WEAVER: A KNOWLEDGE-BASED ROUTING EXPERT

Rostam Joobbani, Daniel P. Siewiorek

Department of Electrical and Computer Engineering
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

ABSTRACT
In this paper we describe WEAVER, a channel/switch-box knowledge-based routing program. WEAVER considers all the important routing metrics such as 100% routability, minimum routing area, minimum wire length, and minimum number of vias simultaneously. It allows pre-routed nets, and user interaction throughout the entire routing process. It also relaxes unnecessary constraint of assigning different layers to different directions, the constraint imposed by all of the current channel and switch-box routers. WEAVER is a grid-based router that utilizes two interconnection layers and can be easily expanded to route any shape routing area such as 'T' or '+'. Implemented in OPS5, a production system language, WEAVER routinely produces routings requiring less area than routers that focus on a single routing metric.

1 Introduction

Routing[1, 2] is one of the last steps in the design of VLSI circuits. It involves interconnecting pins* of a net, which are scattered on the surface of an IC chip, under a set of constraints. For example, the number of layers available for interconnection is one of the encountered constraints.

Due to the complexity and size of the problem, especially in the case of VLSI chips, the problem is broken into several steps. The first step defines rectangular areas called routing channels through which nets are routed and determines the order in which these channels should be routed. The second step called loose routing decides on the channels that each net will cross. The third and final step does detailed routing of each channel. In this paper we will concentrate on the last step of routing, the detailed routing of each channel.

The quality of the overall routing and the last step of the routing, can be measured by using several metrics, including: the area taken by the routed nets; the routing completion rate; the length of the interconnection wires; and the number of vias. Most of the current routing algorithms often can not complete the routing of all nets and eventually require manual intervention which is very tedious and time consuming.

Numerous detailed routing algorithms have been reported in the literature.[3, 1, 4, 5, 6, 7, 8] The majority of these algorithmic approaches have the following characteristics:

1. They are applicable to certain types of routing, for example there are algorithms that can only route channels** or switch-box*** routing areas.

2. As was mentioned previously, there are several metrics that can be used to measure the quality of routing. Due to the complexity of the routing problem, and the often conflicting nature of these metrics, current algorithms consider one or at most two of these metrics and ignore the rest. For example minimum routing area and 100% completion rate are the most frequently considered, and often not satisfactorily achieved, metrics.

3. The algorithms are over constrained. In addition to the fact that algorithmic approaches can only be applied to one specific routing area, these algorithms constrain themselves even further by assigning different layers to different directions. This means inefficient use of the available layers and consequently more routing area, longer wires, more vias, and less than a 100% completion rate. In addition this constraint prevents routing of the critical nets on one layer.

4. They are oversimplified. Some of these algorithmic approaches are graph theory based and generate topologies that can not be physically realized.

5. Almost all of the existing routing algorithms are structured such that user interaction with the program during the routing process is not possible.

6. Often one wants to route the critical nets manually using the shortest length of wire and minimum number of vias. Very few of the existing algorithms allow pre-routed nets or in general any obstruction in the routing area.

7. All of approaches use brute force and a small number of heuristics[6, 7] that work only in a few specialized cases.

*Pins represent interconnection points for nets.

**Here channel means a rectangular routing area with fixed pins on two parallel sides of the rectangular area. Nets can exit the other two sides of the rectangular area, but, their location is not fixed and is determined by the channel router.

***The switch-box is a rectangular routing area with fixed pins on all four sides.

In this paper we present a knowledge-based routing program called WEAVER. WEAVER attempts to eliminate some of the short comings of the current algorithmic approaches. More precisely some of its characteristics include:

1. It can be applied to a wide range of applications. Even though, presently, it is implemented as a channel/switch-box router, it can easily be expanded to route any general shape area.

2. It considers all routing metrics simultaneously. These include: minimum routing area; 100% completion rate; minimum wire length; and minimum number of vias.

3. There is no assigned layer, for different directions. So it can use all layers for all directions. This means one can route critical nets on a single layer.

