

VYUHA : A Detailed Router For Multiple Routing Models

BY

C.P. Ravikumar and S. Sastry

Technical Report 90-17

May 1990

Electrical Engineering - Systems Department

University of Southern California

Los Angeles, CA. 90089-0781

VYUHA : A Detailed Router For Multiple Routing Models

C.P. Ravikumar and S. Sastry
Department of Electrical Engineering Systems
University of Southern California
Los Angeles, CA 90089

May 26, 1990

Abstract

This paper presents a general-purpose tool called VYUHA for detailed routing. Unlike many of the existing programs for routing, the routing *model* is accepted as an input. Our router permits horizontal, vertical, 45° and 135° wires. The user can turn off the use of any of the four types of wires. VYUHA is capable of routing in multiple layers. Unlike the conventional Manhattan model, a wiring layer is not associated with a particular direction. This means that all the three wiring slopes are possible in the all layers. The user can either explicitly specify the layer for a pin or request the router perform the assignment. The routing region is assumed to be rectangular, although the algorithm can be trivially modified to overcome this restriction. Pins can be present on any of the four sides of the rectangle. Thus both channel routing and switchbox routing are possible. Obstacles, if any, can be specified in the routing region. Two-point as well as multipoint nets are handled by the algorithm. The "multiple layer and multiple pin" problem which arises in 3-D chips is easy to handle using our router. The algorithm was implemented in Pascal to run on a Sun 3 workstation.

1 Preliminaries

It is customary to perform circuit wiring in two stages. A *global router* is first used to assign nets to rectilinear wiring regions known as channels and switchboxes. A detailed router is then used to wire individual regions. The term "detailed routing" refers to the actual assignment of wiring tracks to nets such that design rules are satisfied. The track

assignment is influenced by the *routing model*, which is characterized by the following considerations.

1. Shape of the wiring region : After the placement of modules, the routing region formed between the modules is a rectilinear polygon. To simplify the routing problem, it is customary to break the polygon into rectangular regions known as *channels* and *switch boxes*.
2. Pin topology : A channel, or a "street", is a rectangular wiring region in which pins are placed only on the top and bottom sides. A switch box is a rectangle in which terminals are allowed on all the four sides. A switch box is typically formed at the intersection of two streets ; the "pins" on the left and right hand sides of a switch box are actually wires that enter from the intersecting channels.
3. Number of layers : Depending on the technology, one or more metal layers are permitted for routing. If an integrated circuit being routed, the polysilicon layer may also be used for routing short wires. Some authors refer to a poly layer as a " $\frac{1}{2}$ layer", since it is shared between cells and interconnect.
4. Wire Slopes : In most of the existing literature on routing, it is assumed that wires run in the horizontal or the vertical direction. More recently, routing models with diagonal wires (45° and 135° slopes) have been proposed.
5. Slope/Layer Binding : For simplicity of the routing algorithm, it may be desirable to restrict wires of a single slope to a layer. For example, the Manhattan model uses two layers and permits horizontal wires in one layer and vertical layers in the other layer. The diagonal routing model proposed by Lodi and associates [5] uses two layers, restricting 45° lines to layer 1 and 135° lines to layer 2. When the Manhattan model is extended to three layers, we get VHV and HVH routing models. The VHV model, for example, allows vertical wiring in two layers and horizontal wiring in the middle layer. The multilayer router presented in [6] allows mixed mode routing i.e. both horizontal and vertical wires on the same layer.
6. Pin/Layer Binding : A routing model may or may not restrict the layer in which the pins are placed. In the Manhattan model, the pins must be placed in the "vertical layer". In a less strict model, all the layers are permitted to carry pins. Such a model is necessary to capture the routing problem in 3-D chips, where active devices are stacked on top of each other. In other words, at the same grid point pins belonging to different signals can coexist in different layers. Enbody and Du refer to this problem as "Multiple layer, Multiple pin" (M & M) routing problem.

Existing detailed routers have been developed for some specific routing model or the other. For example, Yoshimura and Kuh used two layers and the Manhattan wiring model [18]. In their model, horizontal wires are placed on layer 1, vertical layers on layer 2, and the pins on layer 2. This routing style is adopted by many other routers such as BEAVER [4]. Chen and Liu extended the Manhattan model for three layers by introducing the VHV and HVH routing styles. In the VHV style of routing, vertical wire segments are placed on layers 1 and 3, whereas horizontal segments are placed on layer 2 [3]. The HVH routing style is defined similarly. Enbody and Du used $n \geq 2$ layers for routing [6]. They relaxed the slope/layer assumption of Manhattan models by allowing both horizontal and vertical segments in all layers. Furthermore, they also allowed 45° wires between two adjacent columns. Lodi and associates introduced a two-layer channel routing algorithm which exclusively using 45° degree and 135° wires [5]. They restricted 45° degree routes to layer 1 and 135° degree routes to layer 2.

The contribution of this paper is that it describes a routing algorithm which operates independently of the routing model. It can handle any or all of the following features.

- Arbitrary shapes for routing region, including channels and switchboxes
- One, two, or multilayer routing
- Optional use of 45° and 135° wire slopes
- Optional use of horizontal and vertical wire slopes
- Stacked or unstacked terminals
- Specification of obstacles to routes
- Specification of priorities for specific nets (e.g. power and ground nets)
- User specified pin/layer binding
- Unrestricted slope/layer binding
- Multiple layer/Multiple pin routing
- Short wire lengths and minimal number of vias.

The next section will discuss the routing algorithm. The section that follows it will present an implementation of the algorithm along with some results. Conclusions and extensions are discussed in the final section.

