Electrocomponent Science and Technology, 1981, Vol. 8, pp. 215–219
0305-3091/81/0804-0215 $06.50/0

MINICOMPUTER INTERACTIVE SYSTEM FOR HYBRID AUTOMATION

W. SANSEN, R. GOVAERTS, W. HEYNS† and H. BEKE

Katholieke Universiteit, Leuven, Department Elektrotechniek, afdeling ESAT, Kardinaal Mercierlaan 94, 3030 Heverlee, Belgium

A computer-aided layout program MISHA has been developed for the layout of thick-film substrates. The program consists of several modules to carry out the various steps involved in the layout process. A complete layout is produced starting from a library of available components and the network-description of the circuit. Manual interference of the user with the program is possible at every step in the layout process. The program is written in standard FORTRAN IV and runs on a PDP 11/40 minicomputer.

1. INTRODUCTION

A number of computer aided layout programs already exist mainly for MOS- and PCB-design.[1–4] In order to make computer aid feasible, they all simplify the layout problem by imposing certain restrictions on the cells, respectively packages, which are used.

1.1 MOS-design

Most MOS-design computer aided layout programs start from a library of standard cells. The restrictions that are usually imposed on these cells are that all cells should have approximately equal heights and that all pins should be situated along the top-side and the bottom-side of the cells. These restrictions make it possible to perform a placement of the cells on the substrate in rows with all the pins facing clearly definable routing-channels. Because the routing-channels can be clearly defined a channel-router[5] can be used. Channel-routers are very efficient because it is very easy to implement the interchanging of equivalent pins on a cell and the mirroring of a cell in order to obtain an optimal layout.

1.2 PCB-design

Most PCB-design programs assume the use of dual-in-line packages. Placement of the packages on the board in rows again results in clearly definable routing-channels that are routed one channel at a time.

† Supported by the IWONL.

A computer aided layout program for thick-film substrates however should be able to deal with chips or packages with the pins situated along the four sides. Furthermore it is impossible to impose restrictions on dimensions, for instance, because of resistors. In thick-film technology resistors form a very distinct class of components. As a result of all this the layout problem has become more general but also more difficult.

The existing algorithms for MOS- and PCB-design and thus also the programs based on these algorithms cannot be used for the design of thick-film substrates. New programs have to be developed especially for this class of layout problems. Therefore we developed CALHYM (Computer Aided Layout of Hybrid Microcircuits) several years ago.[6] This program already featured pins along the four sides of the chips or packages but failed to incorporate resistors. This resulted in the development of MISHA (Minicomputer Interactive System for Hybrid Automation). It is this program that is presented in this paper. The differences between MISHA and CALHYM are evident during the placement and the routing. The placement can now be carried out in an interactive way using algorithms that take into account the real dimensions of the components. For the routing a new algorithm based on the expansion of lines has been implemented. This algorithm searches for a path between points on a substrate with arbitrarily placed obstacles. It guarantees that a path will be found if one exists.

As for MOS- and PCB-design computer aided layout programs the justification for a program like MISHA lies in the reduction of design costs and time. It is

important however that the user should always be able to manually interfere with the design process. In MISHA the automatic use is always complementary to the optional manual use.

2. PROGRAM DESCRIPTION

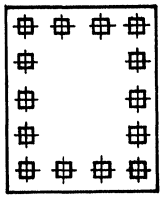
The MISHA program actually consists of two separate programs i.e. a library program and a design program. The library program is used to generate and update a library of available components. The design program produces the layout of a circuit starting from a network-description and information contained in the library. Both programs will be discussed.

2.1 Library Program

Throughout the remainder of the text we will use the word component because it is more general than chip and can be used for instance for dual-in-line packages. Instead of the words diode, transistor, gate, flip-flop, op-amp and so on we will use the word element. Each component consists of (at least) one or more elements. If a component consists of more than one element, the elements are assumed to be identical.

With the MISHA library program one can generate and update a library of component-descriptions to be referenced by the design program. By a component-description we understand an amount of data that contains all the necessary information about a component. The component is replaced by a symbolic rectangular representation with the pins situated on grid-points along the four sides of the rectangle (Figure 1). In the final layout this symbolic representation allows the component to be placed in the middle of the rectangle.