4. It allows pre-routed nets and obstructions, on one or more layers, in the routing area.

5. Since human designers are the best experts, it allows user interaction through out the entire routing process. In particular the user can stop the program at any point and route part or all of one or more nets or delete part or all of one or more routed nets and ask the system to continue.

Section 2 of this paper reviews some of the existing routing techniques; Section 3 describes the rationale for choosing a knowledge-based expert systems approach; Section 4 discuses the architecture of WEAVER; Section 5 presents a step by step example of routing a simple channel; Section 6 compares some of WEAVER's result with that of other techniques; followed by concluding remarks in Section 7.

2 Background on algorithmic approaches for routing

There are numerous techniques for routing a rectangular area. The first and the most general of them is the maze router[3]. In the same class as the maze router are the line routers[1]. Both of these techniques route one net at a time. This means that some of the already routed nets block the un-routed nets and require manual intervention to complete the routing. The next class are the channel and switch-box routers[4, 5, 6, 7]. These routers consider the interaction between the nets before routing them but they still suffer from routing only one row or one column at a time. The result is that the routed rows or columns can interfere with future routings resulting in leftover nets. In the case of channel routers, where the channel width can be increased, some approaches[5, 6] achieve 100% routing by expanding the channel width automatically which may not be acceptable or feasible in manufacturing. The hierarchical router by Burnstein and Pelavin[8] attempts to route one grid at a time and has the same limitations as the previous techniques in the sense that arbitarily made decisions in the early stages of routing affect the routing later in the process.

The global router[9] was the first attempt to avoid the routing of one net, one row, or one column at a time. This router treats all the nets as floating, even when they are routed. All the nets can interact and change as the routing progresses. The draw backs with this approach are: it emphasizes 100% completion rate only; it controls the routing process by using a small number of rules that are only applicable in specialized cases; and it is very costly for problems of realistic size.

3 Knowledge-based expert system rationale

To justify the need for and the use of expert systems we would like to quote two distinguished researchers in the field of routing. J. Soukup in 'Global router'[9] mentions: "The state of the interconnection art has reached a saturation point. We know we should iterate by removing routes and putting them back again. A consistent theory is missing. Little more has been done than a blind trial and error approach, which is slow and ineffective". D. W. Hightower in his concluding remarks in 'The interconnection problem: a tutorial'[1] mentions: "Many routers can get very high yields, but the last few wires which must be edited in add greatly to the overall design time. To cut the design time down, we must do 100% wiring. In order to do 100% wiring (without a 'rewiring' post processor), more intelligence must be programmed into the routers so that future needs have more of a say in current actions...". These two quotations clearly recognize the need for domain specific knowledge. We feel that the time and the techniques for avoiding blind trial and errors and programming more intelligence into routing programs has arrived and WEAVER is the first attempt to address these issues using the Artifical Intelligence[10] techniques of expert systems[11]. The characteristic of problems amenable to expert systems applications are areas in which knowledge of the task domain by an expert increases the solution quality and human experts outperform traditional algorithmic approaches. Obviously routing is one of these task domains.

The following example shows how a very small and trivial amount of knowledge of the routing task can help solve problems that are marked as un-routable by some of the algorithmic approaches. The example in Figure 1 is the switch-box which was attempted by hierarchical wire routing techniques[8]. A human expert can immediately point out why the algorithm failed[****]. There are two reasons:

1. Net 3 at the bottom of the switch-box (coordinates 14,1) changes layer, this completely blocks net 24 from even entering the switch-box area, consequently transforming the switch-box into an un-routable area.

2. If we start from the bottom left corner of the switch-box and route the nets one grid at a time one will realize that there is only one way that that corner can be routed. The partial routing of that corner is shown in Figure 2. Since nets 3 and 24 cross each other at coordinates 1,1 there is only one way to route the corner. Net 24 can not change layer at 1,1 and can not go to the right (2,1) because of the presence of net 17 on the same layer.

[****] Pins with the same number at the periphery must be connected internally. For purposes of discussion the internal area of examples are assumed to be on a coordinate grid starting with (0,0) at the lower left hand corner.

