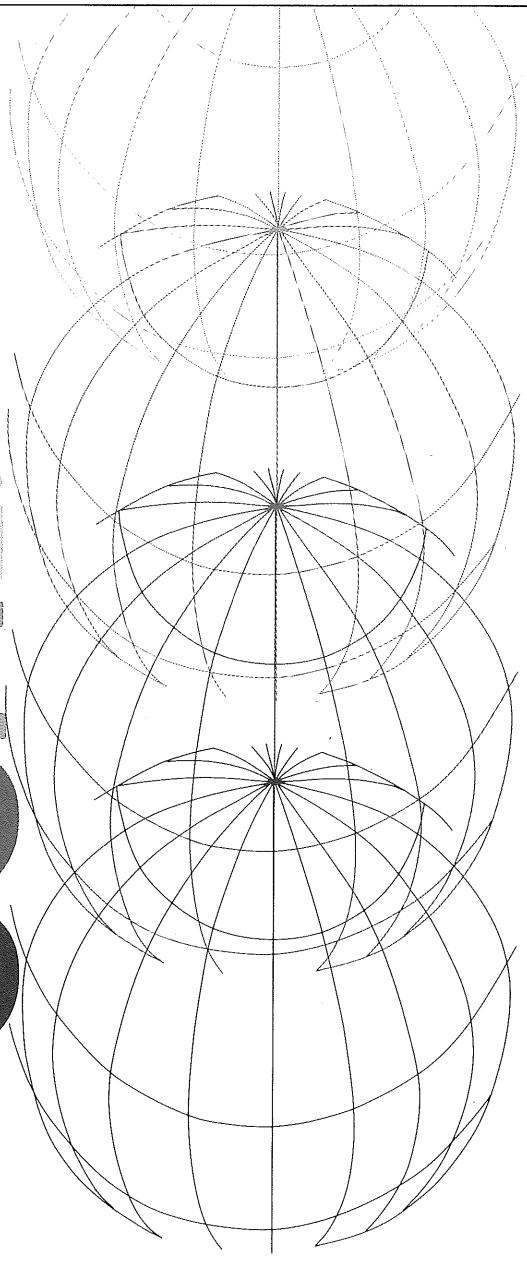
TEK INFORMATION FOR TECHNOLOGISTS

MAY/JUNE 1984

TECHNOLOGY report

COMPANY CONFIDENTIAL



JOINING THE CAD WORLD

Tektronix®
COMMITTED TO EXCELLENCE

CONTENTS

Volume 6, No. 5, May/June 1984. Managing editor: Art Andersen, ext. MR-8934, d.s. 53-077. Cover: Jackie Miner; Graphic illustrator: Nancy Pearen. Composition editors: Jean Bunker and Sharlet Foster. Published for the benefit of the Tektronix engineering and scientific community.

Why TR?

Technology Report serves two purposes. Long-range, it promotes the flow of technical information among the diverse segments of the Tektronix engineering and scientific community. Short-range, it publicizes current events (new services available and notice of achievements by members of the technical community).

HELP AVAILABLE FOR PAPERS, ARTICLES, AND PRESENTATIONS

If you're preparing a paper for publication or presentation outside Tektronix, the Technology Communications Support (TCS) group of Corporate Marketing Communications can make your job easier. TCS can provide editorial help with outlines, abstracts, and manuscripts; prepare artwork for illustrations; and format material to journal or conference requirements. They can also help you "storyboard" your talk, and then produce professional, attractive slides to go with it. In addition, they interface with Patents and Trademarks to obtain confidentiality reviews and to assure all necessary patent and copyright protection.

For more information, or for whatever assistance you may need, contact Eleanor McElwee, ext. 642-8924. □

WRITING FOR *TECHNOLOGY REPORT*

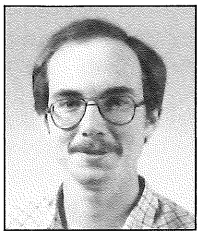
Technology Report can effectively convey ideas, innovations, services, and background information to the Tektronix technological community.

How long does it take to see an article appear in print? That is a function of many things (the completeness of the input, the review cycle, and the timeliness of the content). But the minimum is six weeks for simple announcements and as much as 14 weeks for major technical articles.

The most important step for the contributor is to put the message on paper so we will have something to work with. Don't worry about organization, spelling, and grammar. The editors will take care of that when we put the article into shape for you.

Do you have an article to contribute or an announcement to make? Contact the editor, Art Andersen, 642-8934 (Merlo Road) or write to d.s. 53-077. □

AN INTRODUCTION TO COMPUTER AIDED ELECTRICAL ENGINEERING



Glenn Widener is a hardware/software engineer III in Logic Design Systems (LDS). He is developing a user interface for viewing simulation results. Glenn joined Tek in 1980 after receiving his BSEE from the University of Texas. His extension is 629-1521 (92-826).

In this issue, eight Tek engineers will try to help you overcome the barriers that will stymie your attempts to use present CAD tools, discuss what must happen before CAD tools can achieve their potential, and excite some of you about the opportunities for doing very significant work in CAE at Tek. In other words, nothing ambitious . . .

My article, or our issue of *Technology Report,* alone won't do this, but it's a start. Engineers face some "exciting" times. The whole nature of engineering is changing. As an engineer you will do CAD – there will be no choice.

This article has four objectives:

First, I want to raise your expectations by giving you an idea of what CAD tools can do and how they will change the engineering process.

Then we will discuss the problems with CAD tools as they exist today, and how you can get more out of these tools.

Next I want to share my opinions of what must happen in the CAD world to allow these tools to achieve their potential, and encourage some of you to get involved in the development of CAD tools at Tek.

Finally, we will try to put CAD tools, CAE..CAX..CAY..CAZ..whatever, in their place by presenting a concept of the engineering process and where various CAD tools fit in that process.

You and CAD

Why Should You be Interested in CAD?

To be blunt, CAD will dramatically alter the way you do your job. That fact may be perceived as a threat, but I prefer to see the promise. Let's talk for a minute about some of the things that CAD can do.

Design verification

CAD excels at design verification – or it will. It does the boring, repetitive verification tasks automatically and much more accurately than most engineers care to do manually. CAD has the potential to manage complex systems, taking care of the myriad details that the designer hasn't the time to worry about.

When we in Logic Analyzers started doing our TECLI (ECL) gate arrays, with the cooperation of Monolithic IC Engineering, many of the verification tools now in place were still under construction. Each time a chip was completed, something died. And wouldn't you know it, the failures were always in the area that had been only hand checked – even though five people had checked it! Only when 99 and 44/100% of the array was automatically verified did we start getting chips that worked the first time.

CAD is essential in VLSI because one error can blow a schedule and be very expensive. (You have probably heard the cliche: If at first you don't succeed, give us another $20K.) While an error at the board level is not as serious, it can still add many weeks to the development cycle. Logic Analyzers bought Daisy workstations to help eliminate board-level errors.

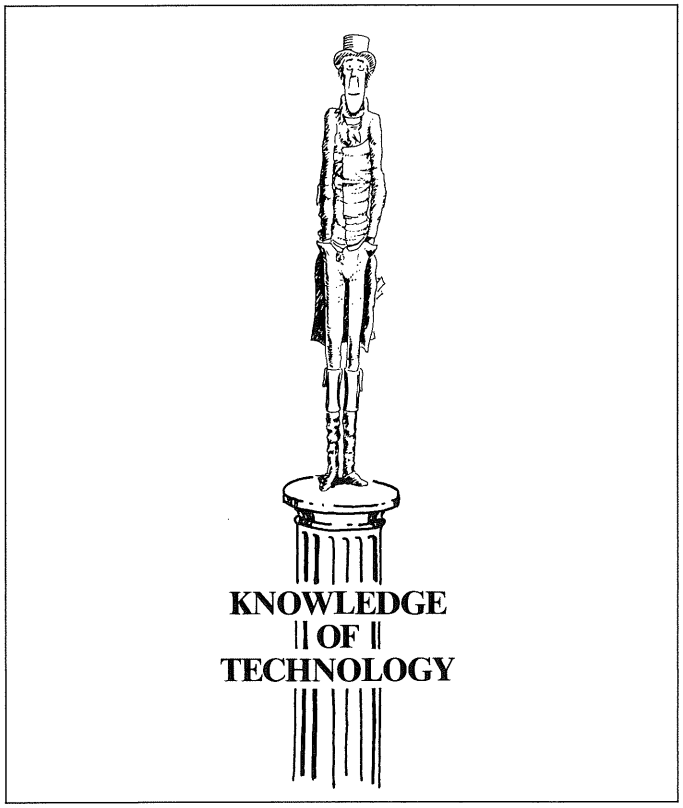
Knowledge-base

CAD tools are the pedestal that the "tall thin man" stands on. They are his connection to reality, the knowledge-base that allows him to concentrate on system design instead of spending immense effort gathering and sorting information about technology.

It won't be easy to be an engineer in the 90s, it's not easy now. The whole nature of electrical engineering is changing.

The concept of the knowledge-base is a major ingredient in Technology Group's *Design Base Extension* (DBX). The DBX program seeks to put complex technology in the hands of division designers. It is the knowledge-base that enables CAE to be a powerful teaching tool as well as a means of overcoming the shortage of technologists. Let me use my experiences with the grandaddy of CAE tools, SPICE, to show how this works.

Most of what I know about high-speed logic I learned through SPICE. When I came to this company four years ago, I knew a little software, a lot of microprocessor hardware, and a little analog – as long as it was below 20 kHz. A year later I was thrust into the magical world of 1-gigabit logic, transmission lines, and custom ECL. Although I had forgotten most of the equations describing transistors, I have designed some of the highest-speed digital circuits ever made at Tek. SPICE made this possible because it embodies the *knowledge of the best transistor experts.*



SPICE is a circuit simulator that allows you to forget about math and concentrate on what a human does very well: intuitively analyze what will and what won't work. SPICE and its transistor models are so good that, with only minor reservations, a circuit designer can treat what SPICE tells him as Gospel. It's as close as an IC designer may ever get to a detailed picture of reality. (Any SPICE user who has tried to probe a real IC can testify to this!)

Communication vehicle

Lynn Conway, in her paper "The MPC Adventures," discussed the importance of computer communications to her work on the Multichip Project at Xerox Park and Caltech. This experiment placed more than 20 designs from geographically scattered groups on one chip. Its success hinged on good communications that linked hundreds of computers at a dozen or more sites.

The net was not only a means of transmitting design databases, it carried gossip. When someone hit a bug or found a fix for a bug, the news spread rapidly to everyone. And too, the net was a forum for discussing ways to improve the MultiChip program itself. We could do much the same if we took full advantage of the UNIX net here at TEK.

Communication sums up the whole engineering process – after all, the entire output of engineering is information – the instructions on how to build the product. Jack Hurt discusses this aspect of CAD in *The Direction of CAE at Tek.*

A discipline of digital engineering

You may notice echoes of Edgar Dijkstra and structured programming methodologies in CAE; this is no accident. When computer tools are introduced to a design process, like it or not, discipline is introduced too. Computers thrive on order. Software and hardware disciplines will converge thanks to CAD. The future digital engineering world will be populated by lots of system designers supported by expert systems. Behind the expert systems will be the handful of technology wizards. Much of the design process will be freed from the underlying implementation, whether hardware or software.

Shorter design cycles

Fewer design iterations, rapid introduction of technology, multiplication of experts, faster and efficient communication, and the promotion of good engineering practices will all speed product development. The pressing goal of management – reduced time to market – can be approached only through CAD. Earlier time to market translates into big bucks for the company which invests in CAD tools.

Do Engineers Really Want This Kind of CAD?

But do engineers really want to work faster? After all, the old way can be pleasant. During our recent gate-array project I found nothing more relaxing than sitting down for an hour or so to layout a gate-array cell by hand. Will we resent losing much of the implementation portion of our jobs? Does every engineer really want to be a system designer?

A few months ago, I finally got around to reading *Zen and the Art of Motorcycle Maintenance,* something I've wanted to do for sometime. In this book, an English teacher/philosopher/technical writer named Robert Persig, addresses part of the question we have just asked. Persig explores the philosophical relationship between a craftsman and his craft. In his examination of this relationship, Persig's key question becomes "what is quality?" That is our question too: What is quality?

I am convinced that satisfying this human need . . . for a caring relationship between the designer and his tools is the real promise of CAD . . . So the task is a joy, not drudgery.

When we think of quality, we usually think about such things as failure and defect rates, production yield, safety, meeting specs, and other such concrete terms. But there is another aspect of quality, an aspect often neglected because it does not appear directly on anybody's balance sheet. It has to do with things like an engineer's pride, satisfaction, fulfillment – the customer's pride of ownership – the employee's pride in the quality of his work. This is the kind of quality that Persig is interested in – the kind that makes a job have meaning, the quality that in itself makes a job enjoyable and rewarding.

Thus, according to Persig, quality is far more than a number such as field-failure rate. It is the product of a *caring relationship* between the craftsman and his product, the artist and his art – a relationship that is realized *through his tools.* The most important thing a tool can do is to support and build this kind of quality, the quality of the craftsman-product relationship, so that the task is a joy, not drudgery. When this kind of relationship exists, when the designer puts a piece of himself into his work, the numbers that are the evidence of quality will be there. Just achieving the short-term goal of time-to-market is not enough.

It will be the recognition of the principles, the human values, of quality that will keep a company like Tek pre-eminent in business. I am convinced that satisfying this human need that is at the heart of what CAD can do for the designer at Tek. This caring relationship between the designer and his tools is the real promise of CAD and should not be overlooked in the rush toward ever shorter design cycles.

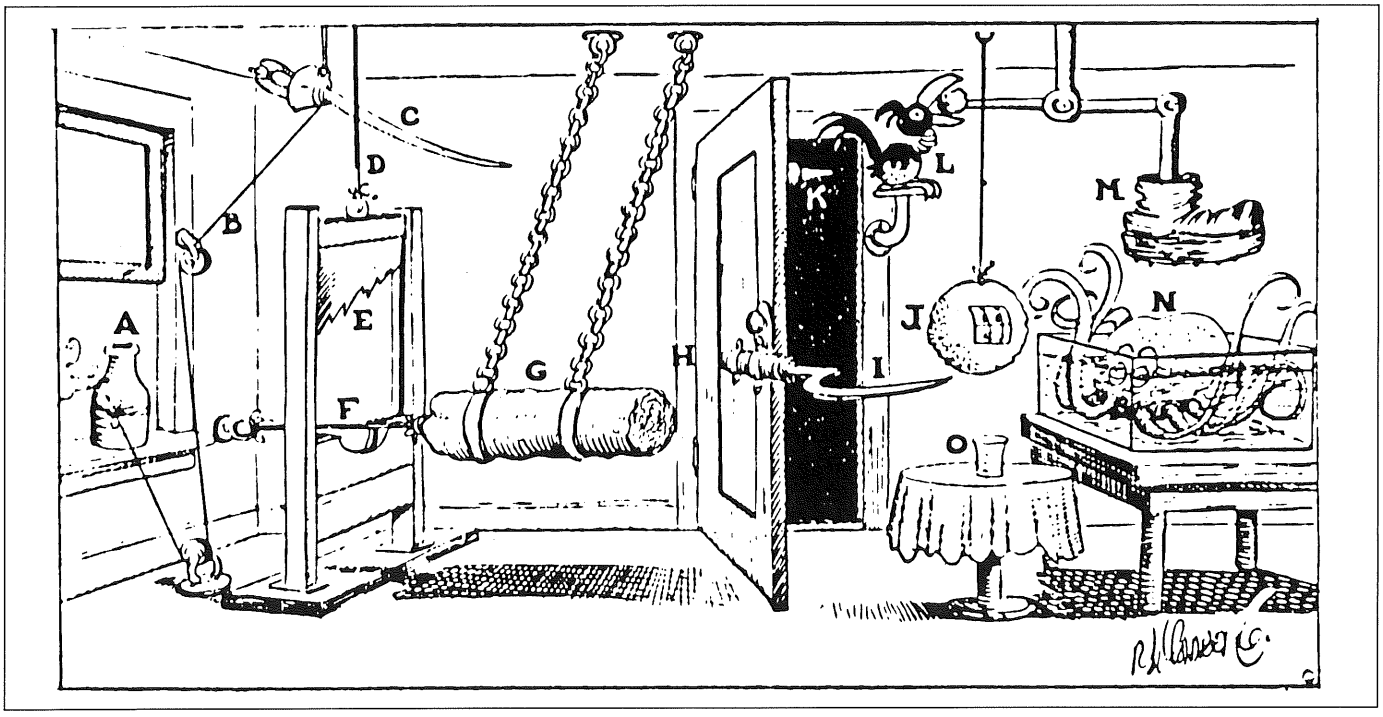
Caveat Emptor

We have discussed what CAD can or will do, but what can it really do now? Well for every ideal, there is hard reality. The fact is, no CAD system exists – at least nothing that I would call a CAD system. No engineering organization – anywhere – has more than a *collection* of tools, each of which ranges from excellent to lousy. Where attempts have been made to produce a system, the process usually follows Rube Goldberg's methodology. Rube was a cartoonist who made a good living depicting wildly imaginitive kluges in nationally syndicated cartoons. He was a master at integrating things that were never designed to work together.