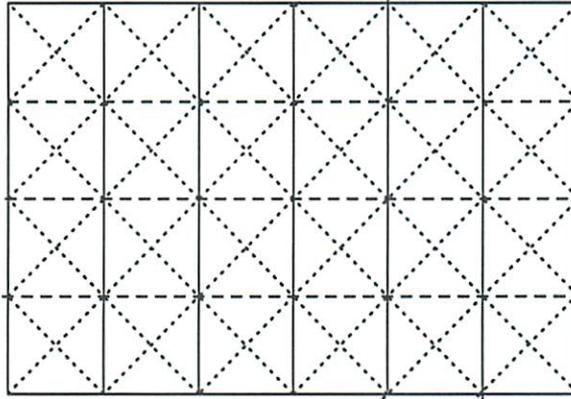


Figure 1: Routing Grid in a 4×7 Rectangular Switchbox

2 A Model-Independent Routing Algorithm

It is convenient to superimpose a grid structure on the wiring region. Wires are only allowed along the lines of the grid. Therefore, if the grid spacing is chosen appropriately, the “minimum separation” constraint between two wires is automatically satisfied. When there are multiple layers, the grid is simply duplicated as many times as there are layers. Note that it is always possible to draw such a grid on a rectilinear region, irrespective of the shape of the wiring region. Figure 1 shows a rectangular switchbox with 7 columns and 4 rows along with a grid consisting of horizontal, vertical and 45° lines.

The inputs to a detailed router are a netlist, a specification of the wiring region and the routing model. The output of the router is a geometric realization of each net, under the restrictions posed by the wiring model. In addition, the router attempts to optimize the following objective functions :

1. Since VLSI real estate is expensive, most routers attempt to minimize the area of the routing region.
2. Reducing the length of wires minimizes signal delays.
3. Reducing the number of via holes improves the reliability of the chip and also reduces the resistance of signal paths.

The complexity of detailed routing depends on the routing model. Some specific routing models have been studied in the literature. In the two-layer Manhattan model, channel routing for minimization of channel height is known to be NP-complete [10]. Lodi and associates have examined a *diagonal routing model* for two-layer channel routing. This model allows only 45° and 135° wires – 45° lines on layer 1 and 135° lines on layer 2. It is unknown if the diagonal routing problem is NP-complete or otherwise. In this paper, we

shall allow the most liberal routing model which we shall refer to as *mixed-slope routing*; this means all the four routing slopes are allowed – horizontal, vertical, 45° and 135° on all the layers. Mixed-slope routing is a hybrid between Manhattan routing and diagonal routing. It offers a great deal of flexibility to a router. To establish the complexity of mixed-slope routing is an open research problem. This paper presents a heuristic algorithm for routing in the mixed-slope model.

Our router, VYUHA, is based on Lee's maze routing algorithm [12]. The original algorithm due to Lee finds a shortest-path algorithm between two points in a maze, if such a path exists. The algorithm has received widespread acceptance by the design automation community. It has been used by a number of existing detailed routers [1, 15, 16]. The original algorithm has been extended by many authors to include a number of features such as multiple layers and multiple-pin nets [1]. Many coding techniques have been invented to improve the performance of Lee routing [1, 15]. Hardware accelerators have been built for efficient execution of Lee-based routing algorithms [2, 9, 16, 17]. Before hierarchical routing techniques were developed, Lee's algorithm was the only available detailed router. The importance of the algorithm has been somewhat shadowed by the arrival of global routing techniques. A global router plans the routing and subdivides the netlist by allocating particular nets to specific routing regions. The smaller routing regions (channels and switchboxes) are handled by special-purpose routing algorithms. These algorithms use heuristics to optimize routing objectives. All the same, they do not guarantee 100% routability. If a channel router or a switchbox router leaves some nets unrouted, Lee's algorithm is used as the last resort to complete the routing. Since Lee's algorithm uses a brute-force search, it is guaranteed to find a path for a net, if such a path exists. When Lee's algorithm fails to route a net i , it either means that local congestion of nets has caused a blockage or that the netlist is unroutable in the given area. There is no easy way to conclude which is the case. The only resort under this circumstance is to assume routability, rip up one or more nets and reroute them after the net i has been completed. Rip-up and reroute techniques can be manual or automatic [16].

2.1 Achieving Model-Independence

In this section, we demonstrate that Lee's algorithm can be readily extended for almost every conceivable routing model. To start with, we examine how diagonal slopes can be allowed. In the conventional Lee router, a diamond-shaped wavefront is sent out from the source in the outward direction until the sink is caught by the wavefront. The reason for the diamond shape of the wavefront is that it expands in four orthogonal directions – north, south, west, and east. In order to include diagonal routes, the wavefront must consider four additional directions, namely north-east, north-west, south-east and south-west. If multiple-layer routing is permitted, then the top and bottom directions must also be taken into account. A simple precaution must be exercised when expanding the wavefront in

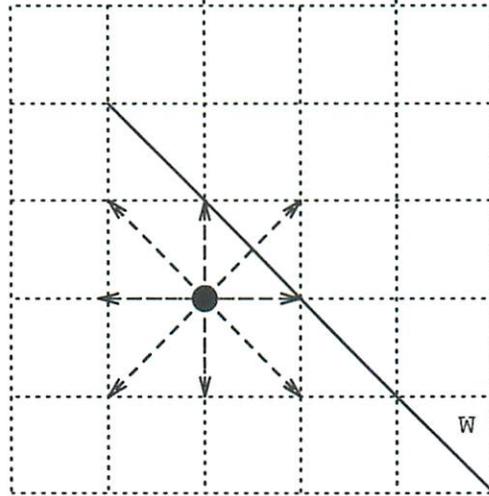


Figure 2: Avoiding wire crossing.

diagonal directions. This is shown in Figure 2. W represents an existing wire. The shaded circle represents the grid point at which expansion is being carried out. The wire W poses an obstacle to the wavefront in the north direction, which is easily detected since the north point is marked “blocked”. The wire W also poses an obstacle in the north-east direction although the north-east point is not blocked. The diagonal expansion must check for the possibility of crossing wires.