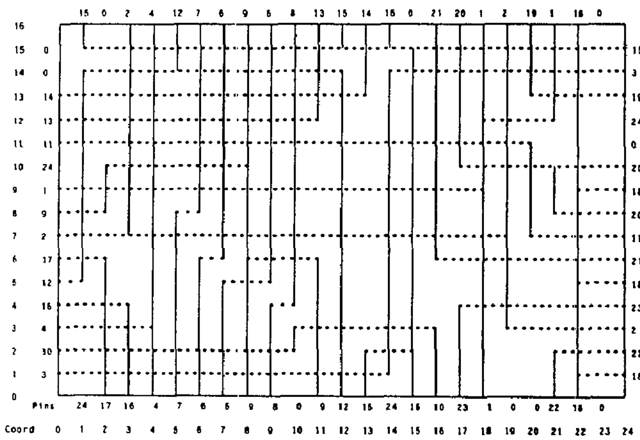

Figure 1: Partial routing of Burstein's difficult switch-box, routed by Hierarchical Wire Routing. Net 24 could not be connected.

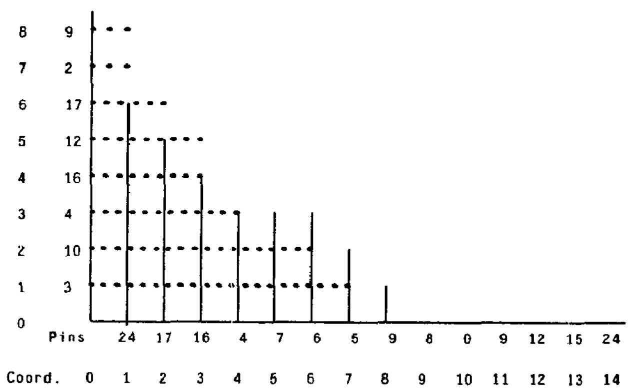

Figure 2: The only possible partial routing of the lower left corner of Burstein's difficult switch-box.

> Consequently, the only way net 24 can be routed is to continue in the vertical direction on its current layer to coordinates 1,2; based on the same type of reasoning logic net 3 has to extend horizontally on its own layer to (2,1); if we follow this line of reasoning we reach the partial routing of Figure 2.

WEAVER has captured simple deductive knowledge as well as other expert knowledge in combination with some algorithmic approaches such as the vertical constraint graph. WEAVER is a global router in which all nets are routed simultaneously when constraints require and individual nets are routed in part or in entirety when there are many choices. The example in Section 5 will illustrate this approach.

4 WEAVER architecture

WEAVER's architecture is shown in Figure 3. It takes the form of HEARSAY II and HEARSAY III[12,13] in which a set of knowledge-based experts are organized around a communicating medium called a blackboard. Each expert decides, based on its own knowledge and metric criteria, what should be done next. A focus of attention module decides which expert should be allowed to give advice at a given time.

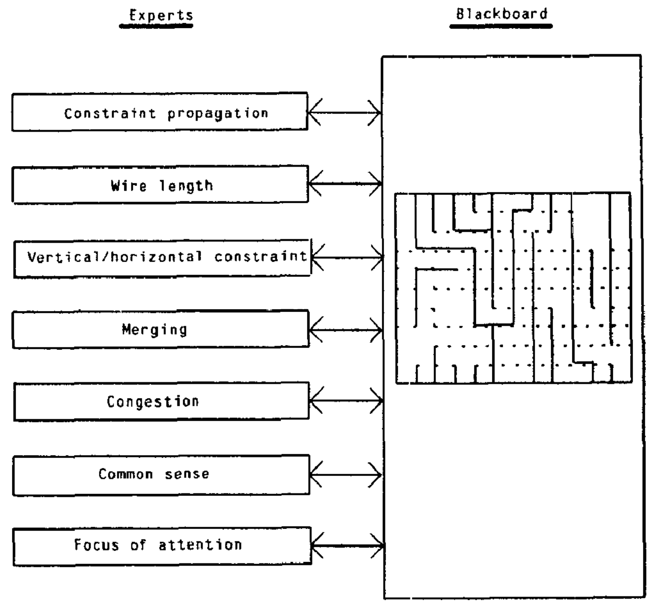

Figure 3: WEAVER architecture

WEAVER's experts consist of the following:

