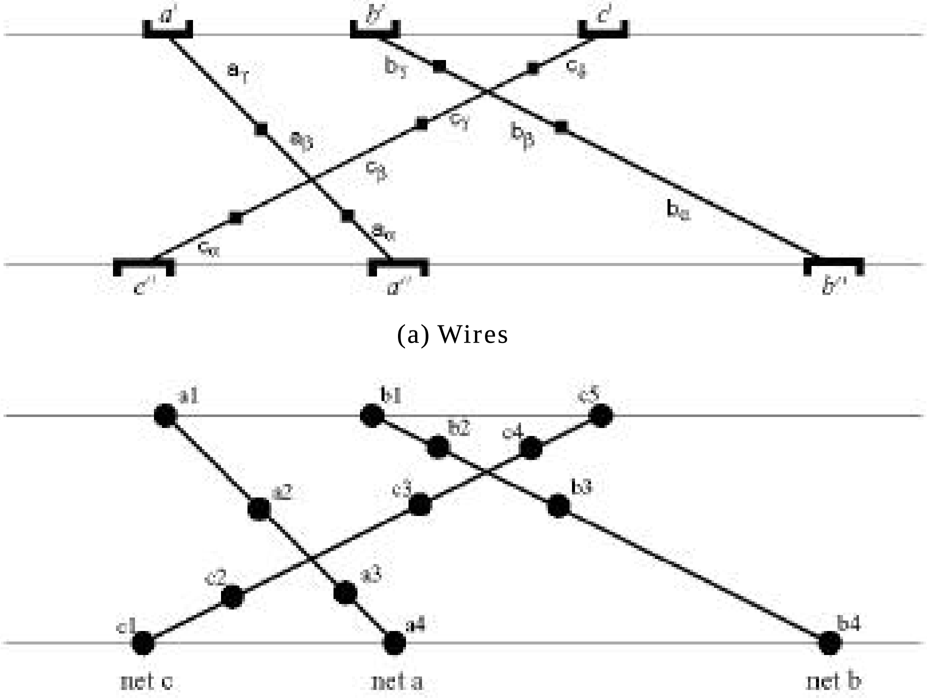


Figure 6: Two-dimensional Topological Model

Since the wires are portions of a net, they must lie on the line segment for the net and be line segments themselves. The points which separate adjacent wires of the same net are the *wire endpoints*. For the wires to lie on the net's line segment, the wire endpoints must also lie on the net's line segment. Thus, a net's line segment is divided into subsegments corresponding to wires with these subsegments separated by points corresponding to wire endpoints. The model specifies the relationships between the nets, wires, and wire endpoints in two-dimensional terms; i.e. a wire line segment may be below, above, to the left, or to the right of any given wire endpoint.

Figure 7 shows a labelling of wires and wire endpoints for the example channel routing problem. Wire labels are subscripted with Greek letters, while wire endpoint labels are indexed with numbers. Once a topological model of the net is established, the topological definition subproblem can be solved.



(a) Wires

(b) Wire Endpoints

Figure 7: Wire and Wire Endpoint Labels

2.2 Topological Definition

The topological definition subproblem has two components: deriving an interval representation of the wires and assigning wires to layers. A solution to these components relies on and maintains the equivalence to the topological model and determines: (a) the number of vias, (b) the minimum possible channel height, and (c) the parameters of the physical placement subproblem (Section 4). The components of the topological definition subproblem are independent in that they can be solved in any order. They are discussed in detail in the following subsections.

2.2.1 Interval Representation

A necessary condition for an optimal channel routing is that any vertical line drawn across the channel intersects each net only once [RiFi82]. This condition can be met if the wires are topological equivalent to one-dimensional intervals. Figure 8 shows an example of an interval representation of a wire together with a plausible physical layout. Notice that the physical layout is topologically equivalent to the interval in that any vertical line can only intersect the wire once. Consequently, wires are represented as one-dimensional intervals, and in the routed channel all wires will have this topological equivalence.

