

Macro Generation Algorithms for LSI Custom Chip Design

Presented in this paper are macro generation algorithms which have been developed and implemented for the optimization of physical and electrical designs of LSI macro circuits. Any type of logical macro circuit for a custom chip design can be automatically generated and optimized through the use of the concepts and numerical techniques for algorithmic layout generation and electrical network design which are described. The application of this design method to programmable logic array macros, which allow the generation of very attractive designs, is also discussed and illustrates the efficiency and flexibility of the macro generation concept.

Introduction

The ever-increasing circuit densities of LSI chips, corresponding to thousands of basic logical functions, lead to the necessity for new design methodologies and automation capabilities. Existing design techniques such as master slice [1], programmable logic arrays [2], and custom design [3] do not achieve the best tradeoffs between circuit density and chip development cycle time.

A design methodology has appeared by means of which an LSI chip can be efficiently designed by assembling building blocks (macros) which generally represent functional entities (registers, multiplexers, arithmetic and logic units, etc.) [4]. This concept has been intensively applied to the designs of microprocessors and associated components [5]. These experiences showed that macro-based chip designs using standard design aids for electrical network analysis (such as simulators [6, 7]) and for layout generation (such as graphics applications [3]), led to highly optimized designs—referred to as custom chip designs—with circuit densities two to three times higher than those possible with other approaches. Unfortunately, design cycle time and manpower effort were increased by a factor of five to ten. The result is that the custom design approach has been suitable only for those design applications, for example, where large numbers of one chip design are needed or where there is no alternative because of system performance requirements.

Therefore, feasibility studies of a comprehensive design automation for functional entities were undertaken. These have been generally restricted to assembling of predesigned cells stored in a library [8]. The time saved in chip design activities was significant; but, on the other hand, circuit densities were generally much lower than those achieved in custom design because the set of fixed macro circuitry existing in a library cannot exactly fit a designer's needs for a specific application unless the number of available macro types in the library is extremely high. This misfit impacts the logic design (which must conform to available functions), the electrical designs (where oversized circuits are used to meet performance requirements), and the physical design (where unused macro features cannot be removed). This is particularly true for FET technologies where the range of variation in circuit performance and in device size is quite large.

The study concluded that what was desired was a design system which would optimize a wide range of electrical and physical designs. On the basis of prior work [9-17], a system satisfying these objectives for an NMOS FET technology was developed.

The concepts, algorithms, and numerical techniques for this macro generation system are described in the following sections of this paper; most significant of these are

Copyright 1980 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

algorithmic layout generation and electrical design based on equations. When implemented in appropriate algorithms, these provide the capability of an electrical and physical macro design tailored to a user's requirements. In the last section of the paper an overview of an application of this approach to programmable logic array macros [18] is presented.

Concepts

The basic components of circuit design activities are illustrated in Fig. 1; these generally fall into three categories: process information (electrical and physical designs), design data (electrical and physical network data), and control information (layout ground rules and designers' requirements).

The main concept to be noted in Fig. 1 is that any integrated circuit may have two representations: electrical and topological. Each representation has its own set of elements: diode, transistor, capacitor, etc. for the electrical representation; and rectangle, line, polygon, etc. for the topological representation (mask data). In both representations, certain elements have constant values throughout the design process, while others are modified by the process. It should be noted first that there exists an interdependence between the physical design and the electrical design of the macro. The reason for this is that the input and output data of the physical design are, respectively, the output and the input of the associated electrical design. Secondly, the only data handled by a specific process are the elements of its related representation. In other words, the solution to the macro design problems, as mentioned in the previous section, lies in the analysis of macro network representations.

• Electrical design

Designs of electrical networks are generally considered under two aspects: a functionality aspect (dc design)—where variable element values are set to satisfy the electrical rules of the technology, and a performance aspect (ac analysis)—where the switching speed of the network is compared to the designer's requirements. Usually, both aspects are taken into account by preliminary design followed by comprehensive electrical simulation with a design tool such as ASTAP [6].

In the case of an automated design approach, the problem is to find a way to compute the set of values associated with the set of variable elements while guaranteeing functionality as well as performance. The problem is significantly simplified if the macro is no longer considered as a network of electrical elements (*i.e.*, transistor, diode, etc.), but rather as a network of basic circuits corresponding to basic logical functions (*e.g.*, NOR, NAND). The rea-

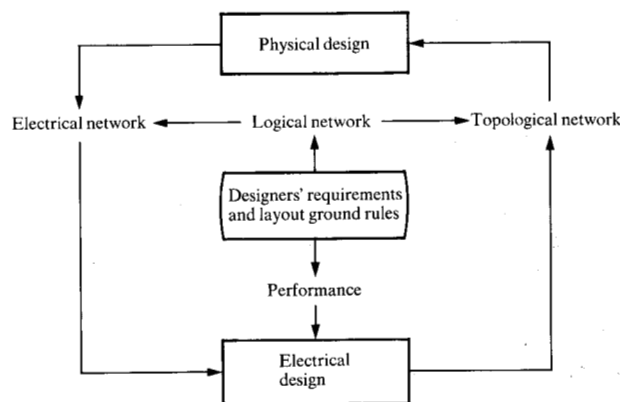


Figure 1 Basic components of circuit design activities.