Wiring Slopes

The mixed-slope model is the most general routing model. Given this model, it is easy to implement a more restricted form of routing model such as diagonal routing or Manhattan routing. The wavefront expansion is carried out conditionally in the eight different directions. It is also easy to implement a restricted slope/layer binding. For this purpose, an 8-bit mask is associated with every layer. The eight bits correspond to eight different directions. The j th bit of a mask is set to 1 (or 0) if routing in direction j is allowed (or not allowed). This mask is used by the wire expansion phase to curtail (if need be) the growth of the wavefront in a particular direction in a particular layer.

Irregular Boundaries

Conventional Lee routers assume a rectangular wiring region. It is straightforward to extend the algorithm for regions with irregular boundaries. Let B be a bounding rectangle which covers the entire wiring region. The maze data structure is a three-dimensional array of size $L \times C \times R$, where L is the number of layers, C is the number of columns in B and R is the number of rows in B . Each entry in the maze array is a record with

several fields, such as the "blocked flag", "visited flag" and "path number". The "blocked" flag is set to *true* if (and only if) the corresponding grid point is occupied by an obstacle. The "visited" flag is marked if the grid point is free but has been already included in the wavefront. The path number indicates the name of the wire which passes across the grid point. To implement an irregular wiring region, all but the cells which are included in the wiring region are marked "blocked". This technique does waste storage, but it makes the coding elegant and easy.

Pin/Layer Binding

Although not dictated by the routing algorithm, it is assumed that pins are at the boundaries of the wiring region. One of the fields in the maze record is a flag which indicates if the grid point corresponds to a pin. If this flag is true, the "path number" field indicates the net to which the pin belongs. If the user specifies a pin/layer binding then a pin is completely defined (layer, x -coordinate, y -coordinate) and the corresponding grid point is marked as a pin. The user may let the algorithm decide the pin/layer binding. In this case, a pin is incompletely specified (x -coordinate, y -coordinate); all the grid points which satisfy these coordinates are marked as pins. Therefore the wave expansion phase will detect any one of these grid points as a sink. The expansion phase is also modified to place all the pin points in the wavefront before the wavefront is expanded.

Multiple Layer, Multiple Pins

The term M&M routing was introduced by Enbody and Du [6] to refer to routing in 3-D chips. In this type of routing, multiple pins may be present at grid points of the form (z, x, y) where x and y are fixed and z (layer) is a variable. No special modification is required to VYUHA to handle M & M routing. It only requires a modification of input data to specify multiple pins at appropriate grid points.

Multi-pin Nets

While the original Lee-router [12] was intended for a pair of points, the algorithm can be extended for connecting multiple points. There are many variations possible, but the aim of the extension is to construct a Steiner tree of small length to connect all the points. We used a "multiple sink, single source" algorithm for handling k -point nets, $k > 2$. Two points A and B are selected as initiator points and a connection is established between them. All the grid points along the path from A to B are then marked with a special indicator. A wavefront is then initiated from each of the $k - 2$ remaining points of the net. The wavefront expansion from a point p stops only if one of the following conditions is fulfilled.

1. a grid point marked with the indicator is found
2. the wavefront is empty.

In the latter case, no path exists to connect p to the rest of the pins. In the former case, the path from p is also marked with the indicator. The name “multiple-sink, single source” is derived from the analogy that wavefront expansion always begins at a single point but may terminate at any of the indicated points.

The above algorithm for routing multiple-pin nets is guaranteed to find a Steiner tree if one exists. However, it may or may not be a tree of minimum length. Two factors will affect minimality – the choice of the two initiator points and the order in which the remaining $k - 2$ points are selected for routing.

3 Order of Routing

Authorouters based on heuristics do not guarantee 100% routing. The routability problem is more pronounced in Lee-based routers since they lack a global view of the problem. Nets are routed one by one and it may happen that a previously routed net makes another one unroutable by blocking it. A global view can be provided to some extent by ordering the nets for routing. In addition, the path selected for a net must affect the remaining unrouted nets as little as possible. In our experience, the ordering of the nets is found crucial to achieve 100% routing.

In the current implementation, we support four methods for determining the routing order. The first method is the simplest of all, namely random ordering. When tested on benchmarks, random ordering performed quite poorly in terms of routability. The second method is known as “minimum span first” ordering. The *span* of a net is defined as the semiperimeter of the bounding rectangle. Nets which have small spans are routed first. The third method routes nets with maximum span first. Almost as a rule, the “minimum span first” rule outperformed the “maximum span first” heuristic in terms of routability. The fourth ordering method is user-specified ordering.

If an automatic ordering method fails to complete routing, VYUHA resorts to rip-up and reroute techniques [16]. We are currently examining better routing orders which are based on graph theoretic principles. For each incomplete net i , the router computes the set $B(i)$ of the nets which block i . For each net $j \in B(i)$, the router also computes the number of blockages b_{ij} caused by j while routing net i . $B(i)$ and b_{ij} are easily computed during the wave expansion phase. The routing of net i is guaranteed to be completed if it is routed before all the nets which block i . But such a strategy may render many of the nets in $B(i)$ unroutable. The scheme which we have observed to be effective is to pick the net $j \in B(i)$ such that b_{ij} is maximum and then interchange the order of nets j and i .

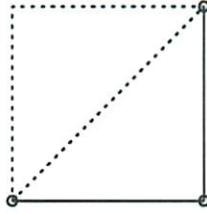


Figure 3: Advantage of diagonal paths.

4 Results

The routing algorithm described in the previous section was implemented in Pascal to run on a Sun 3 workstation. The source has about 2000 lines of code including comments. The program was tested on some of the benchmarks available in the literature. This section will discuss the results on some benchmarks. The program was especially helpful in understanding the advantages and disadvantages of 45° routing. Some of the observations are summarized below.