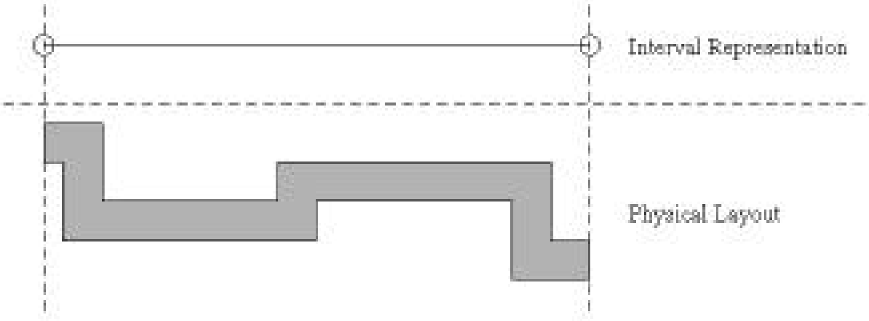


Figure 8: Interval with a Physical Layout

The above goal can be satisfied by mapping the two-dimensional topological model to a one-dimensional set of intervals. A *wire endpoint sequence* is such a mapping. It is a total ordering of all of the wire endpoints in the topological model. There are many such mappings, each corresponding to a valid sequence of the wire endpoints. Only one is necessary . The set of possible endpoint sequences is limited by the topological model. Figure 7(b) shows the wire endpoints from our example problem for which a sequence is to be determined.

The two-dimensional topological model imposes partial orders on the endpoint sequences. There are three sources of such partial orders:

1. The terminals have known horizontal locations which can be sorted from left to right. In Fig. 7 this is $\{c1 \rightarrow a1 \rightarrow b1 \rightarrow a4 \rightarrow c5 \rightarrow b4\}$. If two terminals have the same horizontal location then a pessimistic ordering which requires the attached wires to overlap is chosen for their corresponding wire endpoints.
2. The endpoints for all the wires from the same net have an established order. The topological model requires that this order be preserved. For our example these orderings are $\{a1 \rightarrow a2 \rightarrow a3 \rightarrow a4\}$, $\{b1 \rightarrow b2 \rightarrow b3 \rightarrow b4\}$, $\{c1 \rightarrow c2 \rightarrow c3 \rightarrow c4 \rightarrow c5\}$.
3. There are four wire endpoints adjacent to each intersection between nets in the topological model. These wire endpoints are grouped into an upper pair (above the intersection) and a lower pair (below the intersection). Each pair forms a partial order. There are two intersections in Fig. 7, and these yield the partial orders $\{a2 \rightarrow c3\}$, $\{c2 \rightarrow a3\}$, $\{b2 \rightarrow c4\}$, and $\{c3 \rightarrow b3\}$.

These partial orderings are summarized in a directed graph, the *endpoint constraint graph*. For the running example, the graph is shown in Fig. 9.

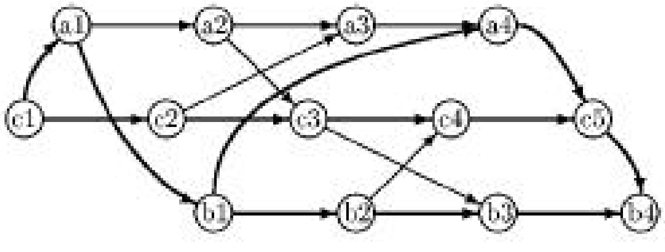


Figure 9: Endpoint Constraint Graph for the Data in Fig. 7

Specific total orders compatible with all the partial orders are generated by complete traversals of the endpoint constraint graph [Golu80]. A search tree describes the possible traversals. The graph limits the permutations of the endpoint sequence and the search tree enumerates these permutations. Different sequences are different paths from the root of the tree to the leaf nodes. One possible sequence for the example problem from Fig. 5 is $\{c1 \rightarrow a1 \rightarrow a2 \rightarrow c2 \rightarrow a3 \rightarrow b1 \rightarrow a4 \rightarrow c3 \rightarrow b2 \rightarrow c4 \rightarrow b3 \rightarrow c5 \rightarrow b4\}$.

Each wire is associated with the interval defined by its endpoints. These intervals are completely defined by the endpoint sequence. Thus, the endpoint sequence maps the two-dimensional topological model to a set of intervals. A set of intervals induced by our example endpoint sequence is illustrated in Fig. 10. This set encodes information about which wires overlap. It is possible for a wire to overlap another wire in the interval representation while not intersecting it in the topological model. The converse is not true — if a wire intersects another wire in the topological model, it must also overlap it in the interval representation.