It is not until the end of a session that the library is actually updated and only if the user requires it. Otherwise the session has had no effect on the library whatever actions were taken during the session.



a.

```
component-description xxxDM74ØØ  xxx
type      = 4.NAND
#inputs   = 2.IN
width     = 4
height    = 5
#pins     = 14
#classes  = 4
sequence =
in clockwise direction starting top left
4.IN 4.OUT 1.IN 1.IN 1.OUTØ.GND 2.OUT &
2.IN 2.IN 3.OUT 3.IN 3.IN Ø.VCC 4.IN
```

b.

FIGURE 1 a) Symbolic rectangular representation of a DM7400; b) the corresponding component description.

2.2 Design Program

The design program produces the layout of a circuit in several successive steps starting from the network-description of the circuit. Additional information is to be given about the resistors in the circuit in order to have the program calculate their dimensions. The next step in the design is the selection of components from the library and the assignment of the network-elements to the selected components. This step is followed by the placement of the components (including the resistors) and the bonding pads of the circuit. The placement is followed by the prerouting. The assignment of the elements within a component is optimized and the coordinates of the pins that are to be interconnected are calculated. The design is then completed with the routing of the interconnections and the drawing of the final layout. The program can be rerun from any step in the design. All relevant data is stored in a workfile. Before executing a requested step the program first checks if all necessary data is available.

2.2.1 Network-description To describe the network one has to specify for every element the nets that are to be connected to it. The output of this step consists of the dual form of the input i.e. for every net the elements it is connected to and a list of all elements arranged according to their function.

2.2.2 Selection of components and assignment of elements The selection of components from the library and the assignment of network-elements to the selected components can be partly or completely specified by the user. He can also specify whether or not a selected component is to be considered full i.e. all elements are supposed to have been used. If possible the program will fill up the user-selected components first and select a minimal number of additional components to implement the not yet assigned network-elements.

The selection of components results in an initial assignment of elements to components. This assignment is optimized by a min-cut algorithm.[7,8] This algorithm clusters strongly connected elements and minimizes the number of interconnections between components. It is easy to prove that the min-cut criterion is almost identical to the minimization of theoretical routing-length criterion.

2.2.3 Calculation of resistors The user can immediately specify the dimensions (in grid-units) of some or all resistors. Some additional information about the resistors is to be given if the user wants the

program to calculate their dimensions i.e. the nominal resistor value and the average power dissipation. A minimal number of pastes will be selected by the program out of a set of available default or user-specified pastes. The user however has the possibility to impose the selection of a specific paste for a specific resistor. The resistor dimensions are calculated according to default and/or user-specified design rules.

In the following steps the resistors are treated as components. In contrast to the other components however the program does not reference to the library but to the workfile.

2.2.4 Placement of components and bonding pads Placement[9,10] can be done manually and/or in an interactive way on a graphic display. The algorithms available to the user during the interactive placement phase carry out a force-directed placement, a force-directed spreading, a pad-assignment optimization and a resistor-orientation optimization. They can be called by the user in any order at any time.

The force-directed placement algorithm clusters the components that are strongly interconnected. However only a relative positioning of the components results because the actual dimensions of the components are not taken into consideration (Figure 2.a).

From the actual dimensions of the components and the total substrate area the program calculates effective dimensions for the components. These will be used to resolve, or at least minimize, any overlapping between the components by the force-directed spreading algorithm. This algorithm spreads the components over the substrate by calculating an optimal displacement for every component to minimize overlapping with other components. It cannot guarantee that all overlapping is resolved (Figure 2.b).

The assignment of the pads is optimized in order to minimize the expected total routing-length. For the pair of pads under consideration the minimal additional routing-lengths to connect them to the pins of the respective nets are calculated. The additional routing-lengths are calculated as the Manhattan distances to the enclosing rectangle of the pins of the respective nets. They are calculated for the two possible assignments of the pads and compared to select the best assignment.

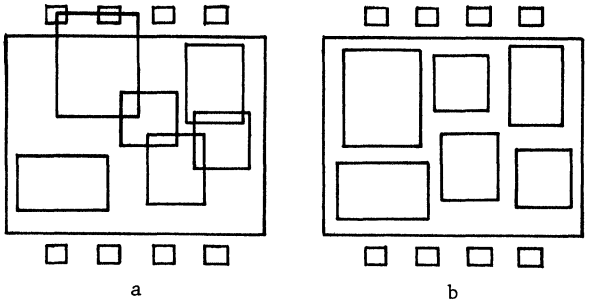


FIGURE 2 a) Results from the force-directed placement; b) force-directed spreading.

