AN AUTOMATED, LOW POWER, HIGH SPEED COMPLEMENTARY PLA DESIGN SYSTEM FOR VLSI APPLICATIONS

Authors: Scott Powell, Elaine Iodice, and Eby Friedman

IEG TECHNOLOGY CENTER
HUGHES AIRCRAFT COMPANY
6155 EL CAMINO REAL
CARLSBAD, CALIFORNIA 92008

ABSTRACT

An automated Programmable Logic Array (PLA) design system that is fully compatible with the density and power constraints of VLSI is described. A low power CMOS version of the PLA has been integrated into a technology independent, automated PLA generation system to provide a self-contained, highly functional and low power, dense cell design capability. A description of the Complementary Programmable Logic Array (CPLA) technique is provided, particularly in terms of its integration into the automated PLA design system. Details of the system's input formats, minimization and verification capabilities, design flexibility, CAD compatibility, and processing speeds are discussed.

INTRODUCTION

With the arrival of very large scale integration (VLSI) as a mainstream component of present day technology, design methodologies must be created which are capable of generating random combinatorial functional blocks quickly and accurately. These combinational blocks should be described symbolically in software, thereby permitting the system to be technology independent while still satisfying density and performance requirements. The design system must be completely automated to provide maximal accuracy and minimal design turnaround while still providing flexibility in input format, layout orientation, technology, and compatibility with automated layout systems. This fully integrated, cohesive design system must be able to generate highly functional and dense macrocells which satisfy the low power requirements necessary at VLSI levels of integration.

This paper describes an automated Programmable Logic Array (PLA) design system that is fully compatible with the density and power constraints of VLSI. Special attributes of the PLA design system which maximize either performance or density are described in detail, particularly in relation to their implementation in the overall design system. This technology independent, automated PLA generation system is used with a low power CMOS version of the PLA, thereby minimizing the effective power dissipation per device and eliminating the need for any special thermal removal systems. This low power PLA, the Complementary Programmable Logic Array (CPLA), permits the use of PLAs in both VLSI and power sensitive applications. Together with the automated PLA generating system, the CPLA provides a self-contained, highly functional, low power, dense cell design capability embedded within an accurate and fast design system for application in custom VLSI circuits.

THE COMPLEMENTARY PROGRAMMABLE LOGIC ARRAY DESIGN TECHNIQUE

Array logic techniques which use structured matrices of transistors or gates have long been known to give rise to designs which utilize area efficiently and minimize design turnaround. PLAs consist of an AND plane cascaded with an OR plane to directly implement Boolean logic equations in the sum-of-products form, as illustrated in figure 1. In practice, the AND/OR operation is performed by the logically equivalent NOR/NOR or NAND/NAND operation, resulting in two NOR planes or two NAND planes. PLAs have proven to be very useful for designing random logic[2-3] and many automated PLA design systems exist which exploit the algorithmic nature of PLAs. However, due to the high power dissipation inherent in standard PLAs, their use has been limited to less power sensitive applications. PLAs implemented with NMOS-type wired-NOR structures consume too much power to be considered as useful design elements in a low power or VLSI environment. The excessive power dissipation of the standard PLA can be reduced by using gated load techniques, but these techniques

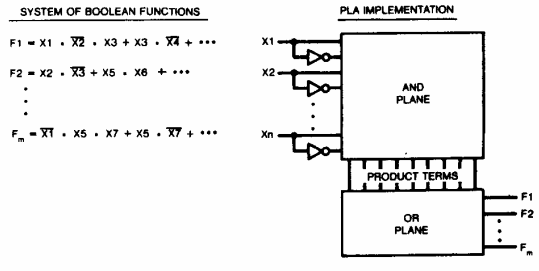

Figure 1: Two-level PLA

require the generation of additional timing signals and extra hardware to latch the outputs.

The CPLA is a dense matrix of transistors in a fully complementary arrangement configured to implement combinatorial logic functions. This complementary arrangement of transistors gives rise to a power dissipation one to two orders of magnitude lower than that achieved by the standard PLA. The area advantage of the CPLA is not as great as for a standard PLA. Due to its fully complementary nature, the CPLA can occupy two to three times more area than a standard PLA. The CPLA thus permits the insertion of array logic into applications where excessive power dissipation would otherwise prohibit the use of PLAs. Due to the fully complementary arrangement of the CPLA, very low power dissipation levels are achieved without requiring special synchronous timing signals and/or extra hardware to latch the PLA outputs.

