scribed. At present this technology is in its infancy. Many problems exist, but it is at least theoretically possible to describe any circuit with a silicon compiler language. The greatest promise of this technique is perhaps in our knowledge of creating large software programs by assembling many small subprograms.

Correct the First Time

The difficulties in discovering errors later in the design process were discussed previously. Stan Mintz of HP's Corvallis Division in Oregon has proposed a "correct-the-first-time" methodology. Stan proposes that the functional description is the control document for both the development of an IC and any tests that will be applied to it. In addition, as the design progresses through logic, circuit, and mask design, Stan proposes that each stage be compared to the functional description. Furthermore, he states that the test program used to evaluate the IC be generated from the functional description. In theory, if each path rigorously follows the verification procedures, when the finished IC and test program meet at the tester there should be no reason for a failure. Stan's proposal has many implications for both methodology and tools, but if enforced, should relieve the disastrous effects of discovering an error late in the design sequence.

Finally, as an extension of Stan's proposal for establishing the functional description as the premier description of an IC, the potential exists for completely synthesizing the design from this description. There is some very encouraging work progressing at Carnegie-Mellon University. Their work involves synthesis of a design from a high-level ISPS language description into implementable components.[2] Their techniques involve high-level decisions as to the selection of a design style and implementation of a design based on that design style.

References

1. C. Mead and L. Conway, "Introduction to VLSI Systems," Addison-Wesley, 1980.
2. C. Bell and A. Newell, "Computer Structures: Readings and Examples," McGraw-Hill, 1971.

William J. Haydamack

Bill Haydamack is a native of the province of Alberta, Canada. He attended the University of Alberta in Edmonton and received BSEE and MSEE degrees in 1963 and 1966. After three years of automatic test system design, he joined HP's Automatic Measurements Division in 1969 and served as project manager and group leader. He's now section manager for design verification with HP's design aids group. He has authored eight papers on computer-aided design and his work has resulted in three U.S. patents. A resident of San Jose, California, Bill is married and has two sons. He is active in YMCA youth activities, having served in various offices and as a baseball and basketball coach. He enjoys various sports and outdoor activities and does his own photographic darkroom work.

Daniel J. Griffin

Dan Griffin received his BSEE degree from Massachusetts Institute of Technology in 1971 and his MSEE degree from the University of California at Berkeley in 1972. With HP's Desktop Computer Division since 1972, he was on the project teams for the 9845 Computer and the NMOS processor chip set used in the 9825 and 9845. He was in IC production engineering for two years and is currently project manager for three NMOS LSI chips. He's a member of IEEE. Dan was born in Denver, Colorado and now lives in Fort Collins. He's married and enjoys photography and hot air ballooning.

Design and Simulation of VLSI Circuits

by Louis K. Scheffer, Richard I. Dowell, and Ravi M. Apte

A VLSI CIRCUIT is a complex structure containing tens of thousands of individual transistors. The design of VLSI proceeds from a high-level architectural description down to the layout of the artwork. The circuit design process is the activity that occurs between the functional description and the start of the physical layout. However, for integrated circuit design, there is usually no clean separation at either end of the process. System architectures are specified with a final implementation in mind, and layout and device designs often dictate changes in the system design. The responsibilities of the circuit designer are to insure that the circuit performs its intended function, verify that the function is performed at the required speed, and guarantee that the circuit can be manufactured and tested for a reasonable cost.

Circuit design may be divided into several stages. In the first stage, the designer considers the functional specifications and produces a logic design. Various portions of the

allowable delays and the power consumption are allocated to parts of the total circuit. This step requires a significant amount of experience, since optimizing a VLSI design is far different from optimizing a design in a technology such as TTL. Speed, power, space, and design time can all be traded off against each other, but a wrong decision can lead to a great deal of backtracking, since all portions of a design may be interdependent. At this stage of the design, a logic simulator (such as TESTAID-IC) may be used to verify that the desired function is indeed performed.

This translation step from functional to logical circuit description is of great interest to current researchers. To lessen the required integrated circuit knowledge and reduce the design time and errors, structured methodologies have emerged that permit the computer to assume a greater role in the design.