It's bad enough that CAD systems are put together this Goldbergian way; what's worse is the kind of picture they present to the user: disordered, bewildering and *complex.* The dismay fostered by this complexity is compounded because the system bears no resemblence to the way an engineer thinks about the job.

Not exactly conducive to quality design! In fact, Persig would have a field day with such systems. In *Zen and the Art of Motorcycle Maintainance,* Persig described an art he calls *gumptionology* – that is, the art of keeping your gumption in the face of frustration. CAD today is full of gumption traps, things that make you want to throw up your hands, quit, go home and sleep it off. I can't count the times that I have walked away from a computer terminal mumbling something like "why can't it do this, it knows everything it needs to do it!"

One of the bigggest problems facing the uninitiated is simply finding the right tool for the job. At one of the recent EAC Seminars on CAD I surveyed the crowd by asking, "How many digital hardware types do we have in this room? Raise your hands



Inventions of Professor Lucifer Butts

Professor Butts steps into an open elevator shaft and when he lands at the bottom he finds a simple orange squeezing machine. Milk man takes empty milk bottle (A) pulling string (B) which causes sword (C) to sever cord (D) and allow guillotine blade (E) to drop and cut rope (F) which releases battering ram (G). Ram bumps against open door (H) causing it to close. Grass sickle (I) cuts a slice off end of orange (J) at the same time spike (K) stabs "prune hawk" (L). He opens his mouth to yell in agony, thereby releasing prune and allowing diver's boot (M) to drop and step on sleeping octopus (N). Octopus awakens in a rage and seeing diver's face which is painted on orange, attacks it and crushes it with tentacles, thereby causing all the juice in the orange to run into glass (O).

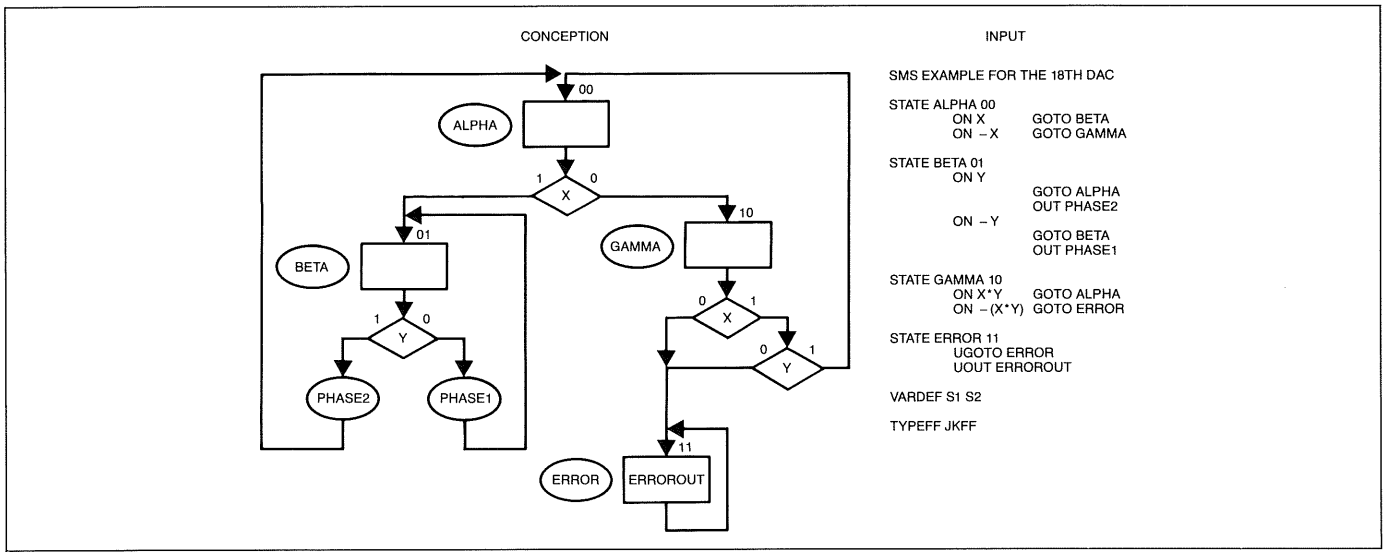


Figure 1. SMS (for State Machine Synthesizer) is a pretty good tool for minimizing logic expressions. But despite its proven ability, too many hardware state-machine designers don't know of it. For details, see *CAX Center Tools for EEs* later in this issue. (This figure does not show the minimized expression that is the product of SMS.)

please." About 30 responded. Next, I asked "How many of you digital types have designed at least one hardware state machine at Tek?" Most raised their hands. "Now, how many have heard of SMS? Actually used it?" By this point, only a couple of hands were left!

SMS is a tool to describe a state machine in terms of state transitions and outputs; it produces minimized logic expressions. These expressions can be inputs to programs like PALASM, which builds a programming image for programmable array logic (PAL), the hardware for implementing the state machine. If you are designing more than a 2-bit state machine without SMS, you are in for a lot of grunt work! (Jim Murphy describes SMS in *CAX Center Tools for EEs.*)

Yes SMS is useful, but it has its problems, problems typical of much of Tek CAD software. Tek software tends to be very like university software: Some excellent work on algorithms – after all that's what makes for a good graduate thesis, and algorithms are more fun and challenging than such mundane tasks as user interfaces and integration with other tools.

The input to SMS is text (see figure 1), not something nice like a state diagram. The output has to be edited before sending it to PALASM for creating PAL. What's more, there is no connection to any simulation, no integration. This lack of integration of CAD tools is why it is hard to find the tool you need – there is no system to lead you to it. Without integration, each step in the design process is an end in itself. The steps deliver products that without some modifying won't mesh with the tools used in the next step of the process.

What You Should Do About It

So what can we as users do about the immaturity of CAD?

From my experience in Logic Analyzer Engineering, I would like to offer several suggestions:

Recognize up front that CAD tools are immature and learning about them and using them will not always be pleasant and easy. Real gumption traps exist that are deepened by excessive expectations!

Spend the time to evaluate the tools carefully. It can't be done in hours, it takes days and weeks. Use the tool on real samples of your work – the results will be enlightening.

Work with your vendor. He is struggling to produce a solid product, and needs your help. The way to get good tools, at least the next time around, is to work with the supplier, taking the time to define the problems. Criticize, but be sure to suggest how to make the tools better. CAD companies, including TEK, are spending an incredible amount on R&D, and you'd be surprised how fast your ideas can get implemented. You are the expert on how you do your job, so offer some expert advice!

Go ahead and join the CAD world in spite of the problems I have talked about. CAD is already producing, significantly in many areas. Even if you don't find a tremendous gain the first time around, the education itself will be worth your sweat.

The CADEE world cries for interdisciplinary engineers – people that understand both hardware and software design. Too often the CAD-tool designer struggles, ignorant of the way you, the EE, will use the tools he writes. At Tek, right now, experienced hardware types who can also do software have some spectacular opportunities to directly influence and build the CAD tool standards of the 21st century. If you don't like what you see and you know what you want, maybe you should try your hand at it.

What Must Be Done

Now I will voice my opinions on four things that must happen for these tools to achieve their potential.

One, *complete the tool set* – The gaps need filling. It's the major, obvious tools that exist, while the little tools that should link

the major tools are missing. As we have seen with the SMS-to-PALASM connection, and the Daisy-to-ECB-layout connection, the little tools are often left to the user to cobble up. Every phase of engineering needs coverage – a tool for every task, no exceptions! Particularly weak now are tools for the higher levels of the design process.

Two, *close the loops* – We need tools that talk to each other, no more data re-entry! We must understand data flow and organization as well we understand how to get to work in the morning. This understanding is the basis for developing common data bases and good formats for transfering data between data bases. And please, let's get everything running under one operating system with one text editor!

Three, *make REAL expert systems* – I define a real expert system as one that "understands" not just technology, but the engineering process too.

Real expert systems don't exist yet, but they are coming. If we are not alert, the "experts'" idea of how we do our jobs may fail to match our ways of doing things. Again, it is our responsibility to get involved in developing the tools that will shape the way we do our work. We, as Tek engineers, know that each engineer's approach to the job is individualistic. The system must accommodate reasonable individualism so that styles can differ without the design process degenerating into chaos. My favorite example of this is the schematic bubble-logic convention – some designers worship it, others hate it!

We must deeply understand how we do our jobs: the ways each of us is different, what parts of our individual methods are essential, and what parts can be changed without damaging the craftsman-tool relationship.

Four, *design a good user interface* – A good user interface means one unified interface for all applications. Designers must spend less time learning and more time designing. The interface is where the system really comes together. It's the thing that helps users find their way around the "workbench," It's the handle on the screwdriver and it must feel right. We are creators in a creative business, and the tools we use must fit our hands and our jobs comfortably.

The preceding four imperatives, in fact, describe the major efforts in CAD development, at TEK and elsewhere. Logic Development Systems (LDS), for example, is focused on completing and integrating the tool set. Tek's long-term goal is to connect the tools, making a system to run on one operating system with common data bases, and one user interface.

The Engineering Process

Now let's attempt a brief introduction to the world of CAD and a few of its tools. Rather than talk about the tools individually, I'd first like to see if we can come up with a general idea of what the engineering process looks like, and then fit these tools into that process. Let's try to diagram your job.

The representation of electrical engineering in figure 2 is obviously too simple for a profession it may have taken you six years to get into. In the horizontal direction are a number of different technologies that we can use to implement design. In the vertical direction we have a top-down design starting with an idea at the top and, optimistically, ending up with a working system at the bottom. We work down through various design processes, increasing detail and hopefully getting more like reality and less abstract.

One of the first things we must decide is whether the problem is digital or analog. If we get into the digital world we have to decide whether we implement with hardware or with software, and so on down. One way to think about the different levels of design is to look at what questions are being asked at each level.

At the top level (in figure 2), we start with the IDEA. We ask what is it we're trying to accomplish? Why are we doing this in the first place? Next we ask WHAT? In the computer world, the answer to "what" is data; it's information. Our objective is to manipulate and move information around and that is the prime focus in the early stages of the design.

Once we know what we're trying to accomplish, we start to describe the actual logical devices, the modules, the gates, the wires, the interconnections, where the data is going to flow.

In LOGIC, the designer uses a schematic or whatever it takes to describe the interconnection of a bunch of devices. Next, we have to think about how these devices are to be used in a PHYSICAL IMPLEMENTATION; that is, the kind of an ECB we are going to build, how the wires look, what's in silicon, what are the transistor characteristics. All this relates to physics.

Then we get into actually building PROTOTYPES, and eventually into MANUFACTURING parts.

At the bottom of the representation we get down to the OOPS phase. We've done all this wonderful work, but the system we've implemented doesn't work. We have to back up. And we discover the process isn't quite as top down as we thought! One of the key things CAD tools will do for us is to move the oops phase earlier into the design process, so we'll find our mistakes a little sooner than we have in the past.

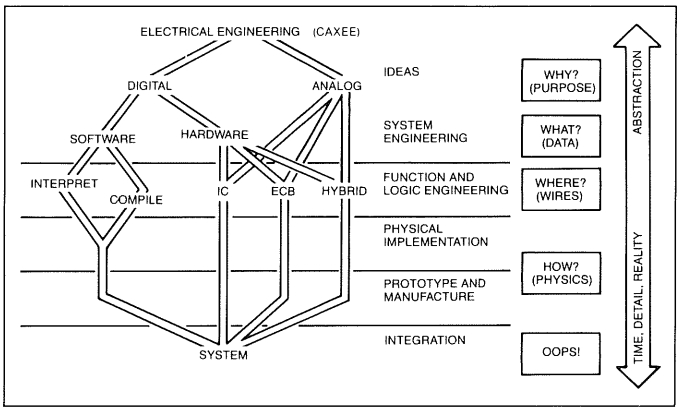


Figure 2. The electrical engineering process from idea to integration. Computer aided engineering will eventually affect every segment. Figures 3 and 4 represent what is now in place to support hardware development.

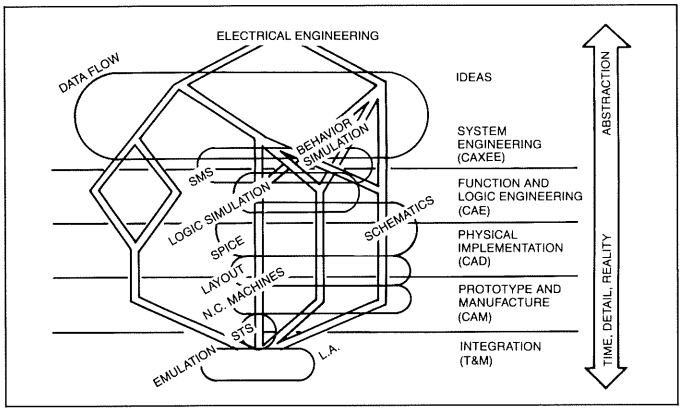


Figure 3. Terms and tools of CAXEE. Many of these tools are applicable across technolgies, hence the term "horizontal integration."

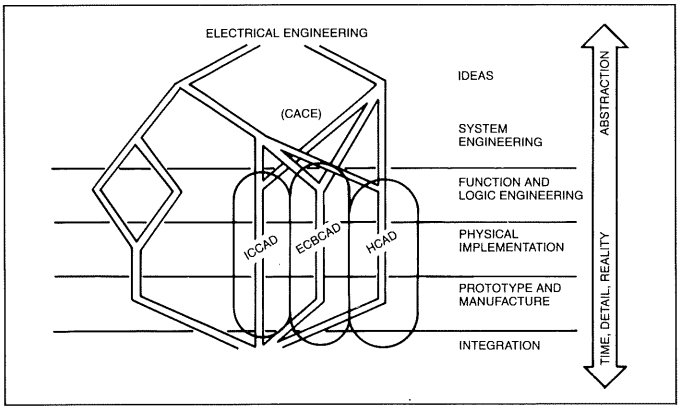


Figure 4. Vertical-integration has been the first step towards a true CAD system. Various tools are "cobbled" together and applied to a particular technology. The systems shown are in use at Tek today, and will be discussed in other articles in this issue.

Now let's start to drop in some alphabet soup and the examples of tools on our representation. Let's start at the bottom of figure 3 with Tek's traditional business, *test and measurement tools.* Examples here are the kind of tools we produce in the Design Automation Group: logic analyzers and emulators and so forth, which are aimed at integrating the system, and test systems, which are tools for checking out production parts. As we move into computer aided manufacture (CAM), we see things that take information that has been produced by a computer and build a part, things like numerically controlled machines.

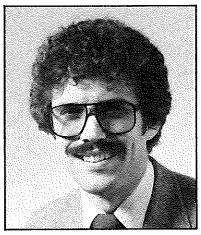
Physical implementation was the area where the electronics industry first applied CAD, particularly to IC or ECB layout. We all have heard of companies like Applicon and Computervision; this is where they started. Because of this history, the term CAD is often associated with the layout level of the design process, but we also use "CAD" in a broader sense because it's the only acronym in the bunch, that is the only easily pronounceable "word."

