CONTRASTS IN PHYSICAL DESIGN BETWEEN LSI AND VLSI

William R. Heller

International Business Machines Corporation
P. O. Box 390
Poughkeepsie, New York 12602

ABSTRACT

In the last five years, there has been rapid growth in logic and memory chip circuit density. The number of different digital processors and the typical size of such processors has also grown. With all this growth, alternatives in VLSI design style as well as packaging have to be considered. These consist, on the one hand, of powerful automated placement and wiring routines, indispensable on large regular package images, and, on the other, of techniques facilitating rapid, interactive adaptation of functional logic design to the layout and interconnection of "macros" on large chips. Some results from study of each method are presented.

INTRODUCTION: GROWTH TRENDS IN SYSTEMS AND TECHNOLOGY

Recently, every year has seen a new achievement in circuit count and circuit density on silicon chips and a growth in the typical content of newly announced digital systems. The latter growth rate is not as high as the chip circuit growth. Figures 1 and 2 illustrate these points. Together with these rates has come an increase in concern with methods of design. In part, this concern comes from the increased difficulty and the duration of the logical and physical design efforts, now that one has to consider more circuits on a single chip, as well as more circuits in a given system.

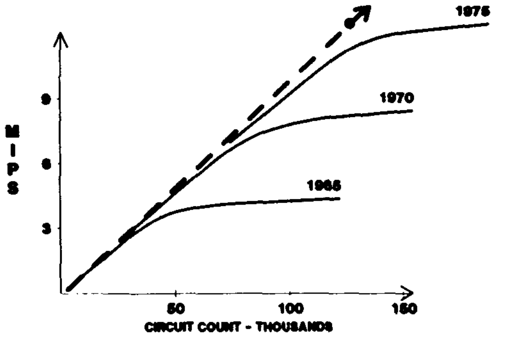

Figure 1. Processor circuit growth

An implication is that complex systems are usefully discussed as hierarchical structures in order to manage their design. In this paper, we aim first to discuss in some detail the mutual influences of the logic design hierarchy and the physical design hierarchy under the pressure of VLSI. This is followed by a treatment of the effects of VLSI in modifying the design aids available in LSI, even when regular physical structures form the package hierarchy of choice. Then we discuss, using examples as well as a recent algorithmic proposal, the effects of VLSI when functional differences in subsystem parts force the use of a combination of physically quite distinct part numbers or "macros" to make up the system, which may consist of only one or a few chips. Finally, we summarize key issues and point to some unresolved design-aid problems to which the next few years may bring solutions.

INFLUENCE OF LOGIC DESIGN HIERARCHY ON PHYSICAL DESIGN HIERARCHY

A previous paper [1] sketched the design flow for a large computer using LSI parts and a regular package hierarchy. Design of such machines may involve twenty-five to one hundred technical people, each taking responsibility for a share of what is a complex task, measured either in circuit count or in the number of different technical and economic issues which must be considered. One may compare