Allowing 45° degree routing can help in reducing the total wire length significantly. A diagonal line between two opposite points of a square grid occupies about 30% less wiring when compared to Manhattan routing (see Fig 3). The diagonal line AB occupies a single edge and two grid points, unlike the Manhattan path ACB, which takes up two edges and three grid points. Since Lee's algorithm attempts to minimize the number of grid points and edges consumed by a path, it will prefer diagonal routes whenever possible. Therefore, it is possible to conclude that the total length of a path in mixed mode routing is less than or equal to the total length used by Manhattan routing. On a similar note, it can be seen that Lee's algorithm minimizes the number of via holes used to construct a path. In Lee's algorithm, a via hole amounts to two grid points of the form (x, y, z) and $(x, y, z \pm 1)$. If there exists a shorter path on the same layer, the algorithm will never make a layer change.

When compared to the diagonal routing model proposed by Lodi and Associates, the mixed-slope routing model offers many advantages. A serious limitation of the diagonal routing model is that it allows layer changes to be made at non-grid points. This happens if the length of the Manhattan path from source to destination is an odd number of units. See Figure 4(a). A moment's reflection will show that there is no path from point X to Y which uses only diagonal edges of the grid. When layer changes occur at non-grid points, design rules can be violated. The mixed-slope routing model avoids this problem by using diagonal edges as much as possible and then using a short vertical or horizontal stub to complete the connection. Figure 4(b) illustrates the point.

Mixed-slope routing can pay off significantly when it comes to minimization of routing area. Figure 5 shows ¹ the wiring of Burstein's difficult channel with a density of 4. Using

¹In all the figures, the routing grid is shown in dotted lines for reference. Dark lines are used for layer

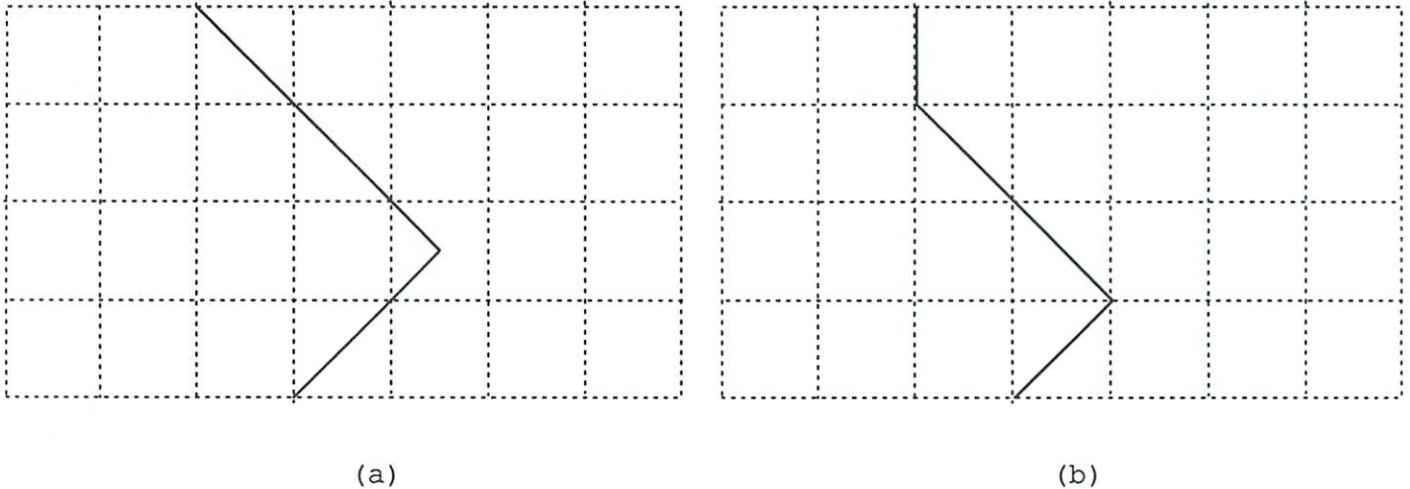


Figure 4: Problems with Diagonal routing.

mixed-slope routing, it took 4 tracks to complete the routing. The router SPYDER [6] required 5 tracks. Both solutions require two vias. Lee's algorithm required only 63 units of wire length. SPYDER required 104 units of wire. Recently, Gerrez and Herrmann have reported a router called PACKER which routes Burstein's difficult channel using 4 tracks, 10 vias and 82 units of wirelength [7].

The lower bound on the channel height is unknown in the mixed-slope routing model. Figure 6 shows a channel with density 3. SPYDER's solution of this channel requires 4 tracks. A two-track solution was obtained by Lee's algorithm. Figure 7 shows a benchmark from Yoshimura and Kuh's paper [18]. The original paper used 5 tracks, whereas VYUHA required only 4 tracks. The channel density of this example is 5. While [18] required 16 vias, our program uses only 2 vias. The wiring length of the original paper was 75, whereas in the present implementation it was 47. Figure 8 shows the solution to a benchmark problem provided by Joobbani [11]. WEAVER routed this channel using 7 tracks [11]. PACKER, a detailed router based on the concept of stepwise reshaping, solved this example using 6 tracks, 25 vias and a wire length of 167 units [7]. PACKER's solution is an improvement over other routers such as WEAVER and SILK. As can be seen in Figure 8, our solution is a further improvement; it uses 6 tracks, 13 vias and 114 units of wirelength. Further, while PACKER required 710 seconds to solve this problem, VYUHA required only 0.2 seconds of CPU time. The channel routing results of VYUHA are summarized in Table 1.

Mixed-slope routing has advantages when coupled with multiple-layer routing. Figure 9 shows the first benchmark from [3]. (This benchmark was first used for two layer channel routing by Yoshimura and Kuh in [18].) Using 3 layers, VYUHA completed the routing in

1 and dashed lines are used for layer 2. In examples which use 3 layers, the wires in layer 3 are shown using dotted lines and the grid is omitted.