It is well known that a NAND gate can be equivalently implemented using either: 1) parallel p-channel devices on the inputs and a pull-down load or 2) serial n-channel devices on the inputs and a pull-up load. Thus, one can envision a NAND/NAND PLA, with appropriate load devices, where the gates within an array are implemented with either parallel or serial devices. In the parallel PLA, the NAND gates actively pull their outputs high and passively pull their outputs low. In the serial PLA, the NAND gates actively pull their outputs low and passively pull their outputs high. In the CPLA the load devices of a standard parallel PLA are replaced by serial devices, resulting in a fully complementary PLA in which the NAND gates actively pull their outputs both high and low. Area reduction techniques such as folding[4], input decoding, input/output splitting, and partitioning[5] all may be applied in precisely the same way in a CPLA as in a standard PLA. Either or both NAND planes of the CPLA can be folded. By breaking an input line into two non-adjoining segments, CPLA inputs can share common rows. Similarly, by breaking an output line or a product line into two non-adjoining segments, CPLA outpus and product lines can share common rows and columns. The presence of both n-channel and p-channel transistors poses no additional constraints to the folding problem. The CPLA offers all the advantages of a structured, regular design methodology with CMOS power dissipation levels (albeit with a loss of cell density as compared to the standard PLA).

As in a standard PLA, the CPLA uses a NAND/NAND configuration to implement Boolean AND/OR operations. Each CPLA NAND plane is constructed from a combination of two arrays: an "input array" identical to a standard parallel PLA excluding the load devices and a "load array" identical to a standard serial PLA without any load devices. Figure 2 illustrates how the two arrays are meshed to reduce the area occupied by the input buffers 20-50% and also cut the polysilicon parasitics in half. A product line is pulled to a logical one by one or more of the parallel transistors connected to that line. Unlike the standard PLA, a product line is pulled to a logical zero by a serial chain of transistors. The only time every transistor in this serial chain is turned "on" is when every associated parallel transistor is turned "off". This configuration eliminates the passive pull-down devices of the standard PLA which greatly reduces power dissipation by eliminating all low resistance paths between power and ground.

Both CPLA NAND planes require parallel and serial transistors to form a product term or PLA output signal. A NAND configuration is used instead of a NOR configuration to take advantage of the inherent drive advantage of the n-channel devices in the serial device path. The maximum size CPLA is determined both by the available area and by performance constraints imposed by the maximum length serial chain of transistors within the

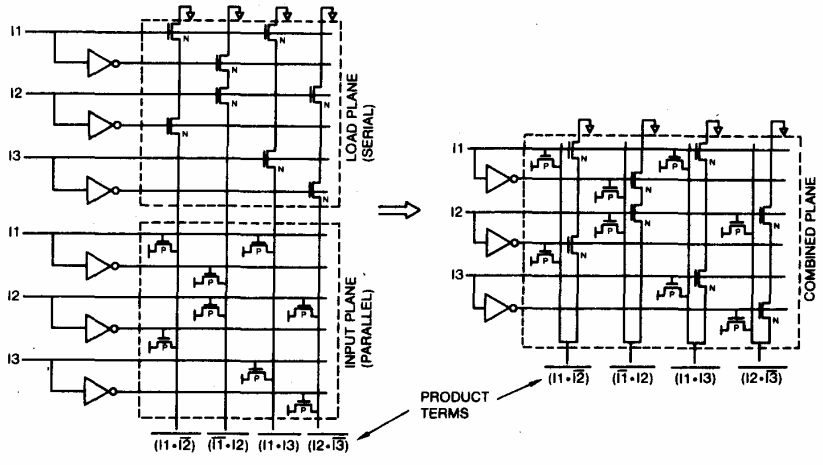

Figure 2: Combination of serial and parallel NAND planes forming a CPLA NAND plane

CPLA. Appropriate area and performance constraints are determined by considering the related tradeoffs appropriate to the intended chip application.

DESCRIPTION OF THE AUTOMATED PLA DESIGN SYSTEM

The automated PLA design system is intended to provide the VLSI designer with a tool to quickly and accurately generate the physical layout of a combinatorial logic block of arbitrary complexity. The input to the system consists of a high level symbolic description of the desired combinatorial Boolean logic. The PLA design system is used to minimize the logic equations and subsequently to generate the physical layout of the functional block along with descriptions of the PLA suitable for higher level logic simulation and automated routing. The generated artwork includes input and output buffers, the NAND arrays with appropriately sized transistors, and power/ground connections. Manual additions or manipulations are completely eliminated.