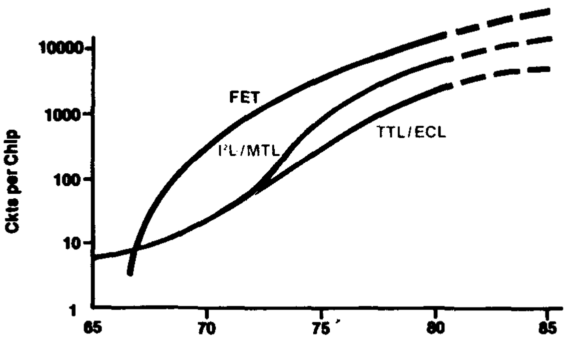

Figure 2. Logic circuits per chip

0146-7123/81/0000-0676$00.75 ©1981 IEEE

this activity to that of the development of the S-1 machine [2] at Stanford and the Lawrence Livermore Laboratory by MacWilliams and Widdoes. Essential differences come from the choice by the latter of the simplest feasible physical hierarchy for a high-speed special purpose machine (plugged-in standard components on a wire-wrapped board), as well as the necessity in commercial machines for extensive preparation and build-up of production, testing, and assurance activity involved in large-scale manufacturing. The latter tasks are all made particularly exacting when technology development must accompany system design and evolution of product design aids. From the point of view of organization of this complex whole, the ideal arrangement is that the various tasks making up the total job are "loosely coupled," meaning that they can be done individually by members of a product team in one or a few passes over the entire machine. It is a chief virtue of regular packaging in LSI and VLSI, that this loose coupling can be readily achieved, and yet permit savings (in both design and manufacturing costs and schedules) which come from the similarity of component parts. Thus, for example, a single design system, including both logical and physical design aids, can be simultaneously used on many gate array (master slice) chips, together with their higher level packaging components. Figure 3 shows the design flow in such a procedure.

It has been characteristic of logic design thus far, that the earliest phase of architectural realization is done informally and with few computer aids. These are used for instruction level simulation: pipelining for fast arithmetic, the problems of cache size and structure, and, in general, the minimization of the number of cycles per instruction in a synchronous machine. Partitioning of logic to chips or chip carriers has traditionally been done with minimal assistance from computer programs. When a known logic structure (gate level) is to be adapted and revised, programs permitting "remapping" parts of existing machines may be used [3]. These may be interactive and semiautomatic, so that the designer may guide successive transformations so as to balance the number of logic levels, the number of logic devices, their interconnection count, and the number of inputs and outputs in a cluster which may be destined to be a single chip. Donath and Hitchcock [4] have shown that there is an inverse relationship between the Rent exponent characterizing a partition of combinational logic, and the number of logic levels contained in it. Statistics from logic parts bear this out as a trend. Thus logic speed and limited package connector count must be traded off.

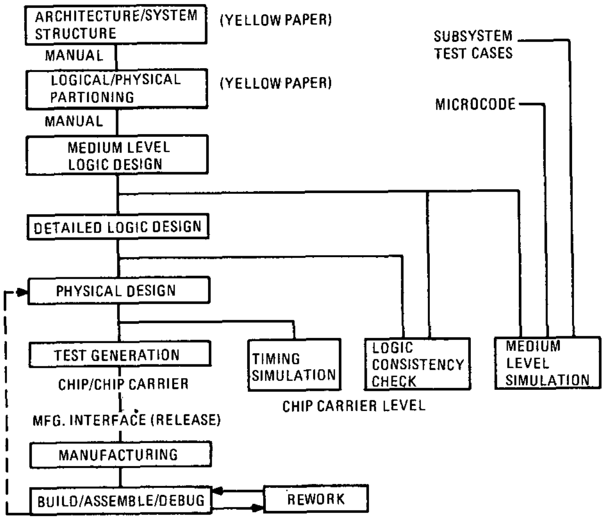

Figure 3. Hardware design process

When a new section of a design or an entirely new design is to be evolved, it is possible to give the designer the capability to work with flow charts which express logic relationships. Programs then are required to compare these to gate level implementations, to establish equivalence, or reveal discrepancies. The flow chart cannot be directly related to usable partitions of gate level logic, without the transformation tools just mentioned.

A paramount necessity in large machine design is the capability for making incremental changes in the logic. These modifications may arise from changes in design intent, correction of logic errors, or, especially in the later stages, from timing or density constraints. The latter are usually influenced by wiring delays or wiring congestion, again highlighting the relation between physical and logical design. It is clear that, in VLSI, these changes, which now require iteration of physical design of entire chips and/or their carriers, must be minimized. Materials processing which permits incremental and automated changes to already manufactured logic chips is not yet available, although VLSI could benefit by use of it.

Still other constraints placed by logic design on physical design come from the sometimes conflicting requirements of testability and reliability. A subject of increasing interest in the era of VLSI is that of incorporation of circuits on the chips aimed at testing, the logic design. One such tactic, which of course also requires extra chip input and output pads, is LSSD [5]. Experience with this design procedure, limiting the design to combinational logic contained between shift register latches or package connectors, shows a cost of five to fifteen percent in extra circuits. It is clear that error correction circuitry carries an analogous penalty. In VLSI, it is likely that fault-tolerant designs will evolve which require redundant logic [6]. One has the additional difficulty here that testability is difficult or impossible to guarantee unless special testing equipment or connections are permitted beyond the normal functional connections.