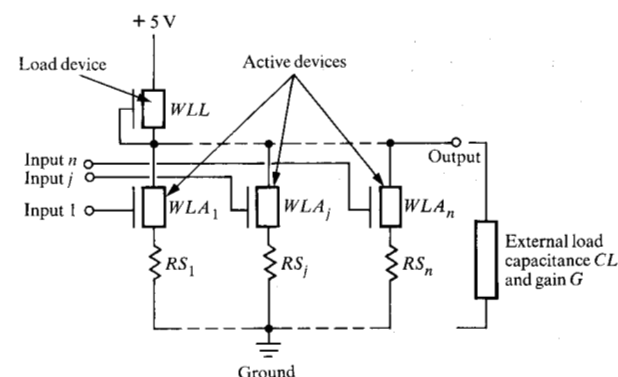


Figure 2 Basic FET circuit model.

son for this is that the number of different logical functions needed to define a macro is very small (two to three), and their electrical behavior can be expressed by a set of analytical formulas.

dc design equations Figure 2 illustrates the circuit model upon which the dc design equations are based. In Fig. 2, WLL is the size of the load device, WLA_j is the size of the j th active device, and RS_j is the source resistance related to the j th active device. The design consists of computing the values of the WLA and WLL parameters which ensure that the electrical levels are correct. The main idea is that these levels are not fixed, but continuous. This means that each circuit can be designed just to meet the requirements of the circuit it is driving. As a result, circuits are not overdesigned and gates can be smaller.

To accomplish this continuous design, a new design parameter, referred to as the "gain" G , is introduced. This

parameter, defined as

$$G = \sum_{j=1}^n \frac{WLA_j}{WLL}, \quad (1)$$

represents a measure of the least positive up level which can be observed for worst-case leakage currents. From the familiar FET electrical equations [19], and from the above gain definition, the dc design equation can be formulated as

$$\begin{aligned} \frac{1}{WLA_j} = C_1 + \frac{1}{WLL} & \left[C_2 + C_3(WLL)RS_j + \frac{C_4}{G^{1/2}} \right. \\ & + \frac{C_5(WLL)RS_j}{G^{1/2}} + \frac{C_6}{G} + \frac{C_7}{G^{3/2}} + \frac{C_8(WLL)RS_j}{G} \\ & \left. + \frac{C_9}{G^2} + \frac{C_{10}(WLL)RS_j}{G^{3/2}} + \frac{C_{11}(WLL)RS_j}{G^2} \right], \quad (2) \end{aligned}$$

where $C_1, C_2, \dots$ are derived from a simulation of the basic circuit model using ASTAP. In short,

$$WLA_j = F(WLL, G, RS_j). \quad (3)$$

In addition, it is known [14] that

$$P = K_p(WLL), \quad (4)$$

where P is the power dissipated in the circuit, and K_p is a constant determined by the particular FET process and is a function of device parameters such as the transconductance, the voltage across the device, etc.

The accuracy of Eq. (2) is a function of the constants $C_1, C_2, \dots$, and it has been verified by comparing results to dc electrical simulations. Further, hardware designed from this equation has proved reliable.

Under these conditions, macro dc design can be considered as the computation of the device sizes of the set of circuits composing the electrical macro network.

ac design equations The delay of an FET circuit is determined, to a first-order approximation, by assuming that a constant current from the circuit charges the load capacitance. This approximation is

$$D_j = \frac{(CL)V}{I},$$

where CL is the load capacitance, V is the amplitude of the output voltage, and I is the current from the circuit. However, this approximation is not accurate enough for design because the output current of the FET is dependent on its input switching time.

Therefore, the ac behavior of a basic FET circuit must be defined by both its delay (turn on-turn off) and its

switching time (rise or fall time). In practice, the ac behavior is determined by analyzing the results of simulations performed with tools such as ASTAP [6]. An electrical analysis leads to an acceptable formulation, such as

$$D_j = G(WLL, WLA_j, CL_j, SS_j), \quad (5)$$

$$S_j = H(WLL, WLA_j, CL_j, SS_j), \quad (6)$$

where D_j is the circuit delay for the j th gate of the circuit, S_j is the switching time of the circuit when activated by the j th gate, CL_j is the load capacitance seen by the j th gate, and SS_j is the switching time of the input signal which activates the j th gate.

On the basis of multiple regression techniques applied to simulation results, an accurate formulation of the ac equations may be obtained as

$$\begin{aligned} S_j = K_6 + K_7(SS_j) + K_8(RS_j)CL_j \\ + \frac{K_9(CL_j)WLA_j}{WLL} + \frac{K_{10}(CL_j)(SS_j)WLA_j}{WLL}, \quad (7) \end{aligned}$$

$$\begin{aligned} D_j = K_1 + K_2(SS_j) + K_3(RS_j)CL_j \\ + \frac{K_4(CL_j)WLA_j}{WLL} + \frac{K_5(CL_j)(SS_j)WLA_j}{WLL}, \quad (8) \end{aligned}$$

where the K are a set of constants provided by regression methods. Comparisons of computed data with experimental measurements have been made for a large number of designs. In most of the cases (90%), no major discrepancies were observed. The remaining cases generally corresponded to defects created during the manufacturing process.

• Design optimization

Optimization of complex electrical networks results in device size and power dissipation reductions leading to increased circuit densities. The result is lower product cost, and therefore design optimization is of utmost importance.

Optimization consists of designing networks such that performance meets the requirements with minimum power dissipation. A number of methods have been implemented for circuit optimization (Hessian matrix, fast gradient, etc.).

The optimization of macro-based chip designs is generally achieved by creating, for each macro, three or four designs corresponding to different power levels, i.e., performance levels, and by selecting the appropriate design according to the required performance. It was our objective to provide a more flexible and more accurate optimization capability. In addition, the computational time

consumed by the optimization must be as short as possible. Therefore, due to our specific problem formulation and our particular objectives, we were led to develop numerical methods presenting an acceptable accuracy (error <10%) and a good computational efficiency.