The method by which the AND/OR operation is implemented is completely independent of the automated PLA design system. NAND/NAND, NOR/NOR, or AND/OR PLAs can all be used with any technology without modifications to the design system. The NAND/NAND CPLA is used for its low power attributes and resulting increased applicability to VLSI applications.

The regularity of a PLA is of fundamental importance for automated artwork generation. Both NAND planes of the PLA are divided into rows of equal width cells, one cell per product term. There is one row of cells for each input, one row of cells for each output, and one column of cells for each product term. In the first NAND plane, each row contains both the input signal and the complement of the input signal. There are three possible cell types in the first NAND plane: a "1" cell which includes the row input in the product term, a "0" cell which includes the row input complement in the product term, or a "-" cell which excludes both the input and the input complement from the product term (a "don't care" cell). In the OR plane, there are two possible cell types: a "1" cell which includes the column product term in the output or a "0" cell which excludes the product term in the output. Thus, both NAND planes of any CPLA can be completely built from these five basic cells. Additional cells are needed for the input/output buffers, power/ground routing, and for the interface between the NAND planes. The cell type and cell location information is transferred to a turnkey graphics system where the physical artwork for each cell is inserted automatically.

The PLA design system consists of four stages as depicted in Figure 3. The first stage generates a complete exhaustive truth table from various input formats. Mathematical expressions are permitted as one form of input to the system. Functional blocks which implement addition, multiplication, trigonometic, and other mathematical functions can be generated directly from a mathematical equation. Using a logic simulation program, truth tables can be generated from simplified or unsimplified Boolean equations. Functions described by complex Boolean equations with nested parentheses, intermediate variables, and exclusive-OR operators can be generated in this manner. Additionally, if a logic schematic has been captured in a digitized format, a logic simulator can be used to generate a truth table automatically from the connectivity description. Using this technique, PLA artwork can be generated directly from a logic schematic.

The second stage of the PLA design system performs a logical minimization of the truth table generated by the first stage. Since the size of the PLA is directly dependent upon the number of product terms, it is essential that all redundant terms are removed. The logic minimizer program reduces the number of product terms by identifying all "don't care" states. The output of the minimizer is in the same format as the truth table with "don't care" states signified by a "-". At this point in the design process, the minimized truth table is compared to the original truth table. The verification program accepts a minimized truth table as an input and recreates an exhaustive truth table as an output. This complete truth table is then automatically checked against the original truth table to verify proper minimization. The verification program also permits a manual logic minimization to be verified so that the first two stages of the PLA design can be bypassed, permitting direct entry to the next stage of the design system.

As mentioned above, the PLA can be described in terms of adjoining rows containing equal width cells. Each cell type within a row is given by the corresponding "1", "0", or "-" entry in the column of the minimized truth table. Figure 4 illustrates this relationship as well as showing that every other row is reflected about its x-axis and that every other cell is reflected about its y-axis. These reflections permit a single supply

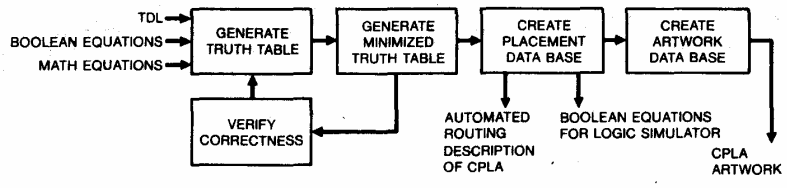

Figure 3: Operation of PLA automated design system

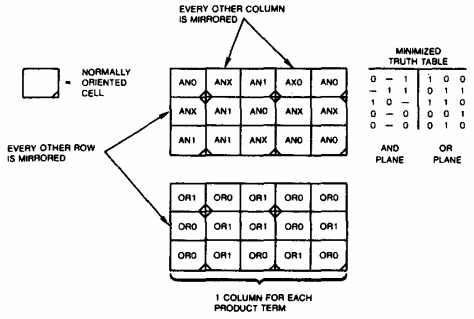

Figure 4: Cell orientation for AND/OR planes

line to be shared by two adjacent columns in the first NAND plane and by two adjacent rows in the second NAND plane. Additionally, these reflections permit adjacent contacts to be shared. Note that at this point, the various cells are represented by a rectangle defining the boundary of the actual cell artwork.