One must mention the constraints imposed on physical design by electrical limits; these indirectly restrict the logic designer. Examples are: (a) fan-out or wire length limits due to device drive capability, (b) wire length minima or cluster configurations set by line reflections of voltage pulses, (c) noise coupling from extended wire proximity, (d) nearly simultaneous switching of circuits which drive off a package, creating inductive voltage swings (Delta-I effect). The last-mentioned is of increasingly vital importance as computational switching speeds reach the gigahertz range.

All the preceding has general validity for the interaction of logical and physical design. Special additional problems arise when, as in VLSI, one is led to design digital systems and subsystems as single- or few-chip packages. So far, we have discussed the situation in which the logic design can be partitioned so as to have a whole group of designers working on different but similar parts, with identical design aids and design algorithms. In contrast, the system which can be accommodated on one or a few chips demands distinctive design tactics for its different functional parts. Here the central fact of VLSI design becomes obvious: the silicon must bear practically the full burden of sharing space among package connectors, logic devices and their contacts, power bussing and signal wiring, as well as any extra components constituting or substituting for lumped parameters. The existence of numerous fabrication levels complicates chip processing while helping to simplify the separation of the above requirements.

Matching these changes in physical realization of design from MSI and LSI, a different organization of talent and a different set of computer-oriented design aids are now required. It is essential that only a relatively small design team be entrusted with <u>both</u> the logical and physical design. Now the coupling among these design aspects is tight, and the designers, as well as the design aid developers, must understand and be able to make compromises between logic function and speed on the one hand and layout restrictions on the other. This is particularly obvious when one considers the assignment of input and output connectors on a layout "macro" incorporating a typical logic function (e.g., ALU, register file, control PLA). It is now essential that the embodiment of the logic permit assignment of the input/output (I/O) pins to suit global wiring requirements to other layout macros. In general, this will require iterative design of the interconnected macros, if one is to minimize the wiring penalty internal to the individual macros, as well as the global area requirements.

Among the most critical tasks are those assuring the correctness of both the logic and the layout with respect to function and to known tolerances, respectively. These two tasks force the development of relatively novel design tools in VLSI. A hierarchical structure makes possible checking of logic function at and between the hierarchical levels, and interactive semi-automated tools facilitate this in system design [7,8]. Timing can be assured during logic design, again with interactive tools [9]. An important set of recent developments permits passing automatically to actual cell layouts obeying processing tolerances [10,11]. Use of these tools speeds up what still remains a time-consuming task--the completion of a complete, though structured, custom design. Lattin [12] has pointed to the importance of reducing this design duration, and hence to the importance of versatile and efficient design aids in VLSI.

<u>ALGORITHMS AND ANALYSIS ENHANCING WIRABILITY IN REGULAR PACKAGE IMAGES</u>

A familiar image to designers and users of digital systems is the card or board with plugged components in a regular array. The hierarchy of physical structure exemplified in this image does not necessarily lose relevance in VLSI. As discussed in earlier sections, growth in machine size accompanies growth in the circuit density of package components. A critical aspect of such components is their pin/subpackage ratio. As seen in Figure 4, when the number of logic components at a given level of the physical hierarchy is

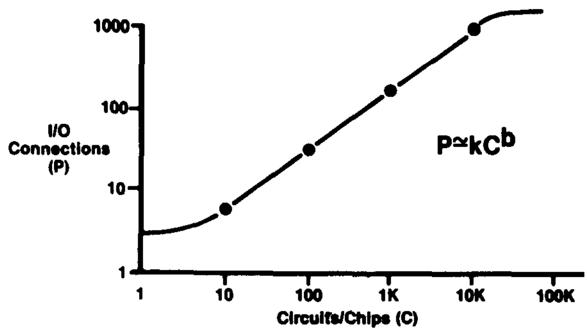

Figure 4. Interconnections

sufficiently large that a single system function requires numerous subpackages for its implementation, the package connector count rises as a fractional power of the number of subpackages contained on the package (Rent's rule). An important deduction from this result was made by Donath [13]. He showed that the direct implication of Rent's rule in hierarchical two dimensional logic arrays is that the average length of wires interconnecting members of a regular array increases with the number of logic subpackages in the array. One can see [14] that the average length of "wires" interconnecting randomly placed points in the plane grows with the 1/2 power of the number of points, N. Donath's result is that, in hierarchically placed two dimensional structures of logic units, the growth rate goes as N to the power (p-1/2). The symbol p is the Rent exponent, commonly found to be about 2/3. We can see that the error made in assuming random placement, about twice too high with ten placed objects, rises to seven times larger at 100 objects, of course ten times larger at 1,000, and so on.