The next step is to turn the logic design into a transistor level design. In many cases, this is not done explicitly, for it is simpler to design the artwork directly from the logic specification. Where performance is critical, however, detailed investigation of the transistor level designs is required. The circuit simulator SPICE is used to do verification on this level. It allows the designer to look at speeds, powers, noise margins, best and worst cases, and the effects of process variations. By using the results of these detailed simulations, designers can significantly improve the reliability and yield of the final product.

Another necessary step is the definition of the tests for the chip. The designer must guarantee that the chip can be tested. This is partially achieved in the logic design stage, by making sure that the storage elements can be read and controlled, but the actual definition of tests must take place after the logic design is completed. Logic simulators (TESTAID-IC)offer considerable help with this task.

All of the preceding steps deal with logic diagrams and schematics, which must be converted to machine readable form so that simulators may use them. Furthermore, several different representations of a circuit usually exist, and a hierarchical design is almost a necessity to keep all the details straight. DRAW is a program that helps the designer in these areas. With a graphics terminal and a digitizing pad, the designer can enter and edit schematic diagrams interactively. The hierarchy is easily maintained and used, and the program can convert conventional schematics to the textual forms required by the simulators.

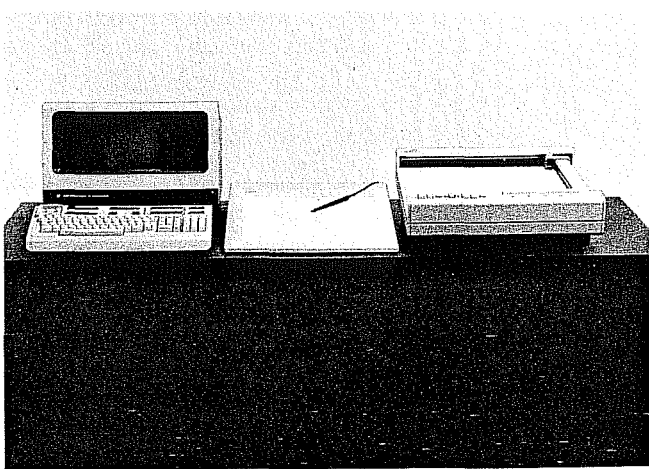


Fig. 1. *DRAW* circuit design station consists of an interactive graphics terminal, a digitizing tablet, and a four-color plotter, all tied to a timeshared computer. *DRAW* is a graphics editor designed for editing schematics.