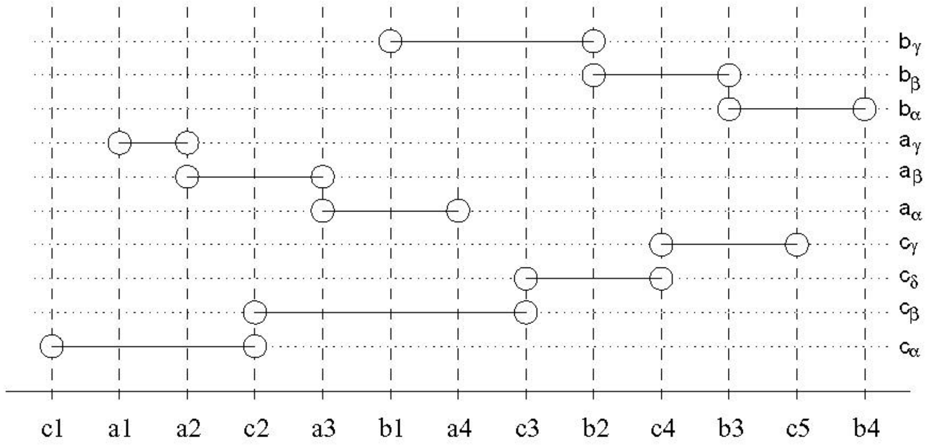


Figure 10: Set of Intervals

2.2.2 Layer Assignment

There are two constraints on assigning the wires to layers: if two wires intersect in the two dimensional topological model (Fig. 6), then the wires must be assigned to different layers; and if an endpoint of a wire corresponds to a terminal, then the wire must be assigned to that terminal's layer (all terminals have a defined layer). These constraints are encoded in the *layer assignment constraint graph.* Any two nodes connected with an edge cannot be assigned to the same layer. The layer assignment constraint graph for the running example is shown in Fig. 11.

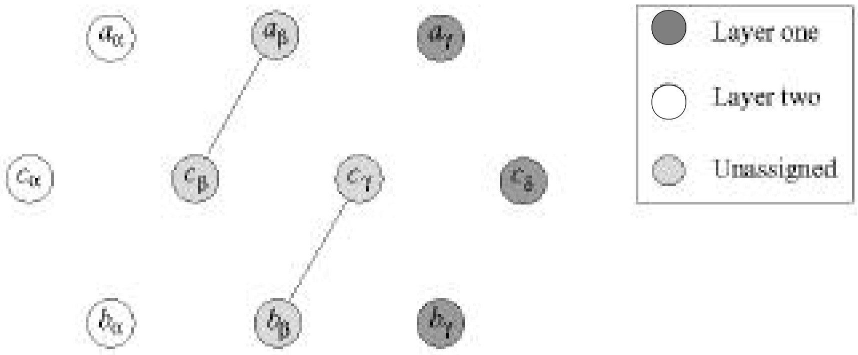


Figure 11: Layer Assignment Constraint Graph

The assignment of wires to layers is equivalent to "colouring" the layer assignment constraint graph for the problem. Each colour corresponds to a routing layer. A specific layer is assigned to a node if its wire is attached to a terminal. Other wires are free to be assigned to any layer subject to the constraints from intersecting wires. For example, nodes a_β and c_β are connected by an edge requiring that they be assigned different layers. In Fig. 11 a colouring is possible with two colours; i.e. the channel is routable in two-layer technology. An assignment of layers to wires in the running example is a_γ, b_γ, c_δ, $c\gamma$, a_β on layer one, a_α, b_α, b_β, c_α, c_β on layer two.