If a "top-down" approach to a hierarchical structure is adapted in VLSI, we can see that the I/O pins of the substructure should be assigned for the benefit of the higher level, as one progresses down the hierarchy to the chip level. While this means that the longest wires in the hierarchy (cables) are made to approach minimum length, there is a penalty in that the I/O connector assignments at successively lower levels are then not optimized for the wires internal to that level. This will be less serious, the larger the number of units which must be placed at a given level, since the ratio of pins to units at a given hierarchical level goes down as the number of units increases, so long as Rent's rule holds true. Experiment with effective algorithms for pin assignment and placement shows that this penalty is about five percent in required extra wiring tracks or in average wire length on a master slice chip with about 700 circuits to be placed, and about 100 I/O pads. Conversely, the relative savings in average global wiring length at the next level up of the hierarchical structure is larger since the number of units at a given hierarchical level in the system must go down as one goes toward the top of the hierarchy. Figure 5 shows how the part number count may vary in machines of different size.

In practice, things are more complicated. First, it is often true that the system, or parts of it, for reasons of economics and schedule, must be designed "bottom-up" so that chip pads are assigned for the benefit of the chip wiring. Second, even though one must be primarily concerned about average wire length in optimizing wirability, the designer, in practice, concentrates during the design on certain nets he knows to be critical for performance reasons. These nets are set as short as possible throughout the hierarchy. The penalty is that this constrains placement and pin assignment and lengthens the placement of the great majority of wires, which are not judged as critical. Third, it is inevitable that package electrical characteristics at various hierarchical levels are quite different. This leads to problems in impedance matching, in off-level drive capability, and in shortening of critical nets, particularly at that package level with the least favorable propagation speed. Again, a penalty is taken in the reduction of overall wirability in order to favor critical nets.

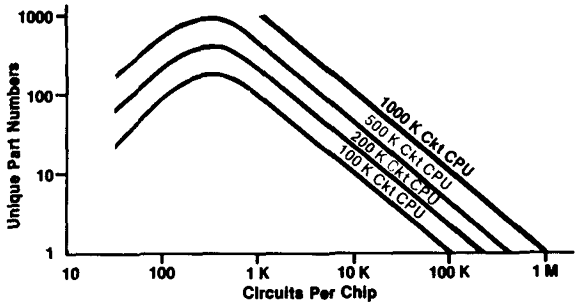

Figure 5. Part number proliferation

Overall, we can characterize the joint problems of pin assignment, placement, and wiring at a given hierarchical level by three variables useful in estimation of required wiring capacity. These are the number of units to be placed, the number of wires per unit placed, and the average length of these wires. Thus the placement and pin assignment enter the model developed to predict wiring capacity via the length estimate. Early work on wirability was done in IBM by L. Poch, W. Thompson and W. Vilkelis. A probabilistic model incorporating the variables above mentioned was presented by Heller, Mikhail and Donath [15]. Figure 6 shows estimates for wiring track capacity deduced from this

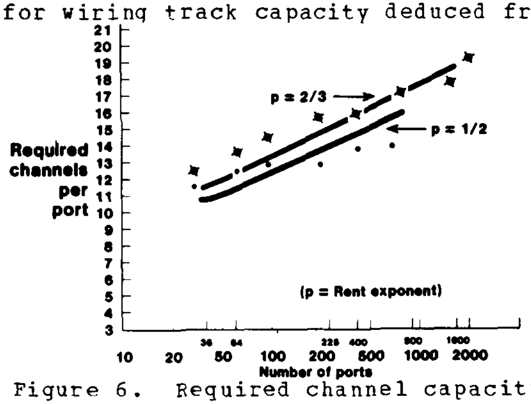

Figure 6. Required channel capacity versus number of ports

model, together with experimental results from an efficient program.