SPICE bridges the gap between the physical and implementation levels and the function and logic level. We use SPICE to define the behavior of the models used at the functional level. There are many such tools that help us transition from one level of the design to the next. The function and logic area is now the domain of the CAE workstation folks, who use the term CAE to differentiate themselves from CAD efforts concerned only with layout. In CAE we have schematic entry systems, logic simulators of all shapes and sizes, and things like SMS, all of which move toward higher levels of the design process, but still deal, to a large extent, with interconnecting logical devices.

Now, the Rest of the Story

I've attempted to introduce EE CAD, tried to convey my enthusiasm and frustration. My fellow authors continue the story with these articles:

HCAD: CAD TOOLS FOR HYBRIDS



Clayton Mohr is a senior engineer. From 1976 until recently, he has held various positions in the Hybrid Circuit Engineering (HCE) (now called Hybrid Microelectronic Development (HMD). Initially, he was involved in laser-trimming engineering. Next, he was involved in modeling the laser trimming of hybrid-circuit and wafer-level components for two years. Then, until his recent move to the Logic Analyzer division, he has been in the HCAD development project.

Most of you are familiar with hybrid circuits, but I will include a brief overview for those who are not. Figure 1 shows several typical hybrid circuits. Hybrid circuits are usually larger than ICs and smaller then ECBs. Hybrid circuits are formed by screening or depositing conductors and circuit components on a ceramic substrate. Integrated circuits and passive components may also be attached to the substrate. Designers usually choose to use hybrid circuits to increase performance, save space and weight, and to increase reliability.

Although many of the CAD tools at Tek, such as SPICE and other simulators, can be used during the design of a hybrid circuit, I will address only the HCAD system. HCAD is a set of tools that Hybrid Microelectronic Development (HMD) has developed specifically for implementing hybrid circuits. With HCAD, the designer can lay out and analyze a circuit interactively on the screen of a graphics terminal or workstation. The user interacts with the HCAD program by typing commands on the terminal keyboard and places components by positioning graphic representations with a thumbwheel-controlled crosshair.

Figure 2 shows the major physical components of the HCAD system. Most of the HCAD system resides on a VAX 11-780 in building 59; the analysis programs which are called under program control from the VAX reside on the Cyber mainframe. The HCAD program supports most Tek graphic terminals, including the 4107, 4109, and 4115B terminals. The major outputs from the system are pattern data for generating artwork and for hybrid-fabrication and documentation plots used for fabrication-process control. The pattern data are stored on magnetic tape.

HCAD performs four major functions: component synthesis, interactive layout, layout analysis, and documentation-plot creation. I will address each in the order in which the designer would usually use them to design a hybrid. The user may revisit any of the functions at any point in the design and has much control and flexibility in the redesign process.

Component Synthesis

Figure 3 shows examples of the components that are automatically synthesized by HCAD. At the top of the figure are laser-adjustable resistors, in the center, laser-adjustable capacitors, and, at the bottom, inductors and a number of standard patterns such as IC attachment pads and process alignment marks.

In figure 4, you see the information that the designer must supply to HCAD for component synthesis. The schematic symbols are shown on the left, the data to be entered into the program is shown in the center, and the resulting patterns displayed on the screen are shown on the right.

To gain some insight into the factors HCAD considers during component synthesis let's look at the resistor-design section. After the designer enters resistor values, tolerances, and power dissipation, as shown in figure 4, the program assigns resistor shape, size, laser-trim cut type, and sheet resistivity, taking into account laser-trim adjustment range and tolerance, max-power dissipation, max-voltage gradients, and resistor/conductor termination effects. The resistors are designed to minimize area. The design variables are:

- Type – rectangular, tophat, and precision trim
- Laser cut – L-cut, single plunge, and double plunge
- Size – length and width

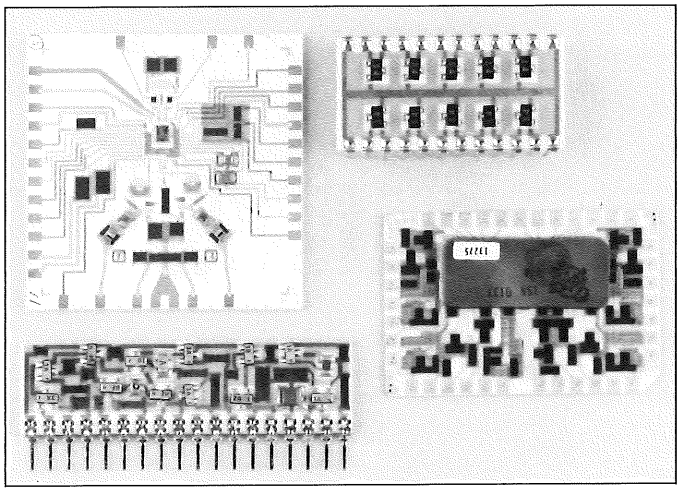


Figure 1.

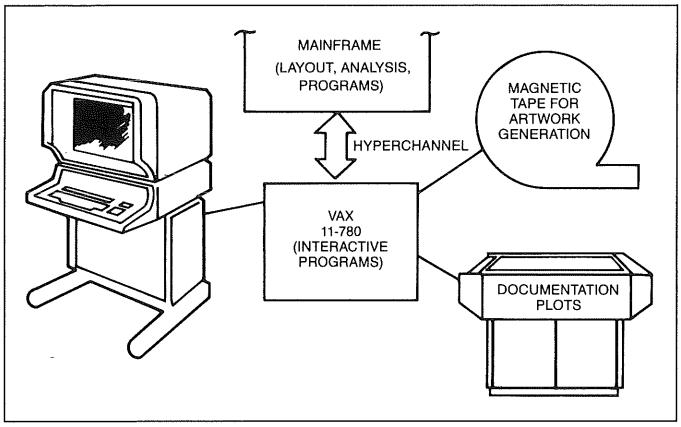


Figure 2. The Hybrid Circuit Development System (HCAD). The major output of HCAD is pattern data used to generate fabrication artwork and documentation plots for fabrication-process control.

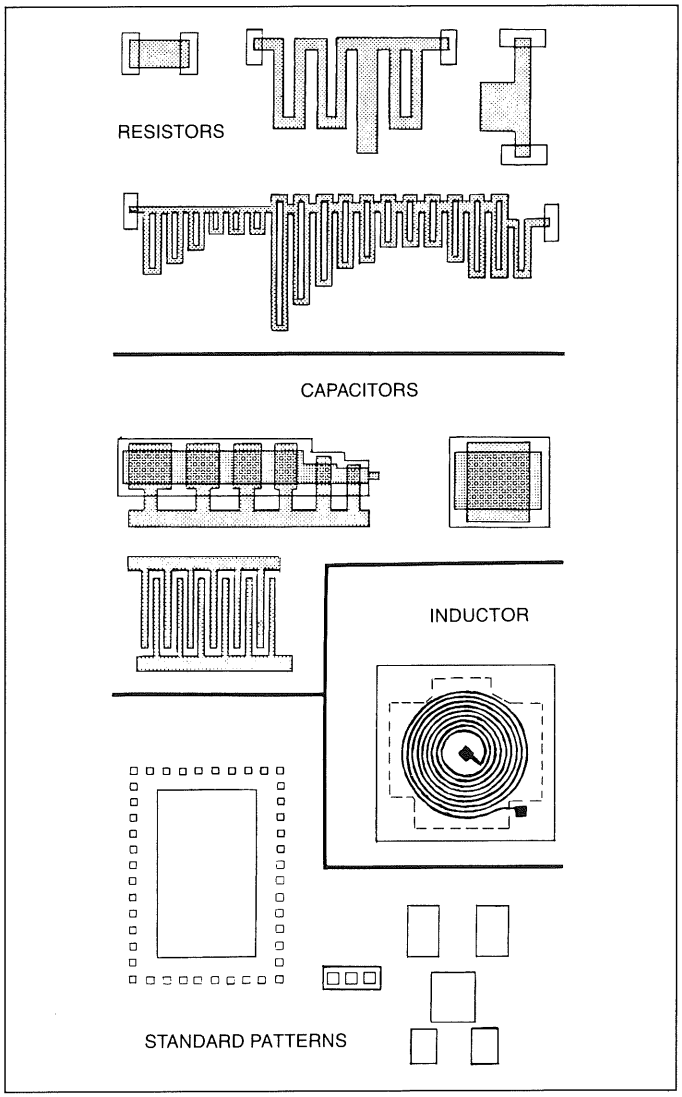


Figure 3. Examples of HCAD components and standard patterns such as IC-attachment pads and process alignment marks.

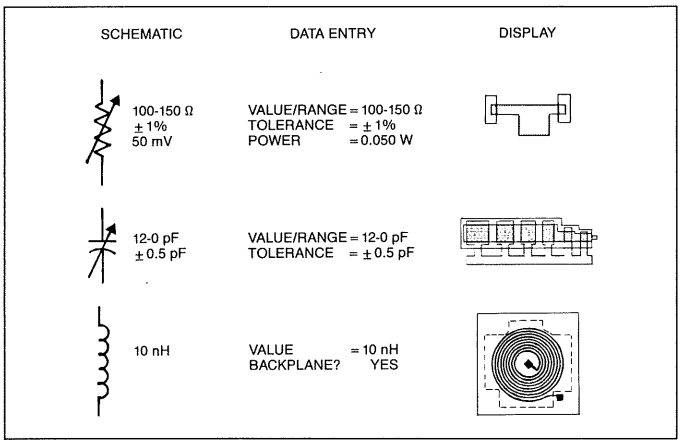


Figure 4. To create the physical component on the right the designer enters the data in the center and gets the components shape, on the right.

Interactive Layout

After synthesizing all passive components, the rest of the hybrid is laid out. Because HCAD does not automatically place components and route wiring, all placing and routing is done interactively with graphics-editor commands. HCAD provides the designer with many commands to aid in adding and manipulating artwork on the terminal screen. Some examples of the commands are translate, rotate, mirror image, duplicate, redisplay, magnify.

Figure 5 shows the interactive layout of a hybrid circuit on HCAD. In 5(A), HCAD has synthesized resistors and added pads for attaching soldered-on components. In 5(B), HCAD has added all the I/O pads and a dolly of the attached IC. Runs are interactively added, see 5(C), leading to the completed hybrid layout in 5(D).

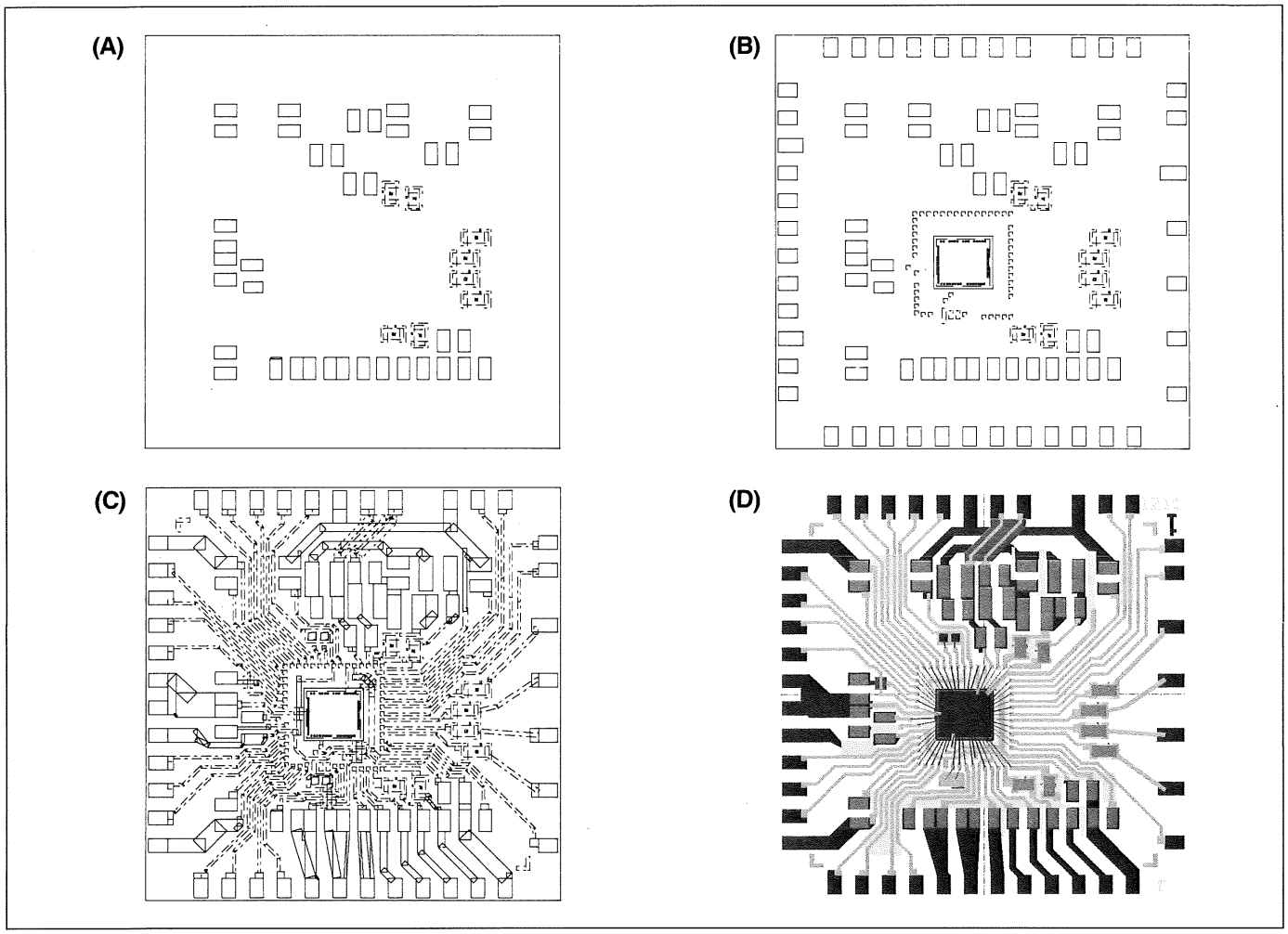


Figure 5. In the four interactive-layout steps of a hybrid circuit, HCAD synthesizes resistors and adds I/O pads and a dolly for the soldered-on components (A); HCAD adds I/O pads and a dolly for an attached IC (B); runs are added interactively (C); and the hybrid layout is completed (D).

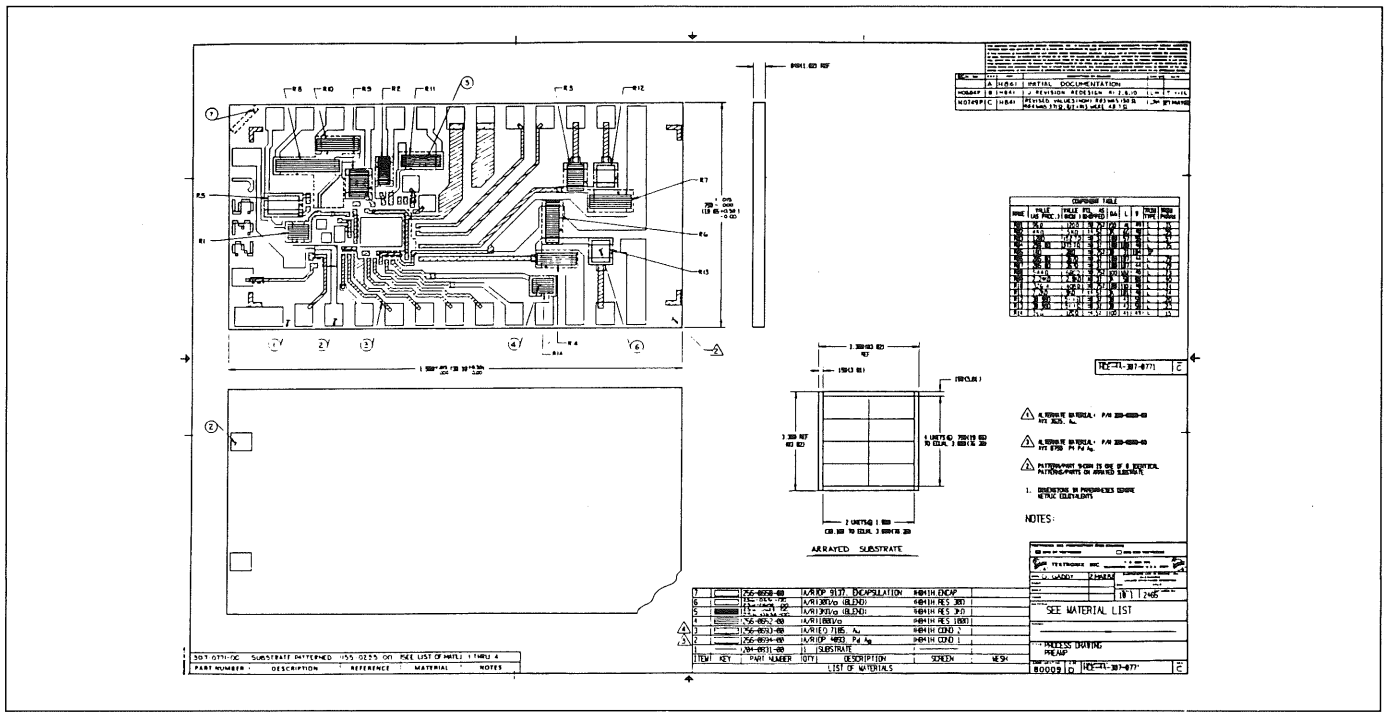


Figure 6. A documentation plot of a hybrid. HCAD provides an interactive editor that automatically creates drawings. Drawings may be manipulated. They also follow drafting standards.