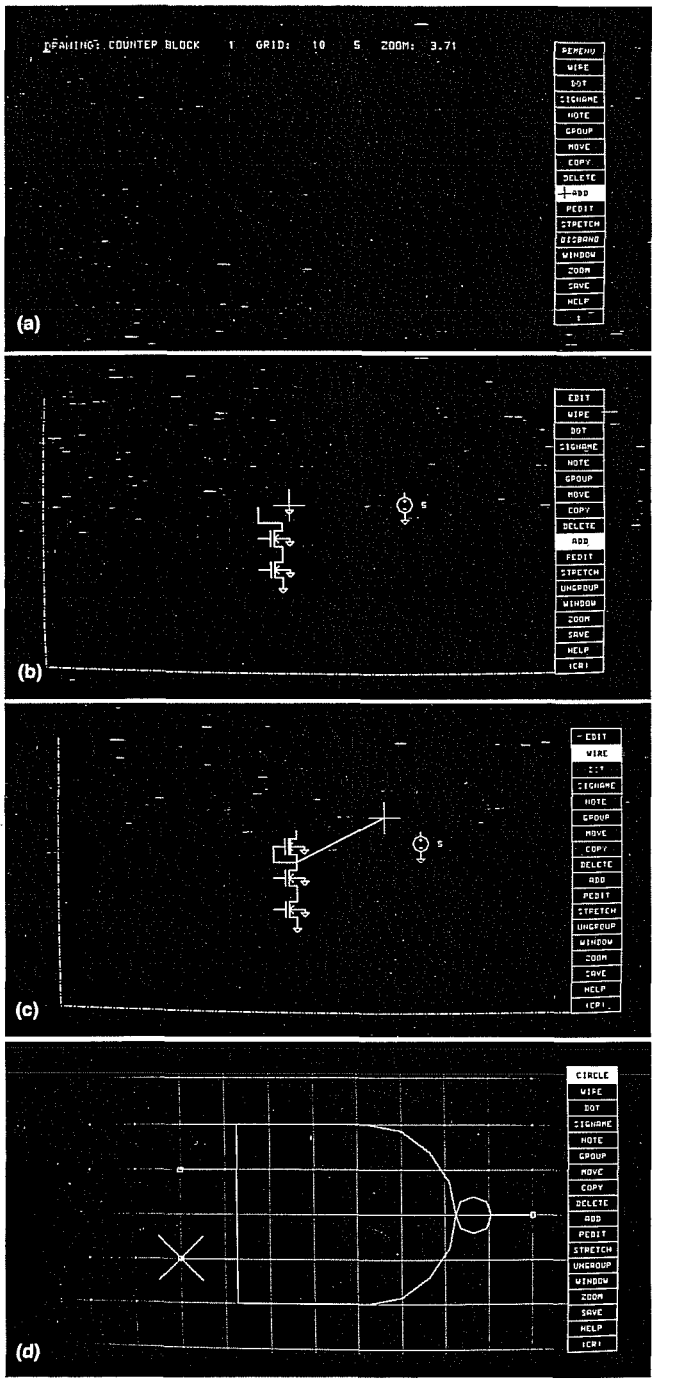


Fig. 2. Using the *DRAW* schematic editor. (a) Select the *ADD* command. (b) Place a transistor. (c) Wire two points together. (d) Define an external *VIEW* drawing to be used on higher-level schematics.

Schematic Input Using DRAW

Schematic entry using DRAW involves a combination of tablet and keyboard input (Fig. 1). Schematics are kept in a