The success of the probabilistic wiring model in characterizing physical design has been borne out in a recent VLSI master slice chip designed by Davis, et al. [16,17]. The algorithms developed for placement and wiring this chip had been developed originally for less dense chips [18,19], and the new work [20] shows that the rate of increase of required computer time for completion of automatic placement and wiring goes roughly as a power 1.1-1.2 of the number of units dealt with. The essential features of these automatic algorithms are: (a) they work hierarchically, first handling groups of circuits; (b) both placement and wiring smooth global wiring demands initially, before proceeding to detailed wiring; (c) the final detailed wiring is essentially done by channel routing, followed by a "clean-up" maze runner restricted by wiring "window" size. Improvements in this approach will have to come from more detailed consideration of electrical constraints, some schemes for iteration among different packaging levels, and, possibly, special purpose machines to handle the characteristic algorithms.

ANALYSIS AND ALGORITHMS FOR CUSTOM PHYSICAL DESIGN OF STRUCTURED CUSTOM CHIPS AND PLANAR PACKAGES

In characterizing the physical design problems for custom planar packages comprising digital systems, one must deal with the varying pin and space demands of functional units (e.g., read-only stores, arithmetic logic units). Chips incorporating field-effect transistor (FET) technology have had to deal with these problems since the advent of the first microprocessors (INTEL 4000 and 8000 series, MOTOROLA 6800 and 68000, etc). Designs of so-called "special" cards, on which are accommodated chip carriers of different size and pin count, form a problem set of geometrical and topological aspects similar to those of custom chip designs. Figure 7 shows some of the differences between designs in custom and regular packages.

	Memory Microprocessor Unit Logic/MSI	VLSI Random Logic
Design Cost	High	Low
Part Number Count	Low	High
Batch Size	Large	Small
Mfg. Cycle	Long	Short
Inventory	High	Low
Testing	Functional	Stuck Fault Pattern Generation

Figure 7. VLSI implementation alternatives

A question which immediately occurs to the system and package designer of custom planar packages concerns the connector requirements, both within and on the package. As already pointed out, sufficiently near the top of a hierarchically organized system structure, where functional characteristics become prominent, one can expect Rent's rule to fail. That is, the number of I/O connectors will no longer rise as a positive power of subpackage circuit count. Instead, it must actually decrease--after all, only the power lines need come out of the box, if communication to other boxes is not a prominent feature of the system.

Work to check quantitatively on the above remarks was done by the author [21] in a study of the INTEL 8085 processor and of three processor component chips designed at Caltech. Figure 8 shows how the Rent exponent behaves at three successive hierarchical levels in the structure of the Caltech OM-2 data flow chip, bearing out the earlier description of behavior at the functional level. It is clear that a related influence on reduction of pin count at any level of a hierarchical system structure can come from a compromise between speed and internal storage. That is, one may serialize successive signals from numerous internal devices if one sacrifices overall speed by providing circuits to store results of partial calculations until sufficient data are accumulated to pass along a complete message string through one or a small number of output connectors.

Overall, the benefits of shaping the layout and tuning the communication of functional units sharing a custom FET chip come from three sources: (a) even if each functional box is built by regular repetition of a cell structure, the cell belonging to a function can be tailored in device and pad arrangement to suit that function; (b) switching devices, rather than NAND or NOR gates, can be interconnected directly to define a cell function; (c) advantage can be taken in the global layout of different power requirements for the different functions (macros) making up the entire planar package. The net reduction in device area alone, using the first two capabilities in

	Top Level	Middle Level	Lowest Level
Caltech OM-2 Data Flow Chip (Dave Johannsen)	T_1 = Number of ≈ 56 Chip Signal Pads	T_2 = Av. Function = 78 Block Connector Count	T_3 = Av. Logic Cell Connector Count = 11
	M_1 = Number of ≈ 18 Function Blocks	M_2 = Av. Number of = 143 Logic Cells Per Function Block	M_3 = Av. Device Count Per Cell = 9
	p_1 = Rent Exponent = 0.067	p_2 = Rent Exponent = 0.515	p_3 = Rent Exp. = 0.583

Figure 8. Best fit to Rent's rule at three hierarchical levels of design

FET microprocessor chips, is abcut two to four over a gate array implementation of the same functions. It is found that global and local (intercell) wiring together require about 50 percent of the available space, including all three physical levels (diffused silicon, polysilicon and metal) of a typical FET chip containing about 10,000 devices. The relative importance of the wiring requirement will increase at higher circuit densities.