Layout Analysis

Once the hybrid circuit has been laid out there are five programs within the HCAD program for analyzing the layout:

- 3D Steady State Thermal
- 3D Capacitance Coefficient Calculation
- 3D Inductance Coefficient Calculation
- 2D Potential Fields
- Stray Capacitance to Ground

These programs reduce the number of design iterations by revealing problems before prototypes are built. These analysis programs automatically use the geometric layout data generated earlier.

The user interface for the analysis programs is within the HCAD program on the VAX. The programs themselves, however, may be compute-intensive and therefore run on the Cyber mainframe via the Hyperchannel data link; the results are returned to the VAX upon run completion. This approach offers the most efficient use of computer resources, freeing the VAX for interactive work while using the mainframe for CPU-intensive jobs. All this is transparent to the HCAD user.

Documentation Plots

After layout analysis and completion, other HCAD programs allow the development of engineering-check and documentation plots. Figure 6 shows a plot generated in the HCAD system.

HCAD History

HCAD development began in 1977 and has continued in varying degrees every since. HCAD has been used throughout its development and as a result has been shaped by user comments and suggestions. Currently, the system contains about 125,000 lines, (1700 subroutines) in Fortran. Designers continue to increase their use of the package with present usage at around 600 hours of connect time per month. (See figure 7.)

The primary users of HCAD are shown in figure 8. ECO (Electronics Component Organization, formerly part of the Technology Group, is the heaviest user with the rest of the usage distributed among the groups.

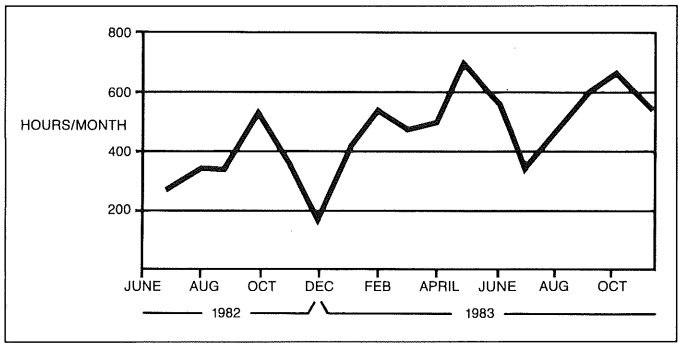


Figure 7. Hours of HCAD usage.

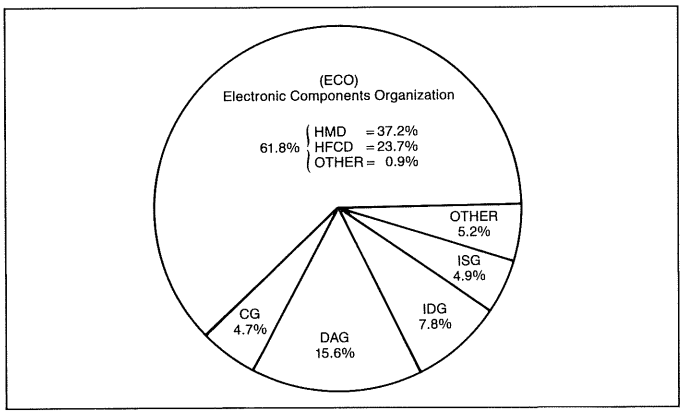


Figure 8. HCAD usage by organization.

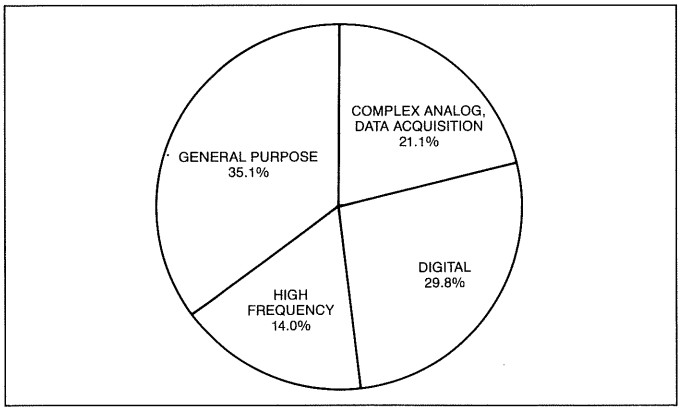


Figure 9. HCAD design usage by circuit type. Digital types are almost all A to D or D to A converters.

Figure 9 shows the types of circuits being designed on HCAD divided into four very arbitrary divisions. *General-purpose* hybrid circuits are simple hybrids such as resistor networks. *Complex analog* and *data acquisition* types are mostly probes. The *digital* division is a slight misnomer as our digital hybrids are mostly A/Ds and DACs, plus a few GaAs test patterns.

The Future

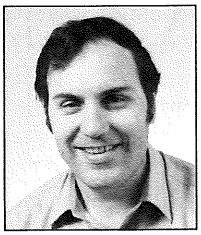
The general direction of future HCAD development would be to move from being mostly a physical implementation tool to a system covering the entire hybrid development cycle. I see Tek moving away from a mainframe-based system to a distributed workstation system. However, there are no concrete plans for this development at this time. The HCAD organization is currently working on a hybrid-circuit-data-management system and a design-rule checker. These will be completed in the first half of FY500.

For More Information

The HCAD programs reside on the HCAD VAX in building 59 and are accessible via the Develcon network. User numbers and information about HCAD development are available through Marv Abe, 627-4038, 59-345. For information about HCAD instruction contact Mal Gilbert, 627-4021, 59-431. □

A GRASS-ROOTS EFFORT

This issue would not have come to be without a grass-roots effort . . . and contagious enthusiasm. The source of contagion can be traced to the Engineering Activities Council. Last fall, the Council decided to do a seminar on EE CAD. Nick Fkiaras of the EAC chaired this effort. Mark DeSpain, Geoff Herrick, Preston Seu, and others helped plan, schedule, and worry the seminar into shape. In February and March, the seminar was presented three times.



Nick Fkiaras, EE CAD Seminar Chairman

The eight articles in this issue were derived from the material presented at the seminars plus updated and additional material. The authors, by describing CAD activities in Wilsonville, Walker Road, and Beaverton, give a pretty good view of the *breadth* of Tek's efforts in using CAD – and the considerable efforts that will make CAD even more useful in the future.

Tek's efforts in CAD also have *depth,* but we haven't gone into the details here. Earlier articles in *Technology Report* have done this for some elements; future articles will continue do so. In this issue, we have only touched on what is coming in Tek CAD products. This aspect of CAD will be covered in future issues as material is submitted or made available to the editor.

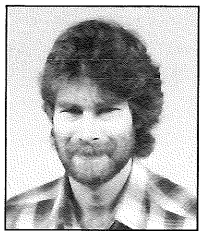
In addition to the authors' work, Al Zimmerman and Jack Hurt contributed suggestions and supplementary material. Al manages the Computer Science Center. Jack manages the CAX Center; he also authored one of the articles.

Some readers may wonder why Nick Fkiaras, a software engineer working on mechanical engineering projects, chaired an effort aimed at EEs? Well, in addition to being active in the EAC, Nick is a member of the CAX Center; he is immersed in CAD and has a deep interest in any discussion of CAD between engineers, EE or ME.

EAC members chose to do a seminar on EE CAD rather than mechanical CAD because it's the hotter topic now. Perhaps something similar to this effort can be initiated by the ME crowd. Although outnumbered by EE and software types, Tek has a lot of MEs. Certainly enough to do something on ME CAD. How about it MEs?

Art Andersen, Editor

CAX CENTER TOOLS FOR EEs



Jim Murphy is a project leader in the CAX Center. He joined Tek in 1978. Jim has worked on the 4081 graphics system, touch-based CAD system interfaces, and data integration. Jim graduated from the California State University, Chico in 1976 with a bachelors degree in computer science. He received a masters degree in computer science in 1978.

This article focuses on Tekwide electrical CAE and the CAX Center. The CAX Center is part of the Computer Science Center of the Technology Group. The CAX Center is responsible for helping to develop Tek-wide resources for three of the functions of CAE: mechanical design, electrical design, and design data management.

The CAX Center has three primary functions:

1. Providing or acquiring CAX tools which are usable Tekwide; getting these tools into the hands of the people who need them; and integrating these tools into a usable system.
2. Educating people about existing tools and getting feedback about what is lacking.
3. Providing any data transformations necessary to integrate the various tools so that together they act like a consistent system.

Our group has some knowledge about the tools used by designers and others for circuit simulation, design synthesis, and ECB design and fabrication. We are particularly involved in the integration of these tools so that data can flow easily from one tool to another as the design develops and becomes a manufacturable item. In addition to these primary functions, the CAX Center has been and will continue to be involved in business opportunities in EE CAE.

Data Entry

To function, all CAE tools need information from the user. Several years ago, we thought the Tek 4081 graphic system could be used as the engineer's interface to CAE tools. We were wrong. Needs differed too much. In light of divisionalization, we decided not to provide a CAE graphic editor for Tektronix. Today, there are many systems for schematic entry and graphics layout. Engineering groups are free to choose the system that best fits their needs. Our task has become one of enabling dissimilar systems to communicate with our tools. More on that later . . .

Simulation Tools

Circuit simulators come in two flavors, analog and digital. The CAX Center has both types. All our simulators use text files as input.

SPICE

SPICE is one of the original circuit simulators. The program originally came to Tek in the early '70s from the University of California, Berkeley. It is an analog simulator which uses mathematical models for the circuit elements to simulate circuit behavior. There are many models in its library: diodes, BJTs, JFETs, MOSFETs, resistors, capacitors, inductors, lossless transmission lines, and mutual inductors.

SPICE can analyze linear AC, nonlinear DC, and nonlinear transient responses. Circuits may have independent voltage and current sources and four types of dependent sources. The output can be displayed on a graphic terminal, or as a text file. The program runs on the Cyber 760 and 855.

Super-Compact

Super-Compact is a simulator for high-frequency and microwave analog circuits. It was purchased from COMPACT Software, Inc. in December, 1982. Originally it performed analyses in the frequency domain, but due to the nature of Tek's business, we have added time-domain analyses. Super-Compact characterizes circuit elements by impedance, inductance, capacitance, and transmission-line length.

The program can do many things here are a few: linear AC circuit analysis and optimization, frequency and time-domain analysis, matching-network synthesis, statistical failure prediction of elements, transmission-line analysis, power or voltage gains, and input-output reflective coefficient calculations. It can plot Smith charts or X-Y graphs on color or monochrome graphic terminals.

Super-Compact runs on the Cyber 760 and 855. (Some SPICE output files can be converted into Super-Compact format.)

TEKSIM

TEKSIM is a general simulation language that provides a common input format for multiple simulators. It has on-line help, extensive error checking, and explanatory error messages. The user can access operating sytem features, such as editors, without leaving the program. Circuit descriptions can be hierarchical (circuits within circuits), and the expanded circuit may be saved, once it is correct. Saving expanded circuits allows different analyses to be run without the overhead involved in expanding the hierarchical netlist each time. Subcircuit descriptions can have fixed or variable parameters; variable parameters allow the same subcircuit-configuration to act differently each time it is referenced.

For those familiar with UNIX, TEKSIM is a filter. It "filters" incoming general simulation descriptions, to produce a simulator-specific description. The program runs on the Cyber 760 and VAX/UNIX systems. TEKSIM makes possible distributed simulations where the circuit description is processed on a local machine and the circuit is analyzed on a mainframe. Distributed processing can be done now for the logic simulator TEKSIM/TLOGS. TEKSIM/SPICE is available now for experimental testing. Production release is scheduled during FY500.

TEKSIM/TLOGS

TEKSIM/TLOGS is a digital-logic simulator. It has 13 defined elements, including AND, OR, XOR, NAND, NOR, and ROM. Each node of an element exists in one of three states, high, low, or undefined. Signals in the circuit may have one of four strengths, so that the strongest simulated signal wins when more than one signal contends for a node at the same simulation time. Input patterns can be entered in the form of sequences or test vectors. Outputs can be timing diagrams or vector printouts.

The program can be used for fault simulation and random-fault sampling to grade the quality of a test pattern. Typical applications are bipolar gate array design and HMOS standard cell design. TEKSIM/TLOGS runs on the Cyber 760 and 855, VAX/UNIX systems, or a combination of the two.

Design Synthesis Tools

Design synthesis is a higher level of circuit design. The engineer uses equations and high-level languages rather than circuit descriptions to interact with these tools.

FUNMINI

FUNMINI is a boolean-function minimizer. It accepts one or more boolean equations as input, and outputs each equation in its simplest form. Its purpose is to reduce the number of gates needed to implement a given function. The program runs on the Cyber 760.

As an example, here are two equations, followed by the minimized equations.

```
2    ONE(OUT1) = (A * (B * C) + C) * ((Q + P) * H) + P      (BEFORE)
3    ONE(OUT2) = (A * C) + -(Q * -A)

NO ERRORS DETECTED

NAME                                          REFERENCE TYPE
--------------------------------------------  --------------
A                                             INPUT
B                                             INPUT
C                                             INPUT
Q                                             INPUT
P                                             INPUT
H                                             INPUT
OUT1                                          OUTPUT
OUT2                                          OUTPUT


OUT1 = P + C*Q*H      (AFTER)
OUT2 = -Q + A
```

SMS

SMS stands for State Machine Synthesis. This tool transforms a description of a state machine into a minimized set of equations for logic gates and flip-flops that implement the state machine.

The interesting part of state-machine design is deciding how the states relate to each other to accomplish an overall task. Deriving the boolean equations from a state-machine diagram is tedious, error-prone work, expecially when there are more than eight states. SMS relieves this tedium.

The program accepts a high-level description of the state machine; The language describes the inputs, outputs, state variables, and state transitions; SMS outputs the boolean equations for implementing the circuit. The output assumes a sum-of-products form for JK and D-type flip-flops.

Another program turns SMS output into the data to supply a PLA programmer (see Figure 1).

ECB Tools

The CAX Center developed PIRATE and CRAFTS to enable electrical engineers to design circuit boards.

PIRATE

PIRATE is an automatic circuit-board-layout program. It is a product of the CAX Center that can design two-layer and multi-layer circuit boards. It works best for digital boards; although it does a good job with mixed circuits.

The user supplies PIRATE with a text file describing the circuit interconnections and parameters such as the number of layers, board shape, and areas where runs should not be placed. Components need to be described in the file also; much of this detail is already stored in a component library.

