CHAPTER 6

Managing Differential Signal Placement

BIFURCATION PROBLEM

6.0 General Thoughts

Although the general concept behind bifurcation is easy to comprehend and almost intuitive, the details of implementation have been shown to be quite complex. As is the case with most difficult problems, the solution to bifurcation was not immediately apparent. Instead, the solution to the bifurcation problem evolved over time. This evolution progressed through several stages. The first of these was the feature vector which was presented in chapter four. As the feature vector concept was solidifying, the finite state machine theory of chapter five gradually began to evolve. It ultimately provided the theoretical underpinnings of the research. Finally, with the finite state machine theory in place, the various components of the solution were ultimately shown to be linked through the use of regular expressions. This chapter focuses on bringing these tools to bear on the bifurcation problem and the development of the linear time solution. Beginning with the problem definition, a generalized algorithm is presented for solving it. The difficulties arise from uncuttable topologies. Next, a linear time algorithm for bifurcation is postulated. Since the foundations for the functions specified have been demonstrated in earlier chapters, the algorithm can be shown to be implementable. The result makes the differential routing process not only viable, but optimal for the problem space. A proof of optimality is incorporated at the end of the chapter.

6.1 BIFURCATION PROBLEM

The basic concept of bifurcation was introduced in chapter three, and an example of how it would operate is provided in Fig. 3.5. But before a generalized algorithm can be formulated, the problem must be completely specified. An analysis of typical single layer metal nets revealed that there is a class of topologies which are unsplittable, if correct signal polarities are to be propagated. Fig. 6.1 provides two examples of SLE nets which cannot be bifurcated in the allocated fat wire tracks.

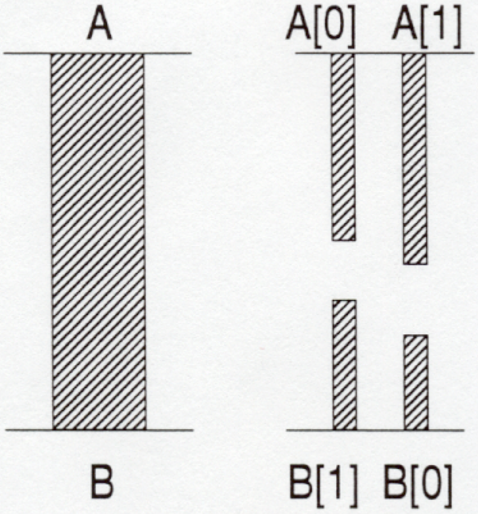


FIG. 6.1(a) UNSPLITTABLE SINGLE METAL CONNECTION

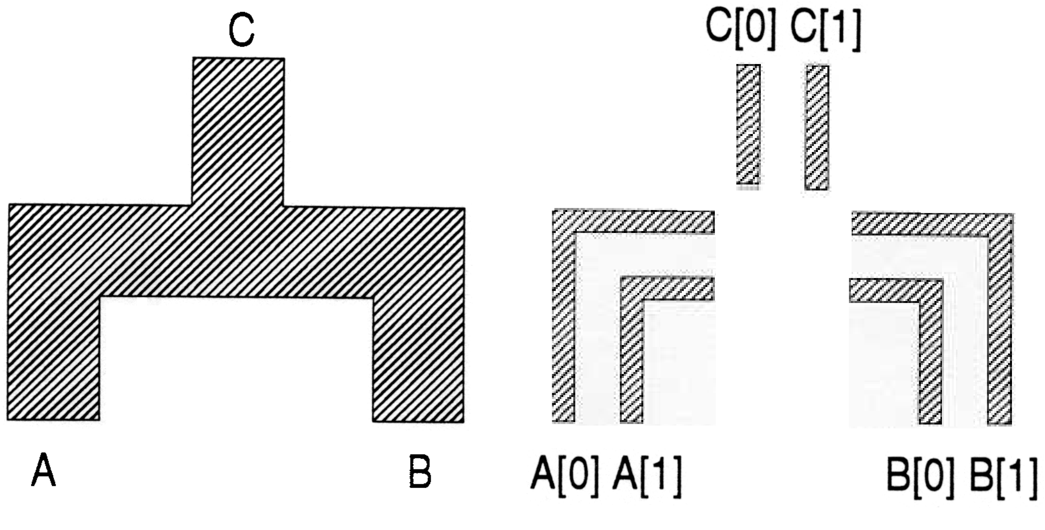


FIG. 6.1(b) SINGLE METAL UNSPLITTABLE WYE CONNECTION

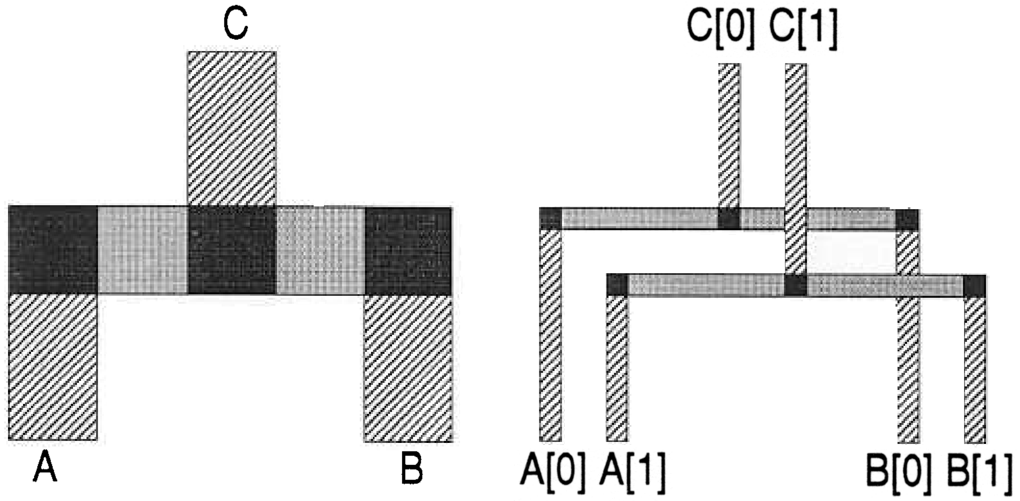


FIG. 6.1(c) SPLITTABLE WYE IN TWO LAYER METAL

Imposition of the Manhattan constraint resolves the class of unsplittable single metal nets shown in Fig. 6.1(b), as demonstrated in Fig. 6.1(c). This constraint introduces a degree of freedom for polarity propagation at each fat via. An example of this was provided in Fig. 3.3. But even with the Manhattan constraint, the degenerate case of a single metal connection between two ports may prove uncuttable, depending on the polarities at the ports and whether or not an inversion occurs.

6.1.1 GENERALIZED ALGORITHM

If the Manhattan constraint is applied to the problem, the generalized bifurcation algorithm is easily formulated using brute force search[Horo86]. Polarity is propagated from a starting point through each junction of the net until the correct polarity has been propagated to all ports. If the correct polarity does not arrive, backtrack to the last decision point where polarity could be assigned and reverse it. The process is repeated until success is achieved or until all settings for all decision points have been explored. The shortcoming of this approach is the potential degradation in algorithm performance as net complexity grows.

Fig. 3.3 provided a glimpse of the functioning of the algorithm for a simple SLE net. Applying the algorithm to a more representative net, Fig. 6.2(a), the backtracking effort necessary to accomplish the splitting process is apparent. The splitting progresses through junctions A, B, C, and D, Fig. 6.2(b). The final connection cannot be completed because the polarities do not match. The algorithm must backtrack through the junctions, exploring alternative settings, until finally arriving back at A, Fig. 6.2(c). After resetting A, the polarities are pushed forward to successful completion, Fig. 6.2(d). The algorithm makes no provision to handle inversions, which can occur at any junction throughout the net.