Optimization of the orientation of resistors is also done to minimize the routing-length. Because the dimensions of resistors can be fairly large compared to the dimensions of the other components, optimization of the orientation is very important to avoid long detours in the routing. Detours can be the cause of congestion and thus lead to difficulties during the routing. The interchange of the two sides of the resistor to where the net is to be connected, is almost identical to the problem of the pad-assignment optimization. The same principles are used to determine the optimal orientation of a resistor.

2.2.5 Prerouting In one of the previous steps the network-elements were assigned to components. However since a component consists of (at least one but possibly more) elements, the network-elements have yet to be assigned to a specific element within the component. This is done during the prerouting. The optimization of the assignment of network-elements to a specific element within a component is important in order to minimize the expected total routing-length and the expected number of cross-overs.

Last but not least the program assigns the nets that are to be connected to an element, to the pins of the elements and calculates the coordinates of the pins.

2.2.6 Routing The last step in the design process is the routing of the interconnections. This can be done by the user in an interactive way on a graphic display and/or by the program. As it is already indicated a channel-router cannot be used. A maze-router[11] would find a path if one exists but the disadvantages of a maze-router are the memory- and time-requirements. A line-search router[12] would be able to find a path if one exists but cannot guarantee a path to be found even if one exists. The MISHA program however includes a new line-expansion router that always guarantees a path to be found if one exists.[13]

The line-expansion router has been developed to generate interconnections on two layers. We assume that the first layer contains all the horizontal lines, while the second layer contains all the vertical lines. Interconnections can cross if they are on different layers. Extension to multilayered interconnections is evident.

The basic principles of the line-expansion router are given in Figure 3. Because a line can be crossed, obstacles to the expansion are vias, borders of components and the border of the substrate. A solution is found if one reaches the target-point or, if the expansion was also started from the target-point, if an active line belonging to the expansion of the target-point is reached. Active lines are generated if a solution is not yet found. They border the area reached by the expansion i.e. the area containing all the possible ending-points of lines starting on the expanded line and perpendicular to this line. If a solution is found, a trace-back routine is called to generate the actual path. Otherwise another line from the list of active lines will be selected for expansion. A complete layout is given in Figure 4.

A very powerful feature of the routing algorithm implemented in MISHA is the ability to make a contact with a resistor along the whole top or bottom side without specifying beforehand a specific coordinate. This means that the program is able to select the optimal coordinates for contacts in order to minimize the total interconnection length, the number of vias and (most important) the congestion probability.

It is easy to implement routing underneath the components. In this case not the borders of the components but the pins that block an expansion should be generated as obstacles.

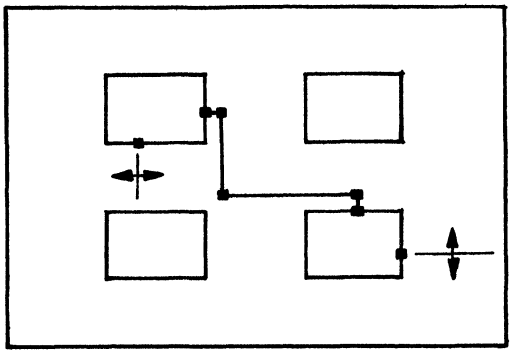


FIGURE 3a Expansion from the starting points. The arrows on the active lines indicate further possible expansion.

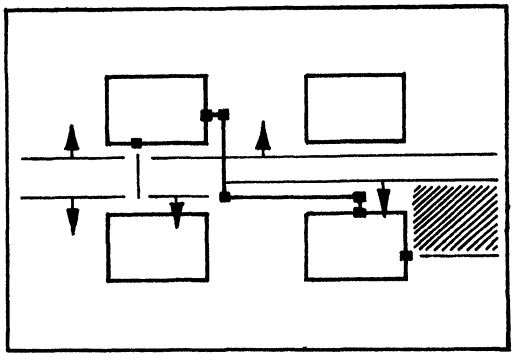


FIGURE 3b Expansion of the active lines. A solution will be found in the shared area.

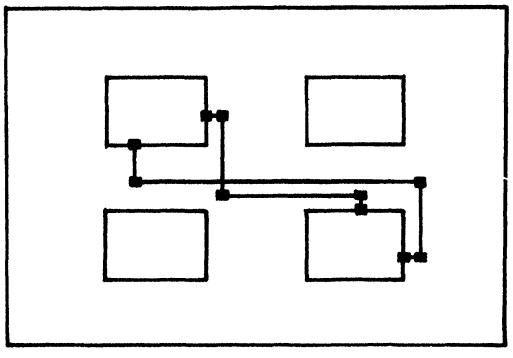


FIGURE 3c Actual path after trace-back of solution.