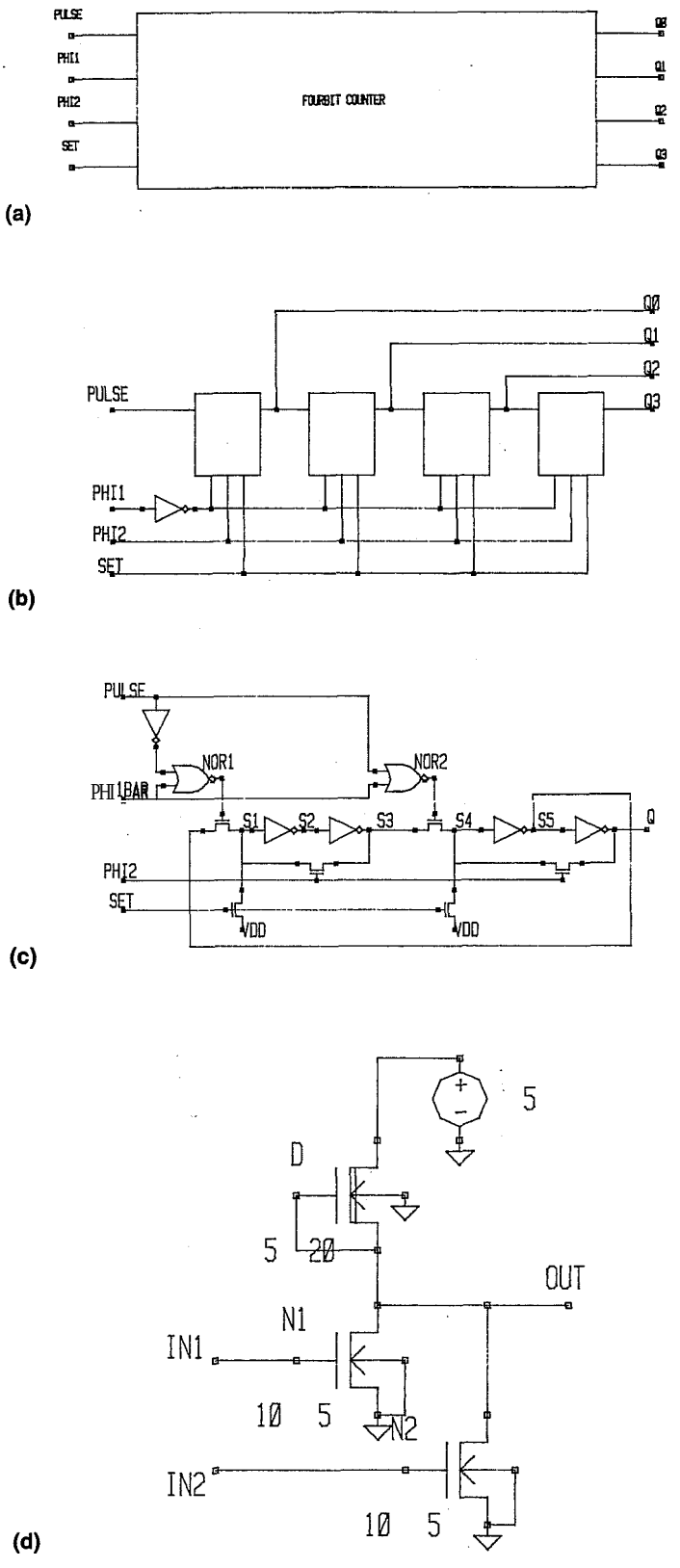


Fig. 3. Schematics for a four-bit counter, created *by* DRAW. (a) VIEW-level plot of the counter. (b) Block-level plot. (c) Block-level plot of a one-bit cell. (d) ***HP-SPICE*** plot of a NOR gate.

library which is managed by DRAW on the user's account. A "clean sheet of paper" is created by asking to edit a presently nonexistent drawing. The screen of the HP 2648A Terminal provides a window to the work area, as shown in Fig. 2. Both text and graphics displays are used. In Fig. 2a, the cursor has been placed on top of the ADD command and the stylus depressed. DRAW interprets the position of the cursor to be a request for the ADD command and highlights it as shown in the figure. The type of component to be added is typed at the keyboard together with its parameter values, if any. The placement of an NMOS transistor onto a partially completed schematic is shown in Fig. 2b. The WIRE command is used to interconnect the components. The use of this command is aided by the rubber-band line mode of the terminal. The cursor with a loose wire attached is shown in Fig. 2c. User-defined components that are to be used in higher-level schematics are defined by an external VIEW. The description of an external VIEW drawing is shown in Fig. 2d. The grid is provided to aid in the creation of an outline for the component, which is drawn using the WIRE command.

A four-bit counter can be used to illustrate the hierarchical description of DRAW. The highest level of description is shown in Fig. 3a as an implementation-independent four-bit counter with four inputs and four outputs. Figs. 3b, 3c, and 3d show schematics at different stages of design. Both user-defined components such as the NOR gates and SPICE components such as MOS transistors have been used. These schematics are easily incorporated into larger systems and can be ussd to drive simulators such as TESTAID-IC and SPICE.

Logic Simulation Using TESTAID-IC

TESTAID-IC is a multilevel digital logic simulator for verifying that a proposed design is logically correct and meets timing requirements. Circuit descriptions permit components to he described at Boolean, logic, and transistor levels. Data not appropriate for schematic display is kept in an associated text file. Such data typically consists of transition delay times, Boolean equations and other nongraphic descriptions. In response to a command, DRAW creates a text file containing a TESTAID description. Fig. 4a shows the output generated by TESTAID-IC after simulating the four-bit counter shown in Fig. 3b. At this level the designer is interested only in checking out the functional behavior of the circuit. An output format using 1s, Os, and Xs is chosen to display the results. The circuit can be run for all the possible states for checking out complete functionality.

Fig. 4b shows the output waveform generated by TESTAID-IC after simulating the one-bit latch shown in Fig. 3c. The waveform shows the functional response of the circuit to the user-specified stimulus at the logic level. The stimulus applied has a very precarious timing relationship between the SET and PULSE inputs and the two-phase clock. The close timing of the PULSE and PHI1BAR signals is shown circled in Fig. 4b. The waveform in Fig. 4c shows the consequences of the short delay between PULSE and PHI1BAR inputs. The NOR gate, NOR2, does not respond fast enough to the changes on the input side because of its inertia, and consequently it fails to toggle the Q line. The SPICE output shows the NOR2 signal starting to rise but returning to the