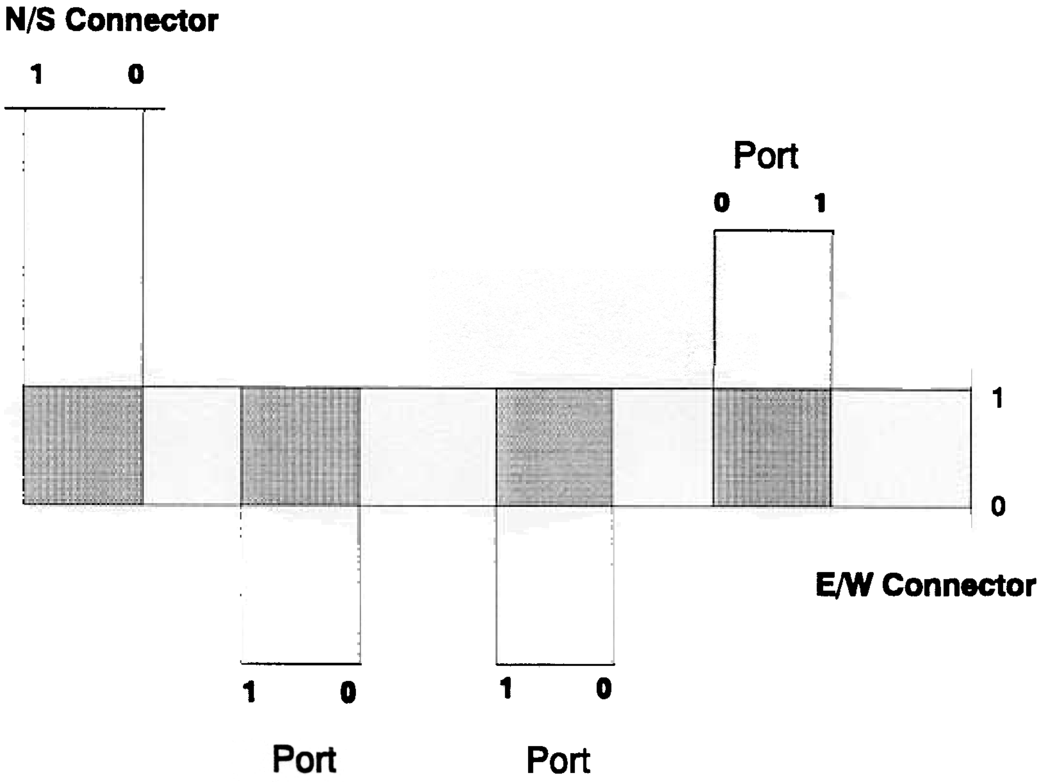


FIG. 6.2(a) SLE NET

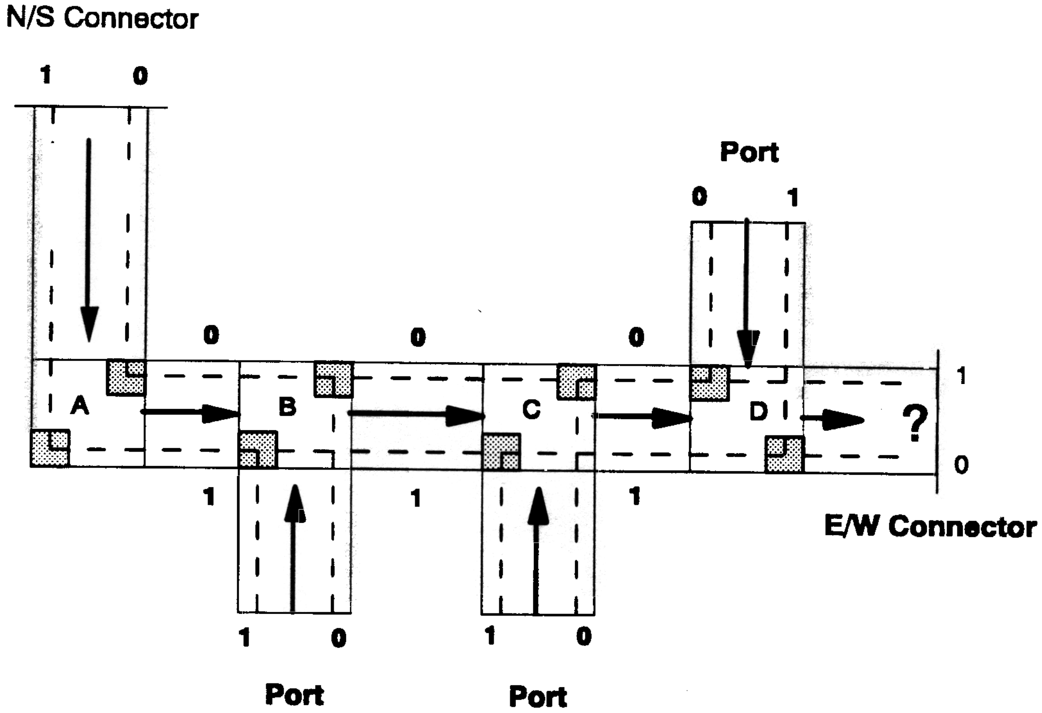


FIG. 6.2(b) POLARITY EXPLORATION

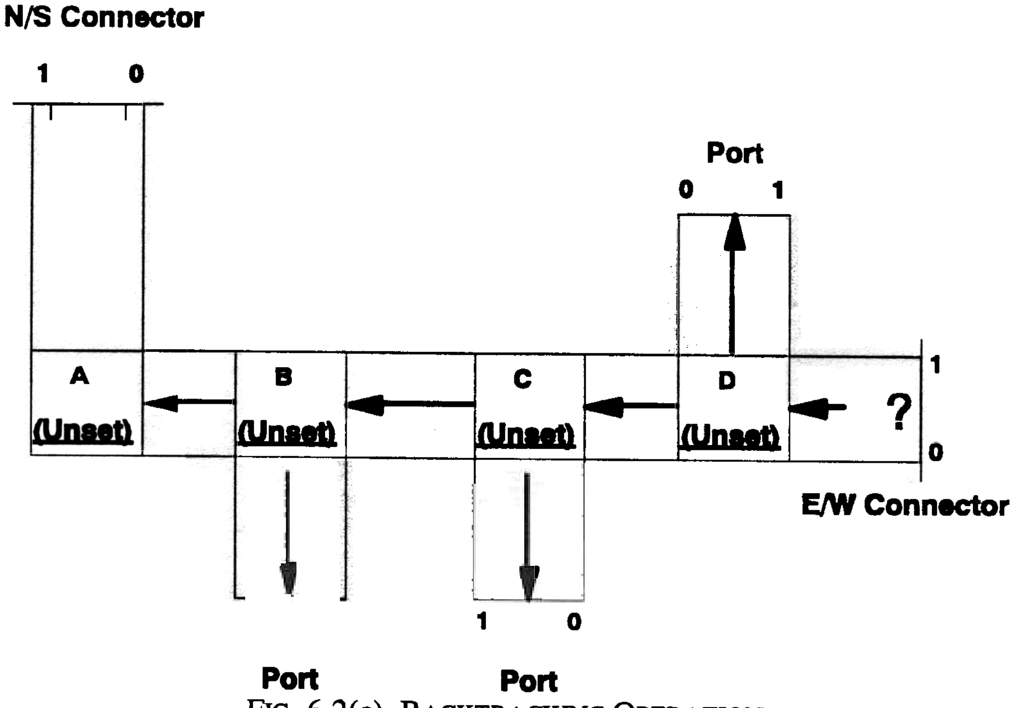


FIG. 6.2(c) BACKTRACKING OPERATION

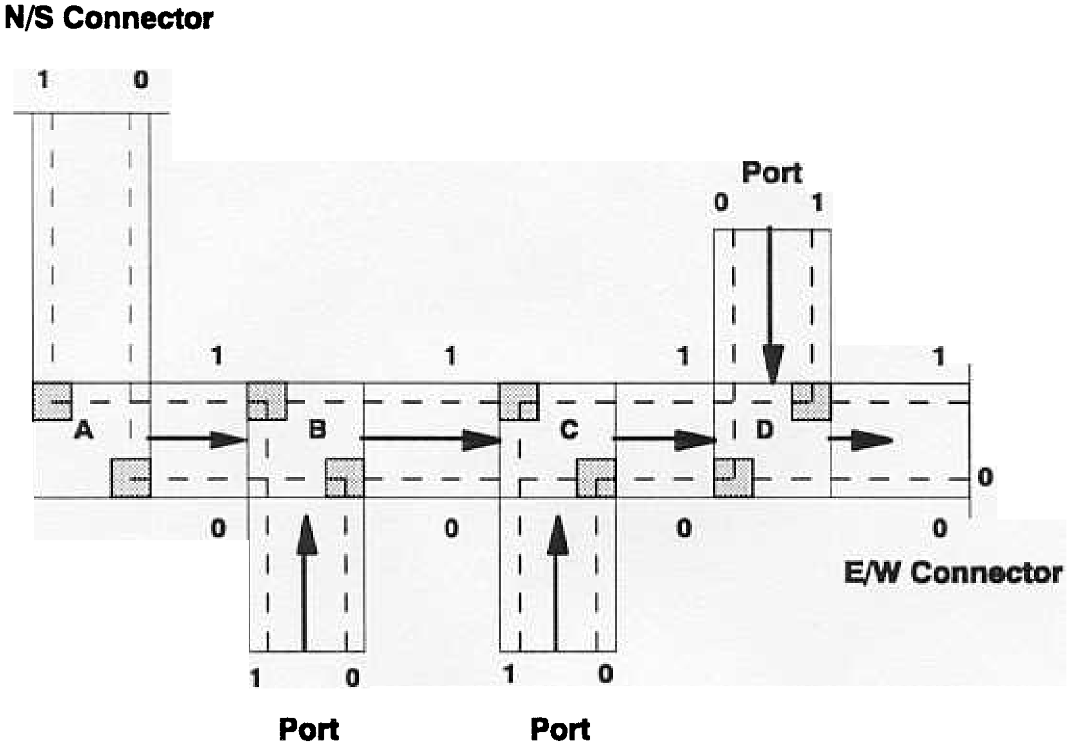
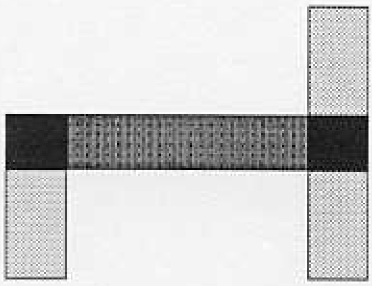


FIG. 6.2(d) FINAL SOLUTION

If presented with the SLE net in Fig. 6.3, the algorithm will explore all setting options in an effort to resolve conflicts, but will fail to find a solution. This analysis reveals that unsplittable configurations exist even with the Manhattan constraint. The configuration which causes the problem is an over-constrained via. Since each via has a single degree of freedom, it can assure proper connection to a single port regardless of pair ordering. When two SLE port segments converge at a via, sufficient freedom does not exist to insure proper connections. This configuration can be eliminated by a local reroute, which introduces a jog to offset the segment junctions.



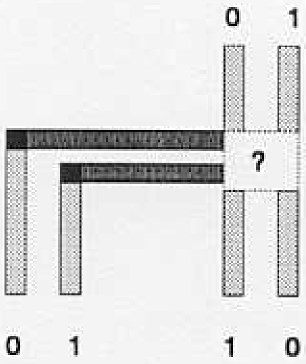


FIG. 6.3(a) OVER-CONSTRAINED VIA FIG. 6.3(b) UNRESOLVABLE POLARITY SITUATION

The obvious penalty associated with the introduction of a jog, is the addition of three fat wire vias. This translates to six additional vias at fabrication time.

In both of the example cases, where the generalized algorithm either has exponential running time or fails completely, the solution to be presented successfully categorizes the net as to its type, and if possible, successfully splits it, re-introducing inversions where appropriate. In the case of an unsplittable net, it is recognized as such, and in the context of this research, the offending configuration is marked for directed editing or local reroute. A general solution for over constrained via configurations is provided in chapter eleven.

Exponential running time of the generalized bifurcation algorithm is a serious impediment. But in a design regime where one improper connection results in a meaningless circuit that cannot be repaired, software testing is crucial. By definition, the generalized algorithm should handle all nets. Yet to fully validate and test software designed to split an arbitrarily configured net, while correctly propagating signal polarity is formidable. Enumeration of all test cases would be theoretically impossible[Beiz83]. Generation of even a modest fraction of the high probability test cases would take months. These factors are major liabilities.

6.1.2 LINEAR TIME ALGORITHM

A basic divide and conquer strategy is employed to reduce the computational complexity of Stage III from exponential to linear, while at the same time insuring complete test coverage. Normally, when such an approach is taken, the reduced complexity of the smaller sub-problems yields a speed up in their individual solutions, but there is recombination cost. Quite uncharacteristically, the proposed algorithm does not suffer from the usual recombination penalty.

