PLA GENERATION USING A SILICON ASSEMBLER

K. Dickson, D. Smith, P.G. Gulak

Materials and Devices Research Laboratory
Department of Electrical Engineering
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

Abstract

The PLA generation procedure in the University of Manitoba Silicon Assembler (UMSA) is described. The user oriented syntax of the standard cell based assembler allows the designer to describe a PLA layout in a natural and convenient fashion.

1. Introduction

A silicon assembler translates a structural description of a VLSI design into a geometric description. Sophisticated versions are language-based and allow a hierarchy of functional modules. It is with these objectives in mind that the University of Manitoba Silicon Assembler (UMSA) has been developed. UMSA syntax, described in [1], is loosely based on a Backus-Naur metalanguage convention. Macros, analogous to the mnemonic macros of assembly language, which represent the standard layout blocks to be repetitively called in the assembly of a chip design, are defined and maintained in a library file accessible to all students or other designers. The machine language corresponding to these macros or cell symbols is the CIF file definition of the layout, which is a statement of the detailed geometrical coordinates for the patterns in the various levels or masks. Among the basic cells are the bonding pads and pad driver circuitry which connect the integrated circuit to the package pins and the outside world. The library of standard cells, together with a placement algorithm which keeps track of the ports or connection points on the four connection layers, forms the heart of the silicon assembler. Wire routing algorithms may be added to complete the automation of the chip layout. A typical layout generated by UMSA is detailed in Figure 1. In any new design, some proportion of the layout must be generated for the first time, but any cell definitions from previous projects which are utilized need not be reinvented. Pascal and Fortran versions of UMSA have been developed.

2. A PLA Generator

A programmable logic array (PLA) is versatile functional block in a digital system which can realize any combinational logic function. With input and output registers and feedback, the PLA can also implement an arbitrary finite-state machine. One common application for a PLA is as a microprogrammed controller in a structured computer design.

The PLA generator which resides in UMSA includes all of the macros or cell definitions required in implementation. The specification of a 'new' PLA corresponds to the presence of short 'stubs' of diffusion lines at approriate locations in the layout (Fig. 2) and represents the fixed 'program' of the device.

To invoke the PLA generator package within UMSA, a PLA definition statement must be encountered in the standard input data stream. The definition statement is then followed by the boolean logic equations to be be implemented. UMSA automatically removes redundant terms and simplifies these equations. These equations are then interpreted by UMSA to determine the exact location of each programming transistor. Output is in the form of CIF code and is defined as a new cell structure which ultimately can be chained together to form a much more complex design.

3. Future Improvements

Currently, PLAs are laid out on a rectangular grid with all transistors the same size. However, each gate in the PLA, a horizontal slice in the AND-plane or a vertical slice in the OR-plane, drives a different load, depending on the number and the size of the transistors on it. In addition, large PLAs have large parasitics associated with the wires. Collectively, these effects cause PLAs to be wasteful of speed on the heavily loaded paths and wasteful of power on the lightly loaded paths. UMSA , in the near future, will address this problem by automatically scaling gate sizes for delay and power optimization.

In addition, PLAs can take an inappropriately large amount of area; the area equals approximately the product of the number of input terms and the number of AND terms. We are attempting to address this issue, in UMSA, via two approaches: hierarchical PLAs [3] and muliple column and row folding [4].

4. Conclusion

The structure and capabilities of the University of Manitoba Silicon Assembler (UMSA) has been briefly sketched. The use of the PLA generation subroutine, encompassed within UMSA has greatly simplified the task of circuit layout.