Example	VYUHA	PACKER	SPYDER	Yoshimura	WEAVER
Burstein's channel	(4,2,63)	(4,10,82)	(5,2,104)		
Enbody and Du	(2,0,19)		(4,0,33)		
Yoshimura	(4,2,47)			(5,16,75)	
Joobbani	(6,13,114)	(6,25,167)			(7,27,137)

Table 1: Comparison of Two-layer Channel Routing Results. Entry (t, v, l) stands for $(tracks, vias, wirelength)$.

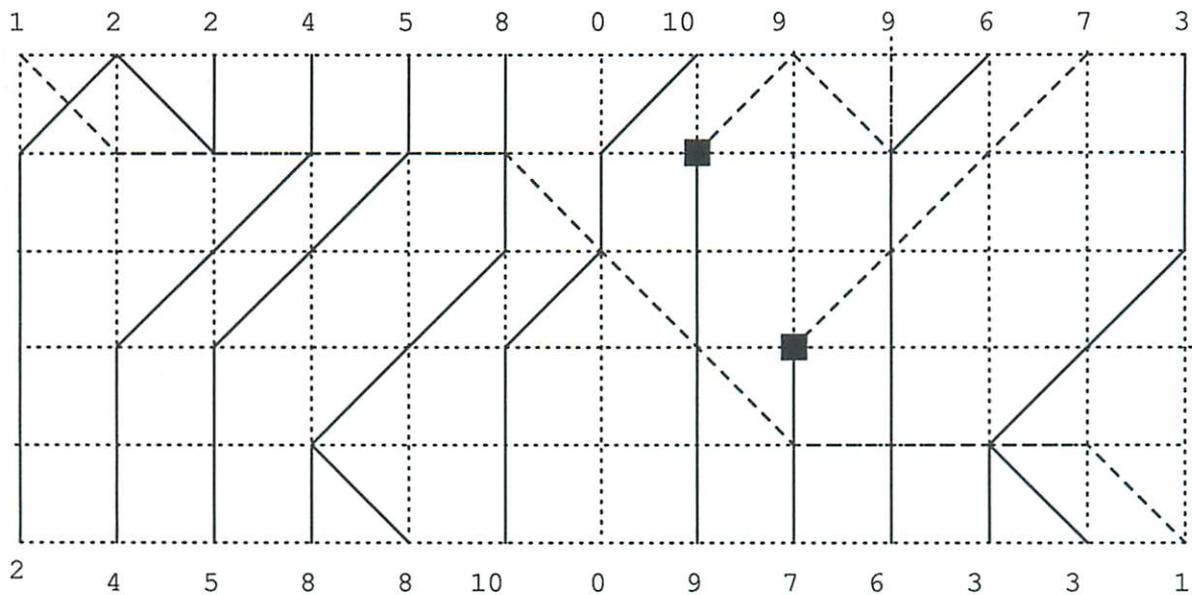


Figure 5: Burstein's Difficult Channel.

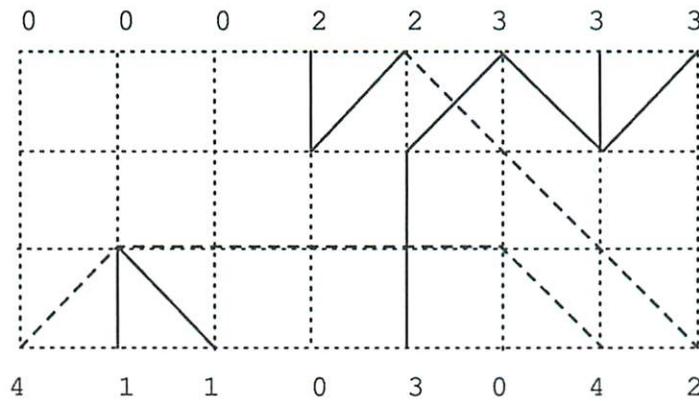


Figure 6: Example from Enbody and Du's paper.

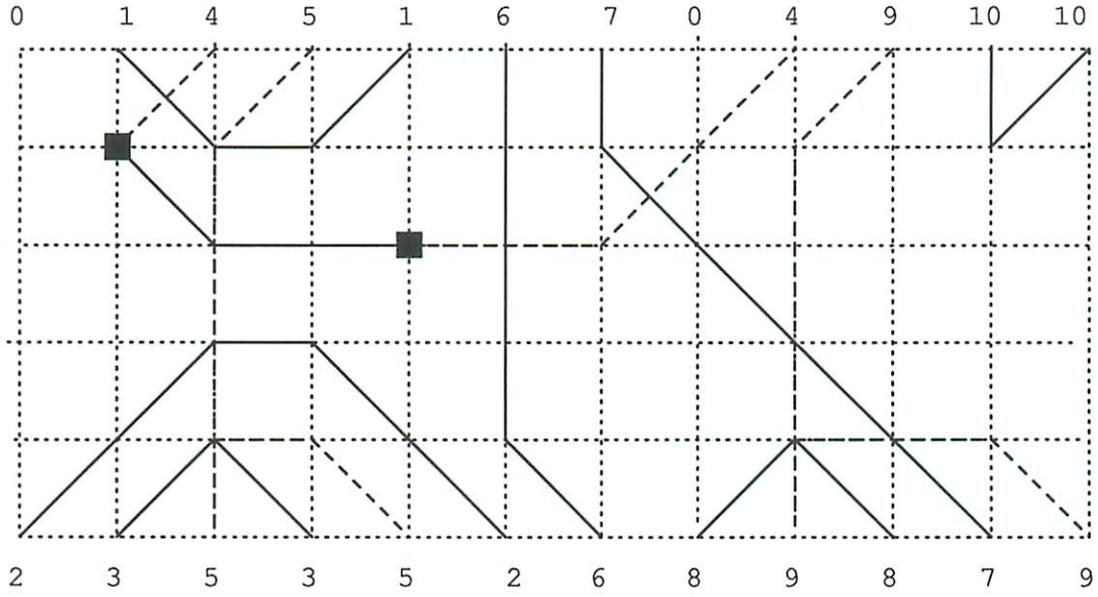


Figure 7: Example from Yoshimura and Kuh's paper.