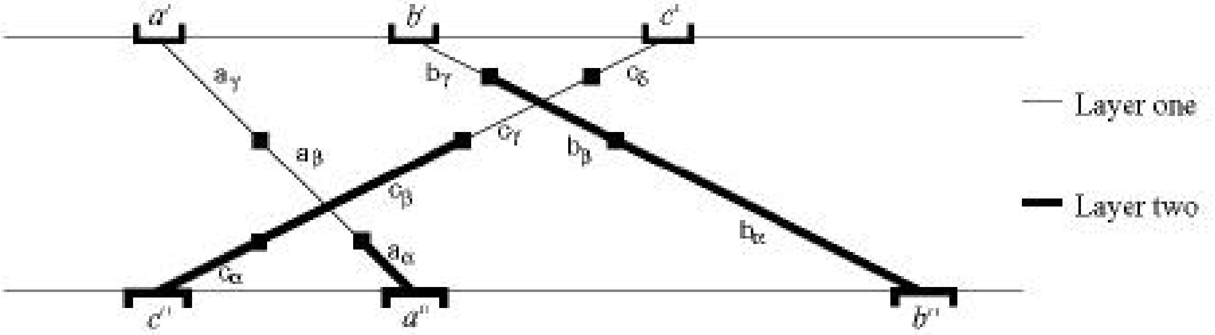


Figure 12: Topological Layer Assignment

Figure 12 shows the effect of this layer assignment on the topological model (solid cubes show the separation of wires from the same net). This assignment of layers to wires results in the division of the interval set into one subset for each layer. Figure 13 shows this division. Overlapping intervals must be on the same layer (in the same subset). Before the division of the interval set, all intersecting wires overlap. After the division, all intersecting wires are assigned to different layers and thus cannot overlap. This assignment of wires to layers satisfies the layer assignment constraints. For example,in Fig. 10 wire intervals a_β and c_β overlap (and are connected by an edge in the layer assignment constraint graph, Fig. 11). When the layer assignment is complete, they are assigned to different layers and thus cannot overlap (Fig. 13).

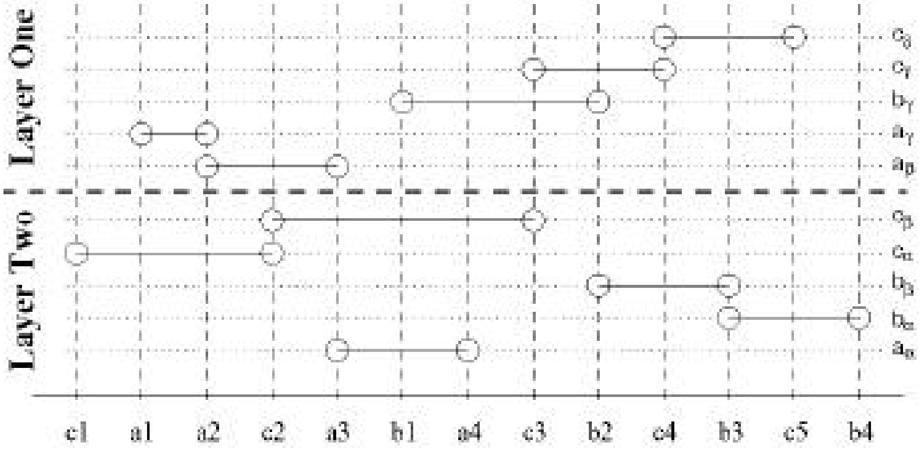


Figure 13: Wire Intervals by Layers

2.2.3 Optimization

The quality of a solution to the topological definition subproblem is determined by two factors: the number of vias and the height of the channel. The number of vias is determined by the assignment of wires to layers and is the sum of all the layer transitions of all nets. The channel height, on the other hand, is dependent on the algorithm used for physical placement. Since our routing technique separates physical placement from topological definition, a heuristic measure for channel height is required. The chosen heuristic is the *minimum possible channel height* (MPCH). It is dependent on the assignment of wires to layers and on the sequence of the wire endpoints, but not on the physical placement scheme used; it is the minimum height in which the physical routing is theoretically possible regardless of the employed algorithm. A given physical routing algorithm may require more height than this minimum, but it cannot use less.

In Fig. 13, for an interval where two wires on the same layer overlap, the trunks of the wires must have different vertical locations. The minimum vertical space required to route two such wires is the sum of their widths. This concept can be extended to sets of mutually overlapping wires by an *overlap graph*. This is a graph with nodes representing wires and an edge between two nodes if the corresponding wires are on the same layer and overlap. A set of mutually overlapping wires will have edges from each node in the set to all other nodes in the set; i.e., all wires in the set are on the same layer and every wire in the set overlaps every other wire in the set. The set of nodes forms a *clique*. The required vertical space for a given clique is the sum of the widths of all the wires (nodes) in the clique.