The program automatically gathers all gates into packages. The placement of the packages on the board may be done automatically or interactively. Often the interactive mode supplies the best placement, since people are more adept than the program at knowing what looks right. Once the placement is finished, PIRATE uses its routing algorithms to connect the component pins with traces.

The routed board may be viewed on a graphic terminal, or on penplots generated for detailed inspection off-line. PIRATE runs on the Cyber 760.

CRAFTS

CRAFTS is the Computer Resident Automatic Film and Tooling Stream. From an EDF (described later) text-file input, it can produce penplots of ECBs and schematics. For ECB plotting and drilling, it can produce line art, mylar photoplots, and numerical control files. These are optimized for best use of the machines which do the actual plotting and drilling.

ECB fabrication engineers use CRAFTS for automatic or interactive flat layout of multiple board images on the large ECB flats used in production. The system runs on the Cyber 760.

Data Integration Tools

Component Library

The component library contains data about components used in custom electrical designs. A component is a part with either a Tektronix part number or some other designator, such as a manufacturer's part number.

The descriptions for each component may contain its logical characteristics, its physical characteristics, and its simulation characteristics. Logical descriptions tell how many elements, such as gates, are in a component, how the pins behave, and

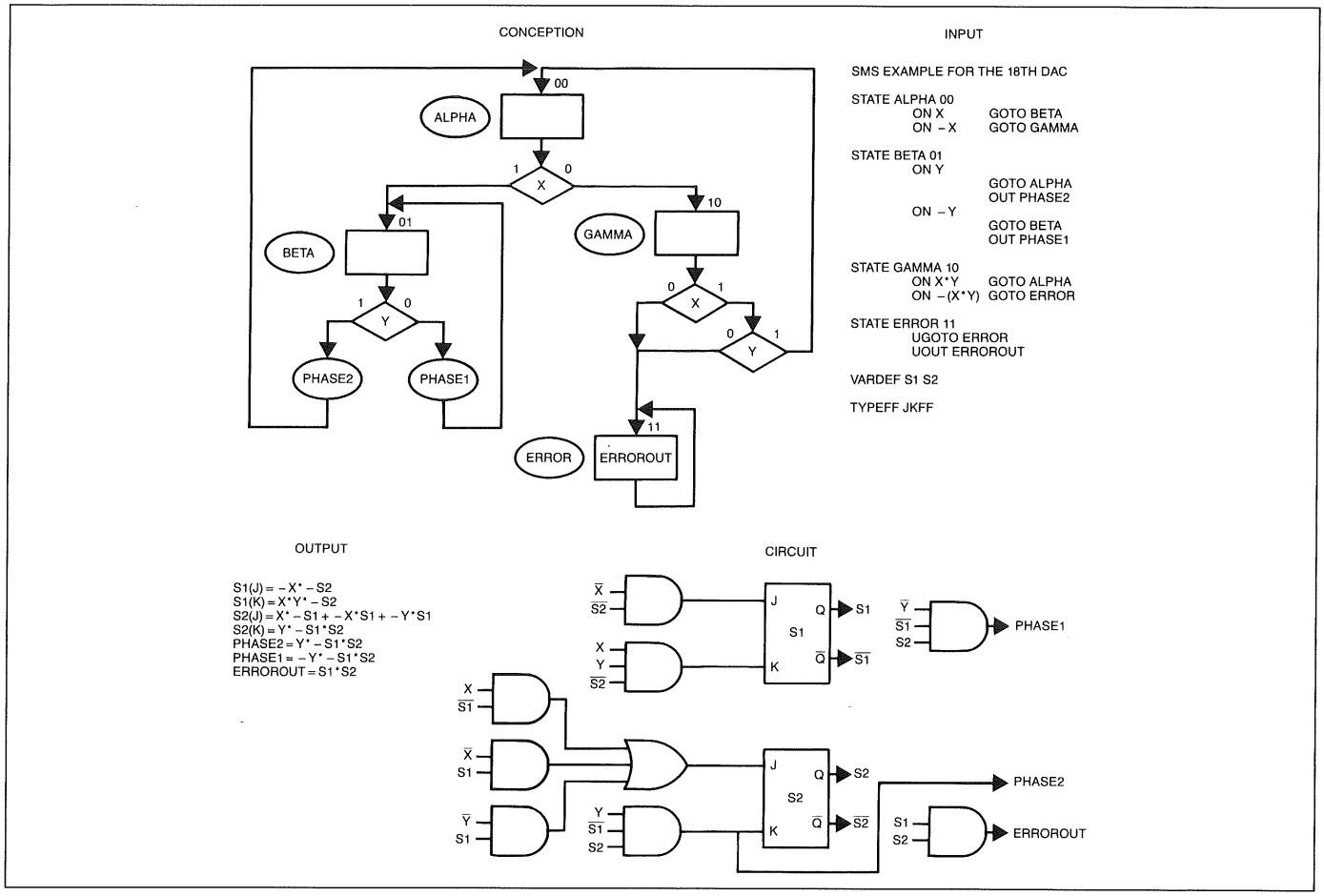


Figure 1. One function of the State Machine Synthesizer (SMS) is reduce the number of gates required to implement equations. In addition to minimization, SMS can help a designer go from algorithm to input to drive a PLA programmer.

how it should be treated by an automatic place and route program such as PIRATE. The physical description details how to use the part on a circuit board, including the drill size to use for the type of pins, solder mask patterns, and metalized shapes. The simulation description is text used by the SPICE simulator. It is not complete at this time.

Besides providing data for simulation and ECB design, the component library system allows people to view data as text and graphics. The component library is on the Cyber 760.

EDF

The EDF stands for Electrical Design Format. EDF is a Tektronix-defined format for describing electrical designs at interconnect, schematic, and circuit-board levels. By design, the format can be used in the future for other technologies: integrated circuits and hybrids.

The format was conceived to solve the problem of sharing design information between various CAE systems and tools. Its primary purpose is for data transfer, but it can be used for archiving designs.

The CAX Center has developed or assisted in developing translators between the EDF and CAE systems and tools. To date, one-way and two-way transfers can occur between EDF files, on one hand, and Applicon CAD systems, 4081 graphic systems, NOMAD systems, Daisy CAE systems, the component library, PIRATE, CRAFTS, and TEKSIM/TLOGS, on the other hand. Ultimately, most CAE systems will be able to share designs; tool developers will not have to define their own input/output formats.

Future Plans and Directions

More integration

The biggest task of the CAX Center in the future will be integrating Tek tools into a coherent system. Tek has many tools, but since most of them operate independently, each tool has its own special format for input. Connecting these diverse tools together requires software "glue."

A way around this problem is standards, such EDF. Other standards are on the horizon: EDIF (from LDS and others) is an alternate to EDF's way of representing circuit information. IGES is a national effort to extend a format designed for mechanical information to encompass electrical data. To help provide the best solution for Tek, the CAX Center must keep up to date with these and other outside efforts, as well as our own.

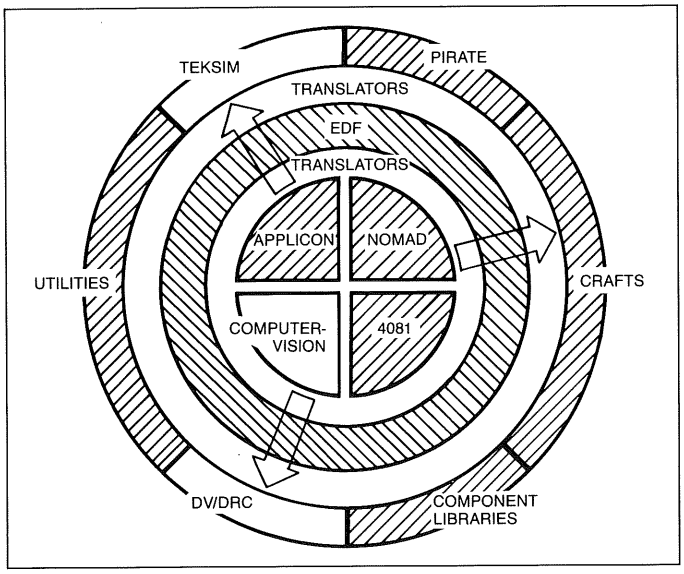


Figure 2. With EDF, a consistent data format will enable designers to use tools and systems smoothly from design conception to board manufacturing. Eventually, this smoothness will be extended to IC and hybrid implementation. With EDF files and CAX-Center or division-provided translators, diverse CAE systems will be able to share design data throughout a design cycle.

One step beyond standard formats would be an engineering-information resource that would assist design engineers in doing their best. Such a system might include on-line information about components, so that engineers can get the latest information on parts to help in component selection.

Simulation

The trend is for simulation to become distributed. Small circuits could run on a local VAX or workstation. Larger circuits might be analyzed on a mainframe. Or a job could be shared by both systems.

TEKSIM/TLOGS already runs entirely on VAXes and mainframes. CYTLOGS is a shell script for UNIX machines to process the input (TEKSIM) on the VAX and run the simulation on the Cyber. TEKSIM/TLOGS will also be linked to the automatic test pattern generation program being developed by the digital test-tools group.

TEKSIM/SPICE, or SPICE with the TEKSIM circuit description language, is being tested and will be released in mid FY500.

CFMS

The CAX File Management System (CFMS) is being built to allow several people to work on the same design, although not simultaneously. CFMS will run on VAX/VMS machines initially. CAE files may reside on any computer in the system and the Tek Engineering Network will be used to transport the files between machines. There is nothing special about files to CFMS; conceivably, people could use the system to share and manage any kind of file. CFMS will be available the second quarter of FY500.

PIRATE

The ECB layout program will be enhanced for better layout of ECL circuits. It will be transported to a virtual-memory mainframe to give it fineline capability in FY500. We will provide support for EDF files and surface-mounted components sometime in FY500.

EDF Applications

Programs to do design-rule checking of circuit boards, ECB-schematic validation, and buildability analysis are planned for the near future.

CRAFTS

Enhancements planned for FY500 release are the support of surface-mounted devices, flexible circuit boards, and tab routing. Composing manufacturing flats from different circuit boards will be worked on in the future.

Component Library

The biggest plans for the Component Library result in storing more data. We are investigating logic simulation models from an outside supplier; this will enable the EDF-TEKSIM/TLOGS link to have a much wider usage. Schematic symbol capability and data are currently under way. Mapping between the symbols and simulation and autoplace and route nodes are also planned.

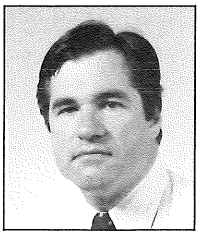
For More Information

Feel free to talk to the following people for further information on any of the programs or topics presented in this article.

Area	Name	Phone
Simulation	Jim Kimball	627-1488
SMS	Phil White	627-1816
PIRATE	Janice Rostal	627-2603
CRAFTS	John Bradshaw	627-2619
Component Library	Phil White	627-1816
EDF	Jim Murphy	627-2266

□

IC TOOLS



Paul Smith is the manager of both IC CAD development and outside marketing in IC Engineering, part of ICO. He joined Tek in 1979 from RCA's Solid State group. Paul has a BA in engineering and applied physics from Harvard, an MSEE from Northeastern University, and an MBA from Rutgers University.

This article gives an overview of IC tools at Tek. There are many different tools around the company in different groups. These tools fit into the overall IC-development process from behavior simulation to the engineering of the IC down to manufacturing the final design.

I would like to describe how IC tools are used at Tek in physical implementation and verification, and then in photomasking and manufacturing. Other articles in this issue discuss simulation tools such as SPICE and TLOGS – we in IC Engineering use both tools heavily.

IC CAD is a major effort at Tek. Of the six organizations that use CAD, five are users of IC CAD tools. And one group, LDS, will become a supplier of IC CAD tools:

- IC Engineering – Part of ICO
- Integrated Systems Lab – Part of the Computer Research Lab
- Advanced Development Group – Part of the Instrument Group
- IDG IC/CAD Group
- Silicon Tools Group – Part of Logic Design Systems

Unfortunately, there is no such thing as a generic IC design tool. How IC CAD tools work and how the are used depends heavily on the IC design methodology and on how the IC is to be physical implementated. Tools for full custom HMOS differ, for example, from tools for CMOS gate arrays.

Let's review some terminology.

The methods of IC design range downward in flexibility from full custom through structured custom to semicustom. *Full custom* is complete design from the device level up. Nothing is predetermined, nothing is *constrained.* Full custom is the typical analog-circuit design procedure.

Structured custom, is like full custom, except that some things are constrained, device layout for example. Examples of structured custom are the Meade-Conway integrated-systems process in the digital world (see figure 1) and gridded layout, which Tek uses for analog circuits. These constraints cut down the circuit-development time and development lost.

Even more constrained are *semicustom circuits,* which include the familiar gate array and standard cell in digital ICs, and the quick chip in the analog world. Devices in semicustom circuits are pre-designed and fixed at either device level or cell level and the designer connects them to achieve the final circuit.

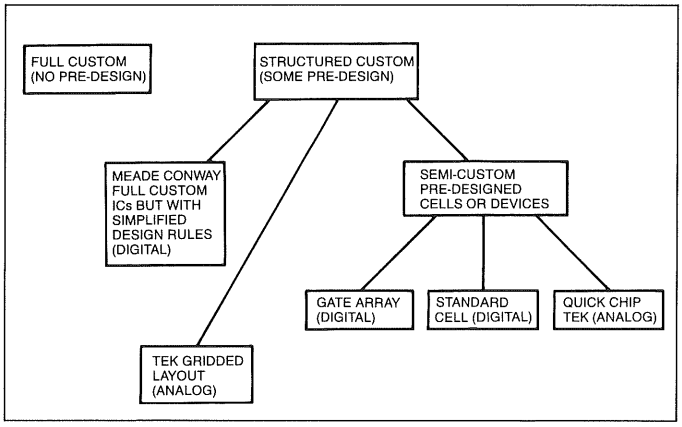


Figure 1. IC CAD systems are specialized to support a design method. The Meade-Conway integrated-design method encourages flexibility by allowing designers freedom of cell design and system innovation. The more constrained semicustom approach provides out-of-the-library components to speed up design and reduce costs.

Integrated Systems Group (ISG)

The Integrated System Group, managed by Kit Bradley, is among the three most active IC CAD users. ISG has two objectives: One is to develop applications for VLSI technology. The second is to develop innovative IC design methodologies.

ISG primarily uses the Meade-Conway method to do integrated systems with a structured-custom approach, but they also use standard cell and do some work in full custom. They use a modified Meade-Conway approach because it enables them to develop innovative system architecture, as well as more efficently design an IC to do a function.

Most of ISG's tools came out of university research. For switching and timing simulators, they use NET/RNL for transistor gate-level access. They also use SPICE and occasionally TLOGS. They do some automated synthesis and analysis of functions. Because their tools fit ISG design methodologies, their tools work quite well for the Meade-Conway method but not so well for random logic designs.

Figure 2 shows the interelations of the ISG's tools by function. The tools follow standard computer-aided design procedures from simulation to layout. When they have developed the physical layout, PRIDE extracts the circuit parameters or other information. These are fed back to the simulators to calculate the effects of parasitics. Once the simulation is satisfactory, the tools check electrical rules and verify the layout against the simulation input. The tools also have programs to figure power consumption and verify the design rules. All these checks increase the probability of manufacturing a working chip that has no design problems.

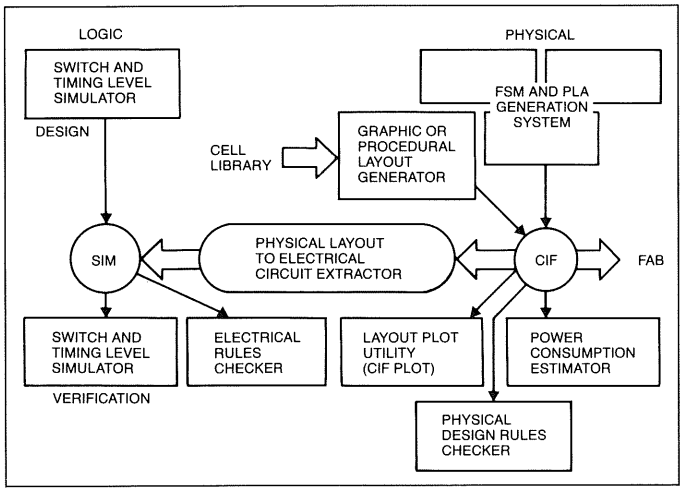


Figure 2. The functional interrelation of IC tools used by the Integrated Systems Group, ISG is part of the Computer Research Lab, Advanced Research.