1. A constraint propagation expert. This expert is the most frequently used expert and it adjusts the rest of the nets based on the currently routed net (partially or totally). For example, given the initial switch-box of Figure 1 the constraint propagation expert, based on the design rules, routes part of the switch-box as shown in Figure 2. This is the only way that that corner of the channel can be routed.

2. A wire length expert. This expert decides which nets should be routed closer to which side of the channel based on the minimum wire length. For example a net with only 2 pins on the bottom of the channel should be routed closer to the bottom of the channel than a net with 3 pins in which two of the pins are connected to the bottom of the channel and the third one connected to the top of the channel.

3. A vertical/horizontal constraint expert. This expert utilizes a vertical constraint graph[5]. The vertical/horizontal constraint graph specifies the order of the nets from bottom to top or from left to right of the channel. For example Figure 4-a shows a typical channel and Figure 4-b shows the vertical constraint graph for that channel. As an example, the directed edge from node 5 to node 3 of the graph indicates that the horizontal segment of net 5 should be routed on a row above the row on which the horizontal segment of net 3 is routed and, in turn, the directed edges from nodes 1 and 4 to node 5 of the graph indicate that the horizontal segments of nets 1 and 4 should be routed on rows above the row on which the horizontal segment of net 5 is routed, etc.

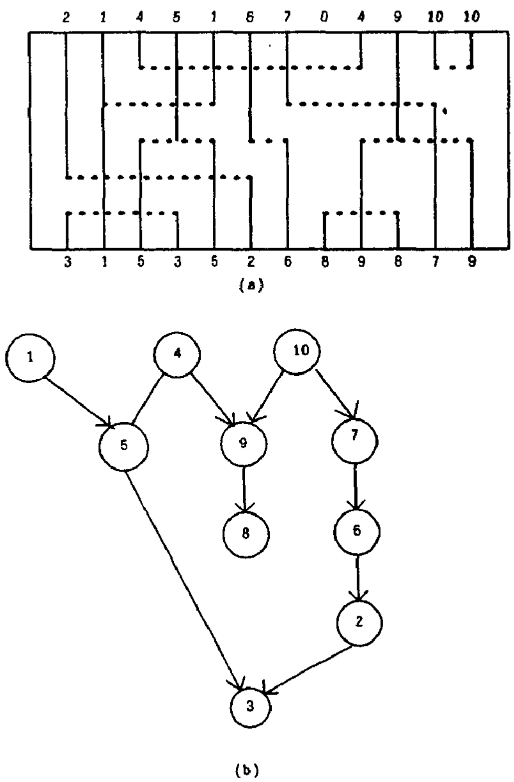

Figure 4: (a) An example of a routed channel,
(b) vertical constraint graph for the routed channel which specifies the order in which the horizontal segment of nets should be routed from top to bottom of the channel.

4. A merging expert. This expert decides which nets can be routed on the same row or column. For example if we have the bottom of a channel as shown in Figure 5 then one can route it in the two ways shown in Figures 5-a and 5-b. The merger expert suggests the routing of Figure 5-a because nets 1 and 2 can be in the same row and as a result the wire length will decrase by 2 units.

5. A congestion expert. The congestion is defined for each row and column in a channel and it is equal to the number of nets crossing that row or column. Each net can cross the most congested area of the channel at most once. This means that the net can not have two horizontal parts crossing the most congested area of a channel. For example net 1 in Figure 6 can be routed in the two ways shown in Figures 6-a and 6-b. But the routing in Figure 6-a needs an extra row since net 1 crosses column 4 which is the most congested area of the channel twice, once in the fourth row and once in the first row.

