AN AUTOMATED INTERCONNECT DESIGN SYSTEM

W. E. Pickrell
Automation Systems, Incorporated

INTRODUCTION

This paper describes a system for automatically designing and producing artwork for interconnect surfaces. This system consists of a number of computer programs which can be a subsystem of a general design automation effort. The interconnect design programs deal with the problems of artwork production and interconnection of electronic components frequently experienced in computer system design and construction.

One such system was developed for a former employer and spanned a period of about 18 months. During this period of development numerous experiments comparing the automatic interconnect design method to the manual method were performed. This paper discusses the results of these experiments along with some details of the programs and their operation. Before a discussion of automated interconnect design can proceed, some preliminary definitions are required.

The laminate of this paper is a series of single-sided etched boards or layers. Each layer has its own series of conductive interconnect paths. Layer-to-layer communication is accomplished through plated holes drilled through the layer. A group of layers are stacked and bonded together under heat and pressure to form a laminated interconnection board. Electronic components which require interconnection are mounted on the surface (or both sides) of the laminated board according to the scheme of the logic.

The “feed-thru” or “drill-thru” implies a second form of layer-to-layer or side-to-side communication in addition to component terminals. The system accepts either fixed or “floating” locations on the interconnect media.

MANUAL INTERCONNECT DESIGN AND ARTWORK

As with many computer applications the beginning of the analysis is the “present” manual method of how things are done, since this situation is usually the one that requires some improvement in speed, accuracy, or cost. This holds true equally well for the interconnect design effort. A typical circuit board design task can be divided into the following steps:

1. Prepare a schematic wiring diagram.
2. Prepare a cover (master design pattern).
3. Place components in the best pattern.
4. Assign the external pins.
5. Check the artwork design.
6. Tape the artwork.
7. Check the artwork against the design layout.
8. Photographically reproduce (chronoflex) and reduce the artwork to desired size; produce necessary negatives and positives.
9. Deliver finished negatives to laboratory for fabrication.

Analysis of these steps shows that some of the work is particularly suited to automation with computer programs. If some or all of the functions could be performed for less cost, time and greater accuracy with computer programs, a significant gain in computer technology would be realized. The following section describes the program system and programs developed to meet that end, concluding with a section of remarks and statistical results for review.

THE PROGRAM SYSTEM

The system includes the following general program areas:

1. **Generation** of the simulated physical composition of the surfaces in digital form (mapping or cover layer generation).

2. **Organization** of the data to be interconnected (routed). This may include: selection processes, determination of a minimal connection tree for a signal net, construction of various sort keys and selected sequences of sorts. These are all designed to facilitate and increase the yield of the routing phase which normally follows.

3. **Routing** of the organized data strings against the simulated physical environment of the surfaces taking into account any special constraints imposed by the hardware system being processed.

4. **Editing** and generating inputs to graphic devices such as plotters for covers and routed surfaces.

5. **Auxiliary** program providing secondary passes or continued processing such as:
   (a) routing update — to allow manual intervention,
   (b) net change — to pass first run failures onto alternate paths,
   (c) drill through — a subsequent run to the router phase if further laminate interlayer communications are required.

THE PROGRAMS

This interconnect design system is modular and independent of other systems and, as such, operates on a fixed placement of components on the board as provided by a Logic Assignment and Placement Subsystem which precedes it. The programs can be best described by function, input and output, in the sequence they normally retain in operation.

1. **Title:** Cover Layer

   **Function:** Simulate the environment in digital form for interconnection (grid, size of board, obstacles, number of layers, external connectors, component pin arrays).

   Apply grid coordinates to all necessary data.

   Provide for special requirements such as clock, voltage singularities, deletion of grounds, etc.

   **Inputs:**
   (1) String list (signal nets with component-pin identifications) as assigned and placed.
   (2) External connectors available.
   (3) Obstacle descriptors, drill-thru locations, module or chip arrays.
   (4) Signals to be deleted and signals to special layers.

   **Outputs:**
   (1) Cover layer tape (include all routine obstacles).
   (2) Special layers signal nets tape.
   (3) General layers signal nets tape.
   (4) External connectors mesh.
   (5) Printer plot of cover layer; errata list.

2. **Title:** Organizer

   **Function:** Inspect each signal net. Select an external where required. Produce the minimal tree for inter-connect based on straight-line distances.

   Establish sort keys for data organization for the Router. Compute slope class, distance, number of pins/net and signal priority.

   Sort as directed.
AN AUTOMATED INTERCONNECT DESIGN SYSTEM

Inputs: (1) Signal nets tape.
        (2) Control data for external selection.
        (3) Slope class criteria.
        (4) Sort criteria (sequences, keys).