The Integrated Systems group and the Oregon Graduate Center are heavily involved in training engineers to do IC design. Kit Bradley has been one of the mainstays of the OGC program. OGC teaches two courses each year. In the first one, the student learns how to design ICs. Each student makes a chip as part of the course. The chip is then fabricated. In the second course, the students find out whether their chip really works or not. (If you are interested in this training, call Kit Bradley, 627-1497.)

The Advanced Development Group (ADG)

The Advanced Development Group is part of the Instrument Group. ADG's charter is to develop technologies for future instruments, three to five years out. Right now, ADG's major focus is on digital signal processing, both hardware and software, not just ICs. In addition, ADG is doing some work involving video-image processing as part of an automated calibration and quality control system for scopes. Chuck Saxe (627-3089) is ADG's manager.

I find ADG's systems approach very interesting. They are taking a hard look at designing overall systems top-down, and then fitting component designs into that system. Not from the bottom up where you design your components first and then see if they fit into a system or not. ADG does most system simulation and architecture development first and then they partition the design down to a chip specification. They use a highly custom approach, more custom then Meade-Conway, in most cases full custom. They expect the design cycle to be pretty long, two to three years for a chip or a series of chips because they are designing a complex system.

ADG is also assisting other Instrument Group people with TTL-replacement ICs, semi-custom gate arrays, and standard cells based on random logic. In contrast to much of their work in ICs, these gate arrays and standard cells have much shorter development times but much less functional flexibility.

In system simulation, ADG has been using programming languages – APL and FORTRAN – to describe the workings of a system, or board, or whatever. They generate a set of registers or local units and the simulated item's input/output configurations. These are used in determining what the system will do functionally. This simulation is compared against an algorithmic description of the functioning of the system. The basic idea is to describe the system as best you can in terms of what it is supposed to do and then start partitioning it first into blocks and then breaking it down further into components.

After they describe the system, they partition it into the physical blocks. They then take the system simulation and generate I/O data for chips. They use the I/O data in tools such as TLOGS to simulate the chip, then do a physical layout of the chip. Next, using the simulation of the chip as an input into the simulation of the system, they make sure the entire system and each chip will do the desired functions.

ADG believes that a chip-design process should fit into the system design. As one way to do this fitting, they are trying to minimize the all-too-common translation problems, such as hand-off errors between groups of people working on different components of the same system. We in the IC fabrication area see the results of translation problems frequently. For example, a board (system) having seven custom chips, each designed by a different person. One designer had a 16-bit counter output talking to another designer's 17-bit input chip. This kind of thing could have been easily prevented by some system simulation efforts. Resolving the problem caused another design cycle which extended both completion time and cost.

Integrated Circuits Engineering (ICE)

In ICE our objective is not to develop CAD tools. We are supposed to be tool users. Our job is to provide ICs that build technology and cost barriers in the instrument market. We get into providing CAD tools because we need the tools to do the designs. I would much rather buy them off-the-shelf.

We support approaches all the way from full custom, which we generally do on a CALMA, to gate arrays. CALMA is a graphics workstation; our CALMAs are specialized for IC design. The custom designs run from fully unstructured designs (full custom) to designs using structured or gridded layouts (semicustom). These use pre-designed device primitives to reduce layout time and decrease the possibility for error.

At the more structured end of our work are semicustom circuits. These include gate arrays, where the designer starts with pre-designed logic blocks and customizes only the interconnecting metal layers. A standard cell is similar to a gate array, except all layers are customized to reduce size and – sometimes – speed-up IC performance over that of a gate array. The Quick Chip is basically an analog version of a gate array where the resistors and transistors and other devices are pre-designed on a fixed layout. You customize only the metal layer to wire up a desired function. For a listing of the semicustom circuits we support, see the table.

SEMI-CUSTOM DESIGN METHODS

Design Method	Process	Gate	Speed	Available for Design
Gate Array	TECL1 (SHF-3 DLM)	600	250-500 MHz ~300 ps/gate	Now
Gate Array	TECL2 (LBT DLM)	2000-3500	200-400 MHz ~400 ps/gate	6/85**
Quick Chip	SHF-3	18 Xstrs	6.5 GHz fT	Now
Quick Chip II	SHF-3	250 Xstrs	6.5 GHz fT	Now

**Design method generally available

We work with a range of technologies and both digital and analog devices, on a variety of bipolar processes including a mix of digital and analog circuits on one chip.

Tools, Some Good, Some Not So Good

The tools we use ICE reflect the growth of the CAD and the IC industry over the other years. The tools range from reasonable to poor, or even bad. This is the way they have evolved over time. Let's start with what should be the best and work down to the poorest.

Tools: LBT gate array

We are working on our first few LBT designs now, using two purchased tools. Both are reasonably automatic. For placing and routing, we use VR Systems' Merlin G; it looks like Merlin G will do automatic placing and routing very nicely. The second tool is a Phoenix Data Systems' MASKAP extraction and verification system.

Tools: TECL1 gate array

The TECL1 gate array was designed before any CAD tools were available in Tek. If you had designed TECL not to work with a CAD system you could not have done a much better job! We gave up trying to automatically place and route, deciding it was much easier to do it manually. Since the TECL1 gate array is very small, you can do one manually in about a day. But no one has done one without making some mistakes – even with a system of helps and checks. We have written a lot of error checking programs.

The TECL1 gate array is used for both analog and digital applications in high-speed, high-performance circuits. Since the applications tend to push the limits of the ECL process, they are difficult for a program to opimize. TECL1 uses the TCAD system for extraction and verification, and also to generate custom cells. Custom cells are used frequently in the TECL1 designs.

Some IC CAD Realities

As an illustration of the mess CAD can get you, consider that Tek people have written more than 33 programs to tie all these IC tools together – and every program uses a different data base format. If you want translate and go from one program to the other, you have to translate the data base. We use SPICE and TLOGS heavily for simulation. Their input formats do not fit with the formats of the place and routers. And the format of the place and routers do not fit the format for the CALMA.

We too have the problem of different products on different machines. We are using and supporting programs on Cyber, UNIX, VMS, CALMA, and soon an IBM. And all of us would dearly love to get out of having to know six or seven editors and six or seven operating systems.

In custom analog, there are good simulation tools in SPICE, with libraries and circuit descriptions to go with the tools. Layout, however, is a manual job. We, in ICE, do custom-analog layout on a CALMA, which is very good for what it does, but it has no automatic place and route, no automatic device generation, no verification. Those functions are difficult and complicated to do automatically because of the variety of devices that would have to be recognized. We are starting to put a system together to do this type of CAD for analog circuits.

We certainly would like to have schematic capture. All engineers work with schematics; they don't like working with descriptions of circuits. From the schematic-capture program you should be able to derive the circuit description and feed the simulation program.

Once you get your SPICE-work (simulation) done you need to make a physical implementation. It would be nice to have automatic place and route although in analog that's going to be a long time coming.

And then, after place and route, you need to extract parasitics, such as resistances and capacitances, from your metal runs. The resistances, for example, can be fed back into the either the circuit description or the schematic so that you can resimulate with better information and know how the circuit is going to work once its built. The simulated performance of many circuits changes quite a bit when layout parasitics such as run capacitances are factored in by resimulation. This is true for digital as well as analog.

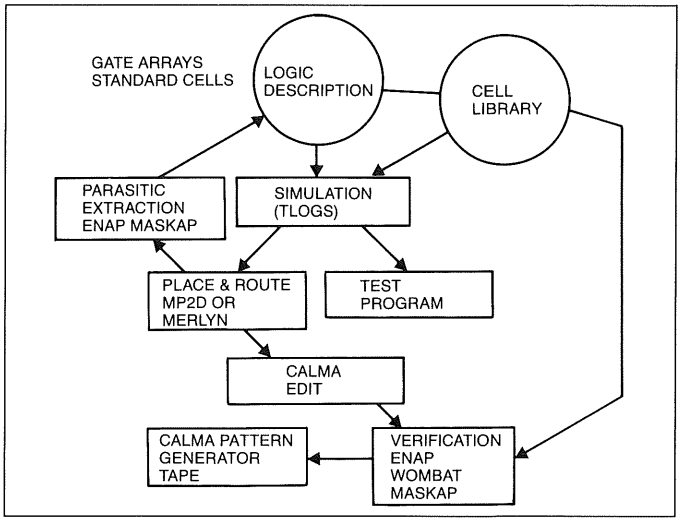


Figure 3. IC design CAD today.

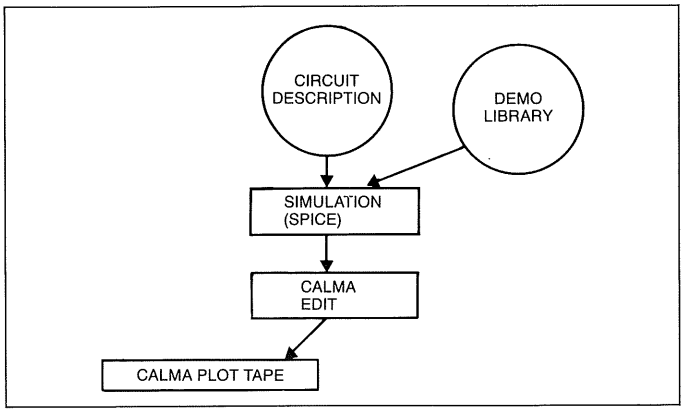


Figure 4. For custom analog, IC designers find a good simulation tool in SPICE. It is well supported with libraries and circuit descriptions.

With the CALMA we can do all of the graphical manipulations to lay out a chip. But since it is so easy to put a metal line in the wrong place, or make some other error, layout match to circuit description should be verified. Even though some circuits will be correct, it is desirable to be able to verify correctness before construction. It's expensive to make IC hardware, very time consuming. You will want to make sure that everything you design is free of errors before it gets into processed silicon.

To do gate arrays and standard cells there are pieces – I emphasize "pieces" – of all of the tools in figure 5 available. Some work better than others. Some hardly work at all, but all the functions are basically there.

For the logic descriptions, schematic capture is not there yet although it's very close. TLOGS is a good timing simulator. Place and routing performance depends on what kind of circuit you're doing and what the technology is. In some cases it's good; in others its not. We can do some things with parasitic extraction, not as much as we would like to do. Test-program generation is certainly in its infancy at this point. Now it's a case of developing the test by hand. Verification is another example of some things done very well while others are not.

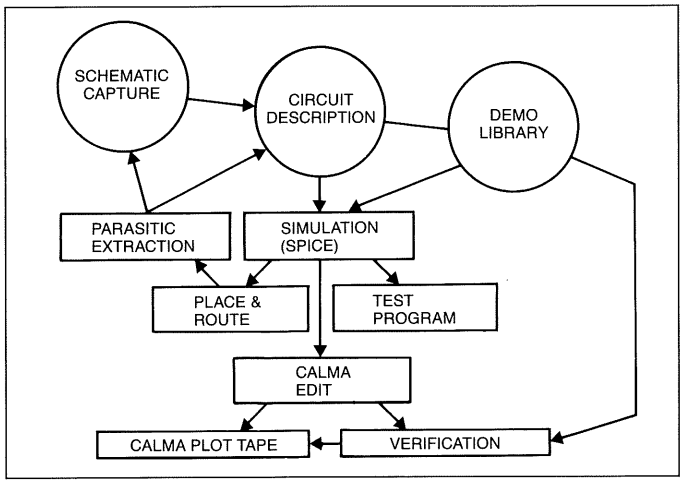


Figure 5. IC design CAD in the future. Just pieces are in place now to do gate arrays and standard cells. Schematic capture is close.

What I'd Like to See

Figure 6 shows what I'd like to see in IC CAD: The "silicon compiler" if I may use that overworked, underdescribed concept. We need to be able to handle arbitrary blocks in gate arrays and standard cells. We need a lot more interaction between the different CAD tools so that an engineer can start with a schematic and have it automatically do a simulation and show quickly what the results are. If a resistor or something is changed, the designer wants to know immediately how it affects a voltage and not have to wait a half hour while the program chugs away. And since circuit place and routing affects circuit performance, you need to do that interactively. CAD programs should calculate the effect of the change on the parasitics, feed it back into your schematic so the simulations are real.

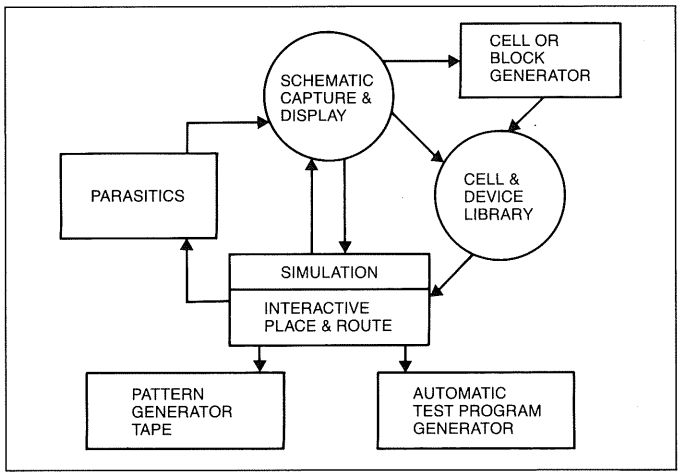


Figure 6. The IC CAD system of the distant future, something that is years away. One reason all this will not arrive is that even a big-machine CPU takes hours to do a place and route something like a 3500-gate CMOS gate array. This is not interactive by any measure.

Figure 6 is a wild dream. We are years away from doing all that interactively. I can illustrate one of the problems. We have a 3500-gate CMOS gate array. It takes a VAX VMS about four to seven CPU hours to do a place and route. That's not quite interactive. Even on a bigger machine like the IBM you may speed things up by a factor of four or five. That's still nowhere near interactive. So there's a long ways to go.

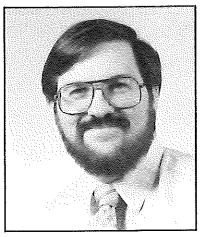
For More Information

For more information, call Paul Smith 627-5457.

ICO gives courses for designing semicustom designs based around the technologies we support: TECL1 gate arrays and Quick Chip. Call Paul Smith (627-4116) or Dick Lynch (627-4116).

Area	Name	Phone
All or training	Paul Smith	627-5457
TECL1	Dave Perkins	627-1378
TECL2	Binoy Rosario	627-4072
Quick Chip	Wink Gross	627-4085

CAD TOOLS FOR ELECTRICAL ENGINEERING IN DAG AND ECS



David Bennett is an engineering project manager in the Logic Analyzer Division, part of the Design Automation Group (DAG). David joined Tek full time in 1977 after earlier summer student employment. Earlier in his career he did digital design at Lockheed and Litton. He is working on his MSCS and holds a BSEE from the Oregon State University.

This article describes some mostly good experiences with the well-known Daisy workstation and with Dash-1 PC-based and internally developed EE CAD software.

What Is a Workstation?

The electrical circuit design process can be viewed as a wheel with design definition as the hub, the coordination point for the entire design process. Although definition is the hub of the design, definition depends on data about the technologies involved in the surrounding wheel. Without this knowledge, the circuit definition process cannot be successful.

The hub includes schematics, netlist, circuit documentation, simulation – all things that heavily involve the electrical designer. The outer part of the wheel generally involves the designer less; this is where physical implementation and productization of the circuit and/or system takes place. However, in some of the outer areas, designers are involved more and more. In hybrid and integrated circuits, for example, designers are doing layouts if not the actual fabrication.

But in many cases the designer is still excluded from the physical implementation and productization; this is particularly true in ECB design. Tektronix engineers seem to look at ECB design as a black art. They hand off a schematic and a few weeks later a perfect circuit board comes back, perfect the first time.