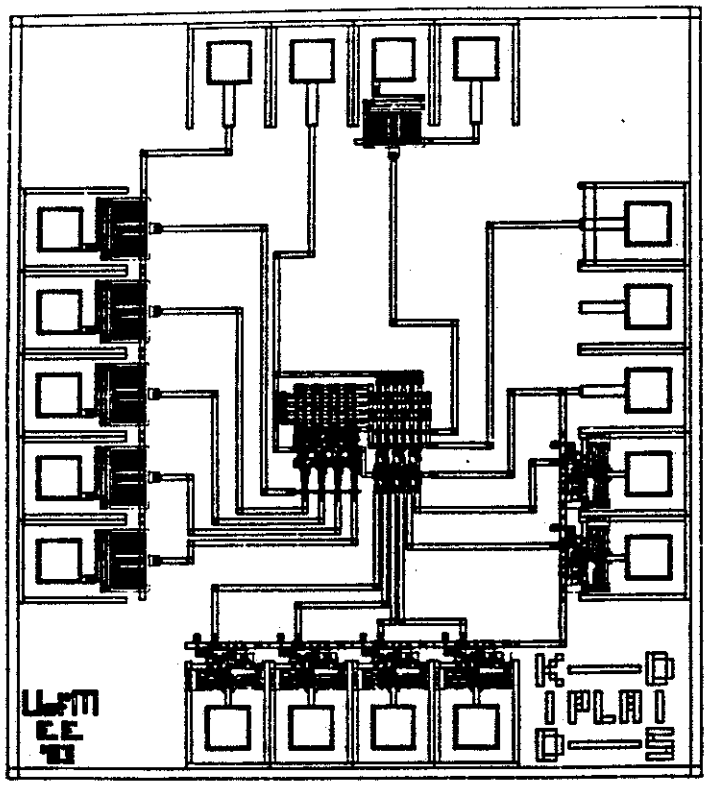

Figure 1: PLA LAYOUT

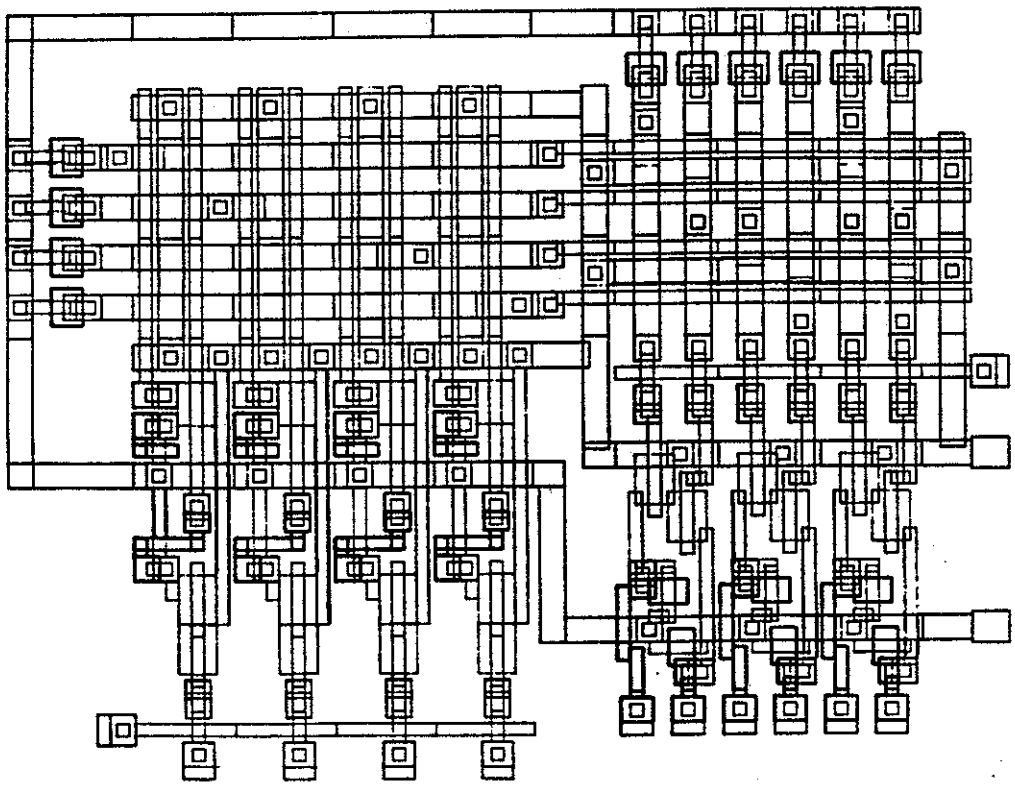

Figure 2: DETAIL OF PLA-4-4-6 (4 INPUT, 4 PRODUCT TERM, 6 OUTPUT)

REFERENCES

1. P.G. Gulak, J.G. Shaw, R.D. McLeod, G.C. McGonigal and H.C. Card, University of Manitoba Silicon Assembler (UMSA), Department of Electrical Engineering, Tech. Report, TR83-1, University of Manitoba, 1983.

2. R.W. Hon and Carlo H. Sequin, A Guide to LSI Implementation, XEROX PARC, 1980.

3. R. Ayres, Silicon Compilation - A Hierarchical Use of PLAs, Caltech Conference on VLSI, January 1979, pp. 311-326.

4. G. DeMicheli and A. Sangiovanni-Vincentelli, Multiple Folding of PLAs, IEEE Int. Sym. on Circuits and Systems, 1983, pp. 1026-1029.