The third stage of the PLA design system uses the minimized truth table to identify cell types and locations within the various rows. The cell placement module of an in-house automated layout system is used to automatically place these cells. This placement system is designed to create cell rows of uniform height and to use the channel region in between the cell rows for routing. The cells are represented as rectangles defining the boundary of the actual cell artwork. The placement system creates a database containing cell type and location information.

In the PLA design system, the cell rows are defined to be adjacent to each other with no channel region in between each row of cells. A translator program was developed to automatically create an input file to the placement system from a minimized truth table description. The input file consists of four parts: 1) a cell library defining the cell type and size, 2) cell classification lists defining which cells belong to which cell type, 3) a row property record defining the row location and orientation, and 4) a row definition record defining the contents of each row and the orientation of each cell within the row. The first module of the placement system creates cell type and location information in a format which can be transferred to the turnkey system. The fourth stage of the PLA design system consists of building PLA artwork from the cell placement description. Note that up to this point, the PLA design system has remained technology independent, with the exception of the cell size defined in the placement cell library. Mask artwork for the various cells is resident on the turnkey graphics system. At this point, the turnkey's automatic cell referencing system places the actual cell artwork at the location and orientation specified by the placement system. Input/output buffer cells, power/ground cells, and AND/OR plane interface cell artwork is automatically placed in the same manner.

In addition to the artwork, PLA boundary and pin location information is automatically generated in a form suitable for higher level routing. The PLA system generates the required input file for higher level automated routing, thereby providing compatibility with our automated layout design system.

A file containing the final Boolean equations being implemented by the PLA is also generated automatically. This file is formatted to permit these Boolean equations to be used in higher level logic simulations to again verify the correctness of the PLA. For testing purposes, the logic simulator description of the PLA consists of a separate set of equations for the AND plane and the OR plane. This configuration permits access to the product lines of the PLA. This access permits fault simulation nodal points to be inserted at the product lines, permitting fault isolation to a particular gate within the PLA.

A layout optimization feature has been added to the PLA design system to meet the special requirements of the CPLA. Multiple versions for each of the five NAND plane cells have been designed to permit variable width transistors within the NAND planes. The PLA design system varies the width of the transistors according to the number of inputs to each of the NAND gates within a given NAND plane. For NAND gates with fewer inputs, transistor widths are decreased, thereby reducing the input load capacitance and optimizing both performance and area. For NAND gates with many inputs, the width of each of the serial n-channel devices is increased when optimizing for performance. To maintain the technology independence of the PLA design system, the number of versions for each type is user programmable.

Other layout optimization features have been added to the design system. Multiple versions of the input buffers exist to allow the widths of the buffer transistors to vary according to the load of the PLA. The maximum load any particular input buffer version can drive is user programmable. This feature saves area for smaller PLAs and improves the performance of larger PLAs.

Several input/output configurations are permitted within the PLA. Inputs and outputs can be placed in any order on the left and/or the right side. This general I/O arrangement is flexible enough to permit folding or partitioning of the PLAs. Included in the CPLA cell library is a cell which

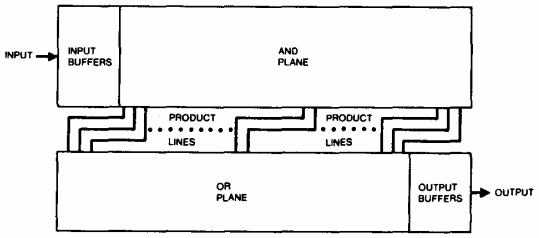

Figure 5: PLA topology with OR plane offset from AND plane

permits the OR plane to be offset from the AND plane. This cell is used when all inputs enter from the left and all outputs exit from the right, as illustrated in figure 5. The cell permits the placement of the leftmost part of the OR plane in the area below the input buffers and the rightmost part of the AND plane in the area above the output buffers.

SOFTWARE IMPLEMENTATION OF THE AUTOMATED PLA DESIGN SYSTEM

The automated PLA design system is written in PASCAL to take advantage of the capabilities offered by that programming language. Such needs as user defined data structures, dynamic allocation, and recursion prompted this choice. This design system is implemented on a VAX 11/750 minicomputer using the VAX/VMS operating system.* The turnkey system is a GDS II CALMA graphics computer.

The PLA design system is subdivided into a number of logically grouped programs. Three command files, using VAX/VMS DEC control language have been created, each corresponding to a different stage in the system. These command files control the sequence of program calls and perform file management tasks. The PLA system is designed to provide maximum flexibility to the user. It permits the user to enter the system at any stage in the design process. The system is composed of a set of individual programs, each capable of stand alone operation.