A pseudo code representation for the linear time algorithm is shown in Fig. 6.4.

```
function bifurcate
begin bifurcation
        for each fat wire net
                identify_category(fat_wire)
                case category
                        1: - split type(1)
                        2: - split type(2)
                        ....
                        13: - split type(13)
                        Default: Unsplittable
                          - local reroute or dynamic cell instantiation
                end case
        end for loop
end bifurcation

function identify_category(fat_net)
begin
        scan components of net
        tally components to build feature vector
        test component relations
        return(net type)
end

function split type_x(fat_net)
begin
        duplicate vertical and horizontal segments
        fix horizontal polarities
        match polarity of port segs to horizontal segs through vias,
                re-introducing inversions where necessary
        modify net list specifics as appropriate for net type
end
```

FIG. 6.4 LINEAR TIME BIFURCATION ALGORITHM

6.2 FEASIBILITY

The success of the linear time algorithm hinges on several interlocking components. First, there must exist a linear time recognition mechanism that identifies nets as either bifurcatable or not. Additionally, even if a net is determined to be splittable, a technique for accomplishing the bifurcation in linear time must be developed. This technique must correctly handle the re-introduction of inversions. Finally, for nets that are not readily bifurcatable, a guided local reroute feature must exist so that ultimately, all nets can be split in an automated fashion. If these components can be constructed, then the algorithm as presented can be implemented.

From the presentations in chapter 4 and chapter 5, both the recognition mechanism and the proof of correctness have been demonstrated. What remains to be shown is that the actual splitting steps and application of inversions can be completed in linear time.

6.2.1 FIXED POLARITY BACKBONES

The steps necessary to split a net were outlined in an effort to discover common routines or operations. From the generalized algorithm, it was apparent that the task of splitting each segment was a recurring operation. The other principal job was splitting of the fat wire vias. If all segments could be split and then the end points appropriately joined at the bifurcated vias, the complete net could be correctly split. Of course, this assumes the ability to re-introduce the inversions where necessary.