Assuming that there are N circuits and M data paths in the macro network to be optimized, the problem can be stated as a constrained minimization:

$$\text{minimize } \sum_{i=1}^N P_i \quad (9)$$

subject to conditions

$$\sum_{i=1}^{N_j} D_{ij} \leq DD_j, \quad j = 1 \text{ to } M, \quad (10)$$

where P_i is the power dissipated in the i th circuit, DD_j is the required delay in the j th data path, D_{ij} is the delay in the i th circuit of the j th data path, and N_j is the number of circuits in the j th data path.

To illustrate, consider the example macro network of Fig. 3, where $N = 3$ circuits and $M = 8$ data paths. In this case, the problem is to minimize the power dissipated ($P_1 + P_2 + P_3$) subject to delay constraints: $D_{11} \leq DD_1$ for path 1, $D_{11} + D_{31} \leq DD_2$ for path 2, $\dots$, $D_{33} \leq DD_8$ for path 8.

A preliminary observation leads to a reduction of the problem. Assuming that E_j is the set of gate delays in the j th data path, if for an E_j there exists at least one E_ℓ such that 1) all the elements of E_j belong to E_ℓ , and 2) the delay requirement for the j th data path is at least equal to the delay requirement for the ℓ th data path, then the j th data path can be removed from the list of data paths. An illustration of this in Fig. 3 is path 1 where $E_j = D_{11}$ and path 2 where $E_\ell = D_{11} + D_{31}$ and therefore E_j is included in E_ℓ . If, at the same time, $DD_1 \geq DD_2$, then if the requirement for DD_2 is satisfied, the requirement for DD_1 is automatically satisfied and the constraint for path 1 can be removed from the set of constraints.

By noting that in the j th data path, the switching time ($S_{i-1,j}$) of the $(i-1)$ th circuit is the input switching time SS_{ij} of the i th circuit, and by combining Eqs. (4), (7), and (8), the delay of the i th circuit can be formulated as

$$D_{ij} = \alpha_{ij} + \frac{\beta_{ij}}{P_i} + \frac{\gamma_{ij}}{P_{i-1}P_i} + \frac{\delta_{ij}}{P_{i-1}}, \quad (11)$$

where the α , β , γ , δ parameters are functions of the WLA_{ij} , CL_{ij} , RS_{ij} , and SS_{ij} .

The problem is then reduced to the minimization of a function of N unknowns (circuit powers) bounded by a

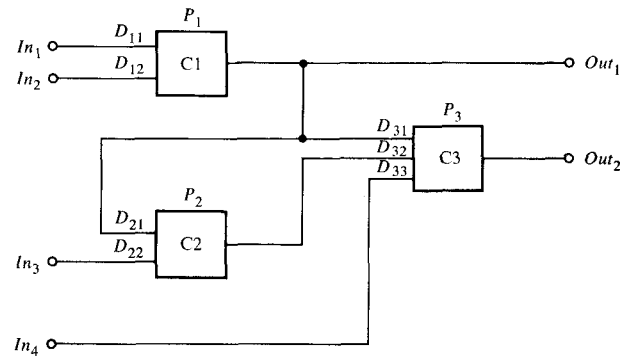


Figure 3 Example macro network for illustrating optimization problem formulation. (Note: path 1: D_{11} ; path 2: $D_{11} + D_{31}$; path 3: $D_{11} + D_{21} + D_{32}$; path 4: D_{12} ; path 5: $D_{12} + D_{31}$; path 6: $D_{12} + D_{21} + D_{32}$; path 7: $D_{22} + D_{32}$; path 8: D_{33} .)

set of M constraint relations (path delays). By the use of Lagrange's method of multipliers, the problem may be reduced to the solution of the following system of $N + M$ nonlinear equations with $N + M$ unknowns (details are given in the Appendix).

$$1 - \sum_{j=1}^M \lambda_j \left(\frac{\beta_{i(j+1)}}{P_i^2} + \frac{\delta_{ij}}{P_i^2} + \frac{\gamma_{ij}}{P_{i-1}P_i^2} + \frac{\gamma_{i(j+1)}}{P_{i+1}P_i^2} \right) = 0, \quad i = 1 \text{ to } N, \quad (12)$$

$$\sum_{i=1}^{N_j} \left(\alpha_{ij} + \frac{\beta_{ij}}{P_i} + \frac{\gamma_{ij}}{P_{i-1}P_i} + \frac{\delta_{ij}}{P_{i-1}} \right) = DD_j, \quad j = 1 \text{ to } M. \quad (13)$$

Obviously, there is not always a solution; however, since a circuit delay is bounded by a minimum value, the existence of a solution can always be ascertained by means of a preliminary computation.

• Topological representation and algorithmic layout generation concept

In attempting to solve the problem of macro design automation, we soon found that the highest priority was to make sure that we were able to automatically generate layouts for masks involving a number of varying elements and that we were achieving high circuit densities. An elementary study of layout structure led to the observations that any layout can be considered to be built from rectangular shapes and that any rectangular shape is fully defined when the coordinates of one corner and the width, the height, and the masking level of the shape are known.