The MPCH must be large enough to accommodate the vertical space required by any clique from the wires' overlap graph, and in particular, the clique with the largest vertical space requirement. Thus, the MPCH is the vertical space required by the clique whose sum of widths is maximal in the graph [Golu80]. Figure 14 shows the overlap graph for the solution to the running example in Fig. 13.

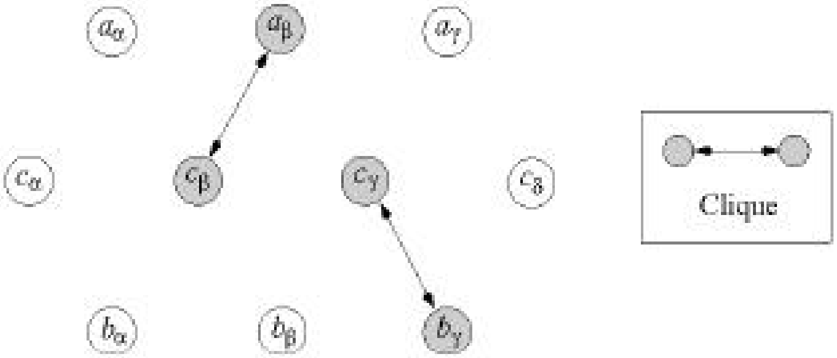


Figure 14: Overlap Graph

3. The CSP Problem

A solution to the topological definition subproblem involves satisfaction of constraints. For instance, when the two-dimensional model is decomposed into a set of intervals, the wire endpoints for the intervals have a fully defined order which satisfies the endpoint constraints. Each interval is assigned to a layer, with this assignment satisfying the layer assignment constraints. Further, solutions to the subproblem are minimized according to an objective function (MPCH and via count). Therefore, the topological definition problem can be cast as a constraint-satisfaction problem, and solved as such.

The topological definition problem can be solved as two subcomponents, where each subcomponent is equivalent to a canonical problem (topological sort and graph colouring). The latter problems are well-studied in the CSP and CLP literature, so their solutions are only briefly described here. Detail on the constraints and algorithms used is provided elsewhere [Hugh98].

The problem posed in Subsection 2.2.1 is how to search for endpoint sequences consistent with the endpoint constraint graph. This problem is equivalent to generating different permutations of a topological sort of the graph [Golu80]. An efficient method of incrementally generating these permutations is to maintain a candidate set from which the next endpoint in the sequence is chosen.

The choice can be randomized or follow a static criterion. The candidate set and the constraint graph are modified when an endpoint is selected. The aspect of choice introduces non-determinism; different endpoint sequences are generated by choosing different nodes from the candidate set. This approach is easily formulated as a constraint-satisfaction algorithm. The non-determinism is readily captured using backtracking, either directly or in the underlying constraint solver. Experimentation [Hugh98] indicates that a randomized selection of next endpoint (from the candidate set) is effective and efficient.

The layer assignment problem is conceptually simpler than the endpoint sequence search. The only constraint in it is that intersecting pairs of wires must not be assigned to the same layer. The problem is an instance of the well-known *map colouring problem* [Golu80]. As a CSP, map-colouring has a straightforward solution: the algorithm imposes the constraints and then proceeds through the list of nodes (i.e. wires) assigning colours (layers) to them. The constraints ensure that no pair of intersecting nodes connected by an edge (wires) will be assigned the same colour (layer). Again there is nondeterminism present, in that a choice of colour (layer) is made from a set of possible choices. A randomized choice once more yields good results [Hugh98].

3.1 Search Control

A constraint-based solution to the topological definition subproblem incorporates the above endpoint sequence and layer assignment searches. These two search problems are independent in that either can be solved before the other. It is only necessary that both be solved for construction of the overlap graph and final calculation of the objective function (Section 2.2.3). The two combined searches must be subject to some overall control regime, however. A branch-and-bound technique, which eliminates unproductive branches, was chosen.