Working with CP files permitted a unique opportunity for abstraction when it came to splitting segments. These components were not delineated by coordinates, but instead by end point pin numbers. Consequently, splitting a segment did not require mathematical manipulation of coordinates, but instead, a renumbering of the pin connections. Clearly, this operation could be accomplished independently of the other requirements.

Propagating the correct polarity through fat via bifurcation became the important task. Once the first port segment attached to a backbone, the backbone polarity was established. Reviewing the generalized algorithm example, it is easy to understand that if this first decision is incorrect, the algorithm will have to exhaustively search the solution space attempting to discover the correct via settings. A way to circumvent this problem was needed.

Connectors linking standard cell areas to other blocks and to pads appeared to be the true source of the polarity inconsistency problem. If nets are properly categorized as bifurcatable, then a degree of freedom exists at each fat wire via for extending the backbone polarity to standard cell ports. Viewing the problem in this fashion, rather than a push the polarity forward technique, another solution presents itself. First, simplify the via splitting decision making by imposing a constraint. Require the horizontal backbone segments to carry an arbitrary, yet fixed polarity. Fig. 6.5(a) shows a representative SLE net, with the fixed polarity backbone established in Fig. 6.5(b).

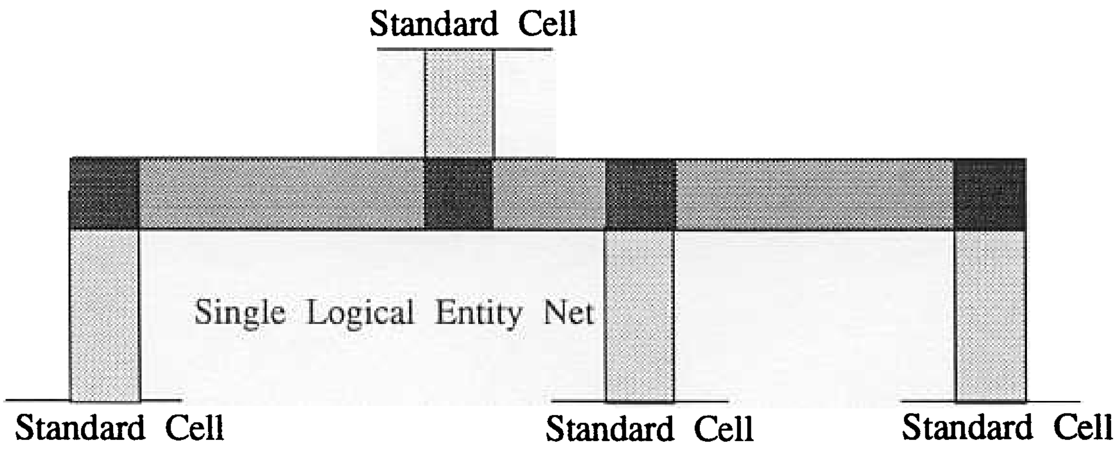


FIG. 6.5(a) EXAMPLE MULTIPLE PORT SLE NET

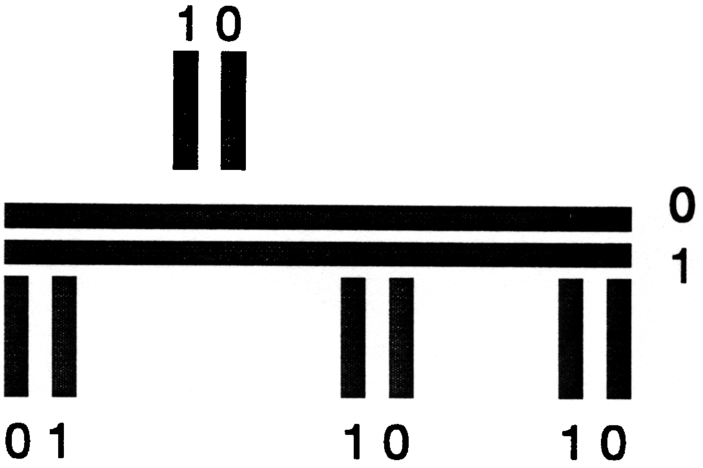


FIG. 6.5(b) SLE NET WITH FIXED POLARITY BACKBONE

With the backbone polarity established in this way, the fat wire splitting problem becomes almost trivial. From a given port, split the emanating segment and attach to the backbone in such a way as to match polarities. The freedom to attach in either way was demonstrated in Fig. 3.5, by splitting the fat wire via in either a NW-SE, or NE-SW manner.

The backbone polarity can be set in either fashion. The "0" signal can be the top wire of the pair and the "1" signal the bottom, or vice versa. With this capability, the polarity at an E-W Connector can be satisfied when the backbone is split. N-S Connectors that link to a backbone can match their polarity just as a port segment can. From this analysis, it is apparent that the constraint which fixed backbone polarity, in no way reduced the number or categories of bifurcatable nets originally set forth in the taxonomy of chapter four. In fact, further review revealed another potential benefit.

If the backbone polarity were still fixed at splitting time, but instead of an arbitrary polarity, a constant one was selected, hierarchical integration among multiple standard cell blocks could be facilitated. Connectors that emanated on the east and west periphery of a standard cell area would have a consistent polarity setting. This would allow immediate marrying of signals from adjacent blocks. In the case of connectors coming from the north or south edge, they could be viewed at the next higher level of the hierarchy as pseudo-port segments. This scheme can be propagated through the hierarchy up through the chip pads.

One stipulation exists at the pad level. When block routing is conducted at the highest level, the Manhattan constraints are relaxed. Long wire runs tend to remain in a single layer of metal, even when direction changes occur. As a consequence, the availability of fat wire vias may be limited or non-existent. To guarantee inversions can be introduced on a wire routed in a single metal layer, two versions of each pad have been generated in the library. For nets of this type, the polarity is examined, and the appropriate pad instance specified to correctly complete the routing. An alternative to generating the additional pad cells is provided in chapter eleven where future avenues of research are outlined.

6.2.2 Introduction of Inversions

Throughout most, if not all of the development to this point, the inversion problem has been pushed to the background. If it can be solved in linear time, then the algorithm of Fig. 6.4 is fully realizable. The notion of inversion, switching the differential wire terminals of the net between the source and sink, is originally identified in Stage I. When the flattened differential net list is translated into a fat wire net list, the inversion information is preserved in a file. As each net is split, the inversion file is checked. If the net being split has been noted in the inversion file, then as they are routed from source to terminus, the polarities must be swapped. This means that at one of the fat wire vias of the net, a split opposite from the normal should be effected. For a two port SLE net shown in Fig. 6.6(a), there are two fat wire vias from which to choose if an inversion must be applied. The first split demonstration depicts an SLE net with no inversion. In this case, the NW-SE fat wire via split option is chosen as the "normal." Having selected this cut technique, providing the cell port connection polarities are the equivalent, then both vias of the net are split in an identical manner.

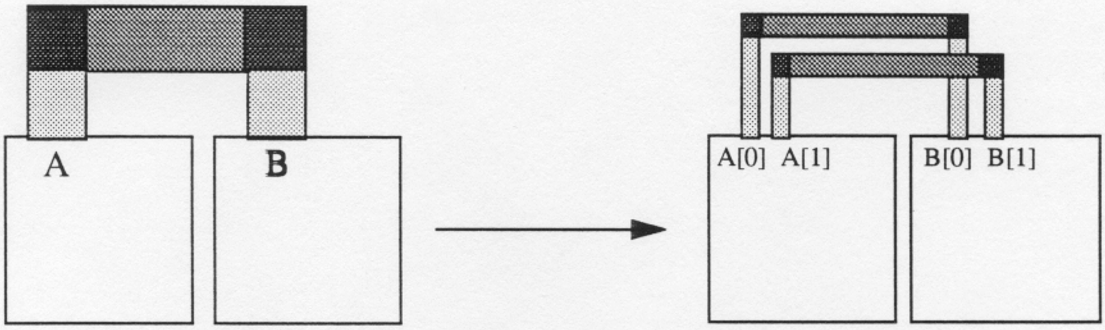


FIG. 6.6(a) SLE NET SPLIT WITH NO INVERSIONS

If an inversion were identified along this net, then a decision must be made about where to apply it. One of the two fat wire via splits must be handled in a way opposite of the other to achieve the inversion. This is demonstrated in Fig. 6.6(b).