Variability of a layout can always be restricted to shape sizes and to macro structure data specified by the de-

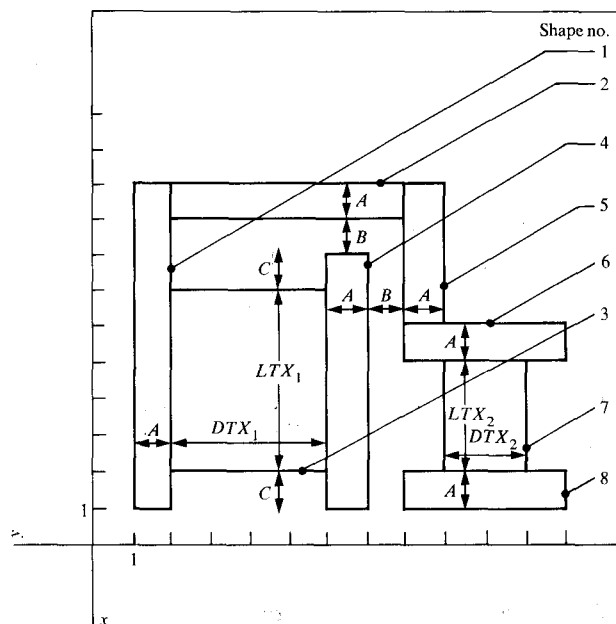


Figure 4 Illustration of topological equation concept. Topological equations for Shape No. 5:

$$XLOW = 1 + 2A + B + DTX_1,$$

$$YLOW = \min [(2A + 1 + LTX_2), (LTX_1 + 2C + B + 1)],$$

$$\text{width} = A,$$

$$\text{height} = \text{abs} (LTX_1 + 2C + B - 2A - LTX_2).$$

signer. These macro structure data are specific to one macro type. They represent the particular values of the parameters describing the structure of the macro, such as, for example, the number of input lines, the type of output (inverted or not), the number of latch cells, etc.

Under these conditions, it is postulated that any data defining a shape can be formally expressed as a function of the layout ground rules, the macro structure data, the values assigned to the variable elements, and the locations and sizes of the other shapes. Thus the entire macro layout can be described as a set of formal equations, and the translation of these equations in a computational form (program) can provide the information to allow an algorithmic layout generation.

Figure 4 gives an illustration of the concept. The A , B , and C parameters are layout ground rule parameters. The DTX_1 , LTX_1 , DTX_2 , and LTX_2 parameters are sizes of variable elements. Each shape has been arbitrarily numbered. The equations given in the figure caption provide, for Shape No. 5, the analytical expression for the coordinates of the lower left corner as well as the width and height of the shape.

Therefore, a program must be developed for each macro type to provide the layout data (shape sizes and locations) corresponding to any configuration of the macro type. Such a program requires several months to be developed, and thus it is clear that this work can be undertaken only if the macro type will have a significant enough usage.

For macros usually encountered with VLSI chips, circuit densities achieved by this method are such that the waste of space, if compared to custom design, does not exceed a few percent.

Figure 5 illustrates the high degree of flexibility and efficiency of this concept as applied to some typical macro types.

Design algorithms

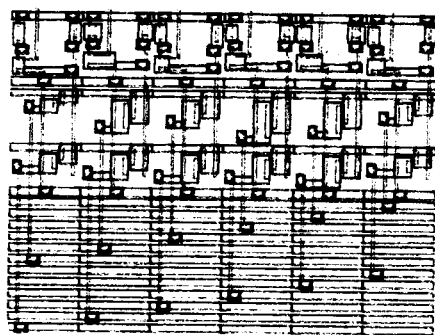
The design concepts just described, namely

1. dc design.
2. optimization based on ac design, and
3. layout generation,

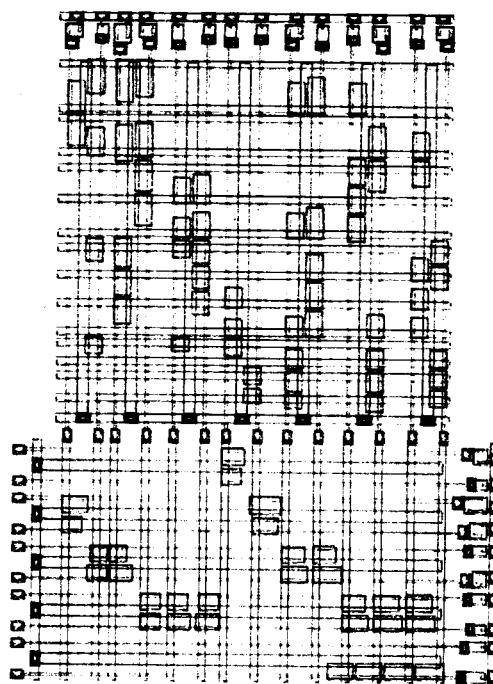
have been implemented in a series of algorithms which constitute an automatic macro generation program (Fig. 6). The second concept listed, to provide optimized designs, or in other words, to determine the set of circuit power values such that the macro performance requirements are met while the power dissipation is minimized, is embodied in the overall flow.

In Fig. 6, the inputs are specified by the designer. *Performance requirements* are the desired set of macro delays. *Macro structure data* define the content of the macro; e.g., in the case of register macros this can be the number of latch cells, the number of inputs per cell, etc. The *external conditions* are the input switching times and the output loading associated with the macro. The first step in designing a macro is to specify these inputs to the program.