6. A common sense expert. This expert uses common rule of thumbs employed by human experts when there is no clear best choice based on the advice of the other experts. For example it routes nets which have only two pins and the two pins are on the adjacent columns/rows on a single layer. Net 6 in Figure 8-f is an example of a net routed by the common sense expert.

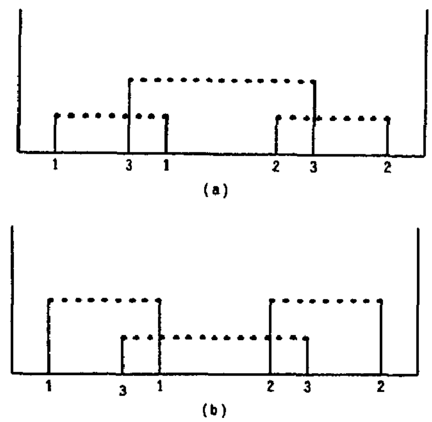

Figure 5: Partial routing of a channel when merging is: (a) considered, (b) ignored.

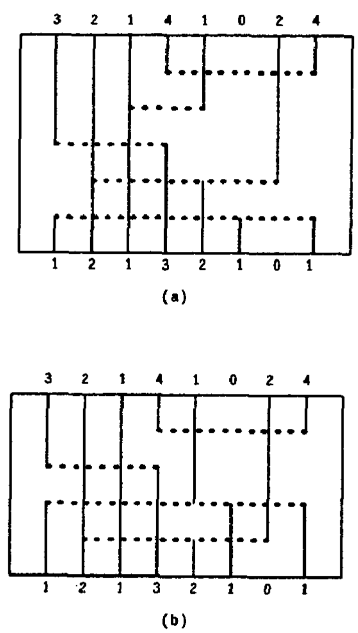

Figure 6: *Effect of considering channel congestion:* (a) is the case where channel congestion is ignored, consequently an extra row is needed; (b) is the case where congestion is taken into account.

7. A focus of attention expert. This expert decides, based on the current active expert and the decision arrived at by the active expert, which expert should be activated next. In the current implementation the *focus of attention expert maintains a priority list for* the experts. This priority list is shown in Figure 7. As Figure 7 shows the highest priority expert is the constraint propagation expert followed by the wire length expert, the vertical/horizontal constraint

graph expert, and the common sense expert. If the wire length expert is activated and it has a suggestion as to which net should be routed next then the other experts such as the vertical/horizontal constraint graph, the merging, and the congestion experts will be activated to comment on the wire length expert's decision.

- Constraint propagation
- Wire length
  - Vertical/horizontal constraint graph
  - Merging
  - Congestion
- Vertical/horizontal constraint graph
- Common sense

Figure 7: Priority list of experts maintained by the focus of attention expert.

Initially, WEAVER's knowledge were acquired by interviewing some of the best designers at Intel Corporation[14, 15] during the summer of 1982. This initial knowledge acquisition revealed that human designers utilize a great deal of trial and error with backtracking that allows them to recover from their errors very quickly. This problem solving strategy reveals itself by the frequent use of an eraser by mask designers. Unfortunately their backtracking strategy does not resemble a strategy which computers can perform well. For example, in their backtracking designers never back up to a previous state and try to follow another track. Human designers often diagnose the situation and take a short cut to a new state which promises success without totally considering the previous states and the procedures that they used to arrive at the present state. Another capability of human designers is the ease with which they can change their representation from one form to another and back. For example a human designer might route a whole net at a time, but when backtracking he/she might not delete the whole net and try to reroute. Instead, he/she will delete portions of a net which are blocking routing and then reroute that portion. This demonstrates that human designers change their representation from line to point and vice versa. Since current computer backtracking techniques are not efficient and the one used by human designers needs a diagnostician which is a research topic by itself, the authors studied various techniques and criteria that can be used to decide when and how a net should be routed. The implementation of these techniques and criteria forms the various expertise explained above. WEAVER utilizes the capabilities of its expertise to avoid backtracking as much as possible.