Outputs: (1) Connect input tape(s) in specified sort.
          (2) Class statistics, errata, external connection list.
          (3) External connection punched cards (optional).

3. Title: Router

Function: Set the environment to memory. Route the interconnect of two points according to the organization, constraints and bounded routing area. Record the path if successful. Record the event of a failure for further routing trials on subsequent layers or update processing. Set a coordinate tape for a plotter editing. Record path in core as a layer history.

Inputs: (1) Cover layer tape.
        (2) Connect input tape.
        (3) Media parameters (grid, number layers, etc.).

Outputs: (1) Coordinate lists (for each layer) tape.
         (2) Fail list (unconnected pairs of pins) tape.
         (3) Layer or side history (cover + paths for each) tape.
         (4) Statistics and layer listing.

4. Title: Plot Editors

Function: Translate data from cover layer tape and coordinate list tape into formats required for the plotting devices (i.e., CALCOMP, GERBER).

Inputs: (1) Cover layer tape.
        (2) Coordinate list tape.
        (3) Control data (scaling, conversion factors, etc.).

Outputs: (1) Plotter tapes (cover + layers) either magnetic or paper, as required.

(2) Signal name tape (optional).
(3) Line length lists (by layer and by signal).

AUXILIARY PROGRAMS

1. Title: Router Update

Function: Reconstruct cover layer and routed history from the initial routing run. Process manually derived inputs through erasure and rerouting to attain total interconnect for plot editing and artwork. Provide sufficient errata reports on fails or invalid conditions to allow feedback through the man-machine loop. Provide outputs for checking continuity of signal interconnection.

Inputs: (1) Cover layer tape.
        (2) Coordinate list (1st run) tape.
        (3) Layer parameters
        (4) Update list (deletions, insertions).

Outputs: (1) Coordinate list tape (updated).
         (2) Listing of connections updated, fails and errata.

2. Title: Net Change

Function: Reconstruct layer history (1st run) and try to route from a list of alternate paths. Use only original failures where alternate paths are available to interconnect the signal net.

Inputs: (1) Layer history tape (1st run).
        (2) Coordinate list tape (1st run).
        (3) Net prep tape (a list of alternate routes).

Outputs: (1) Layer history tape (updated).
         (2) Coordinate list tape (updated).
         (3) Fail list tape.

3. Title: Drill Thru

Function: Apply a subsequent run to the router program with the capability of using “drill thru” locations for layer-to-layer communication.
Inputs:  
(1) Layer history tape (1st run).  
(2) Coordinate list tape (1st run).  
(3) Fail list tape (1st run).

Outputs:  
(1) Layer history tape (updated).  
(2) Coordinate list tape (updated).  
(3) Fail list tape.

RESULTS

The following table illustrates the degree of success in using the design system. Five different laminate boards (in five different hardware systems) were designed. In all cases, no drill-thru pass was used. In one case the manual update feature was employed to attain 100 percent interconnection.

<table>
<thead>
<tr>
<th>Components</th>
<th>Board Size</th>
<th>Layers</th>
<th>Inputs</th>
<th>Paths</th>
<th>Fail</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules</td>
<td>346 × 139</td>
<td>6</td>
<td>593</td>
<td>518</td>
<td>75</td>
<td>87</td>
</tr>
<tr>
<td>Modules</td>
<td>268 × 168</td>
<td>7</td>
<td>1006</td>
<td>965</td>
<td>41</td>
<td>96</td>
</tr>
<tr>
<td>Chips</td>
<td>70 × 84</td>
<td>6</td>
<td>685</td>
<td>670</td>
<td>15</td>
<td>97</td>
</tr>
<tr>
<td>Chips</td>
<td>88 × 105</td>
<td>5</td>
<td>712</td>
<td>659</td>
<td>53</td>
<td>91</td>
</tr>
<tr>
<td>Chips</td>
<td>98 × 78</td>
<td>4</td>
<td>312</td>
<td>304</td>
<td>8</td>
<td>97</td>
</tr>
</tbody>
</table>

Additional items of importance derived from this study are:

1. "Chips" on a board result in a higher yield of interconnect in a shorter period of time. This physical configuration offers a better distribution of pins about the board, simplifying the interconnect problem.

2. Presentation of the data, pairs of points, in the following sequence is the most optimal for the routing phase:
   
   a. Classification by slope of the straight line connecting the two points.
   b. Straight line distance between the two points.
   c. Minor sorts on signal priority and number of pins in the signal net.

Since the deterministic method used in the routing phase does not yield 100 percent interconnect design, in most cases, manual intervention is required. This phase involves an analysis of the failures against the plotted layers. Subsequent introduction of the failures into the machine solution is achieved through the preparation of update and insertion inputs to the Router Update Program.