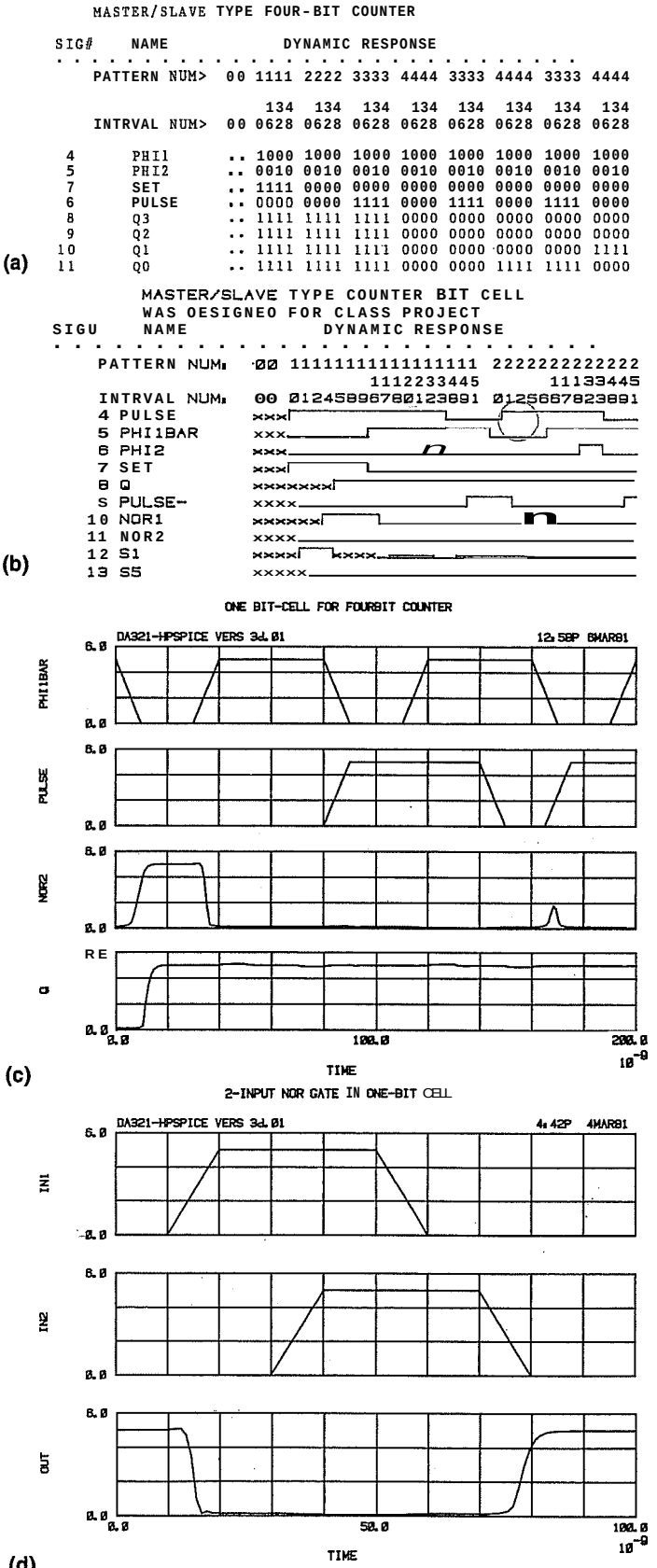


Fig. 4. *TESTAID-IC and HP-SPICE plots.* (a) Functional behavior of a *four-bit* counter. (b) Logic response of a one-bit counter cell. (c) HP-SPICE plot of a one-bit cell response, showing a toggle failure. (d) HP-SPICE plot of a *two-input NOR* gate response.

zero state instead of switching to the one state. Further study using SPICE would probably be done at the single-gate level. Other potential problems such as uninitialized nodes and nodes in the high-impedence state are shown with Xs and double lines, respectively in Fig. 4b.

A means to evaluate the effectiveness of circuit testing schemes is provided by a fault simulation option. Under fault simulation mode, circuit nodes are clamped to logic 0 and logic 1 sequentially. The proposed test vector (set of input values) is applied and the faulty outputs compared with the no-fault output. A difference implies that this fault can be detected. For a given set of test vectors, TESTAID-IC runs the fault analysis and computes the percentage of faults that will be detected. Fault analysis is used as an adjunct to functional testing, since a full functional test is not always economical.

Circuit Simulation Using HP-SPICE

The SPICE circuit simulation program developed at the University of California at Berkeley has become a workhorse for the IC industry. SPICE performs a detailed analysis of circuits using the interconnect data from DRAW. Accurate analytical models are provided for the semiconductor devices.