WEAVER is a rule-based system implemented in OPS5[16] language. Table 1 summarizes the number of rules in different parts of the system. The rules labeled as "Miscellaneous" rules in Table 1 are responsible for reading the input, printing the output, laying out the nets, and house-keeping.

Table 1: Number of rules for each of WEAVER's experts.

Experts	Number of rules
Constraint propagation	106
Wire length	30
Vertical/horizontal constraint	68
Merging	70
Congestion	10
Common sense	31
Focus of attention	26
Miscellaneous	95
Total	436

5 WEAVER example trace

In this section we present a step by step example to illustrate the system performance. The example is shown in Figure 8. To be able to compare the results with other approaches we assumed all the pins enter the channel on the same layer (not a requirement for WEAVER). The major steps are as follow:

- Since the periphery of the channel is not routeable all the pins extend vertically into the channel as in Figure 8-a.
- Two nets 3 and 8 are connected to the bottom of the channel and both can be routed on the same row, so they are routed on the first row as in Figure 8-b (based on the wire length and merging experts' advice).
- After nets 3 and 8 are routed nets 1, 5 and 9 have to be extended upward based on the constraint propagation expert's advice. The result is shown in Figure 8-c.
- Net 10 and part of net 1 are routed in the top row. Again this is based on wire length and mergability (wire length and merging experts' advice). This routing forces nets 4, 5, and 9 to extend downward (constraint propagation expert's advice) and the channel status changes to that of Figure 8-d.
- Net 4 is routed based on the wire length criteria. This action causes a number of activites by the constraint propagation router. These actions include (no particular order):
  - Net 7 should cross column 9 and since only one layer is left for that column, net 7 gets partially routed (the horizontal piece from column 8 to 10 on the 2nd row).
  - Net 9 at column 9 can go left or right but going left does not make sense since all the other pins of net 9 are to the right and the second row is the only free row crossing column 9. So net 9 extends to row 2 column 10 on its current layer.

- o Based on the same reasoning as the two previous cases, net 2 is partially routed (the horizontal piece from column 2 to 4) and net 5 extends from row 2 column 3 to row 2 column 4. This causes net 2 to extend its partial routing to column 5.

- o Net 6 in row 4 column 6 can not change layer (because it can not cross nets 1, 4 and 10), and can not move in any other direction except downward to row 3 column 6. The same reasoning applies to net 7 so it moves downward to row 3 column 7. The result of this step is the partially routed channel shown in Figure 8-e.

- Nets 1, 5 and 6 are routed by the common sense expert.

The program proceeds in this manner until all the nets are routed as shown in Figure 8-f.

6 WEAVER results

In this section we compare WEAVER's performance in two examples with that of algorithmic approaches.

- The examples in Figures 9 and 10 are the same channel attempted by the 'efficient algorithms for channel routing'[5] and WEAVER, respectively. By comparison WEAVER uses 4 tracks, 69 units of wire and 12 vias, whereas the 'efficient algorithms for channel routing' uses 5 tracks, 75 units of wire and 22 vias.

- The example shown in Figure 1 is the one which was attempted by the hierarchical router[8] and failed. Magic[7] could solve the problem with the help of a human expert that pre-routed one of the nets (net 2). Its result are shown in Figure 11. In comparison with that of WEAVER's (Figure 12), WEAVER uses 531 units of wire, and 41 vias where as magic uses 564 units of wire and 67 vias.

- The third example shown in Figure 13 is a switch-box routed by WEAVER. It can be proved that traditional routing approaches, due to the unnecessary constraint of assigning different layers to different directions, will not be able to route this switch-box unless extra rows or columns are added (the congestion at column 6 is 10 whereas only 9 rows are available for routing, consequenetly, two of the nets should be routed on the same row using different layers).

The time and number of rule firings taken by WEAVER to complete these three examples are shown in Table 2. To date, the research focus has been on the application of expert systems to routing and the functionality of the expert router, especially a 100% routing completion rate. The next phase of the research will focus on improving WEAVER's efficiency.