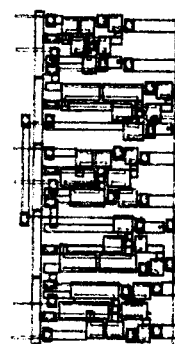
Then, a set of arbitrary power values, from which a dc design and layout are generated, is chosen by the program. This provides the electrical parameter values (source resistance, gain, and load capacitance) and device sizes for the first iteration. These data are substituted into Eqs. (7) and (8) to compute the time delays along all the data paths. These computed delays are then compared to the delays required by the designer. If the difference values do not exceed the designer's specified limits, the macro design is considered to be completed. Otherwise, a next step is processed to compute the α , β , γ , δ coefficients. Then Eqs. (12) and (13) are established and their



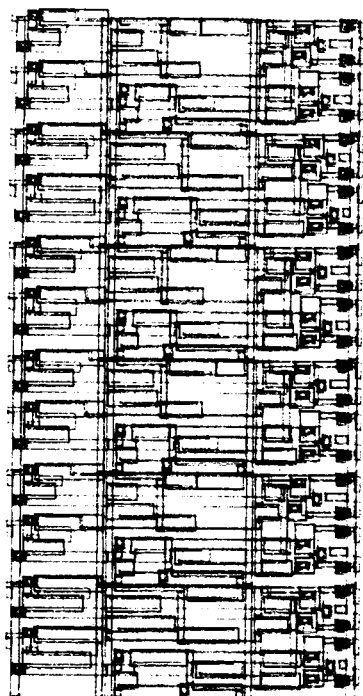
2-bus (6 lines)
multiplexer



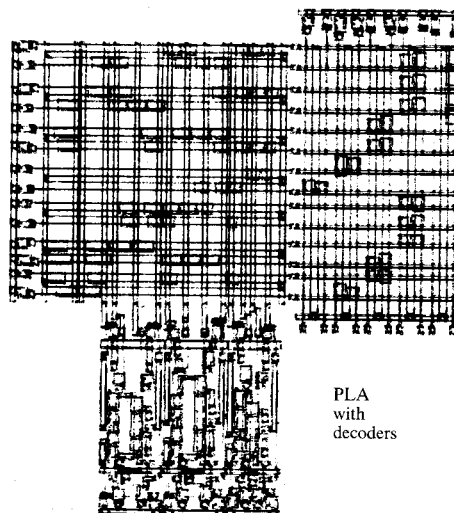
PLA (4-bit accumulator)



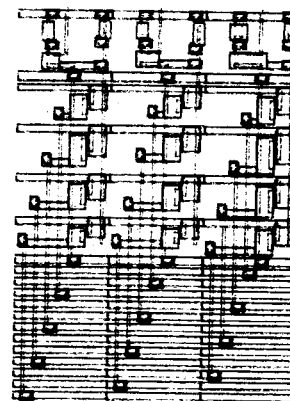
4-bit
parity tree



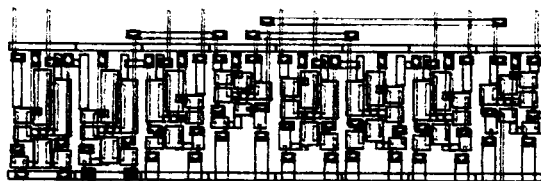
6-bit data register



PLA
with
decoders



3-bus (4 lines)
multiplexer



9-bit parity tree

Figure 5 Examples of typical macros generated by the topological representation and algorithmic layout generation method.

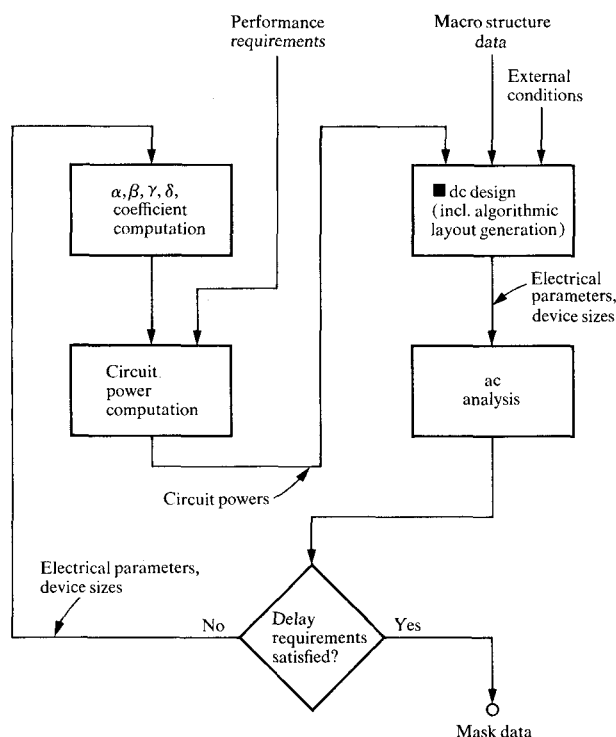


Figure 6 Macro design algorithm (■ indicates start of first iteration).

solutions computed. This provides a new set of circuit power values which initiate a new iteration starting at the dc design step.

The *dc design* block of Fig. 6 is expanded and shown as Fig. 7, which gives the detail of the dc design algorithm.

- *dc design*

Referring to Fig. 7, the layout generation algorithm provides the *layout data* (*LD*) defined as

$$(LD) = F[(WLL), (WLA), (MSD)], \quad (14)$$

where (*LD*) is the set of layout data, *i.e.*, the size and location of each shape of the macro layout; (*WLL*) and (*WLA*) are the sets of load device sizes and active device sizes, respectively, existing in the macro; and (*MSD*) is the macro structure data information defined previously.

Using Eq. 4, (*WLL*) is calculated from the set of circuit powers; the program sets (*WLA*) to arbitrary values for the first iteration, and (*MSD*) is the designer's input.