The PLA design system makes use of both easily accessable and custom software. The TEGAS logic simulator, from TEGAS Systems Inc., is used to generate a truth table either from a logically captured schematic or from unsimplified Boolean equations. The Hughes Automated Layout (HAL) system is used for cell placement and VAX-to-CALMA translation. Custom software has been created which provides various translation operations and special purpose processing. Both the minimized truth table and the logic of the PLA is verified automatically against the original logic description. Additionally, both TEGAS and HAL cell library descriptions of the PLA are automatically generated for logic simulation and higher level automated routing. Lastly, a program for automatically generating a truth table from a set of mathematical functions has been added to the PLA design system to assist in the design of random logic in mathematically oriented chip types.

Each line of a truth table corresponds to a single input/output pair. The I/O pair is stored in a record. Since the length of the truth tables varies widely, a dynamic allocation scheme was used. This technique resulted in the use of a linked list data structure to hold the table in memory during processing.

Table 1 depicts benchmark figures for the automated PLA design system. The benchmarks were taken from a VAX 11/750 with a maximum of eight users sharing system resources. Times are given for CPU usage as well as actual user time for generating PLAs using this design system.

AUTOMATED PLA DESIGN SYSTEM BENCHMARKS

EX	# Input Bits	# Output Bits	# Product Terms	CPU (seconds)	Terminal Time (minutes)
#1	4	4	12	0:62.1	06:08
#2	6	6	38	3:07.66	09:51
#3	9	7	94	13:45.24	19:45
#4	9	9	256	1:55:44.99	2:54:08
#5	9	10	10	9:41.72	30:49

TABLE 1

CONCLUSIONS

A design technique for quick and accurate generation of random combinatorial logic has been discussed. The PLA design system accepts as input a logic description in the form of a graphically captured schematic, a set of Boolean equations or mathematical expressions, or a truth table, and generates a minimized and verified Programmable Logic Array (PLA). While the system is technology independent, it utilizes a CMOS process to maximize the technology's inherent power/density advantages. The automated PLA design system permits designer interactive performance/density tradeoffs. The PLA system is completely compatible with a logic simulator and an in-house automated layout system. Together they provide a fast and accurate CAD capability for the design of custom integrated circuits. Finally, computer processing speeds for generating PLAs are shown to be compatible with the requirements of highly random and complex VLSI circuits. In addition to the design system that has been discussed in this paper, a technique for implementing a low power CMOS version of the PLA has been described. This technique, the Complementary Programmable Logic Array (CPLA), permits the PLA technology to be usable in both power sensitive and VLSI applications. It provides a dense functional cell which dissipates very little power.

Together, the automated PLA system in concert with the low power CPLA design technique provides an overall system for the design of highly random, combinatorial VLSI circuits.

ACKNOWLEDGEMENTS

The authors wish to thank B. Valdez and G. Persky for their helpful discussions and encouragement as well as their assistance in proofreading this paper. They are also grateful to C. Fletcher for his assistance in layout.

REFERENCES

[1] H. Fleisher and L. I. Maissel, "An Introduction to Array Logic," IBM J. Res. Develop., Vol. 19, pp. 98-109, Mar. 1975

[2] B. W. Fung, T. J. Macnee, and G. Rugila, "A high-Density PLA Macro and Its Layout Generator," 1982 IEEE ISSCC Digest of Technical Papers, pp. 58-59, Feb. 1982

[3] R. B. Mayo, "Combining Graphics and Procedures in a VLSI Layout Tool: The TPACK System," Computer Science Division Report # UCB/CDS 84/166, University of California at Berkeley, Jan. 1984

[4] G. Hachtel, A. Newton, and A. Sangiovanni-Vincentelli, "An Algorithm for Optimal PLA Folding," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. CAD-1, No. 2, pp. 63-76, Apr. 1982

[5] J. Hennessy, "Partitioning Programmable Logic Arrays," ICCAD, pp. 180-181, Sept. 1983

[6] J. R. Logan, "Automated Network Synthesizer Compacts VLSI Design," Computer Design, pp. 173-178, Apr. 1981

[7] G. Persky, C. Enger, and D. M. Selove, "The Hughes Automated Layout System-Automated LSI/VLSI Layout Based on Channel Routing," 18th Design Automation Conference, pp. 22-28, July 1981.