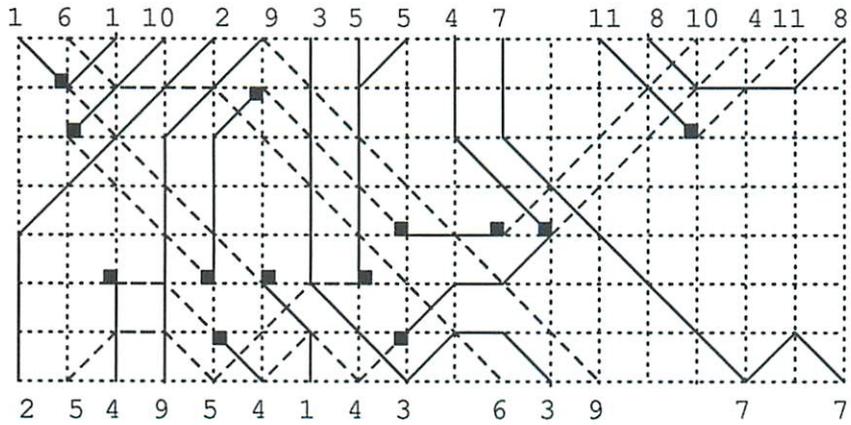


Figure 8: Jobbani's example.

Example	VYUHA	WEAVER	BEAVER	PLANNER	PACKER
Sample Switchbox	(5,45,64)	(4,60,81)	(3,60,81)		
Rectangular	(0,14,24)		(1,31,60)	(12,29,60)	
Thread	(5,46,36)		(6,65,36)		(8,59,36)

Table 2: Comparison of Two-layer Switchbox Routing results. Entry (v, l, a) stands for $(via, wirelength, area)$.

5 tracks. The solution in [3] used 7 tracks and 3 layers. When two layers were used, 11 tracks and 21 vias were sufficient. Yoshimura and Kuh’s router used 12 tracks and 54 vias.

Figure 10 shows a switchbox routed by VYUHA. The example is the “sample switchbox” which has been successfully routed by WEAVER [11], BEAVER [4] and MIGHTY [14]. BEAVER has, by far, the best solution which uses 3 vias and 60 units of wire length. Our solution uses 5 vias and 45 units of wire length. More importantly, our solution has a smaller switchbox area. Unlike WEAVER, MIGHTY and BEAVER, which use a 9×9 grid for the sample switchbox, VYUHA successfully completed the routing in an 8×8 grid. The second switchbox example is also drawn from [4]. Known as the simple rectangular switchbox, it was routed by PLANNER [8] using 12 vias and 29 units of wire. BEAVER improved the solution by using a single via and 31 units of wire. Our solution to the simple rectangular switchbox appears in Figure 11. As can be seen, it uses no vias and only 14 units of wire. Further, our solution uses much less grid space. Both PLANNER and BEAVER routed the switchbox in a grid of dimension 6×10 . Our solution occupies an area of 4×6 . The “thread” switchbox (example taken from [7]) required 8 vias and a wire length of 59 using the router PACKER [7]. The routing for this example is shown in 12; it can be seen that our solution has only 5 vias and a routing length of 46. These results are summarized in Table 2.

On some classic problems, such as shift-by- k permutations and the perfect shuffle permutation, significant improvements are possible when mixed-slope routing is allowed. For example, Figure 13 shows the shift-by-1 permutation routed using a single layer and one track. Using the diagonal routing model of Lodi et al, a shift-by- k permutation requires a channel height of k . However, as noted before, the diagonal routes change directions at non-grid points. In the mixed-slope routing model, a shift-by- k permutation can be realized in k tracks and one layer. Each route is realized using a short vertical stub of unit length and a diagonal path of length k . Figure 14(a) shows the routing of the 16×16 perfect shuffle interconnection. The density of this channel is 8. Four tracks and 3 vias were required for routing the channel; two layers were used. In terms of the routing area, this is a 50By looking at this layout and by observing the properties of the perfect shuffle, a better layout was derived as follows. The odd numbered nets in the perfect shuffle form

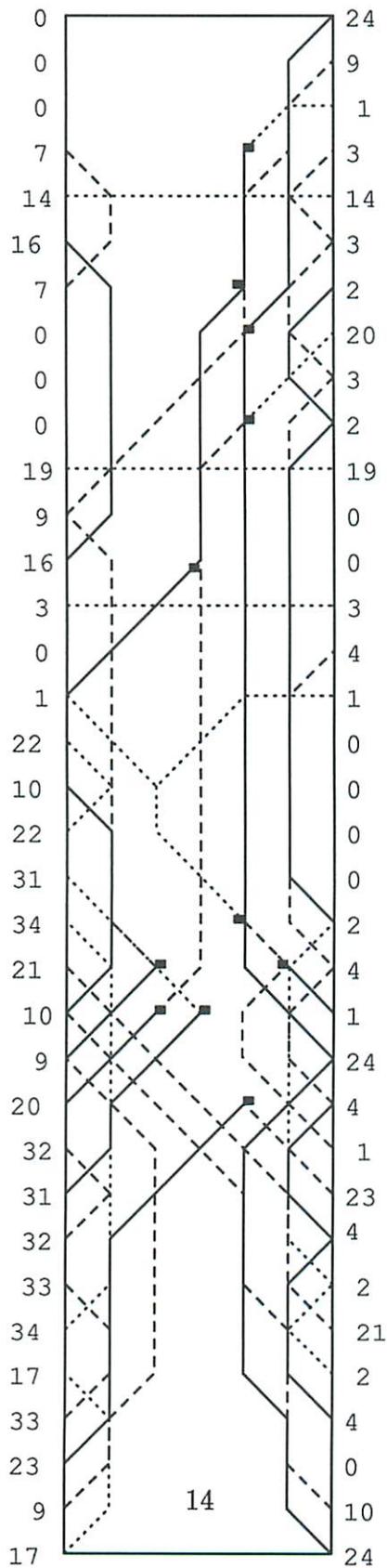


Figure 9: Benchmark from Chen and Liu.

