CALHYM: A COMPUTER PROGRAM FOR THE AUTOMATIC LAYOUT OF LARGE DIGITAL HYBRID MICROCIRCUITS

H. BEKE, S. MAZUR, R. GOVAERTS, W. SANSEN and R. VAN OVERSTRAETEN
Katholieke Universiteit Leuven, Departement Elektrotechniek, Adfeling E.S.A.T.
Kardinaal Mercierlaan 94, 3030 Heverlee, Belgium

(Received April 6, 1977; in final form June 5, 1977)

Using a library of available I.C. chips, and a network description of the circuit in terms of these chips, the program CALHYM produces a 100% complete layout of a multilayer hybrid microcircuit in three steps: the partitioning of the logic functions into library chips, the placement of the used chips on the substrate and the routing of the interconnections.

CALHYM is written in Fortran IV and runs on a minicomputer with interactive graphic facilities.

1. INTRODUCTION

A new program CALHYM has been developed for the automatic layout of digital circuits, built up of monolithic integrated circuit chips, which are interconnected by means of thick film conductors.

The program requires two inputs:

1) A library of available I.C. chips
2) A network description of the circuit in terms of library chips and/or logic functions which can be allocated to a library chip type.

Starting from these, CALHYM will produce a 100% complete layout of the microcircuit in basically three steps.

1) Partitioning of the logic functions into a number of library chips.
2) Placement of the set of used library chips on the substrate.
3) Routing of the interconnections between the different input-output pins of all the chips.

The system is very flexible, since it allows for a wide range of design limitations to be implemented (e.g. fixed distance between chips or not). Moreover a powerful interactive language stimulates the user to manually interfere with the layout process in order to optimize it.

The program is written in standard Fortran IV and only needs 32K (16 bit) words. As a result CALHYM can be run on a wide range of minicomputers, making it a general low cost quick turnaround design tool for hybrid microcircuits.

2. INPUT SPECIFICATION

One of the primary objectives being general flexibility, CALHYM requires both a library description of available chips and a network description in terms of logic functions and (or) complete I.C. chips. In fact this procedure allows any circuit containing a combination of simple logic and complete functions (available as a separate chip) to be accepted by CALHYM without bothering the designer with the difficult job of partitioning the logic in available integrated circuits.

Every library chip must be described in terms of its external dimensions, its gate configuration and its input-output pin configuration. From this data CALHYM will generate a symbolic rectangular chip with all the input-output pins on a gridded distance. In the final layout this symbolic representation (see Figure 3) allows to place the real chip in the middle of the rectangle and to bond the different bonding pads to the pins of the symbolic layout.

The input of library chips is facilitated by an interactive module that asks the designer to sequentially give all the needed characteristics. The network description of the circuit describes for every logic function the different nets that are connected to its inputs and outputs.
3. PARTITIONING OF THE LOGIC

The partitioning algorithm contains two objectives.

1) Minimize the number of I.C. chips required to implement the logic.
2) Allocate the logic in order to facilitate the routing of the interconnections.

The use of homogeneous chips (i.e. chips containing only one specific logic function type) strongly facilitates the selection problem. Therefore CALHYM will only use miscellaneous chips if it is forced to by the designer. Since most available logic chips are indeed homogeneous, one can state that this restriction is not a very severe one.

If only homogeneous chips are allowed, the minimum number of required chips is easily calculated from the set of available library types, the set of different logic functions types and the quantity of every type.

The partitioning algorithm itself is based upon the maximalisation of a "cluster value", CV,

\[ CV = \frac{n}{i=1,j=1} \sum_{(i \neq j)} \frac{C_{ij}}{T_i - C_{ij}} + \frac{C_{ij}}{T_j - C_{ij}} \]

with \( C_{ij} \) = connection strength between elements \( i \) and \( j \)
\( T_i \) = total number of connections to \( i \)
\( T_j \) = total number of connections to \( j \)
\( n \) = number of equivalent gates in the chip
\( i,j \) = logic functions that are assigned to the chip

The formula is a generalisation of the cluster value as defined by Schuler and Ulrich.\(^1\) \( C_{ij} \) is the connection strength as defined by Charney and Plato.\(^2\)

\[ C_{ij} = \frac{m}{\sum_{k=1}^{n_k - 1}} \frac{1}{n_k - 1} \]

\( m \) = number of common nets between \( i \) and \( j \)
\( n_k \) = the number of pins

The maximalisation of \( CV \) favours chips with strong internal connections and weak external connections. Practical tests show that this technique tends to minimize the total theoretical wire length, which, up to now, is the most used criterion for the minimisation of the total substrate area.\(^1-3,5,7\)

4. PLACEMENT OF THE CHIP ON THE SUBSTRATE

The aim of the placement algorithm is,

1) To define the relative positioning of each chip
2) To define an optimal orientation for each chip
3) To assign the \( n \) logic functions selected for a chip to one of the physical gates on the chip.