After the layout is generated, the *electrical parameters* (*EP*) are then computed. The electrical parameter calcu-

lator is a set of equations providing, for each basic gate of the macro, the values of the source resistance *RS*, the gain *G*, and the load capacitance *CL*. These values depend only on the macro *layout data* and on the *external conditions* applied to the macro inputs/outputs. Thus,

$$(EP) = H[(LD), (XD)], \quad (15)$$

where (*EP*) is the set of electrical parameters and (*XD*) is the set of external conditions.

Equation (2) provides the (*WLA*) values from the (*EP*) and the (*WLL*). The (*WLA*) of the present iteration are then compared to the (*WLA*) of the previous iteration. If the maximum difference is less than the specified minimum layout stepping, the process is terminated; otherwise, a new iteration is initiated.

As illustrated in Fig. 7, the relationships of information at the various points in the design process are

layout generation:

$$(LD) = F_1[(WLL), (WLA), (MSD)], \quad (16)$$

electrical parameters computation:

$$(EP) = F_2[(LD), (XD)], \quad (17)$$

electrical design of active devices:

$$(WLA) = F_3[(WLL), (EP)]. \quad (18)$$

By combining Eqs. (16), (17), and (18), the dc design can be formulated as

$$(WLA) = F[(WLA), (MSD), (XD)]. \quad (19)$$

The *F* function is too complex to be represented as a closed-form solution and therefore an iterative process is required to obtain a functional design. The convergence characteristics of the process depend on the nature of Eqs. (16), (17), and (18). Assuming that, when close to a solution, these functions can be approximated by linear functions, the iterative process can be stated as

$$(WLA)^p = Z + M(WLA)^{p-1}, \quad (20)$$

where (*WLA*)^{*p*} and (*WLA*)^{*p*-1} are the matrices of the *WLA* values at iterations *p* and *p* - 1, respectively, *Z* is a matrix of constants, and *M* is the matrix of coefficients of the linear application.

This is equivalent to

$$(WLA)^p = Z(I + M + M^2 + \cdots + M^p) + M^p(WLA)^0. \quad (21)$$

The solution must not depend on the initial set of values (*WLA*)⁰ and must have finite values, which implies that the limit, as *p* → ∞, of (*I* + *M* + *M*² + ⋯ + *M*^{*p*}) must have finite values. If the circuit powers are in the usual

range (less than 1.5 mW), the values of the coefficients of the M matrix satisfy the foregoing condition.

The entire global process may be summarized as follows:

$$(P) = F[(P), (MSD), (XD), (DD)], \quad (22)$$

where (P) is the set of circuit power values and (DD) is the set of performance requirements (macro delays) specified by the designer.

This relation states that from the (MSD) , (XD) , and (DD) —all these data being designer-specified—the process convergence provides the mask data for the macro.

A nonconvergence of the process may be due to one of two causes. First, there may be no solution because the delay requirements cannot be satisfied. This can be detected at each iteration by a direct delay calculation assuming that all circuits have their maximum allowable power dissipations. Second, numerical oscillations may occur as a result of either the approximation used in Eq. (11) or a discontinuity in the macro layout variations. Error from Eq. (11) is only significant when differences between the load capacitance values are quite large. This implies that the macro layout must be implemented with an awareness and control of these factors. However, oscillation must be detected by the algorithm; this is accomplished through an analysis, at each iteration, of the convergence characteristics of successive circuit power values.

An application to programmable logic array macros

One of the most attractive applications of this design approach is the automated design of programmable logic array (PLA) macros. Because of the high degree of variability of layout configurations, PLAs appear to be very appropriate for design automation. The reason for this variability is that PLA macros do not have a specific logical function. Each new function achieved by the PLA generates a new layout with its particular number of input lines, output lines, word lines, personality, decoder types, and folding characteristics. The number of configuration parameters (MSD) and their impact on the layout make manual design of PLA macros highly inadvisable.

However, this variability can be efficiently handled through algorithmic layout generation. Moreover, electrical designs of PLA macros are very sensitive to the personality information and cannot be easily produced without using the electrical equation concept. An automatic tailoring of the electrical design to the personality data is a major advantage of the automated design approach.

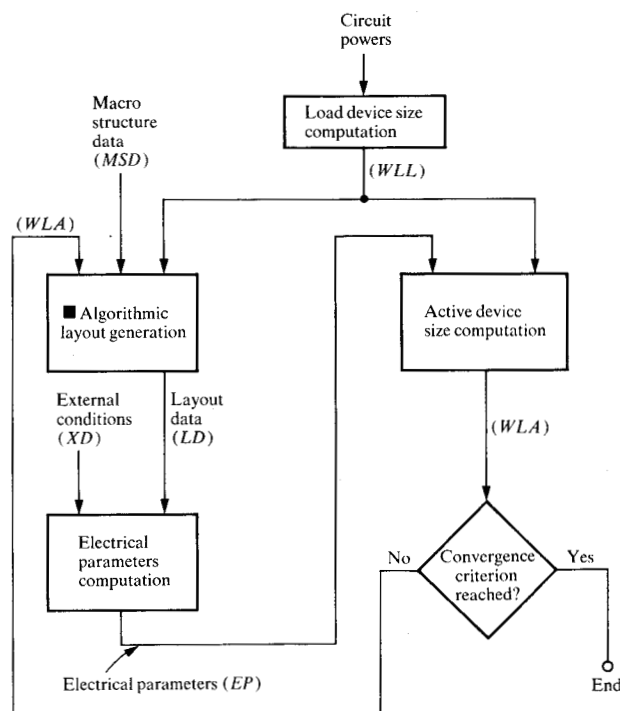


Figure 7 dc design algorithm detail (■ indicates start of first iteration).