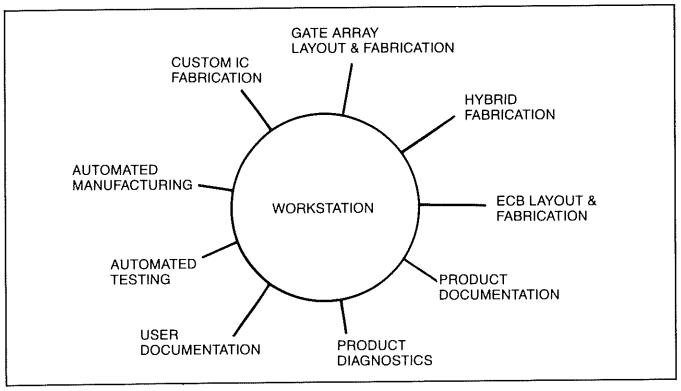


Figure 1. The workstation functions at the "hub" of the design activity where circuit definition takes place.

Most EE workstations in use today are designed to primarily support hub activities, the design definition. And that hub support is scattered. Some aspects of the circuit definition are supported by EE/CAD software on various machines, other design aspects such as system design and bill of material generation are not automated at all. In DAG some technologies, such as TECL gate arrays supported by TLOGS, have heavy simulation support. Others, such as large analog circuits, have none at all. All workstations attempt consistent automated support for all major aspects of the design definition process. But where the designer is involved in the physical design, not all workstations support this additional load.

Recent workstations attempt to support the designer in all electrical-design tasks rather than just concentrating on circuit definition at the hub.

The Daisy Workstation

In the Logic Analyzer Division, we use one stand-alone Daisy system and a second Daisy system of two terminals supported by a computation-card cage.

The Daisy primarily functions within the hub of the wheel. It provides tools for entering and verifying schematics, automated netlist generation, circuit description, and simulation. By automating these processes, Daisy provides a consistent package of information to the circuit-board-design process.

With Daisy, all schematics use an identical general format. This provides a common method of data presentation and a consistent level of data. For example, all schematics use the same name for a signal throughout no matter how often that signal appears on other pages. All components are sufficiently identified to determine physical footprint and connection requirements.

Netlists always have identical formats, with consistent component and signal identification. When automated tools for circuit-board place and route are used, the netlist can be formatted as required for the place and route software with little manual entry required of the circuit board designer.

The Daisy workstation uses a consistent framework for updating schematics with physical annotation such as U numbers and pin information. The automated back-annotation capability of returning the physical information to the schematic by the workstation provides schematics with both logical and physical information in all phases of development. Similarly consistent data entry and return is provided for gate arrays and standard-cell designs. Logic Analyzer engineers have used this data-transfer process successfully for several circuit boards and one vendor gate array.

Daisy also provides a consistent data source for product documentation and testing. These data range from test vectors and their expected response to reproducible schematics.

While we use Daisy for most design-definition processes, we do not use it to address several hub interfaces to the rest of the wheel. We use the Applicon as an interface to manufacturing. Today, our Applicon system provides physical circuit descriptions to our automatic insertion machines and automated board testers. In the future, we expect to use Applicon to address some elements of semiautomatic assembly.

Our application of Daisy does not handle custom-IC circuit description and physical-layout restraints; it does not handle hybrid design either. In fact, we deal with no physical representation on our Daisy workstation stations at all. Gate-array definition is limited to a standard macro cell, which is implemented by a vendor off-site. (Daisy Systems, Inc. offers workstations with physical design capability, but we do not use them.)

The Daisy, Some Opinions Based On Experience

Daisy performance is relatively mature. I stress relatively; it is kind of early adolescent. In contrast, most workstations are barely emerging from their infancy. One indication of this maturity is that the Daisy people often anticipate our needs before we express them. As we got more involved in complex designs and into more designer participation through the full paths of circuit-board design, gate-array design, and so forth, Daisy people kept coming up with the additional tools we needed. For example, they provided automated software for generating component graphics and pin definitions; previously we had to enter these manually.

The Daisy organization has been making their tools better. We have seen notable improvements in schematic entry and interactive simulation over the last year. These have been true improvements, not just fixes.

On the negative side, as a purchased device – whether purchased internally or externally – workstations are rapidly obsoleted. In the year that we have had Daisy, they have increased later-model CPU speed 20%; they have introduced a hardware-enhanced simulator that makes our Daisy's simulation capability obsolete. They introduced a next-generation processing card, dramatically obsoleting what we have in our Daisy. So there will be hardware and/or software obsolescence. To combat this, Daisy offers a program of software and hardware updates, which costs several thousand dollars a year.

Daisy's Structure

A Daisy has three major blocks (see figure 2). The first is used to enter schematics and parametric descriptions of circuits. This block is the source of basic data for all other tools, whether inside or outside the Daisy itself. The second block, the simulation "tower" (on the right), is a set of tools that are applied to simulate a digital circuit. Within the third block are the data-base access tools used to communicate schematic information to the outside world.

The schematic-entry portion of Daisy consists first of all of graphics. Like most workstations, Daisy does not furnish all of the component descriptions we need. Some because they are custom parts, others because they are new – or the parts were little used

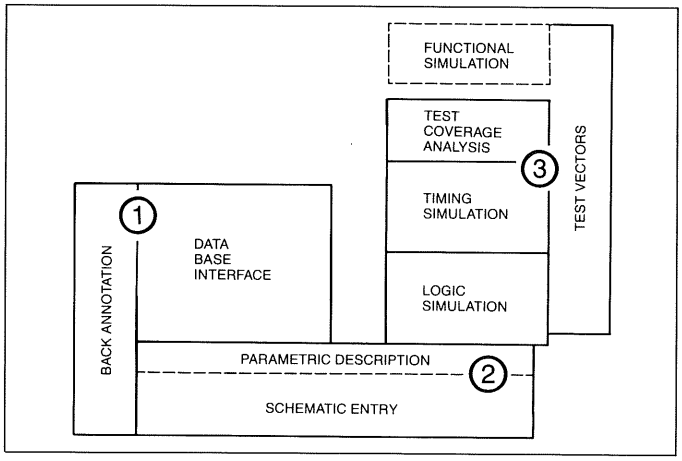


Figure 2. The Daisy workstation has three major function blocks: (1) schematic entry, (2) simulation, and (3) data base access. Functional stimulation is not yet available.

in industry and not worth Daisy's time to describe. So in the schematic entry block we use both Daisy-furnished graphics and Tektronix-user-generated graphics as well as macrocells from gate-array vendors.

Daisy's schematic entry also serves as the host for all parameters needed to describe a component. Some of these parameters are component names and types, physical location and pin numbers, strength and delay of signals during simulation.

Schematics on Daisy are structured as a hierarchy which is physically implemented as a tree of directories (see figure 3). Generally, the top level is a major block diagram. The next level of the hierarchy consists of major functional blocks; these blocks are followed in the hierarchy by low-level schematics. Circuit complexities determine how many levels of block diagrams versus schematics are needed. Now once the schematic is entered, it is run through a set of formatting packages which flatten it to produce a single circuit description (netlist). Lower levels in the hierarchy are portrayed by a named, rectangular block on the schematic. This block is also a means to access the lower levels while in the drawing editor.

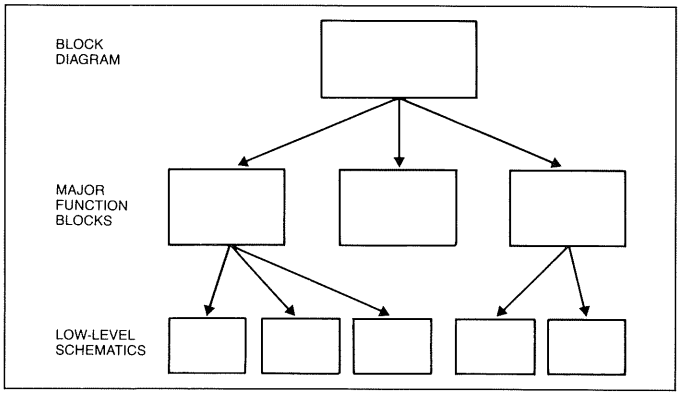


Figure 3. In a Daisy workstation, schematics are implemented as a tree of directories. After schematic entry, formatting packages flatten the entry to produce a netlist.

The simulation consists of levels of simulation, starting with logic simulation, primarily ones and zeros. Next, going up one level, the timing simulation allows close examination of circuit timing, including race conditions, setup and hold, minimum pulse width, and similar analyses. The test-coverage analysis system determines if a node has moved during a simulation run, that is first-level coverage analysis. Functional simulation is a form of behavioral simulation using an algorithmic description of logic blocks rather than a set of nands and nors. (Daisy has not delivered functional stimulation.)

In simulation, we have a problem of component data that resembles what we encounter in component graphics: Daisy furnishes the logical descriptions for their components, but we have to come up with logical definitions and timing for whatever components we have supplied. We require gate-array vendors to supply component information for their macrocells.

Test vectors are logical and timing descriptions of the circuit stimulus. These test vectors combined with the logical descriptions of the various components form a universal input to the simulation. The test-vector inputs for Daisy can be generated discretely or as waveforms. The timing of the simulation goes in "clicks" of time. Any node within a circuit can be examined for its state and forcing level at a particular time. This is invaluable for tracing and pinpointing where an error has occurred.

The data base interface is an access from Daisy to the outside world. Before we bought Daisy, we examined similar workstations, but they lacked consistent access to their internal data base – the parametric description of the circuit. We needed this consistency to interface to outside systems such as Applicon or Pirate. An obvious application of such data bases is netlisting; some less obvious applications are extraction of bills of materials, reliability data, and back annotation.

Back annotation, as mentioned earlier, is a way to update records with reference designators and pin numbers not available earlier in the design process. For example, if you send four two-input NAND gates to the PIRATE place and router, it sends back a single 7400 gate with a U-number and pins assigned. This automated back annotation to the schematic produces a schematic which is "guaranteed" to match the implemented circuit board.

Daisy Does an ECB

Let's trace an ECB through a simplified design path on Daisy. What Daisy furnishes for this function is shown in figure 4. They furnish much component data, but we as users have to define additional components. We then have to do some formatting to interface to our particular place and route package.

This formatting enables Daisy to provide data for ECB layouts and automated manufacturing. It also allows Daisy to provide bills of materials and similar information. Physical data comes down from the layout to automated manufacturing. The back-annotated data that comes back to Daisy from layout also requires a formatted interface on Daisy to function automatically. Schematics come straight out of the Daisy. We manually generate test vectors which are used to automatically verify the circuit via simulation. These vectors come out of the Daisy into automated testing and into product diagnostics.

Daisy Does a Gate Array

Let's trace a path which is followed by a vendor-based gate-array design. The vendor furnishes macrocell graphics and timing to us. We furnish to the vendor a netlist in the vendor's format; or if a translation is required, we expect them to do it. The vendor returns a verification of the schematic and of timing. Once we approve that verification, magic occurs and we receive a finished and functioning gate array.

ECS Chose Dash-1 Software

ECS is moving fast. To address their fast-track requirements, they chose to buy Dash-1. They did this to provide a consistent interface to both their design and support groups. Dash-1 is a system of schematic-entry and netlist-formatting software which runs on a IBM personal computer. Dash-1 can handle start-up documentation and be an interface to semi-automatic manufacturing, and to ECB layout. ECS has not addressed automated ECB layout and fabrication yet. They have used Dash-1 for design documentation and as a semi-automatic interface to manufacturing wirewrap.

Like Daisy, the Dash-1 system has been noticeably improved over the last year with several software releases.

Much as we in Logic Analyzers do when using the Daisy, ECS has to define graphics for any component not furnished with Dash-1.

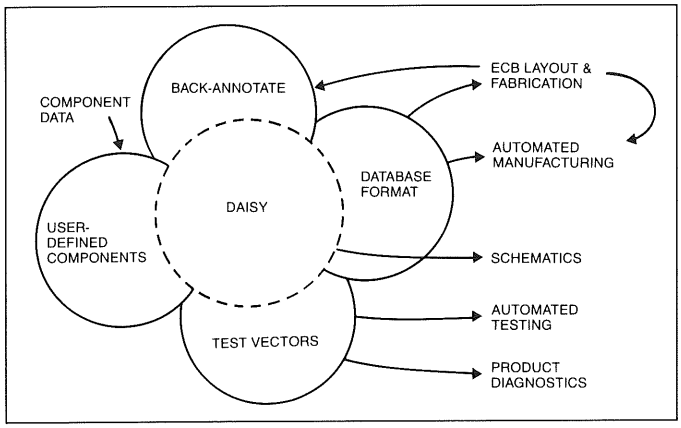


Figure 4. ECB design with Daisy.

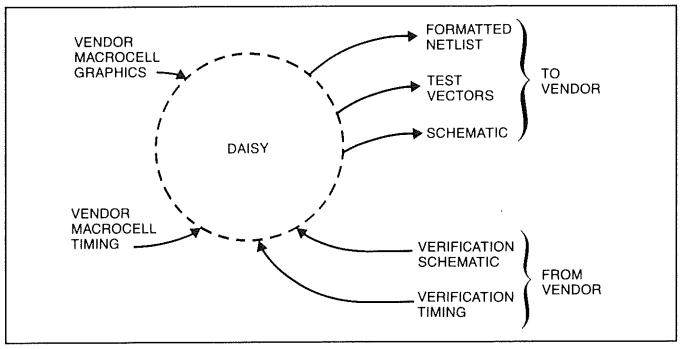


Figure 5. Gate-array design with Daisy.

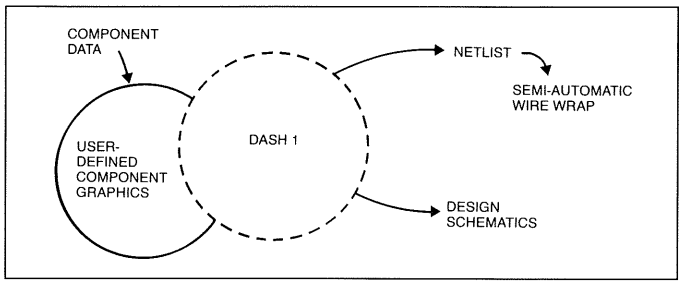


Figure 6. Wire-wrap board design with Dash-1 software.

ECS found they had to do some netlist formatting to do semi-automatic wire wrap. ECS found the netlist-to-semiautomatic-wirewrap process required a small piece of hardware called a VAX. ECS thought it was easier to take a Dash-1 formatted net-list, upload it to the VAX, provide a common data base at the VAX to do translations to wire wrap, and also to sort the information into various lists. The fact that ECS elected to use this involved process says volumes about the need in CAD for supporting utilities on the workstation.

The Future, What We Must Do

It is clear that we need to overcome current barriers between tools. Right now we have diverse tools with just a few linking interfaces; and these interfaces are limited. Some of the barriers between tools are natural; others are intentional. Our challenge is to fit the pieces together!

One place where we need to put the pieces together is in a standard electrical-design format. (Others, in this issue, have discussed standard formats as a means of tying diverse tools together.) Automatic interfaces between tools are a must. These interfaces must be straightforward. Right now, both in Logic Analyzers and throughout Tektronix, interface complexity requires several design iterations just to get rid of the bugs introduced by interface complexity. We have avoided some manual translation, but we have paid for this with a fair amount of initial debugging of the additional workstation interfaces. Of course the fact that the different tools were never designed to work together did not help.

Designers at the division level need support groups. Whether these groups should be central or common to a division is not resolved. Wherever they reside, they must have the resources to investigate and develop the cohesive packages that are needed to support designers. I know it is difficult to create and maintain that environment in a division.

I think the ECS-LDS combination is the best hope we have for combining the set of tools that we typically use in DAG. Again, the overall goal is to put consistent design capability in the hands of electrical designers and to build a coordinated tool set, capitalizing on the tool work Tek has already done. The tasks are large, but the Design Automation Group is addressing them.

For More Information

For more information, call David Bennett 629-1922 (92-731). □

TECHNICAL STANDARDS

Domestic

To holders of ANSI Y1.1 — We have just received a shipment of ANSI Y1.1's that have an Appendix B added. ANSI does not call this a revision. If your copy of ANSI Y1.1 does not have this new information, please contact Technical Standards, 627-1800.