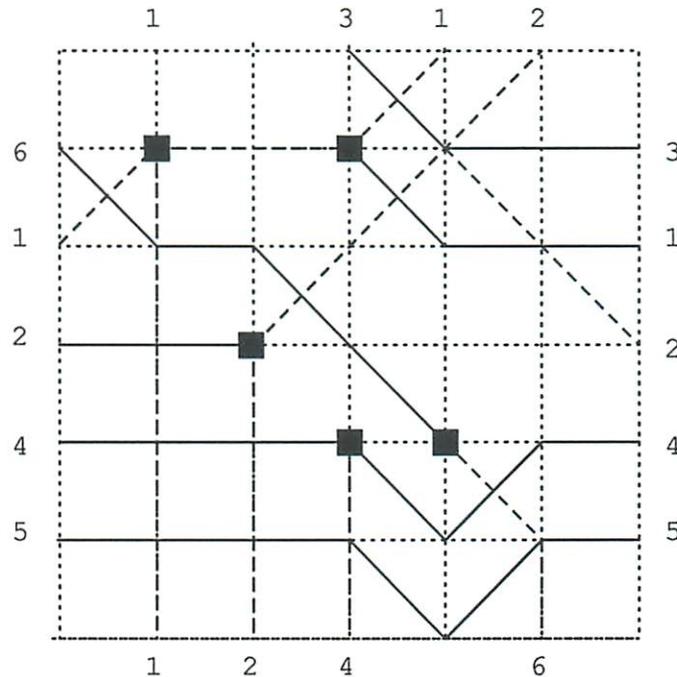


Figure 12: Result for the “thread” Switchbox.

a planar subset and can all be routed in one layer. Similarly, the even numbered nets can be routed on one layer. When this information was provided to VYUHA, it generated a more regular layout without using any vias, as shown in Figure 14(b). Referring to this figure, It is easy to see that for any $2^k \times 2^k$ perfect shuffle, there exists a two-layer, zero-via routing with no more than 2^{k-2} tracks.

There are cases when mixed-slope routing did not help. The reverse permutation of 8 pins, which has a channel density of 8, could not be routed using the mixed-slope model. It was routed using only 6 tracks and 6 vias when only diagonal routing was permitted. Figure 15 shows the final routing.

5 Conclusions

This paper described a simple algorithm for detailed routing. Despite its simplicity, the routing approach has several advantages. Most importantly, the routing algorithm operates independently of a particular routing model. The existing routers are targeted for a specific routing model and cannot be used for a different routing model.

Our algorithm handles a wide variety of routing models in a uniform manner. The router SPYDER described by Enbody and Du [6] comes close in generality. SPYDER

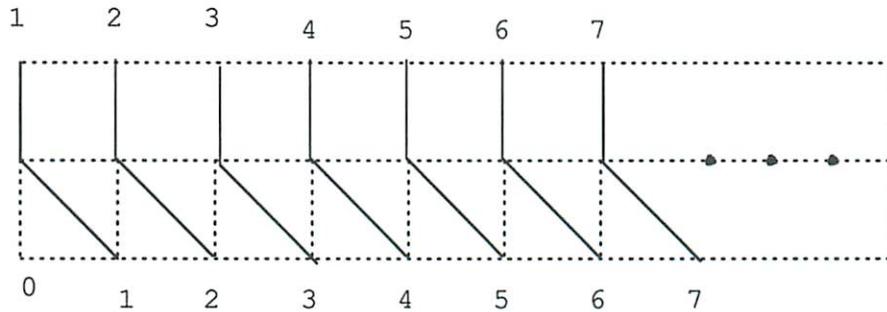
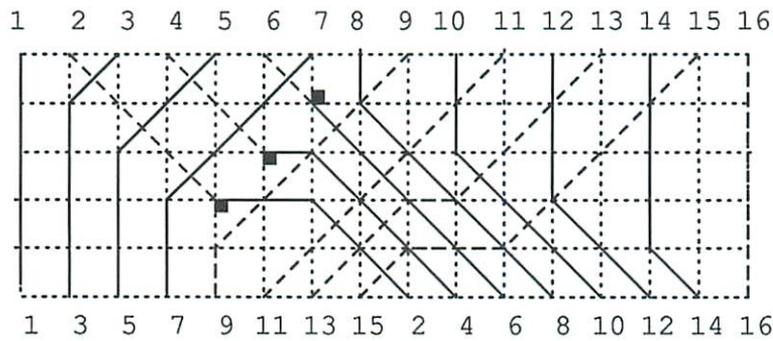
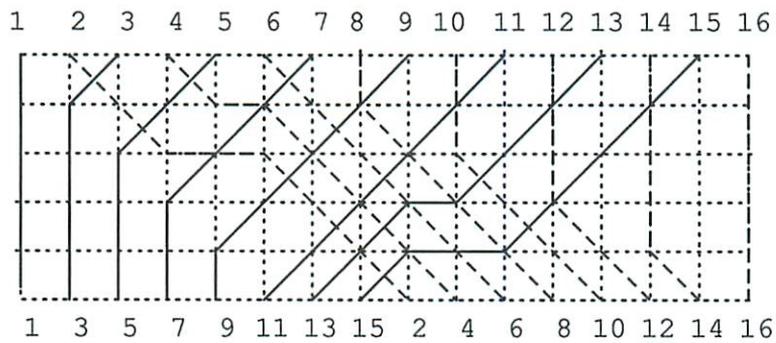


Figure 13: Routing the Shift-by-1 Permutation in Mixed-Slope model.