The DRAW display of a basic NMOS NOR gate is shown in Fig. 3d. The library of circuit primitives provides the transistors, capacitors, resistors and other SPICE elements. Stored in the associated text file are device model parameters, source excitation, and SPICE analysis control information.

A typical output from HP-SPICE is plotted in Fig. 4d. The waveforms show the gate response for worst-case conditions of circuit speed. The signal delay through a typical gate is often studied using these simulated waveforms. Circuit designers typically perform hundreds of similar simulations during the course of a design. The simulations can provide the detailed information necessary for assuring that the final design will be reliable. The program's interactiveness, flexibility, and graphical output have been found to be important contributors to designers' productivity.

While simulators are usually most effective for a certain level in the circuit hierarchy, it is vitally important for simulators to be usable at adjacent levels. On the whole, simulations are usually carried out in the most economical manner, with the assurance that sticky problems can be solved using less efficient but more accurate simulations at the next level of detail. The counter cell shown in Fig. 3c is made up of the gates whose switching performance is shown in Fig. 4d. This simulation could be used to confirm timing calculations done from rough extrapolations from the gate performance.

Conclusions

The advent of VLSI has forced the designers of both systems and design tools to come to grips with the necessity for multilevel system descriptions. The schematic-based system, DRAW, has been offered as one appropriate means for describing VLSI systems. The simulators that have been interfaced to DRAW provide a powerful capability, but are not adequate for evaluation of complete VLSI designs. Simulators and description languages focusing on levels higher than logic states will have to be integrated into a VLSI design package. In addition, the need to simulate

Transistor Electrical Characterization and Analysis Program (TECAP)

by Ebrahim Khalily

As simulation becomes a cornerstone of VLSI design and as technology advances toward fabrication of small-geometry devices for VLSI, greater process control and more accurate simulation models become more important. With the increases in the complexity of device models, an automated characterization system is needed to perform quality transistor measurements and provide reliable model parameters interactively and in a reasonable time.

The Transistor Electrical Characterization and Analysis Program (TECAP) is a characterization system based on the HP 98458 Desktop Computer and several HP-IB-compatible measurement instruments. The system is intended for the IC design environment, where understanding of device and circuit limitations is important. TECAP is designed as a general measurement and characterization tool, with interactive measurements on a variety of devices, including resistors, capacitors, bipolar NPN and PNP transistors, enhancement and depletion mode P and N-channel MOSFET transistors, and JFETs. The model parameters for bipolar and MOS transistors are obtained from these measurements. The extracted parameters are then used in a model analysis program to check the validity of the model through direct comparison with measured results.