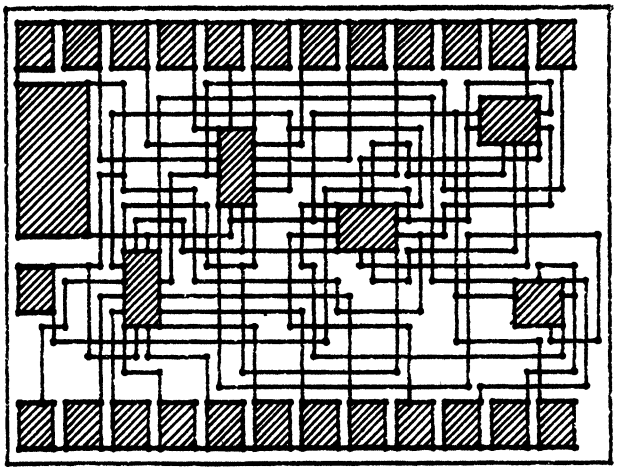


FIGURE 4 Complete layout.

CONCLUSION

The program MISHA is especially suited for the layout of hybrid circuits to be realized by means of thick-film technology. It differs from almost all existing MOS- and PCB-design programs because it imposes no restrictions on the dimensions of the components and allows the pins to be situated along the four sides of the components. These differences are evident during the placement and the routing. Whereas in MOS- and PCB-design everything is simplified to a linear

problem, here quadratic problems are handled. This is possible by the use of several new algorithms, among which the line-expansion algorithm is the most important one. At present work is done to guarantee a placement without any overlapping between the components and to guarantee a 100% complete routing. These problems are related. To solve them it will be useful to impose maximum dimensions rather than absolute dimensions on the substrate. The problem becomes even more difficult if a layout on only one layer is required. Then we have to solve the planarity problem and this implies a completely different approach to the computer aided layout of hybrid circuits.

REFERENCES

1. H. Beke, and W. Sansen, "CALMOS: a portable software system for the automatic and interactive layout of MOS/LSI", *Proceedings of 16th Design Automation Conference*, San Diego (June 1979).
2. H. Beke, W. Sansen, and R. Van Overstraeten, "CALMOS: A computer-aided layout program for MOS/LSI", *IEEE Journal of Solid State Circuits*, pp. 281–282 (June 1977).
3. G. Persky D. N. Deutsch, and D. G. Schweikert, "LTX – a minicomputer based system for automated LSI layout", *Journal of design automation & fault tolerant computing*, pp. 217–255 (May 1977).
4. "CALPCB – User's Manual", Computer aids to LSI and digital systems design, K. U. Leuven, ESAT, pp. 13.52–13.63. (Summercourse 1978).
5. A. Hashimoto, and J. Stevens "Wire-routing by optimizing channel assignment within large apertures", *Proceedings 8th Design Automation Workshop*, pp. 155–169 (June 1971).
6. H. Beke, *et al.*, "CALHYM: a computer program for the automatic layout of large digital hybrid microcircuits", *Proceedings of the European Hybrid Micro-electronic Conference*, pp. 1–9 (May 1977).
7. D. G. Schweikert and B. W. Kernighan, "A proper model for the partitioning of electrical circuits", *Proceedings 9th Design Automation Workshop*, pp. 57–62 (June 1972).
8. M. A. Breuer, "Min-cut placement", *Design Automation & fault tolerant computing*, pp. 343–362 (October 1977).
9. F. Capocaccia and A. L. Frisiani, "An algorithm for the placement of large scale integrated circuit elements", *Alta Frequenza*, pp. 109–113 (February 1970).
10. F. T. Scanlon "Automated placement of multi-terminal components", *Proceedings 8th design automation workshop*, pp. 143–154 (June 1971).
11. C. Y. Lee, "An algorithm for path connections and its applications", *IEEE Transactions on electronic computers*, pp. 346–365 (September 1961).
12. D. W. Hightower, "A solution to line routing problems on the continuous plane", *Proceedings 6th Design Automation Workshop*, pp. 1–24 (June 1969).
13. W. Heyns, W. Sansen and H. Beke,: "A line-expansion algorithm for the general routing problem with a guaranteed solution", Proceedings of the 17th Design Automation Conference, Denver (June 1980).