The most important features appearing in automated PLA macro design procedures [8] can be supported by the topological equation technique. These are the following:

- array folding,
- mixed single- and double-bit decoder blocks,
- power and clock lines bussing,
- gate sharing, and
- input/output line location specification.

A set of parameters can be specified by the designer allowing a high degree of variability and flexibility of design. Basically, these parameters are as follows:

- the number of input lines, output lines, and word lines (MSD),
- the personality data (MSD),
- the macro power of the macro delays (*performance requirements*),
- the folding and gate sharing options (MSD), and
- the input and output locations (MSD).

The global structure of PLA macros can always be pictured as an assembly of decoder and array blocks. Therefore, the overall design must include a sequence of block

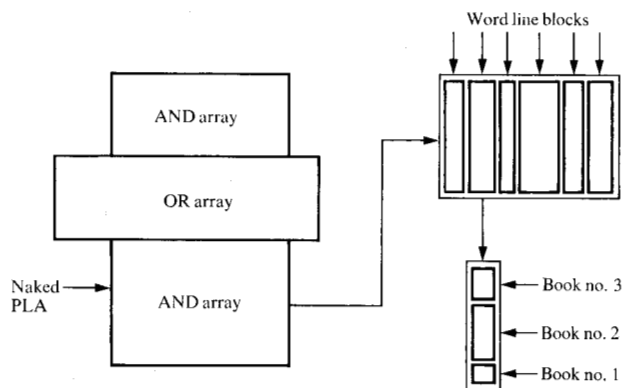


Figure 8 Array layout structure.

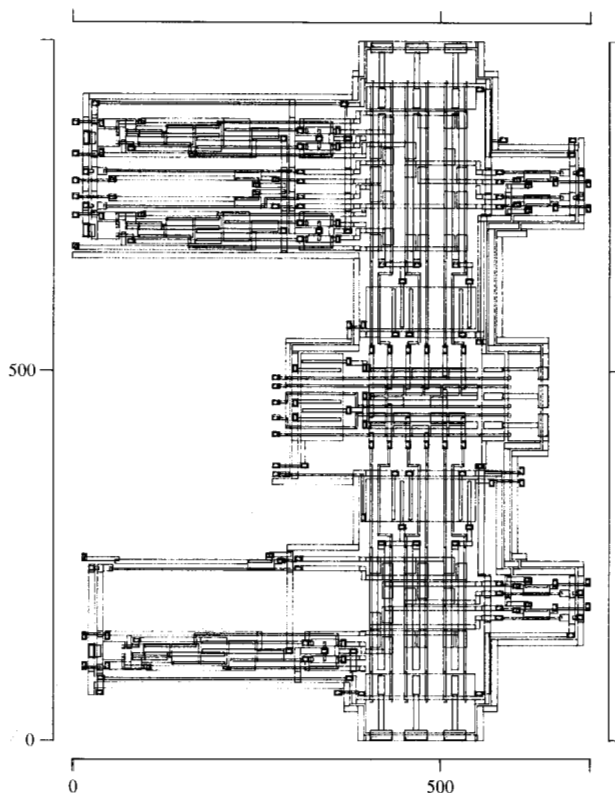


Figure 9 PLA macro layout for a folded, four-bit input-partitioned PLA with gate sharing.

designs followed by an assembly phase. Discussion here will be limited to the algorithmic generation of the array block layout, which probably is the most challenging aspect of PLA macro generation.

The objective is to develop an algorithm as independent of technology rules and circuitry type as possible. Array layouts can be considered as cellular and hierarchical structures composed of a limited set of patterns. Therefore, the structural and hierarchical relations existing in array layouts can be expressed as an algorithmic process.

Figure 8 illustrates this idea and shows its implementation. An array block is composed of two or three arrays. Each array consists of a stack of word line blocks which are built from three books. Each book is described by a set of topological equations. Personality shapes are laid on the array structure after its assembly. Circuitry type and technology rules impact only the design of the books. The main difficulty in designing the array generation algorithm is in the requirement of ensuring pitch compatibility between the OR array and its associated AND arrays. This difficulty is increased by the electrical design provision for large differences in gate size for the macro and by the gate interleaving capability supported by the algorithm.

Design examples and conclusions

Figure 9 illustrates an example of a PLA macro layout produced by the macro generation program and gives an idea of the degree of sophistication which can be supported.

Figure 10 shows the required CPU time (IBM 370/158) for PLA macro generation as a function of the size of the PLA arrays expressed in terms of the number of personality bits. These data have been extracted from chip design experiences and give a good indication of the limitations and efficiency of the algorithm.

Designs of several LSI chips using a macro generation program incorporating the algorithms described in this paper have indicated a significant reduction (two to three times) in design cycle times. Additionally, these designs demonstrated that automated optimization of large networks is possible and provides highly sophisticated and dense layouts. Most of the new concepts developed for this purpose appear to be fitted to a very large domain of applications and FET technologies. In conjunction with macro placement and interconnection algorithms, they could provide a completely automatic method for designing macro chips and thereby significantly reduce the physical chip design time.