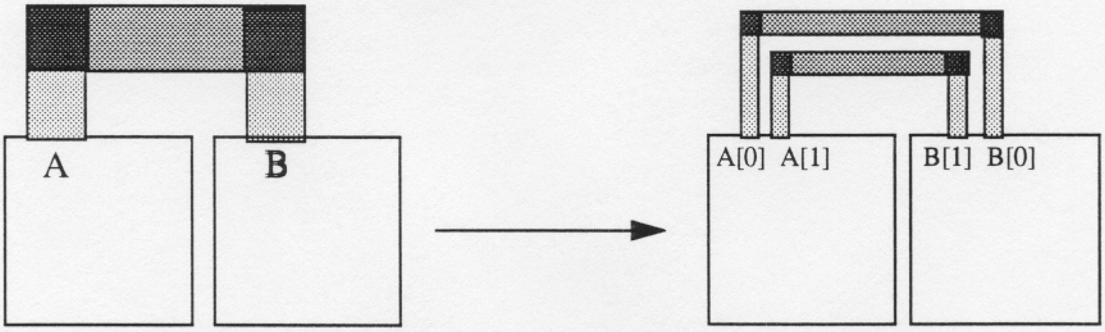


FIG. 6.6(b) INVERSION APPLIED AT SECOND VIA

With more complicated nets, containing multiple vias, the introduction of the inversion can take many forms. Through the application of DeMorgan's Law, the combinations of standard split vias and those split in the opposite direction can be combined in many ways to achieve the desired result. For a three terminal net shown if Fig. 6.7, an inversion between terminals A and B can be accomplished in one of two ways.

Either the cut combinations shown in Fig. 6.8 accomplish the inversion at terminal B, or by applying the inversions at the other two vias portrayed in Fig. 6.9.

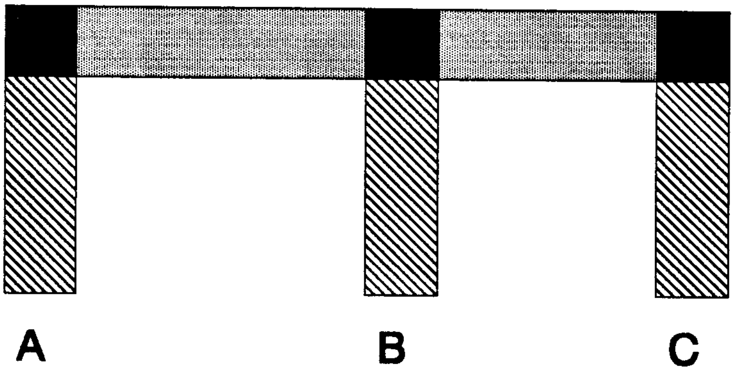


FIG. 6.7 SLE NET WITH INVERSION AT B

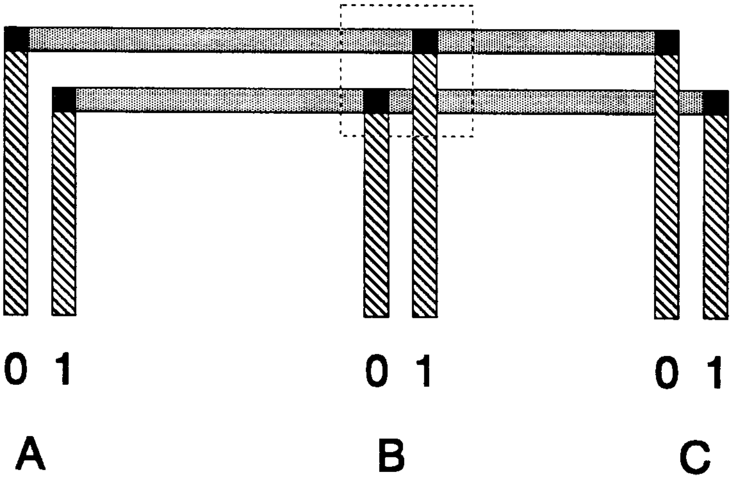


FIG. 6.8 INVERSION AT PORT B THROUGH NON-STANDARD VIA CUT

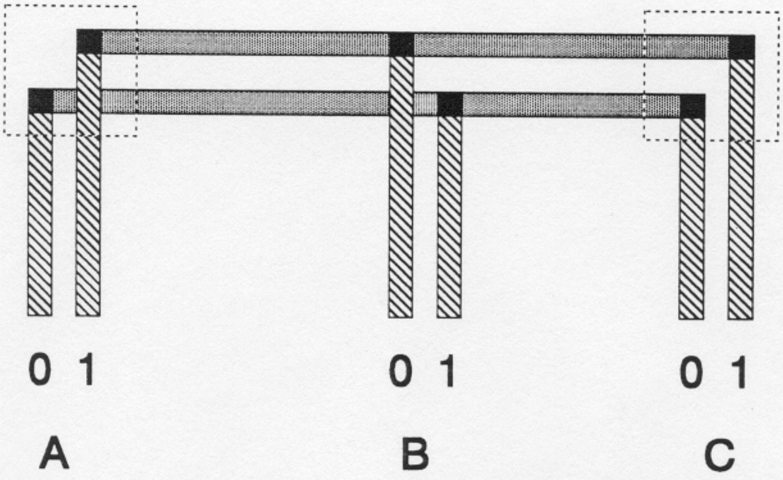


FIG. 6.9 INVERSION AT PORT B APPLIED AT OTHER VIAS THROUGH DEMORGAN

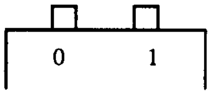
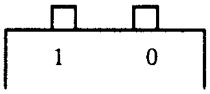
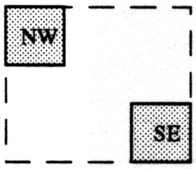
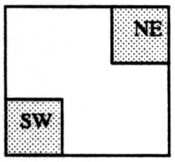
As the number of vias grows, so does the combinations of possible solutions. However, the availability of correct solutions does not necessarily mean that they can be found in linear time. Further study of the mechanics involved produced an additional insight.

Each fat wire via can be split in only one of two ways. Using the fixed backbone polarity, the two via split options allow application of an inversion at any port segment-via junction. Two other factors affect the connection decision. The first is the port polarity ordering on the standard cell. In the GaAs technology, where emitters must retain a given alignment, port connections on standard cells occur only on the north and south side. As a result, the port polarity orderings can only occur in one of two configurations. Finally, the automated placement tool has the capability of reflecting cells at instantiation time. So although the library data on port polarity ordering gives the constructed configuration, the placement system may actually reverse them.

What is common to all of these factors is that each has two possible states. Through analysis, it became apparent that the variables involved might lend themselves to a state variable transformation. The transforms are described in Table 6-1.