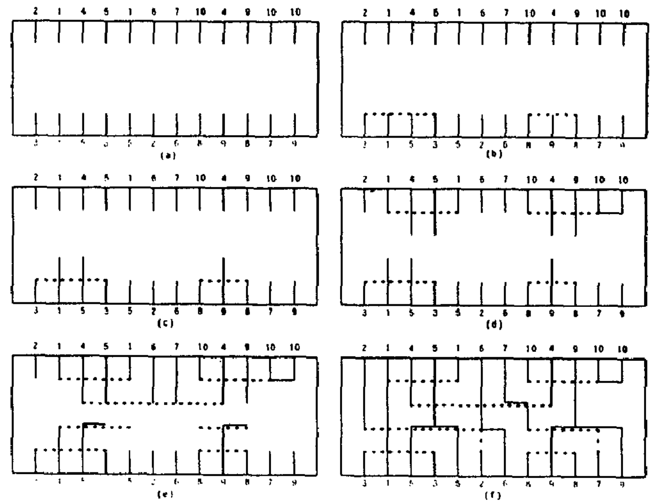

Figure 8: Step by step trace of routing a channel by WEAVER.

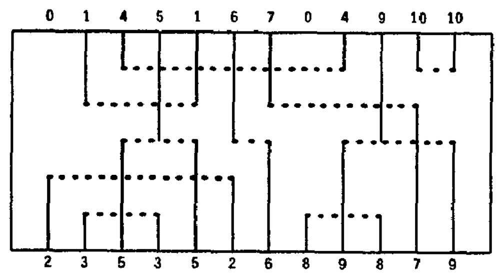

Figure 9: A channel routed by the efficient algorithms for channel routing.

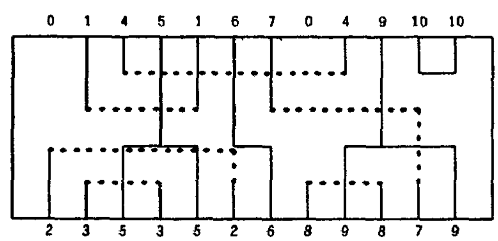

Figure 10: Channel of figure 9 routed by WEAVER.

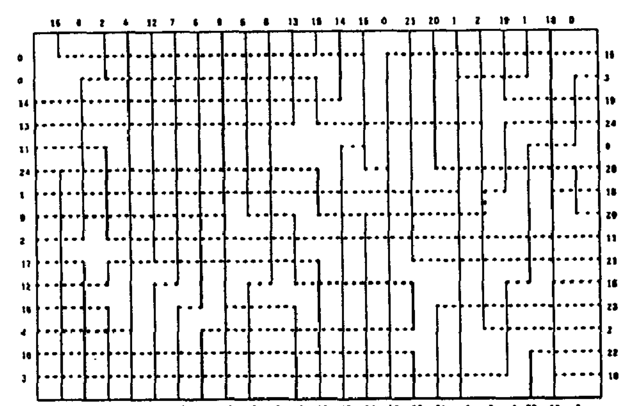

Figure 11: Burstein's difficult switch-box attempted by Magic.

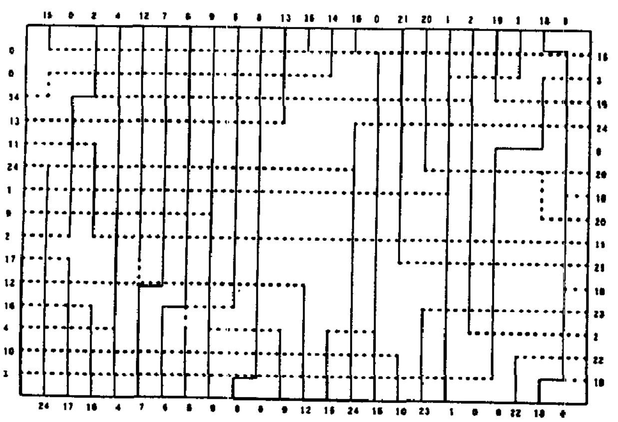

Figure 12: Burstein's difficult switch-box routed by WEAVER.