The chief penalties for this space-saving are fourfold: (a) design checking of such a layout is more difficult because of the increased variety and irregularity of the units; (b) when space is utilized very efficiently in a relatively complete design, engineering changes may ripple through the entire structure in their effects; (c) design times for the entire complex image will greatly exceed the design times for less dense, but very regular implementations, such as gate-array chip assemblies on chip carriers. This last remark ignores, however, the time required to plan the layout and electrical behavior of the original gate array image or images, and hence is not damaging to custom design if the entire system comprises only a few chips of distinctly different functions. A fourth disadvantage is (d) the necessity for special testability provisions, either with external testers having access to different functions on the chip via costly reserved connectors and wires, or by testing circuitry incorporated on the chip and specialized to the different functional units.

The recent work of Mead and Conway [22] highlights the value of structured hierarchical design in taking advantage of custom techniques. Essentially one needs methods to define the system logic and plan its layout in top-down fashion during the first pass of the design. Iteration up and down the structure brings about concordance between the logical and physical hierarchies, and satisfaction of the requirements of correctness and proper cost/performance compromise.

A recent proposal of the author [23] suggests a method by which one can pass from an initial plan of a functional structure, realizing the architectural intentions of the system designer, to an initial planar layout. The method requires initial data as follows: (a) a top-down definition of the key building blocks (subsystem functions) to be accommodated on the chip (e.g. I/O pads, arithmetic logic units, PLAs); (b) a rough estimate of the circuit count and hence, using the ground rules, the silicon area used by each high level function, not including the global wiring interconnecting these functions; (c) an estimate of the number of global interconnections (bussing) between each pair of the functions (macros) defined in (a); (d) an estimate of chip-pad count and a projected set of chip-pad macro locations.

As outlined in the papers cited, one may derive from the above information a dual of a planar graph representing the interconnections as edges and the macros as nodes, except that unavoidable bus crossings in the original graph must first be replaced by global wiring macros of "size" suited to the required numbers of crossing interconnections, in order to assure overall planarity. It was found by experiment with D. Johannsen's OM-2 data-flow chip design [22], for example, that the final layout of the chip corresponded in a useful way with the dual of the architectural realization representing interconnected functions as a planarized graph. Other worked examples have also shown the practicality of the method, which should be contrasted in its approach with the very elegant bottom-up algorithm developed to do functional sub-cell design by Hsueh [11]. In actual layout, iteration between top-down chip-planning and bottom-up cell design should be used to achieve the optimal final layout. Recent Caltech work also deals with combined design techniques with similar intent [24].

CONCLUSIONS AND FUTURE DEVELOPMENTS

We have seen that design techniques used for large planar arrangements of many logic circuits in VLSI need extension and novel concepts whether the logic is arranged in regular arrays of identically sized subunits, or in functional subunits of different size and pin count. We shall see renewed emphasis on the hierarchical approach in both design styles, as well as a gradual climbing of customization up the physical package hierarchy from the chip level. In regular package designs, ways must be developed to incorporate smoothly performance and other electrical restrictions on layout into the operation of automatic algorithms. Also needed are ways to speed algorithm operations. Also, the parametrization of logic design alternatives in a fashion permitting early estimates of physical package wiring requirements is an important development.

In structured custom design, the frontier challenges appear to be the development of design systems and algorithms flexible enough to accommodate easy, rapid and verifiable passage from logical to physical design. To this must be added approaches to functional testability, to reliability and to test generation algorithms which pay direct attention to the defect types and densities characterizing semiconductor processing.

REFERENCES

1. W. R. Heller and S. Triebwasser, "Design of Digital Systems in LSI," Proc. Int. Conf. on Circuits and Computers, 1980, pp. 90-92.