TABLE 6.1

STATE VARIABLE MAPPING TABLE

Variable	State	Description
Cell Reflection	0	Original Cell Orientation
	1	Cell Reflected [x = (-x)]
Inversion	0	No Inversion
	1	Net List Inversion at Instance
Port Ordering	0	
	1	
Via Cut Orientation	0	
	1	

By applying a state variable transformation, the logic operation for deciding the proper via cut can be elegantly defined by the concise formula shown in equation 6-1.

$$\text{Via\_cut} = (\text{Original\_Port\_Polarity}) \oplus (\text{Cell\_reflection}) \oplus (\text{Inversion}) \qquad (6\text{-}1)$$

Using equation 6-1, the via split decision at each via junction can be computed in linear time. At this point, each sub-component of the bifurcation operation can be dealt with in linear time. Nets can be recognized through the use of feature vectors. Those that are categorized as bifurcatable can be handed off to the appropriate splitting routines. At the heart of each splitting routine are the fundamental steps of segment splitting, backbone polarity setting, and finally, computation of the via splitting decision which simultaneously injects inversion information back into the problem space. The linear time bifurcation algorithm is not only feasible, but now a reality.

6.3 PROCESS OPTIMALITY

The metrics of chip area, vias and total interconnect define the optimality of the solution. The fundamental routing solution is generated by the core router. The pre and post processing stages do not affect the relative ratios of the metrics. This implies that the optimality of the differential solution is equivalent to that of the core router, if we consider fat wires as single wires. The designer can tailor the optimality, emphasizing particular metrics by appropriate core router selection. As better standard routers become available, the described approach allows their rapid assimilation. The only stipulation is that the final core router solution conform to the Manhattan Routing requirements with respect to metal layer directions.

This approach will always be equal to or better than a purely differential router. A thorough understanding of the problem space is fundamental to understanding the mechanisms at work that support this premise. The Manhattan Routing problem has been shown to be NP-complete[Joy92][Ohts86]. Since it falls in this category, the solution time for a problem of even modest complexity becomes unrealistic. To discover the optimal solution the entire problem space must be searched. For actual applications a tradeoff is made. Since time must be realistically limited, heuristics are employed to prune the search space and provide a near optimal solution in reasonable time. With respect to increases in chip area, for the thirteen categories identified, the actual differential pair is ultimately routed within the bounds of the fat wire. This includes the introduction of inversions where necessary.

The fat-wire approach reduces the problem space from size *N* to size *N*/2, where *N* represents the total number of nets (counting each wire of a differential pair as an individual net). This is the direct result of the single logical entity (SLE) net concept. Let *N* represent the total number of thin wires. Now, the run time of a traditional router can be expressed by equation 6-2 below:

$$T(R)_{Traditional} \cong 2^{N} \tag{6-2}$$

If the number of fat wires is *N/2*, then the routing time for our proposed solution is given by equation 6-3.

$$\begin{aligned} T(R)_{Fat\_Wire} &\cong 2^{\frac{N}{2}} + T(Bifurcation) \\ &\cong 2^{\frac{N}{2}} + C_1 N \\ &\cong \sqrt{T(R)_{Traditional}} + C_1 N \end{aligned} \tag{6-3}$$

For actual routing problems where the number of nets is in the thousands, equation 6-4 holds true.

$$T(R)_{Fat\_Wire} << T(R)_{Traditional} \tag{6-4}$$

The reduction factor of two used for the standard fat wires can be increased if the process is extended to deal with large busses in a similar fashion. This is discussed in the conclusion chapter where future research is outlined.

6.4 SUMMARY

Bifurcation, or the splitting of the fat wire nets into their constituent differential wires, is critical to the proposed differential routing process. Although intuitively rather simple, the real world implementation must overcome many complex issues. If only a single layer of metalization were available, there exists an entire class of topologies that are uncuttable. With the imposition of the Manhattan wiring constraints, all but the degenerate single wire case, and the over-constrained via net can be cut.

The brute force algorithm can be easily formulated, but it suffers from an exponential computational complexity factor, and prohibits complete software verification and testing. Consequently, for even relatively small problem sizes, the run time would be prohibitive.

A linear time algorithm is postulated. To succeed, both a recognition function and a category specific splitting function are necessary. These components were shown to exist. Armed with a suitable bifurcation mechanism for Stage III, the overall routing process was shown to be optimal from both a space and time complexity standpoint.

It is now time to explore the extendibility of the theory to multi-layer metalization processes and Multi-Chip Module interconnect routing.

CHAPTER 7

Managing Differential Signal Placement

N-LAYER GENERALIZATION

7.0 GENERAL THOUGHTS

The results of the routing process are promising. However, the initial implementation focuses on two layer metalization, with the Manhattan restriction. As device density continues to increase, we are rapidly approaching an interconnect limited design space. To continue to be able to successfully route ever more complex chips, additional routing layers will become a necessity. Fabrication techniques are gradually coming of age, with high yield multi-layer metalization gradually becoming a reality. To have lasting applicability, a routing advance such as the one proposed in this dissertation, must be extendible into the multi-layer regime.

This chapter examines the possibility of generalizing the approach to multi-layer interconnect technologies. It begins by reviewing the fundamental assumptions which underlie the fat wire concept. Next, an additional wiring layer is postulated, and the algorithmic adjustments necessary to deal with it are outlined. Then, a second layer is added to see the effects. At each stage, the accompanying taxonomy trees are generated, and the continued applicability of the feature vectors assessed. By demonstrating that the approach can be extended to multiple wiring layers, the differential routing solution presented in this dissertation has been shown to have general applicability; not just a specific problem niche.

7.1 REVIEW OF ASSUMPTIONS

At the outset of the research several assumptions were made that guided the undertaking. First, do to the limited man hours available and the time constraints on completed chip shipment, a router designed from scratch to correctly manage differential signal placement was ruled out. Instead, an approach that centered on dealing with logical nets, and bifurcating the results was postulated. In order to have broad applicability, the system would have to produce a routing solution comparable in quality to the core router. Next, it was imperative that the software be verifiable, or at a minimum, permit complete test coverage. The computational complexity had to be such that the solution could be produced in near real time. And finally, the initial routing domain consisted of two layers of metalization, conforming to the Manhattan rules with respect to layer orientation.

Guided by these assumptions and requirements, the router architecture of chapter three was proposed. Crucial to its success was the efficient handling of the bifurcation problem. It has been shown in chapters four through six, that given an underlying finite state machine theory, feature vectors could be constructed in linear time to recognize nets. Once recognized and categorized, personalized bifurcation routines could then be invoked to split the net. It has been demonstrated that this too can be accomplished in linear time.

7.2 EXTENDING THE THEORY

Having reviewed the underlying assumptions and system requirements, the question now arises, "Can the theory developed to handle two layers of metalization be extended to additional layers?"

To answer this question, it was necessary to break it into stages. Initially, the theory was reviewed in light of the fact that one additional layer of metal was available for routing. Then a similar analysis was conducted using two additional layers. From the results obtained, it was then possible to generalize with regards to the impact and changes required to adapt the system to an *n-layer* environment.

The methodology employed when examining the effects of adding layers took the following form. First, using a "cut-line" concept similar to that in chapter four, the net segments extending into the additional layer/layers could be partitioned from the remainder of the net. Then, taxonomy trees were generated, building on the knowledge gained in chapter four. Next, the ability to construct appropriate feature vectors was assessed. Finally, the inversion problem was re-examined for correctness.

7.2.1 THREE LAYERS OF METAL

The first multi-layer analysis considered one additional wiring layer. A pictorial example of the use of such a layer is provided in Fig. 7.1. It is drawn in such a way that the excursion into an additional layer is highlighted. The two vias connecting the original metal-2 backbones to the third layer metal are exaggerated. This was done to demonstrate how the addition of the next layer really takes what has been a two dimensional problem and projects the effects into three dimensions. Certainly, if a close-up view were constructed of the connections between metal-1 and metal-2, an argument could be made that you were already in a three dimensional problem space. However, for routing analysis purposes, the 2-D synthesis provided all the necessary detail up to this point.