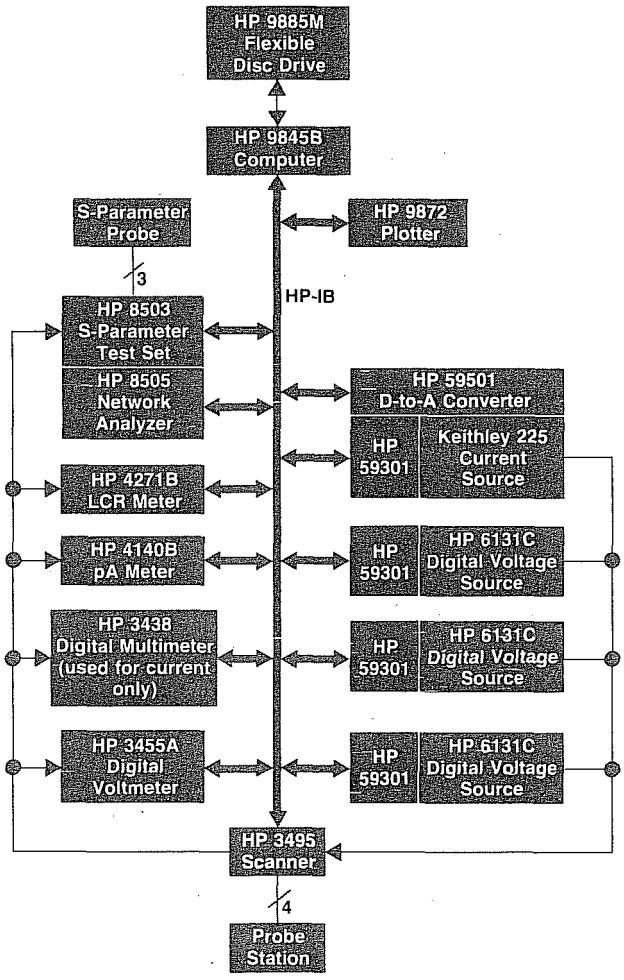


Fig. 1. Block diagram of the TECAP (transistor electrical characterization and analysis program) system. TECAP makes measurements on bipolar and MOS transistors and computes the parameters of models of these devices.

The HP 9845B Desktop Computer is used as the system controller. The enhanced BASIC language provides an easy-to-program, easy-to-use system. A flexible disc drive provides accessible storage for programs and data. The graphics capability of the HP 98458 and the HP 9872 Multicolor X-Y Plotter give fast visual feedback to the user.

The HP-IB* is the communication link between the computer and all of the instruments in the system. The HP-IB commands are easy to program, and the standard connections of the HP-IB provide hardware modularity, allowing any instrument to be replaced by another HP-15-compatible instrument without any hard wiring and with minimum software modifications.

The block diagram of the system is shown in Fig. 1. Three HP 6131C Digital Voltage Sources provide ±100V, 0.5A outputs with a resolution of 1 mV. An HP 59301 ASCII-to-Parallel Converter interfaces each source to the HP-IB. Current sourcing capability is provided by a Keithley 225 Current Source modified for remote programming. An HP 59501 Digital-to-Analog Converter is used to control the current source in the range of ±100 mA with a resolution of 0.1 nA. Voltage measurement is done by an HP 3455A Digital Voltmeter with a resolution of 10 μV. Low current measurements (less than 20 mA) are done by an HP 4140B pA Meter with 1 fA resolution. Currents greater than 20 mA are measured by an HP 3438A Digital Multimeter. The HP 4271B LCR Meter provides capacitance measurement capability from 0.001 pF to 10 nF. The HP 8505 Network Analyzer accompanied by an HP 8503 S-Parameter Test Set enables the system to measure high-frequency (0.5-1300 MHz) parameters. The 3495A Scanner, a matrix switch constructed of 40 programmable two-pole switches, connects the instruments to the four terminals of the system. These four terminals can be either directly connected to the device under test or connected to a probe station for on-wafer measurements.

To maintain system modularity, TECAP uses separate subroutines to control each instrument on the HP-IB. These subroutines enable the user to write or modify the programs using a high-level language. Each section of the program is stored in a separate file on the system flexible disc, and is loaded into the computer and executed only when it is needed. Thirty-two function keys on the desktop computer provide fast access to different tests and measurements.

In general, the application of the system can be divided into two different categories, process control and device characterizations.

A process control engineer is more interested in data on physical parameters of the process. For example, C-V (capacitance versus voltage) measurements can help determine oxide thickness, surface state charge density Q_{ss}, and threshold voltage.

*Hewlett-Packard's implementation of IEEE Standard 488-1978.

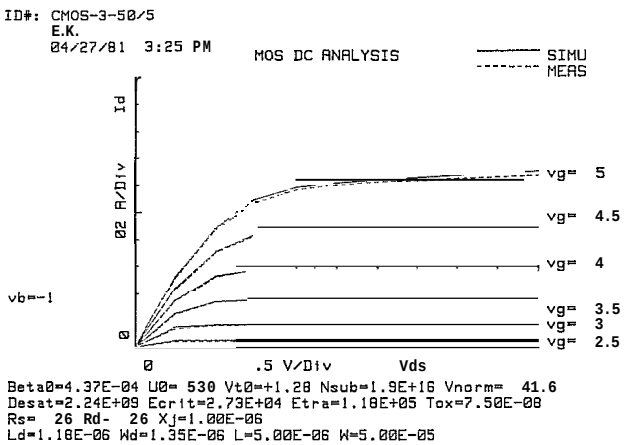


Fig. 2. A TECAP dc characterization of an *MOS* transistor. Model parameters have been extracted from measured data and the simulated result is compared with the measurement.

Also, parameters like sheet resistances and breakdown voltages yield valuable information about the quality of the process. Statistical analysis of the data reveals process variations.