Appendix

Assuming that $S(X_1, \dots, X_N)$ is a function of N unknowns subject to M constraint relations $R(X_1, \dots, X_N) = 0$, there exists a set of M coefficients $(\lambda_1, \dots, \lambda_N)$ such that

the extremum (maximum or minimum) of the S function is the solution of the following system of $N + M$ simultaneous equations:

$$\frac{\partial S}{\partial X_i} + \sum_{j=1}^M \lambda_j \frac{\partial R_j}{\partial X_i} = 0, \quad i = 1 \text{ to } N,$$

$$R_j(X_1, \dots, X_N) = 0, \quad j = 1 \text{ to } M,$$

where the λ_j are the set of Lagrange multipliers.

Application of this method, to Eq. (9) as the S equation and to Eq. (10) as the R equation, leads to the formulation of Eqs. (12) and (13).

The validity of the method requires that the R relations have continuous derivatives; the characteristics of Eqs. (9) and (11) satisfy this condition.

References

1. K. A. Chen, M. Feuer, K. H. Khokhani, N. Nan, and S. Schmidt, "The Chip Layout Problem: An Automatic Wiring Procedure," *Proceedings of the 14th Design Automation Conference*, New Orleans, LA, June 1977, pp. 298-302.
2. H. Fleisher and L. I. Maissel, "An Introduction to Array Logic," *IBM J. Res. Develop.* **19**, 98 (1975).
3. P. Carmody, A. Barone, J. Morrell, and C. Lovejoy, "An Interactive Graphics System for Large Scale Integration Design," *Proceedings of the International Conference on Interactive Techniques in Computer Aided Design*, Bologna, Italy, September 1978, pp. 281-294.
4. J. C. Logue, "Large Scale Integration," *Proceedings of the Symposium on Microelectronics*, Munich, Germany, October 1966.
5. "Single Chip 8-bit Microprocessor," *Electronics International* **48**, 99 (1975). Also, "High Density Bipolar Computer Design," *Electronics International* **49**, 81 (1976).
6. *Advanced Statistical Analysis Program (ASTAP)*, Program Reference Manual, Program Number 5796-PBH, IBM Data Processing Division, White Plains, NY 10604.
7. L. N. Nagel, "SPICE 2: A Computer Program to Simulate Semiconductor Circuits," *Memorandum No. ERL-M520*, Electronics Research Laboratory, University of California, Berkeley, May 1975.
8. R. L. Golden, P. A. Latus, and P. Lowy, "Design Automation and the Programmable Logic Array Macro," *IBM J. Res. Develop.* **24**, 23 (1980).
9. G. C. Brown and G. V. Berger, "Design and Optimization of Circuits by Computer," *Proc. IEEE* **118**, 293 (1971).
10. A. E. Ruehli, N. B. Rabbat, and H. Y. Hsieh, "Macromodeling—An Approach for Analyzing Large-Scale Circuits," *Computer-Aided Design* **10**, 121-130 (1978).
11. G. Arnout and H. J. DeMan, "The Use of Threshold Functions and Boolean-Controlled Network Elements for Macromodeling of LSI Circuits," *IEEE J. Solid-State Circuits* **SC-13**, 326-332 (1978).

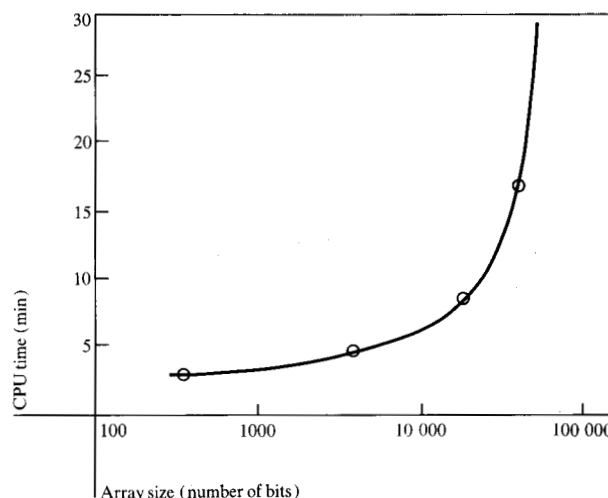


Figure 10 PLA generation algorithm computation time as a function of array size.

12. M. Glesner and K. Aubrichs, "Computer Aided DC Design," *Proceedings of the European Solid State Circuits Conference Symposium*, September 1975, p. 7.
13. F. T. Mellert, "Mathematical Formulation of Switching Delays for the Optimization of Digital Circuits," *Frequenz (Germany)* **32**, 24 (1978).
14. W. J. Sticks, "A New Approach to LSI Design," Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1977.
15. R. K. Brayton and S. W. Director, "Computation of Delay-Time Sensitivities for Use in Time Domain Optimization," *IEEE Trans. Circuits Syst.* **CAS-22**, 910-920 (1975).
16. P. Wolff, A. Ruehli, B. Agule, J. Lesser, and G. Goertzel, "Power/Timing: Optimization and Layout Techniques for LSI Chips," *J. Design Aut. Fault-Tol. Comput.* **1**, 145-164 (1978).
17. S. M. Kang, "An Optimal Design of LSI CMOS Polycells," *IEEE Proceedings International Symposium on Circuits and Systems*, Houston, Texas, April 1980, Vol. 3, p. 1008-1010.
18. J. W. Jones, "Array Logic Macros," *IBM J. Res. Develop.* **19**, 120 (1975).
19. W. N. Carr and J. P. Mize, *MOS/LSI Design and Application*, McGraw-Hill Book Co., Inc., New York, 1972, Chapters 2 and 4.

Received December 5, 1979; revised May 2, 1980

The author is located at the IBM Essonnes Component Development Laboratory, Corbeil-Essonnes, France 91102.