Once again the algorithm will try to minimize the total wire length. The relative position of each chip is calculated with a force directed pairwise relaxation algorithm.\(^9\) The system optimizes a random initial placement by pairwise interchange of chips that are strongly forced to this interchange by their interconnection pattern.

It is the user's responsibility to define the number of columns and rows that are to be used for the placement. He also has to specify the number and the location of the external connector tabs.

In a next step CALHYM will calculate (if the designer allows for this option) an optimal orientation (4 possibilities) and an optimal positioning of the different logical gates associated with the chip.

The principle of this step is shown in Figure 1.

A direction vector is associated with each input-output pin of the chip. The vector is perpendicular to the physical position of the IO-pin. All the nets

\[ \text{FIGURE 1a} \quad \text{Direction vectors of gates} A, B \text{ and } C. \]

\[ \text{FIGURE 1b} \quad \text{Direction vectors belonging to logic functions} 1, 2 \text{ and } 3. \]
belonging to a logic function of the chip come from the chip's neighbours and are also represented by a direction vector, starting from the chip they come from and pointing to the chip being worked on.

Both the orientation of the chip and the gate assignment are now done in such a way as to minimize the resultant vectors from the different gates and the ones belonging to their associated logic function (see Figure 1).

5. ROUTING

The routing problem for hybrid microcircuits is closely related to the problem of interconnecting a set of standard cells in MOS/LSI circuits. Due to the complexity of the connection pattern, a two layer system is necessary. Moreover it is not desirable to use the substrate area under the chips. As a result every chip causes a blocking problem for the interconnections (which is not true for P.C. boards).

CALHYM basically uses the modified branch and bound router. This router automatically gives a 100% complete solution for the interconnection problem in a rectangular channel between two horizontal rows of MOS-cells (see Figure 2). These cells are designed with their input-output pins coming out at only one side.

The routing problem, as arising in CALHYM, differs from this model by the fact that each chip has input-output pins at four sides. However, Figure 2b shows a method to reduce this problem to the previous one by routing first the small local vertical areas between cells.

Although the principle seems easy, a few comments on the method are necessary. First of all, there is the assign problem for "cross-overs" i.e. connections between pieces of routing that belong to different horizontal channels.

If the horizontal distance between the cells is free CALHYM assigns the cross-overs in a way that the horizontal routing is optimized. In the other case the...
dominant factor is the capacity (= width) of each vertical routing domain.

An analog problem arises if the vertical distance between the cells is fixed. To overcome all these problems, CALHYM uses an iterative procedure of four steps.

1) Local routing of the vertical domains between cells.
2) Assignment of cross-overs to vertical domains.
3) Horizontal branch and bound routing.
4) Control of horizontal + vertical capacity; change of boundary conditions for local routing.

Restart of step 1.
The purpose of the iterative procedure is
1) To minimize the substrate area.
2) To force the chips into grid positions.

6. OUTPUT

The output of the program consists of a multicolour plot on paper, representing the different manufacturing masks in overlay. It is also possible to generate a magnetic tape bearing the data needed to automatically cut these masks in rubilith.

Input data for automatic bonding equipment can be provided.

7. INTERACTIVE DESIGN

In structuring CALHYM one of the main objectives was to make it a system that allows both for automatic and interactive design. Automatisation is useful for incidental users, while strong interactivity can help the skilled designer to optimize his results. A common characteristic of all interactive modules is that they basically allow the user to change boundary conditions or initial conditions for automatic algorithms. These algorithms can be run several times and the best solution can be taken as the input for the next step. Interactivity clearly does not mean that some algorithms are suppressed and that their (difficult) task must be carried out by the designer.

8. CONCLUSIONS

This paper briefly described a system for the automatic and interactive layout of digital hybrid microcircuits.

Up to now, several sample layouts have been tried. They all gave satisfactory results in a few days of work (see Figure 3). It can thus be concluded that CALHYM can considerably reduce the design costs and time. Other advantages of the system are:

1) It runs in only 32K of memory and is written in FORTRAN, making it portable to a wide range of minicomputer systems.
2) It is strongly interactive, allowing immediate control (avoiding superfluous computations) and a better optimisation of the result.
3) The routing is always done 100% complete. So the user must not achieve the layout by trying to route manually some difficult nets.
4) The routing algorithm, implemented in CALHYM, yields area savings of about 15% compared with other routers.
5) Masks can be automatically cut.
6) Input data for automatic bonding equipment can be generated.
7) CALHYM can easily be adapted for semiconductor packages (e.g. beam leads) or chip on film carriers.
8) Layout modifications are easily performed due to the interactive control facilities.

Presently most of the research effort is done in the domain of optimizing the iterative procedure of the routing. It is felt by the authors that this procedure could also be applied to MOS/LSI layouts, allowing for cells with input-output pins at all sides.
REFERENCES