An IC designer, on the other hand, uses the electrical characteristics of the device. Parameters like saturation current I_s and forward current gain β at different current levels for bipolar transistors and mobility μ_0 for MOS transistors are some of the essential design parameters. Special phenomena such as saturation, subthreshold, and punchthrough of a transistor are also of great importance in the design of ICs. With an accurate computer model, the only obstacle to successful circuit simulation is the determination of good model parameters. TECAP extracts model parameters through direct measurements on actual transistors. Parameter extraction programs can then be merged with model and parameter analysis programs to compare the simulations to the measurement, thereby checking the validity and accuracy of the model. The MOS and bipolar models used in TECAP are similar to models available on HP-SPICE.

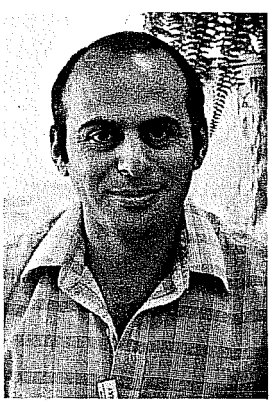
TECAP can also be used to tie the circuit simulators to the device and process simulators. The output of the process modeling program HP-SUPREM, which simulates the impurity profiles in different cross-sections of the process, is used with device modeling programs like SEDAN or GEMINI, which generate device I-V (current versus voltage) characteristics. TECAP can then work with simulated I-V data instead of measurements to extract the model parameters for SPICE. This complete line of process device and circuit simulation allows users to simulate circuits based on process data, and is a powerful tool for studying the effects of process parameters on the performance of the final circuits.

Fig. 2 shows an example of dc characterization on an MOS transistor. The model parameters, including the parameters for short-channel effects, are extracted from measured data and the simulated result is compared with the measurement. The analysis part of the system lets the user simulate and plot drain current and its derivatives and compare them with the measurement.

Acknowledgments

TECAP, SUPREM, SEDAN and GEMINI are programs that have been developed at Stanford University under the direction of Bob Dutton and sponsored by both HP and the Advanced Research Projects Agency (sponsored by the U.S. Army Electronics Technology and Devices Laboratory).

Ebrahim Khalily



Ebrahim Khalily is a development engineer specializing in modeling and characterization with HP's design aids group. Born in Kermanshah, Iran, he received his BS degree in electronics from Tehran Technical University in 1973 and his MS and Engineer degrees in integrated circuits from Stanford University in 1978 and 1979. He joined HP in 1979. He has authored papers on automated characterization and process and device simulation and is a member of IEEE. Ebrahim lives in Cupertino, California and enjoys outdoor activities, sports, music, and bridge.

larger structures in existing simulators has spawned multilevel simulators to deal simultaneously with detailed and logical signal levels. While much work is yet to be done to improve the capabilities for designing with VLSI, one must not lose sight of the designer's fundamental concerns: Will the design work? Will it perform to specifications? Can it be manufactured and tested at reasonable cost?

The computer is able to perform a larger role in providing the information necessary to answer these questions. We must recognize that as tools become more sophisticated, design requirements are keeping pace. If we are to remain successful, we must provide a design environment that has the flexibility to permit plenty of "good old-fashioned engineering" to be added by creative design engineers.

Acknowledgments

The contributions in simulation from Professors D.O. Pederson and A.R. Newton and their students at the University of California at Berkeley are gratefully acknowledged. We appreciate and acknowledge the semiconductor device modeling work done under the direction of Professor R.W. Dutton at Stanford, which plays an essential role in the usefulness of SPICE. The DRAW team of D.B Gorman, K.M. Mann, and J.P. Skudlarek has done a fine job of treating a real design tool, as has the HP-SPICE team of I.D. Klein

Louis K. Scheffer

Lou Scheffer is a member of HP's design aids group and a PhD candidate at Stanford University. He received his BS degree in engineering in 1974 and his MS degree in electrical engineering in 1975, both from California Institute of Technology. He joined HP in 1975. He has designed digital filters for the 3582A Spectrum Analyzer and developed microprocessor assemblers and a high-level language. He has authored three papers on IC design tools. Lou is a native of Rochester, New York. His hobbies are audio electronics and bicycling.