(a)



(b)

Figure 14: (a) 16×16 Perfect Shuffle. (b) A better layout.

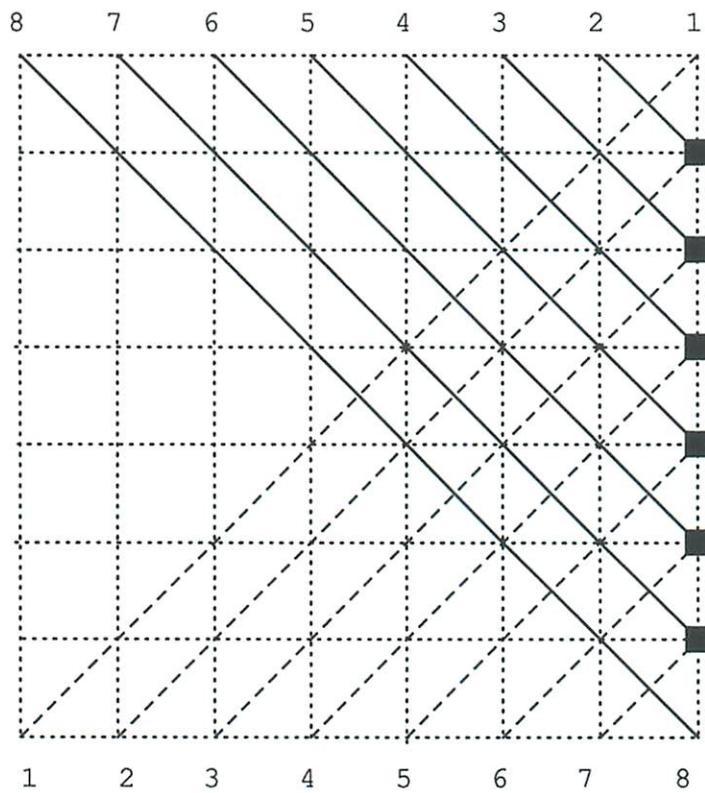


Figure 15: Routing a reverse permutation.

also supports multilayer routing, multiple layer/multiple pin routing and diagonal routing. However, our routing algorithm can handle arbitrarily shaped wiring regions. Unlike SPYDER, which uses a separate layer assignment procedure in order to carry out via minimization, the maze router carries out via minimization as part of the routing. The maze routing algorithm is well suited for parallel processing [17, 13, 16, 2].

References

- [1] S.B. Akers. Routing. In M.A. Breuer, editor, *Design Automation of Digital Systems*, volume 1. Prentice Hall, Eaglewood Cliffs, New Jersey, 1972.
- [2] M.A. Breuer and K. Shamsa. A hardware router. *Journal of Digital Systems*, 4(4):393 – 408, 1980.
- [3] Y.K. Chen and M.L. Liu. Three-layer Channel Routing. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pages 156–163, April 1984.
- [4] J.P. Cohoon and P.L. Heck. Beaver : A computational-geometry based tool for switchbox routing. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, CAD-7(6):684–697, June 1988.
- [5] E.Lodi, F. Luccio, and L. Pagli. A preliminary study a diagonal channel-routing model. *Algorithmica*, 4:585–597, 1989.
- [6] R.J. Enbody and D.H.C. Du. Spyder : A serial/parallel goal-directed router. *Integration*, pages 151–187, July 1989.
- [7] S.H. Girez and O.E. Herrmann. Switchbox routing by stepwise reshaping. *IEEE Trans. on Comp.-Aided Design*, 8(12):1350–1361, Dec 1989.
- [8] W.P. Ho, Y.Y. Yun, and Y.Y.Hu. Planning strategies for switchbox routing. In *Proc. Int. Conf. Computer-Design*, pages 463–467, 1985.
- [9] S.J. Hong and R. Nair. Wire routing machines - new tools for VLSI physical design. Technical report, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, April 1982.
- [10] T.C. Hu and E.S. Kuh. Theory and concepts of VLSI layout. In T.C. Hu and E.S. Kuh, editors, *VLSI Circuit Layout : Theory and Design*. IEEE Press, 1985.
- [11] R. Joobbani and D.P. Siewiorek. Weaver : A knowledge based routing expert. *IEEE Design & Test*, 3(1):12–33, 1986.

- [12] C.Y. Lee. An algorithm for path connections and its applications. *IRE Trans. on Elec. Comp.*, VEC-10:346–365, Sep 1961.
- [13] O.A. Olukotun and T.N. Mudge. A preliminary investigation into parallel routing on a hypercube computer. In *Proceedings of the 24th ACM/IEEE Design Automation Conference*, pages 814–820, 1987.
- [14] Y. Shin and A. Sangiovanni-Vincentelli. Mighty : A rip-up and reroute detailed router. In *Proc. Int. Conf. Computer-Aided Design*, pages 2–5, 1986.
- [15] J. Soukup. Fast maze router. In *Proc. 15th Design Automation Conf.*, pages 100–102, 1978.
- [16] K. Suzuki et al. A hardware maze router with applications to interactive rip-up and reroute. *IEEE Transactions on Computer-Aided Design*, CAD-5(4):466 – 476, October 1986.
- [17] Y. Won et al. Hardware Accelerator for Maze Routing. In *24th ACM and IEEE Design Automation Conference*, pages 800–806, 1988.
- [18] T. Yoshimura and E.S. Kuh. Efficient algorithms for channel routing. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, CAD-1:25–35, January 1982.

**DYUHA : A Detailed Router for
Multiple Routing Models**

BY

C.P. RaviKumar and S. Sastry

**Technical Report CENG 90-17
May 1990**

**Electrical Engineering - Systems Department
University of Southern California
Los Angeles, CA. 90089-0781**