Direct comparisons of manual design versus the man/machine method for multilayer laminates have shown the following results:

1. A calendar time compression to 5-10 days for the design cycle.
2. Cost reductions to the project of 50 percent or more.
3. An accuracy or reliability factor unobtainable by manual approaches.
4. Shorter line lengths for etched paths.
5. By-product provisions for automated tooling for board manufacture.

Figure 1 illustrates in flow form the programs (input/output) comprising the basics of a design system.

Figure 1

The environment of the interconnect medium must be completely described in the following areas: module or chip pin arrays, drill-thru or via locations and identifications, dimensions of the layer or board (it is highly desirable to design the board to an equally-spaced grid in both directions), cutouts, mounting, or fabrication areas where routing is not allowed, and the external connector array where signals enter or leave the board.

In general, for a multilayered laminate, the cover
layer (environment) is constant for all cases. However, some newer manufacturing techniques are being used ("Post" or "build-up") that permit deletion of pins which are net terminals in all layers below that of the interconnection. The concept here is that this deletion "opens up" further areas for interconnect routing as the process proceeds. This is, of course, highly desirable for boards of high interconnect density and tightly packed pin arrays.

This process requires "customizing" the router program to reflect the deletions. In addition, if one desired pen plots as design or update tools, it would require the generation of \( n \) cover layers for an \( n \)-layer laminate.

For the two-sided board, one would normally have two cover layers reflecting the environments on the respective sides.

The function of a Cover Layer Program is to generate this environment and append all coordinates to the net input list for subsequent program processing.

Organization of the data to meet changing requirements of board design, external usage, size, singularities of environment, and so on can very well be the key to the degree of success attained in the routing phase.

The methods employed in the router phase of a system such as this are not iterative. Thus the sequence of presentation of inputs to the router program is extremely important.

Experience in several different environments indicates that so far a typical, practical approach might include some of the following:

1. Inspect the environment for singularities which can prove useful in selecting a set of criteria. For example, a well-distributed pin population on a square board might suggest the use of slope classes in equal angle segments.

2. Also with respect to slope, for laminates it is generally good to equate the number of classes with the number of layers.

3. Specification of pairs of pins selected for interconnection of an entire signal (net) is normally the result of a minimal tree calculation based on straight line distances pin-to-pin.

4. Selection of external connectors available for signals entering or leaving the board can also be accomplished during the tree phase.

5. Other considerations in this stage prior to final sorting are number of pins in the net (this can be important in an update process), and special priorities to be considered in the routing phase.

At this point, an additional function of an Organization Program is to monitor the sorting of the data for which the various keys mentioned above have been included. For example, a typical sort sequence preparatory for routing might be:

(a) Slope class (numeric).
(b) Distance (in grid units).
(c) Number of pins/net.
(d) Priority.

The routing phase of the design system generally has little freedom in changing the sequence of processing the input data. However, a moderate amount of override capability should exist. For example, due to statistics from the organizer phase, it is indicated that several classes should be attempted on layer 1, rather than the single class originally planned. At this point it would be desirable to have the capability to modify the presentation with, say, input cards rather than returning to the organizer phase.

A rather important aspect of a routing program is the ability to "customize" the routing algorithm routines to meet particular requirements of a board or laminate. Certain restrictions or constraints can be imposed in this area which would be germane to a certain type of environment. For example, spacing between etched path and adjacent pad areas (a terminal where an interconnect already exists) might be critical on a board and require a prohibitive action. To reduce computer processing time, only the required logical inspections of the routing space are performed and these are functions of the particular board.

Outputs from a router program are normally:

1. Descriptions of the interconnect paths which can be edited for a plotting device. In previous systems these were in the form of a "from-to" terminal or pad identifications accompanied by a coordinate chain which represented every unit cell in the interconnect path.
2. A listing of all input pairs which the program failed to layout.

Finally, programs which perform the necessary translation of data into plotting device input format are required. There are several ways to develop the graphic output; one of these is to edit the cover layer (environment) and router output separately. If a pen plot is used, the cover and interconnect can be plotted in superimposition using contrasting colors for clarity. Size or scale of the plot may be as desired within the limits of the device.

The “ultimate” at the moment is production of final artwork for the interconnection surfaces. This requires high resolution and optical capabilities in the plotting device.

Within the realm of design automation, automatic interconnect design is in its infancy. Many new things are being done presently, and many more will follow in the months ahead; some of these will include: automatic board design, refinements in the analysis of data organization, and new “customized” router algorithms to accommodate advanced manufacturing techniques.