# Pragmatic VLSI Design Environment
# P R I D E

Thomas Almy
Computer Research Laboratory
Tektronix Laboratories
Tektronix, Inc.

ABSTRACT

This report consists of the overview of the VLSI design "environment" created in Forth. The PRIDE program has been run on computers ranging in size from a TRS-80 Model I to a VAX-11/780 and has been used for the design of over 10 ICs.

## Background

This set of programs, referred to as PRIDE (PRagmatic Ic Design Environment) is a logical extension of the SLAP layout program which the author wrote and used to redesign his content addressable memory LSI chip. The SLAP program was written in STOIC, a derivative of Forth. While the program performed well it suffered from two problems: lack of transportability and incompleteness (SLAP, the text editor, and the plotting program were in three separate environments).

The decision was made to continue the program development in Forth. Forth has the advantage that it is commercially available (in compatible forms) for many different machines. Being an environment, the user perceives the same system interface regardless of what physical system he or she is using. All system components (layout--both procedural and graphic, routing, printing, and plotting, as well as maintenance functions) are immediately available without switching programs or machines.

## Development

The program was originally written and tested in 1971 on the author's TRS-80 Model I using MMSForth. This system has a Z80 microprocessor, 48K of RAM, 2 175k byte capacity mini-floppy disk drives, a 64 character by 16 line display, and an Epson MX-80 printer with graphics option. The total system cost was less than $2500. In addition, the plotter package was tested using an HP 7221B plotter. There was over 350 screens of sources, not including the IC cell library.

The program was transported to a PDP-11/70 running UNIX, and later to a VAX 11/780 using the PDP-11 emulation mode. The program is currently used on a 68000 based UNIX workstations, a TRS-80 Model III, and a CP/M-68k system. Partial ports, to test portability, were made to Tektronix 4112 with the programmability option, Compupro CP/M-86 based system, and an IBM PC.

When the program was ported to the Motorola 68000 processor, a Tektronix proprietary 32 bit integer Forth was utilized. This Forth met the 79 Standard except for integer size. Thanks to the modular programming style of Forth, the conversion of PRIDE for 32 bit integers took less than one day.

**Technologies**

PRIDE was originally implemented for NMOS, with an additional modified version distributed for GaAs. The current version has compilation options (controlled by the constant TECHNOLOGY) for NMOS, HMOS, CMOS, and GaAs. The system has tables for contact generation and design rules, as well as display attributes for the different masks.

Changing technologies or design rules involves recompiling the system, a process that takes about five minutes for the 68000 based systems with hard disks.

**Parts of a PRIDE generated design**

Pride allows creation of entities called cells, which consist of rectangles on different mask layers (called boxes) at specified locations, and other cells at specified locations and orientations. Thus the IC design consists of a hierarchy of cells, the topmost cell being the IC itself.

Additional software layers provide a wire facility (sequences of boxes, with contact cells between mask layers if the layer is changed), a cable facility (collections of wires that can be routed), and a module facility (collections of cells, cables, and wires to implement topologically regular, logic functions.

In addition, the database supports the naming of connection points within cells and the naming of specific cell instances within cells. Printing status flags assist in displaying or plotting designs.

**Components**

PRIDE consists of a number of functional units (groups of Forth word definitions) grouped into three classifications: utilities, tasks, and options. The utilities will tend to be available (that is, in main memory) at all times. Tasks, when loaded from disk, will delete all words defined since the last task load command (thus only one task may be in memory at one time). Options consists of special task features that are loaded when required, and may be removed when no longer needed. These definitions are not rigid; if sufficient memory exists there is no reason why multiple tasks could be loaded at one time, although this has not been done yet.

PRIDE's utilities are either loaded from disk upon running Forth, or are make part of the Forth's loadable core image. The utilities consists of the following units:

1) A kernel wordset that supplies compatibility among all systems. The Forth 1979 standard wordset was settled on, with modifications for the 32 bit integer size in the case of 68000.

2) The file system. This wordset allows management of named files (of up to 6 character names) on one or more virtual disk drives. This system works on top of Forth's virtual memory feature. The file system was implemented to allow identical operation on any target machine, regardless of operating system.

3) The text editor. This wordset must be tailored for the particular terminal being used. Several terminals are supported. The screen editor mimics to some extent the Wordstar editor. Naturally, an editor must be available in order to use the procedural design language, SLAP.

4) Database management. It is possible to examine the "catalog" of cells and change printing characteristics. There is also a command file facility.

Tasks are loaded (but not executed, since they are interactive) by entering the name of the task. Loading a task will delete any words defined since the last task was loaded (or the system was started, if no previous task was loaded). Tasks are:

1) D1INIT is a load screen which will initialize a design by defining the contact cells SLAP requires internally. The user will typically then load the screens containing the SLAP code defining the library cells desired.

2) SF->DB loads the wordset which processes SLAP, the procedural layout language. Several different options are available: the river router, the cable facility, and the module facility.

3) GEDITOR, a graphics editor, allows manual graphic design generation. SLAP designed cells can be converted to graphic files. The graphic file can then be viewed or edited (connection points can be specified or changed). GF->DB, an alias for SF->DB, allows incorporating the graphic file into the design. The intent of GEDITOR is to assist the designer in creating low-level cells. These cells, which typically generate the logic functions in the design, tend to be difficult to lay out using a procedural language since they tend to have little or no structure. The GEDITOR task is very display dependent, so different versions exist for different displays. Current displays supported include Tektronix 4112, 4113, 4115, and the Omega 400.

4) HPPLOT and SCREENPLOT (as well as some dot matrix printer plotters not supplied) allow the plotting of cells. Each cell has associated with it switches concerning whether to plot the cell, plot just its bounding box, or bristles (contents at its perimeter). These switches can be set in SLAP or GEDITOR while the cell is defined, or they can also be set via some functions provided by the plotter overlay. The user specifies which cell to plot and the instantiation depth.

6) DB->CIF, DB->GF, and DB->SF will transfer selected cells (or in certain cases the entire design) to a host CIF file, graphics file, or SLAP source file, respectively. The conversion to CIF allows external processing (mask making, simulation, complete DRC...).

An external utility program exists to convert CIF files into SLAP source screens. This utility has been successfully utilized to move cell libraries into the PRIDE environment.

## Further information

For further information, contact the author at Tektronix, Inc., PO Box 500, MS 50-662, Beaverton, Oregon, 97077, or (503) 627-6188, or the net addresses tektronix!tekchips!toma (usenet), toma%tekchips@tektronix (csnet), toma%tekchips%tek@csnet-relay (arpa).

## Trademark notices