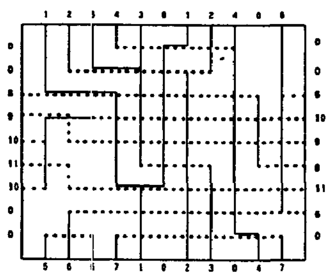

Figure 13: Example of a switch-box that is not routable under the present algorithmic constraints of assigning different layers to different directions.

Table 2: Number of rule firings and number of seconds to complete different examples attempted by WEAVER.

Example	No. of Rule Firings	Time(secs)
Fig. 8	938	115
Fig. 12	3933	1390
Fig. 13	1692	254

7 Conclusion

In this paper we presented the first application of knowledge-based expert systems to the difficult and knowledge intensive task of routing a VLSI circuit. With examples we have shown conclusively that the routing problem is amenable to a knowledge-based expert approach. Furthermore a knowledge-based expert system is a viable techinque that can improve routing quality by considering the many and often conflicting metrics involved in routing. All previous approaches considered at most two of the metrics.

Acknowledgements
This work was funded in part by the Design Research Center of Carnegie-Mellon Univeristy and by the Semiconductor Research Corporation. The views presented here are solely those of the authors and do not necessarily represent those of the funding agencies.

References

1. Hightower D.H., *The Interconnection Problem- A Tutorial*, IEEE Computer Society, 1980, pp. 252-272.

2. Breuer, M. A., *Design Automation of Digital Systems*, Prentice-Hall, Vol. 1, 1972.

3. Lee, C. Y., "An Algorithm for Path Connections and its Application", *IRE Transactions on Electronic Computers*, September 1961, pp. 346-365.

4. Persky, G., D. N. Deutsch, and D. G. Schweikert, "LTX- A minicomputer-based System for Automatic LSI Layout", *Journal of Design Automation and Fault-Tolerant Computing*, Vol. 1, No. 3, May 1977, pp. 217-255.

5. Yoshimura, T., and E. S. Kuh, "Efficient Algorithms for Channel Routing", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. CAD-1, No. 1, January 1982, pp. 25-35.

6. Rivest, R. L., and C. M. Fiduccia, "A Greedy Channel Router", *19th Design Automation Conference, Las Vegas*1982, pp. 418-424.

7. Hamachi, G. T., and J. K. Ousterhout, "A Switchbox Router with Obstacle Avoidance", *21st Design Automation conference*1984, pp. 173-179.

8. Burstein, M., and R. Pelavin, "Hierarchical Wire Routing", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and systems*, Vol. CAD-2, No. 4, October 1983, pp. 223-234.

9. Soukup, J., "Global Router", *Journal of Digital Systems*, Vol. IV, No. 1, Spring 1980, pp. 59-69.

10. Barr, A., and E.A. Feigenbaum (editors.), *The Handbook of Artificial Intelligence (vols 1, 2, 3)*, Los Altos, CA: Kaufmann, 1981, 1982.

11. Hayes-Roth, F., D. Waterman, and D. Lenat (editors), *Building Expert Systems*, Addison-Wesley Publishing Company, 1983.

12. Erman, L.D., F. Hayes-Roth, V.R. Lesser, and D.R. Reddy, "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty", *Computing Surveys, Association for Computing Machinary (ACM)*, Vol. 12, No. 2, February 1980, pp. 213-253.

13. Balzer, R., L.D. Erman, P. London, and C. William, "Hearsay-III: A Domain Independent Framework for Expert Systems", *First Annual National Conference on Artificial Intelligence*1980, pp. 108-110.

14. Kramer, A., "Senior mask designer, Intel Corporation, Private Conversation".

15. Wilde, D. K., "Senior Engineer, Intel Corporation, Private Conversation".

16. Forgy, C.L., "OPS5 User's Manual", Tech. report, Department of Computer Science, Carnegie-Mellon University, July 1981.