Even though the two search components can be solved in either order without affecting correctness, the order does have strong implications for efficiency. This is consistent with other such CSPs. Two main variants exist: solving the search problems sequentially, or interleaving their solution. In the latter case, choices (of layer and next endpoint) are interleaved. For instance, the layers for an intersecting wire pair can be chosen as soon as the right-most endpoint for both wires in the pair have been added to the endpoint sequence. To illustrate, consider wires c_β and a_β in Fig. 7(a). As soon as the right-most endpoint for both wires have been added to the endpoint sequence (i.e. both *c3* and *a3*), layers can be chosen for the wires. Such an interleaved strategy is followed in our solution.

Interleaving the assignment of layers with the endpoint choice allows the branch-and-bound control regime to more effectively prune branches of the search tree [Hugh98]. The interleaving exposes information about the minimum possible height of the channel to the left of the “current” endpoint (in the determination of the endpoint sequence) as soon as the endpoint is added to the sequence. Such information allows the pruning of the search as soon as it can be determined that no layer assignment can produce a channel height (on the left) less than the best found so far, and searching the remaining right side of the channel is futile.

3.2 Constraints Used

The constraint system capturing the search problems and objective function is a combination of boolean and algebraic constraints. This constraint network expresses relationships among the endpoints, wires, and layers for wires. Constraints encode the layer assignment graph, endpoint constraint graph, and overlap graph. Constraints are also used in the calculation of the objective function, e.g. in calculating the maximal clique in the overlap graph. Some of the constraints are imposed prior to beginning the search; e.g. the constraints encoding the partial orders from the endpoint constraint graph. Others are added dynamically to the constraint store as the search is being conducted; e.g. constraints representing an edge in the overlap graph imposed when an endpoint is added to the endpoint sequence (as the endpoint search proceeds).

3.3 Modified Objective Function

The objective function is complex, involving two different measures: number of vias and minimum possible channel height. To simplify solution of the topological definition problem, minimization of the two measures was decoupled. That is, the two measures were minimized consecutively instead of in combination. Of the two, minimum possible channel height is the more important [Sher93]. It is also the more complex, both conceptually and in terms of representation and calculation. Therefore, first a solution is found which achieves a minimum possible channel height. Subsequently, the solution is modified to minimize the number of vias used under the restraint that the MPCH not increase. Investigation of an objective function which simultaneously minimizes both measures is left to future work (Section 5).

3.4 Implementation

The solution to the topological definition subproblem described above was implemented in the CLP system ECL^{i}PS^{e} [Wal+97] using its finite domain and generalized propagation constraint libraries. The three main phases of the program are: an initial phase which reads in a channel routing problem and produces a constraint representation of the topological problem; the *topological solver* which performs the endpoint sequence and layer assignment searches (solutions optimized for MPCH only); and the *via minimizer* which takes a complete solution from the topological solver and permutes the layer assignment to reduce the number of vias. The via minimizer follows a simulated annealing approach. To ensure that the height of the channel not increase, any modified layer assignment that would result in an increase is immediately discarded. The via minimizer is implemented as a C module called from ECL^{i}PS^{e}.

4. Other Work

An extensive amount of additional work [Hugh98] has been performed as part of this research, including the following:

1. A variety of strategies for combining the layer assignment and endpoint search problems were implemented and their performance evaluated. Within each search component, different selection strategies were also explored.
2. A nondeterministic, heuristic layer assignment technique (replacing the nondeterministic one previously described) was developed and evaluated.
3. The performance of the topological solver was compared to a series of benchmark channel routers. “Quality” of solutions under different conditions (e.g. number of layers, channel congestion) were examined.
4. A complete router following the principles espoused above was implemented and evaluated. The complete router required implementation of a physical placement component to transform the abstract wires and endpoints from the topological solver and via minimizer into vias, doglegs and trunks. An existing, well-known channel routing technique was adapted as the physical placement solver.

The evaluations were conducted using large numbers of synthetic problems whose relevant parameters (e.g. congestion profile) were consistent with industrial problems. Standard benchmark problems (e.g. the Deutsch Difficult Example [Deut85]) were also used. Results of the evaluations led to the following main conclusions:

a) The topological router does a good job of solving difficult channel routing problems, often only slightly worse than a perfect router[1]. It achieves a systematic reduction in

[1] A *perfect channel router* makes no concessions to any routing constraints whatsoever and represents the minimum possible channel height which any channel router can achieve. The perfect router is an abstract concept and cannot be realized, but does serve as a useful lower bound on channel height.