2. S-1 Project Staff, "Advanced Digital Computing Technology Base Development for Navy Applications: The S-1 Project," Prepared for the Naval Systems Division, Office of Naval Research and the Research and Technology Directorate, Naval Electronics Systems Command, September 30,1978 (UCID18038).

3. J. Darringer, L. Trevillyan, W. Joyner, L. Berman, "Experiments in Logic Synthesis," IEEE International Conference on Circuits and Computers, 1980, pp. 234-237.

4. W. E. Donath and R. B. Hitchcock, "Path Lengths in Combinational Computer Logic Graphs," IBM T. J. Watson Research Center, Yorktown Heights, NY, 1971, Report RC3383.

5. E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," Proc. 14th Design Automation Conference, 1977, pp. 462-468.

6. J. Boney, "Advanced Concepts for VLSI Design for Testability," Proc. Inter. Symposium on Circuits and Systems, (IEEE), 1980, pp. 265-270.

7. K. A. Duke and K. Maling, "ALEX: A Conversational, Hierarchical Design System," Proceedings 15th Design Automation Conference, 1980, pp. 318-327.

8. G. M. Koppelman and K. Maling, "Verifying Deep Logic Hierarchies with ALEX," Proceedings 15th Design Automation Conference, 1980, pp. 328-335.

9. T. M. McWilliams, "The SCALD Timing, Verifier: A New Approach to Timing Constraints in Digital Systems," Proceedings IEEE International Symposium on Circuits and Systems, 1980, pp. 415-423.

10. J. D. Williams, "Sticks - A New Approach to LSI Design," M.S.E.E. Thesis, Dept. of Elect. Engr., MIT, June 1977.

11. M. Y. Hsueh and D. O. Pederson, "Computer Aided Layout of LSI Circuit Building Blocks," Proceedings International Symposium on Circuits and Systems, 1979, pp. 474-477.

12. W. Lattin, "VLSI Design Methodology, the Problem of the 80's for Microprocessor Design," Proc. First Caltech Conference on VLSI, January 1979, pp. 247-252.

13. W. E. Donath, "Placement and Average Interconnection Lengths of Computer Logic," IEEE Trans. on Circuits and Systems, CAS-26 (1979), pp.272-277.

14. E. N. Gilbert, "Random Minimal Trees," SIAM Journal of Appl. Math. 13 1965, pp. 376-387.

15. W. R. Heller, W. E. Donath and W. F. Mikhail, "Prediction of Wiring Space Requirements in LSI," Journal of Design Automation and Fault Tolerant Computing, 1978, pp. 117-144.

16. C. Davis, G. Maley, F. Simmons, H. Stoller, R. Warren, and T. Wohr, "IBM System/370 Bipolar Gate Array Microprocessor Chip," Proceedings IEEE International Conference on Circuits and Computers, 1981, pp. 669-673.

17. A. H. Dansky, "Bipolar Circuit Design for VLSI Gate Arrays," Proceedings IEEE International Conference on Circuits and Computers, 1981, pp. 674-677.

18. K. H. Khokhani and A. M. Patel, "The Chip Layout Problem: A Placement Procedure for LSI," Proceedings 14th Design Automation Conference, 1977, pp. 291-296.

19. K. A. Chen, M. Feuer, K. H. Khokhani, N. Nan, and S. Schmidt, "The Chip Layout Problem: An Automatic Wiring Procedure," 14th Design Automation Conference, 1977, pp. 297-302.

20. M. Feuer, K. H. Khokhani, and D. A. Mehta, "The Layout and Wiring of a VLSI Microprocessor," Proceedings IEEE International Conference on Circuits and Computers, 1981, pp. 678-679.

21. W. R. Heller, "Wirability of Logic Circuit Packages in LSI and VLSI, Proceedings Third International Computer Software and Applications Conference (IEEE), 1979, pp. 95-100.

22. C. Mead and L. Conway, "An Introduction to VLSI Systems," Addison Wesley, 1979.

23. W. R. Heller, "An Algorithm for Chip Planning," Caltech SSP Memo 2806, May 1979.

24. Proceedings Second Caltech Conference on VLSI, January 1981.