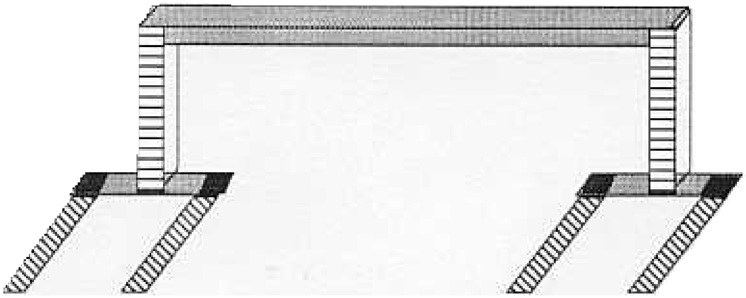


FIG. 7.1 THIRD LAYER METAL EXAMPLE

Using the cut-line concept of chapter four, the example three layer net can be partitioned. The cut lines are passed through each of the metal-2 to metal-3 vias. This results in three nets. Two are immediately discernible as single backbone nets with two port segments (Type 2 in the Taxonomy Tree). The two via "legs" and the metal-3 segment, if projected onto a vertical plane, would also appear as a single backbone net with two port segments. Those segments are represented by the exaggerated vias. A diagram highlighting the cut line process is provided in Fig. 7.2.

From the cut line analysis it was revealed that the third layer of metal would be used to connect nets from the original taxonomy of chapter four. This implies that instead of a single feature vector for a net, it was now possible to have a vector to represent each component net that results from the cut line partitioning of the original net. The constituent nets that lie in the first analysis plane will have feature vectors identical to the taxonomy categories that were developed earlier. The only study remaining to be done concerns the topology of the metal-3 connecting net, which it will be assumed has no connectors.

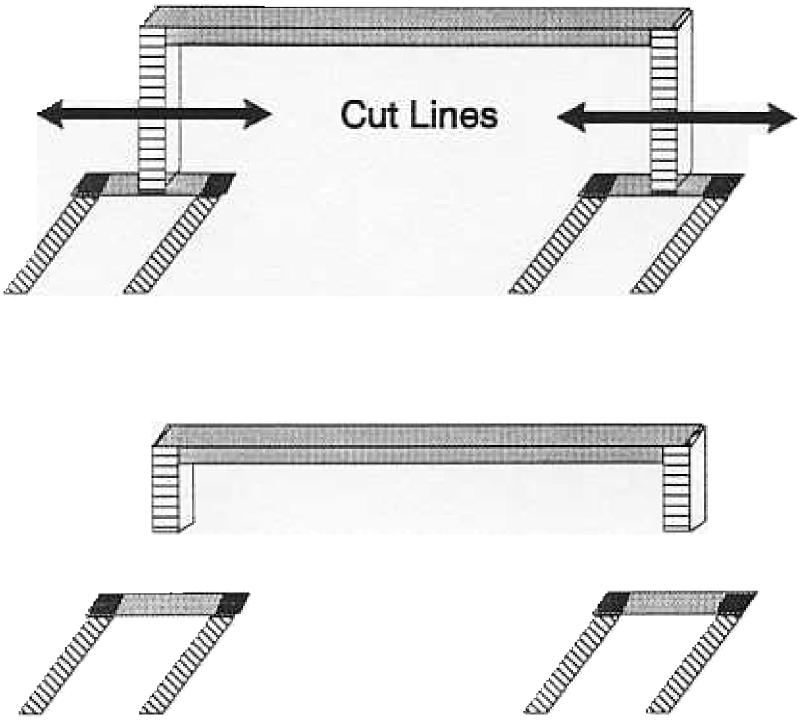


Fig. 7.2 Cut Lines Applied to Three Layer Net

In order that the metal-3 net be bifurcatable, polarities must be properly maintained along the net. Since it must by definition exist in a single layer, any branching of the net is prohibited. It also means that at each bend in the net the connecting pins for the split nets must not cross. This is nearly an identical problem specification to the single layer PCB routing problem. An example of a properly bifurcated third layer net is provided in Fig. 7.3. A third layer net containing a non-allowed branch is shown in Fig. 7.4. The branch stub shows that neither a stub nor a crossover of signal lines comprising the pair can occur in the single additional layer.

From this investigation it is now possible to fully describe the topology of the third layer net. It can possess any number of horizontal or vertical segments, so long as the end of any segment is only connected to a single end of another segment or to a via.

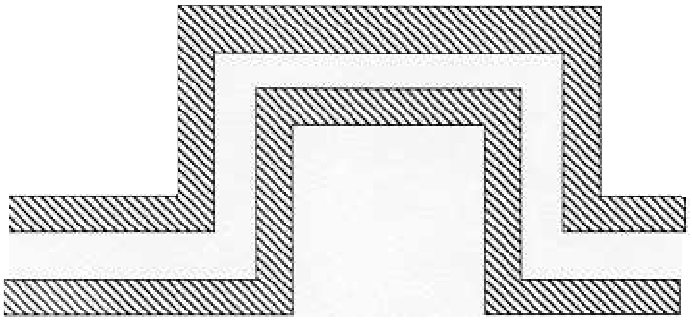


FIG. 7.3 PROPERLY BIFURCATED THIRD LAYER

At each segment to segment junction, the corner, when split, cannot permit a crossover of the signal wires. Consequently, once a polarity is established on the third layer segment, it will be maintained throughout the course of the net. It can only be altered during vertical connection that takes place when the fat wire via between metal-2 and metal-3 is split. As was the case with a fat wire via between metal-1 and metal-2, a degree of freedom exists for making the connection between metal-2 and metal-3. Thus, the limitation on a fixed polarity throughout the course of the third layer net in no way limits the final bifurcation of the complete net.

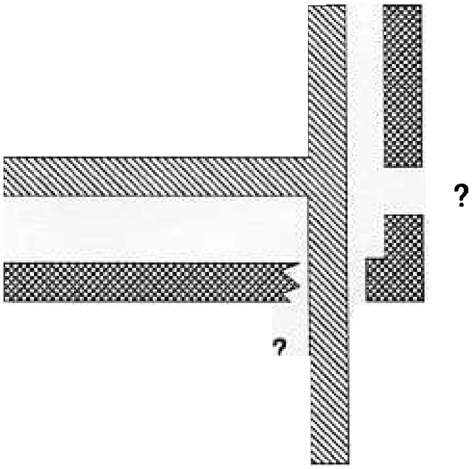


FIG. 7.4 EXAMPLE OF INAPPROPRIATE STUB

In fact, the fixed polarity stipulation on the third layer net serves to simplify the inversion problem solution. Since each of the component nets formed by the cut line partitioning can be recognized by an established feature vector, it is known that the net is bifurcatable, and inversions can be correctly applied internal to each sub-net. By bifurcating each of the component nets in this fashion, the inversion problem for the individual nets can be satisfied independently. Then, when the third layer net is connected, it can be visualized as a transparent extension cord, allowing two separate circuits to be joined by a fixed polarity conduit.

So that all component nets understand the third layer polarity scheme, some standardization is required. To correctly construct it, one assumption is necessary. The metal-2 to metal-3 vias must all occur on the same segment orientation throughout the net. If the starting via connects to a horizontal segment then all of the other vias between metal-2 and metal-3 should also occur on horizontal metal-3 segments. This permits rapid determination of the polarity at any point on the net.