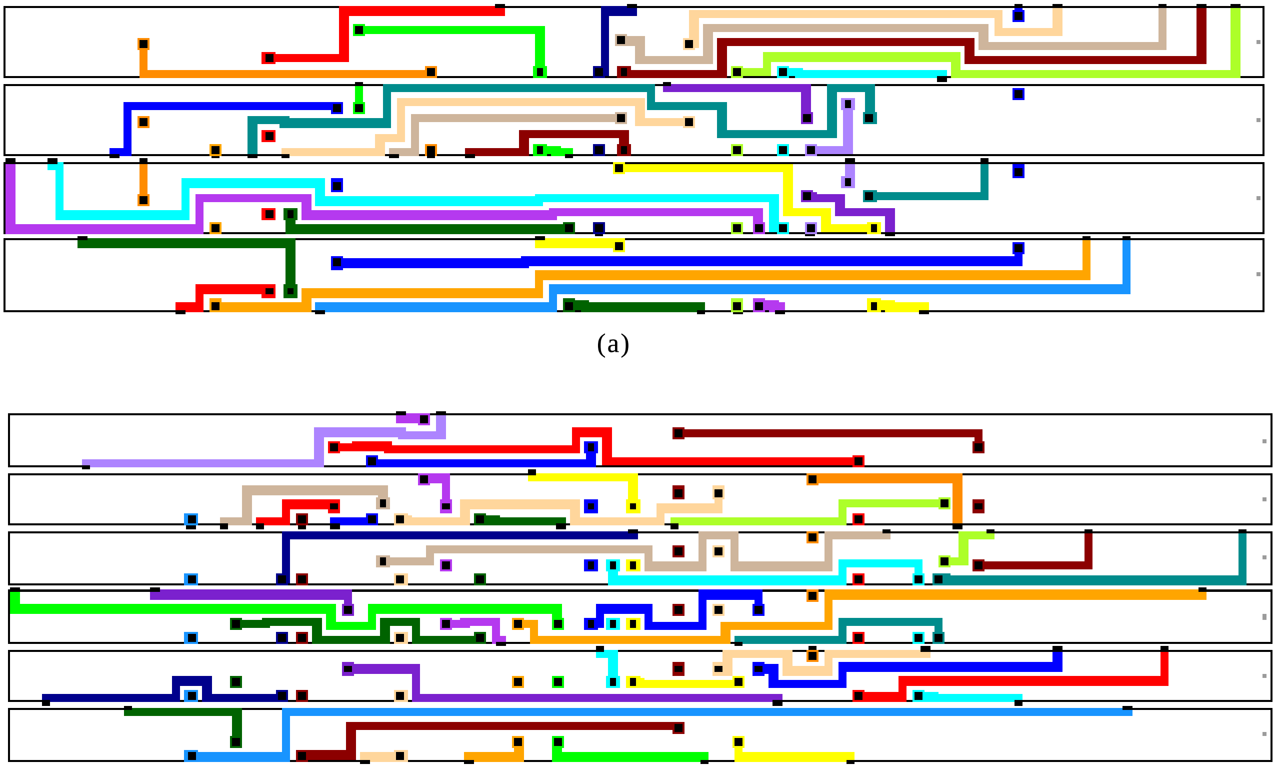
average channel height with increasing number of layers, thus demonstrating that it is effective at exploiting multiple layers to reduce the channel height.

b) The via minimizer incorporated into the design is effective in reducing the via counts required, often approaching the theoretical minimum.

c) Constraint satisfaction problem and the branch-and-bound control strategy were effective in pruning the underlying search trees. It was possible to achieve provably optimal results on problems for which an exhaustive search of the solution space is totally unfeasible.

d) The successful operation of the complete router verified that the decomposition of the channel routing problem into two subproblems can, in fact, lead to a practical router.

To illustrate the operation of the topological channel router, the solutions (complete routed channels) for two test problems are shown in Figures 15(a) and (b). The two channels have the same number of nets and the same congestion, and are drawn to the same scale so the resulting channel heights are comparable. However, 4 layers are available in (a), while 6 are in (b). It is apparent that the additional two layers available in Fig. 15(b) are exploited to shrink the channel height.



(a)

(b)

Figure 15: Complete routing example with 18 nets and congestion of 12. Figure (a) uses 4 layers, while (b) was given 6.