The third layer horizontal end segments (assuming the first via connected affixes to a horizontal segment) will have the same fixed polarity setting as the standard two layer type. The top wire of the pair is considered the zero wire, and the bottom the one. However, since no changes can occur at a corner, a bend in one direction will produce a vertical polarity orientation of one type while a bend in the other direction generates the polarity orientation of the other type. By requiring all vias to connect to a segment with the same orientation, a simple formula for computing the current polarity can be derived. If the number of left bends equals the number of right bends then the current horizontal polarity is identical to the start polarity. If the number of left and right bends is not equal then the polarity of the current horizontal segment is opposite the standard. The same reasoning can be applied to starting with vertical segments, and the same polarity calculation conducted.

Having analyzed the problem for an additional layer of metal, the new taxonomy tree would degenerate into a single node. That node would allow for a single chain type net, consisting of any number of segments, and any number of vias, providing the vias connect to segments with similar orientations. The feature vectors for the partitioned nets would be identical to the original feature vectors. Cutting the third layer net would involve splitting each segment into its constituent components, assessing the direction of bend at each segment junction, and then splitting the fat junction appropriately. These steps can all be accomplished in linear time.

The addition of a single layer of metal has only minimal impact on the already established routines. The stipulation that connecting vias occur on segments with similar orientations is not unreasonable. The cost of an additional bend in a wire may be introduced, but no additional vias. Interestingly enough, when two extra layers of metal are analyzed, this restriction can be relaxed and eliminated.

7.2.2 TWO LAYERS OF ADDITIONAL METAL

The technique for studying the effects of two additional layers of metal follows what was done in section 7.2.1. One can now visualize the net topologies as the joining of two, 2-D routing planes. This is shown in Fig. 7.5, where two typical Type 2 nets are linked together with a Type 2 net that exists in layers three and four.

If the cut line analysis is used to split layers three and four from layers one and two, an elegant hierarchy emerges. The overall net can be viewed as a composite of three or more nets. The net components that reside in layers one and two will have feature vectors identical to those described in chapter four. The taxonomy of allowable topologies will also match exactly. Then when analyzing the net component that resides in layers

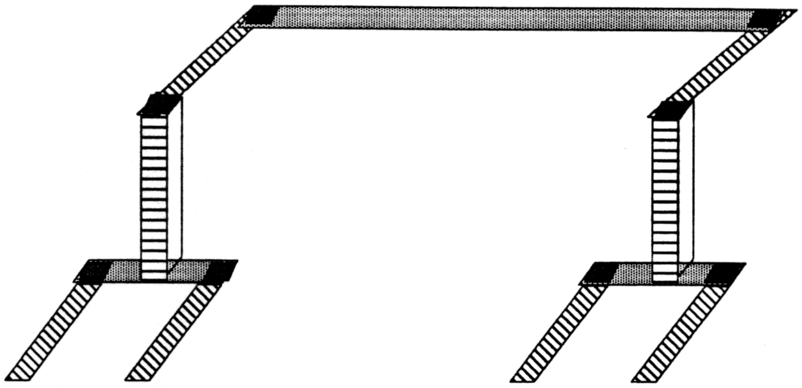


FIG. 7.5 TWO ADDED METAL LAYERS

three and four, a similar finding emerges. Since no connectors occur in layers three and four (basic assumption), the taxonomy tree for these layers will be a considerably reduced version of Fig. 4.13. This revised tree is shown in Fig. 7.6. The feature vectors corresponding to the categories of this hybrid tree are identical to those for the equivalent categories in the original tree.

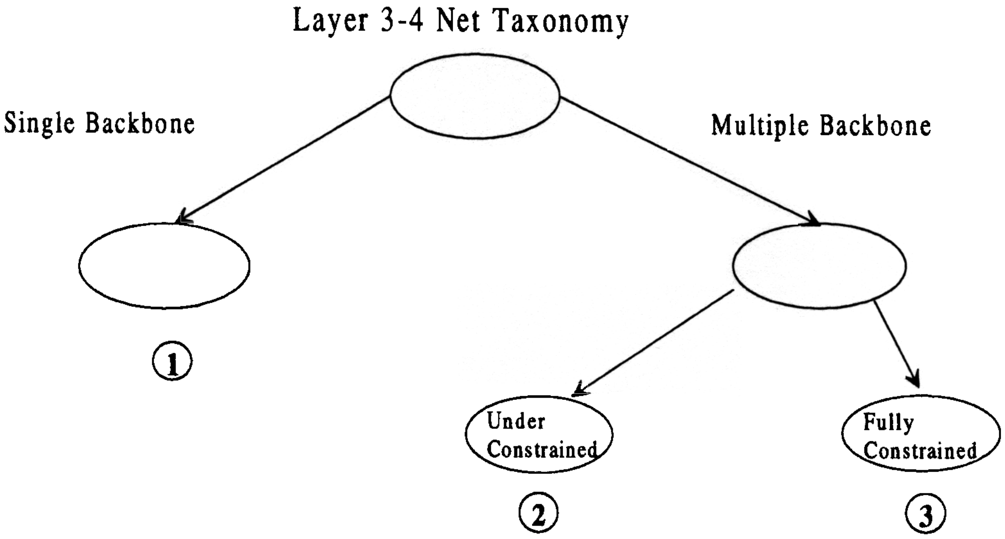


FIG. 7.6 MODIFIED TAXONOMY TREE

Since vias exist between layers three and four, the single wire chain restriction that applied to metal-3 alone is eliminated. As was the case with the layer one and two nets, degrees of freedom exist at each via junction. To establish the transparent wire standard for the third and fourth layers, the traditional layer one and two standards apply directly. For all horizontal segments, the top wire of the pair represents the zero wire, and the bottom the one. For vertical segments, the left wire of the pair is the zero wire, while the right wire represents the one. If the “zero” via cut from Table 6.1 is used for all metal-3 to metal-4 vias, then the standard polarity configuration is guaranteed throughout the layer three-four net.

Net recognition and bifurcation will flow just as if it were a two layer net. Cut line analysis will produce the component nets. Each layer one-two net will be categorized and split following the techniques presented in the earlier chapters. The layer three-four subnet will be identified by one of the feature vectors associated with the modified taxonomy tree. The bifurcation is straightforward, since the inversion problem is dealt with on layers one-two. All fat wire vias on the three-four net are cut using the "zero" cut method. This insures a transparent "extension cord" philosophy, whereby inversions are dealt with prior to the junction leading to the layer three-four net.

7.3 SUMMARY

Specific solutions to particular instances of a problem are important. However, to have lasting impact, a solution should be generalizable. If it cannot cover all instances of a problem, then the greater the breadth of coverage, the more valuable the solution. For the original two layer differential routing problem instance, the fat wire approach that employs feature vectors as net recognizers appeared to be nearly ideal. However, to continue to be of value in the upcoming era of interconnect limited circuits, it would have to be extendible into the multi-layer routing regime.

This chapter explored the possibilities of extending the concepts into that domain. First, a single additional layer was examined. Then, two were researched. The results indicate that the net taxonomies along with the feature vector recognizers will perform well in this extended arena. For a single additional layer, certain minor constraints had to be imposed. However, if a second additional layer can be added, these constraints can be completely eliminated.

Also, it would appear from this research, that adding layers in pairs provides the cleanest transition in routing system upgrades. Layer pairs work together to build net categories identical to the original taxonomy (minus nets with connectors). When pairs are added in this fashion, transparent polarity propagation on upper levels is readily achieved through the zero cut solution applied to each fat wire via.

The results are very encouraging and appear to imply that the overall router architecture proposed in chapter three is easily extendible into the multi-layer routing regime. The primary problem will be selecting a core router that can take full advantage of the added routing flexibility. Again, tradeoffs will be involved; weighing the penalty of total system interconnect against the length of the critical paths.