5. Summary and Contributions

This paper proposes a two-phase channel routing approach. The overall problem is divided into two distinct subproblems: topological definition and physical placement. The topological definition subproblem is solved by determining a topological model which divides the nets in the problem into wires, assigning these wires to layers, and specifying a one-dimensional order for the wire endpoints.

Solutions minimize the minimum required height for the channel and via count. These tasks are accomplished by casting them as constraint satisfaction problems and searching for solutions. The latter problems are equivalent to the standard constraint problems of map colouring and topological sort. Solution of these two problems can be interleaved and controlled by a branch-and-bound control strategy. The end result is a gridless, unreserved layer router which can successfully exploit any given number of routing layers. A successful implementation was achieved using the $\mathrm{ECL}^{i}\mathrm{PS}^{e}$ CLP system and C.

The work has demonstrated that CLP is an effective paradigm for solving difficult problems such as the topological definition subproblem, and channel routing in general, and that a difficult constraint satisfaction problem (the topological definition subproblem) can be decomposed into a combination of simpler, well-understood problems (topological sort and graph colouring).

A useful avenue for future work would be to relax the modification described in Section 3.3 and to have an objective function which directly combines both via count and minimum channel height. It would be useful to investigate how the two performance factors interact and influence the search.

References

[ChSa93] U. Choudhury and A. Sangiovanni-Vincentielli. “Constraint-based channel routing for analog and mixed analog/digital circuits.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits, 12* (2), April 1993.

[Cox+92] J. Cox, K. McAloon, and C. Tretko. “Computational complexity and constraint logic programming languages.” *Annals of mathematics and articial intelligence, 5* (2/4), 1992.

[Deut85] D. N. Deutsch. “Compacted channel routing.” In *Proceedings of the IEEE International Conference on Computer Aided Design*, pp. 223-225, 1985.

[Golu80] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

[Haru92] S. Haruyama. “Topological channel routing.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits, 11*(10):1177-1197, October 1992.

[Hei+92] N. Heintze, S. Michaylov, and P. Stuckey. “CLP(R) and some electrical engineering problems.” *Journal of Automated Reasoning, 9*:231-260, 1992.

[Hugh98] Andrew Hughes. “A Multi-layer Channel Router Implemented using a Topological Channel Routing Model.” M.Sc. thesis, Department of Computer Science, University of Saskatchewan, Saskatoon, Canada, July 1998. Available at http://www.cs.usask.ca/projects/ace/a_hughes/thesis.ps

[JaMa94] Joxan Jaffar and Michael J. Maher. “Constraint Logic Programming: A Survey.” *Journal of Logic Programming, 19 & 20*, 1994, pp. 503-581.

[LiTh93] J. Lienig and K. Thulasiraman. “A genetic algorithm for channel routing in VLSI circuits.” *Evolutionary Computation, 1* (4):293-311, 1993.

[RiFi82] R. Rivest and C. Fiduccia. “A greedy channel router.” In Proceedings of the 19th ACM/IEEE Design Automation Conference, pages 418-424, 1982.

[Sher93] Naveed Sherwani. *Algorithms for VLSI Physical Design Automation.* Kluwer Academic Publishers, Assinipi Park, USA, 1993.

[Simo90] H. Simonis. *Channel Routing Seen as a Constraint Problem*. Tech. report TR-LP-51, ECRC, Munich, Germany, July, 1990.

[Suz+95] K. Suzuki, H. Amano, and Y. Takefuji. “Neural network parallel computing for multi-layer channel routing problems.” *Neurocomputing, 8* (2), July 1995.

[Wal+97] M. Wallace, S. Novello, J. Schimpf. “ECLiPSe: A Platform for Constraint Logic Programming.” Technical report, IC-Parc, Imperial College, London, U.K., August 1997.

[Wilk96] A. Wilk. “Constraint Satisfaction Channel Routing.” Tech. report CS96-04, Computer Science, Weizmann Institute, Israel, March 1996.

[Wolf94] Wayne Wolf. *Modern VLSI Design: a systems approach*. Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[Zhou96] Neng-Fa Zhou. “Channel Routing with Constraint Logic Programming and Delay.” *Proc. 9th IEA-AIE*, pp. 383-388, 1996.