ANSI X3.110-1983 — Set formats, rules and procedures for encoding alphanumeric text, redefinable characters, and pictorial data in many colors. The standard eliminates the need for prior agreements or negotiation dialogs to interchange information efficiently. $20.00

ASTM B628-83 *Specification for Silver-Copper Eutectic Electric Contact Alloy.* $5.00

New Standard

ASTM F816-83 — *Test Method for Combined Fine and Gross Leaks for Large Hybrid Microcircuit Packages* — F816 applies to testing hermetic packages for leaks some of which are too small to detect except with a helium mass spectrometer used in the test. F816-83 provides real-time evaluation of the quality of an in-line sealing process for sealed packages. $10.00

ASTM Publication – STP822 — *New Direction in Molecular Luminescence.* $19.20

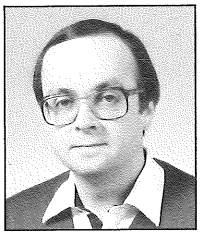
IPC – *Hybrid Microcircuit Design Guide Manual* — A comprehensive 154-page guide seeks to clarify and standardize the design of hybrid microcircuits. Every element, from design concept evaluation through failure analysis, is thoroughly discussed and illustrated. $30.00

IPC – *Chip Carrier Mounting Guidelines* – IPC-CM-78 — A must for packaging engineers, equipment designers, purchasing agents, and project managers. Design considerations and manufacturing/inspection techniques; chip carrier packages available; substrates for surface mounting; surface mounting land pattern configurations; assembly procedures; solder joint reliability data; rework techniques and inspection criteria. $10.00

For More Information

For more information, call Leah D'Grey 627-1800 (78-529). □

HOW WE USE CAE EE TOOLS IN IG



Garey R. Fouts is a software design engineer in CAD Support, part of the Instruments Group. He joined Tektronix in 1973. Garey has been a software designer for Signal Processing Systems (SPS), Semiconductor Test Systems (STS), Microprocessor Development Products (MDP), and Information Display. He majored in Math, English, and Journalism at Portland State University.

The Instrument Group is large; so large that I cannot cover all CAE/CAD tools we use or will use in a few pages. Therefore, I will limit my discussion to CAD tools used in the Portable and Lab Scope electrical engineering. Particularly, I want to discuss how we automate the manufacturing of circuit boards. IC tools used by the Instrument Group are discussed in *IC Tools,* by Paul Smith.

In the Instrument Group, our horsepower comes from three Computervision CAD systems. Two of these machines are devoted to circuit-board design, while the other is used for mechanical design.

The System

Figure 1 shows how we configured the machine used by the Portables business unit (now division) for electrical engineering. Its peripherals include five graphic workstations, a digitizer, a 46″ Versatek plotter, and 600 Mbytes of disk storage. The machine in Lab Scopes is similarly configured.

In the system in Portables, there are also several 'dumb terminals' used for project administration and for such tasks as backup and system maintenance.

Both EE machines, Portables and Lab Scopes, are connected via a RS-232 channel to IG's VAX. This VAX is, in turn, linked by the Hyperchannel into the Tek Engineering Network and to the CYBER and its CAD software and data bases.

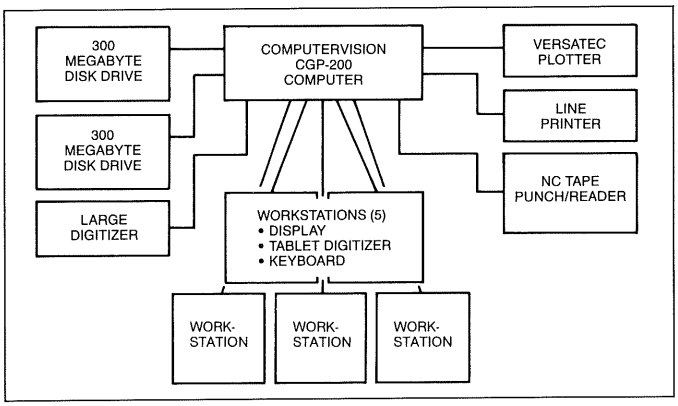


Figure 1. The EECAD system in Portables engineering. This system is used to design circuit boards. A similar system is used in Lab Scopes engineering.

We have an arrangement with the IBM Computer Center that gives us access the Master Parts Library. This gives us on-line access to such details about electrical parts as cost, value, and tolerances. The on-line availability of parts information greatly reduces the time required to generate a circuit description. (See figure 3.)

By having these part-data bases available to our machines, we have greatly reduced the chance of entering invalid information or skipping important information altogether. Now, by merely specifying a part number, all pertinent data for that part can be entered into the schematic or circuit-board data base in one step. I will give an example of how we use this power a little later.

Other tools report on the cost of the materials used, and automatically generate programs for testing machines and component-insertion equipment. Further tools are being planned now that will analyze a board's cooling requirements, check UL rules for voltage differentials between runs on a board, and other circuit-board details critical to quality.

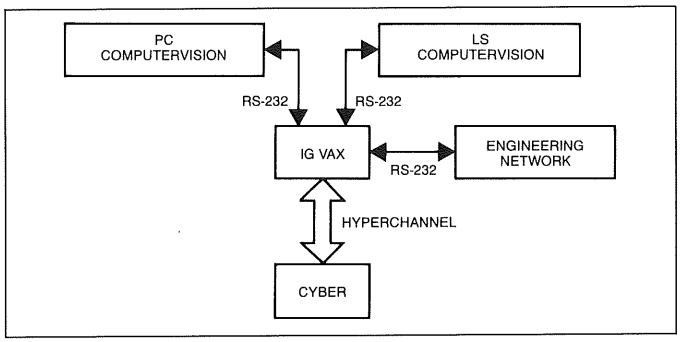


Figure 2. The EECAD system is linked to IG's VAX by a RS-232 channel. Through the VAX, IG designers access Cyber and the Tek Engineering Network.

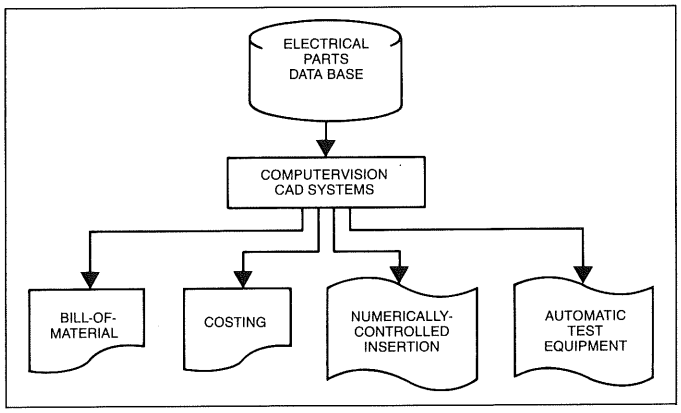


Figure 3. By on-line access to IBM Computer Center data, we can quickly generate a circuit description. By merely specifying a part number, the designer enters all pertinent data into a schematic or circuit-board data base.

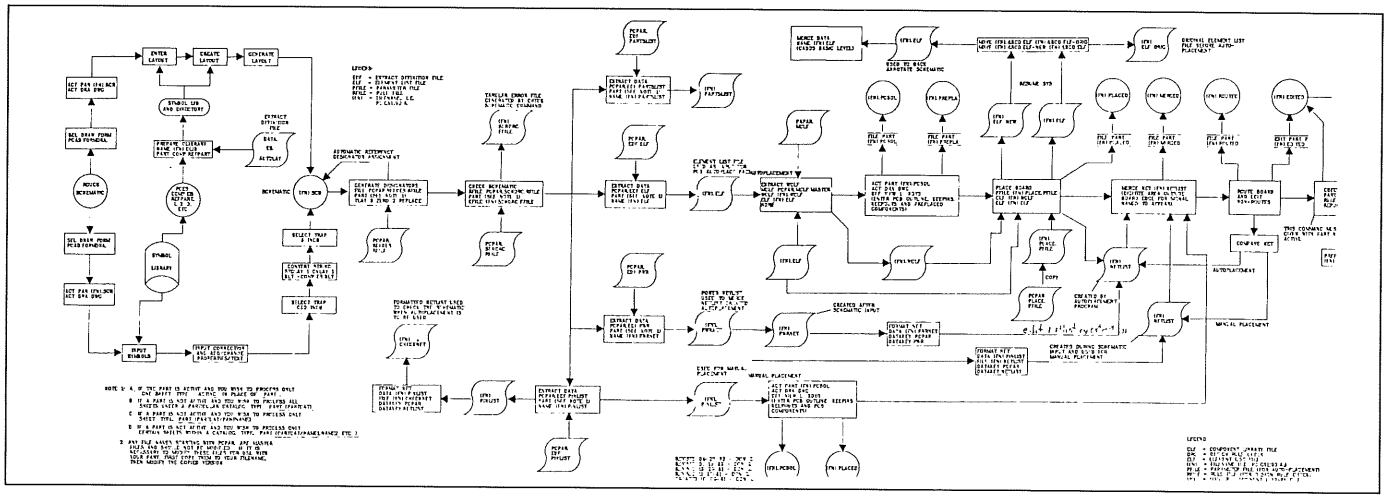


Figure 4. Don't try to read the fine print. This represents how a designer interacts with a Computervision system. This complexity is acceptable now, but it cannot be tolerated in the future.

Using the System

Even with the aid of the CAD system – some even say because of it – errors can be made in the development cycle.

The process of going from a hand-drawn schematic to a finished, stuffed, and tested circuit board is a long one, filled with many details, many steps requiring human interaction. Figure 4 shows each step that requires human interaction as Computervision would have us do it. Although this extensive use of designer time is acceptable now, our goal in IG CAD support is to reduce the dull, tedious aspects of circuit design by automating as many processes as practical. Our long-term goal is shown in figure 5.

We are a long way from being able to drop a hand-drawn schematic into a slot and see a finished product, but we are getting closer all the time.

Most engineers know the phenomemon of computer systems becoming more complex and time consuming than the original problem they were intended to solve. We are constantly trying to avoid that pitfall. One emphasis in our goal is to simplify processes, particularly those the user sees. We try to make our working procedures invisible to the user.

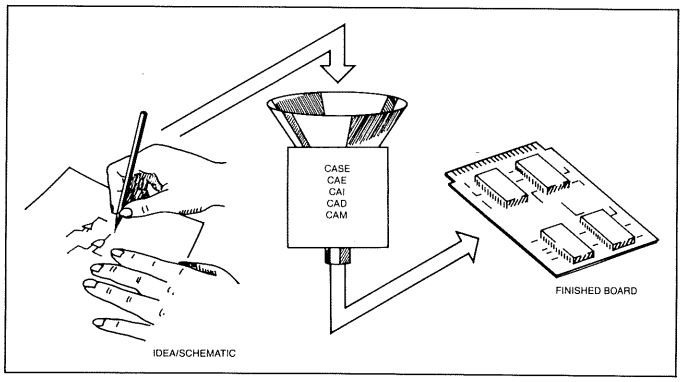


Figure 5. In contrast to the complex design/CAD interaction required now, future systems for circuit board design will be more like a place-an-idea-in-the-slot-and-produce-a-finished-board system.

Unfortunately, it is entirely possible for an automated procedure, fed incorrect data, to screw up an entire project. And with automation, engineering mistakes are made in minutes, rather than the hours or days! But this speed is two edged: one edge producing errors faster; the other discovering errors faster – better yet, discovering them earlier in the development cycle.

Two quiet, powerful programs

To illustrate the power of CAD, I'll describe two of our "quiet but powerful" programs. One is the *Pad Generator.* The second is *Automatic Annotation.*

Pad Generator places heat-relief pads on pins connected to specific power and ground planes. Because it is impossible to place all possible combinations of pads in a library, before Pad Generator the designer has had to input every pin that needed modification; this is a long, boring procedure. The designer no longer has endure this drudgery.

Pad Generator automatically searches for the affected pins and substitutes pads of the correct size. Because our customers, the design engineers in the Instrument Group, expect proper heat reliefs to be in place, they don't "see" the Pad Generator tools directly. What they do see are shorter design times and fewer errors.

The second "home-built" tool – *Automatic Annotation* – saves many many hours of tedious work. Each electrical part on a circuit board has about five properties such as voltages, tolerances, and wattages that should be annotated (printed) on a part. At one time, each of these properties had to be manually inserted on each and every part. The designers hated this task – it was so BORING.

Now, the designer need only supply the part number to the system. Figure 6 shows how the rest of the operation is carried out.

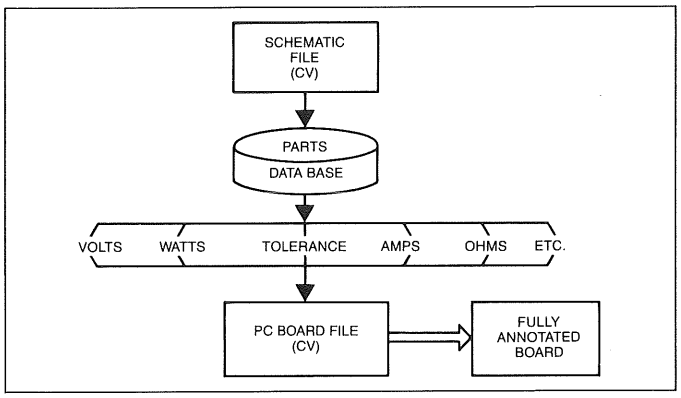


Figure 6. Annotation used to be tedious. Now the designer enters the part number and the system does the work. The process is transparent, that is invisible, to the designer.

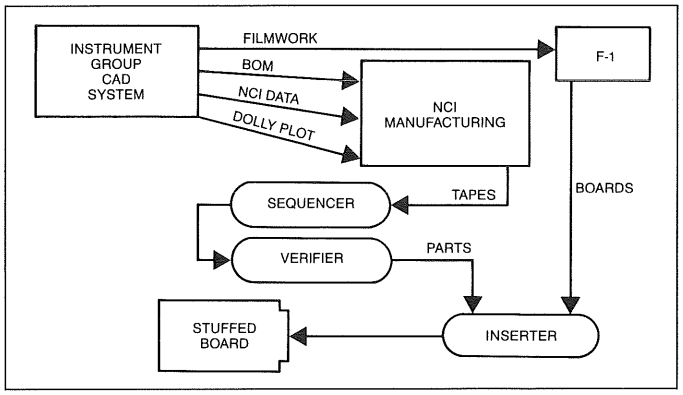


Figure 7. Board-building data flows from the Instrument Group to manufacturing sites, assuring all down-stream steps have correct data.

Making a "Quality" Board Economically

A most important part of board building is making a "quality" board *competitively* – that is, economically and fast. Manufacturing is not only the fabricating of the circuit board itself, it is also inserting and testing each of its components.

In automating insertion and testing, we have had mixed results. On one hand, with Tek-written software, we can drive component-insertion machines, saving the three of four days it takes to hand program insertion machines. But, on the other hand, we have been unable to produce a complete file to drive our GenRad circuit-board testers.

Figure 7 shows the flow of board-building data from the CAD systems in Portables and Lab Scopes, through the various manufacturing facilities, and to the final product. A major benefit of our CAD system is that it supplies correct data simultaneously to the part-insertion group in NCI manufacturing and to the board-manufacturing facility in Forest Grove.

The parallel process of building boards while simultaneously generating numerically controlled insertion (NCI) programs decreases time-to-market by at least one week. For some instruments, this translates into hundreds of thousands of dollars – all profit!

While on one hand we can fully automate insertion programming, neither we nor Computervision (CV) can produce a file that will directly drive our circuit-board testing machines. The CV machines can deliver most of the necessary data on magnetic tape, but ATE operators must make some test-program changes manually. Computervision claims that this deficiency will be "fixed in the next release." I tell my boss a similar story.

The tooling stream, today and tomorrow

Not too long ago, our tooling stream interface to manufacturing looked like figure 8. We had to traverse between the CV, a 4081, and CYBER. The system worked, but it was cumbersome. The direct linkage of the CV to our VAX greatly simplified the process.

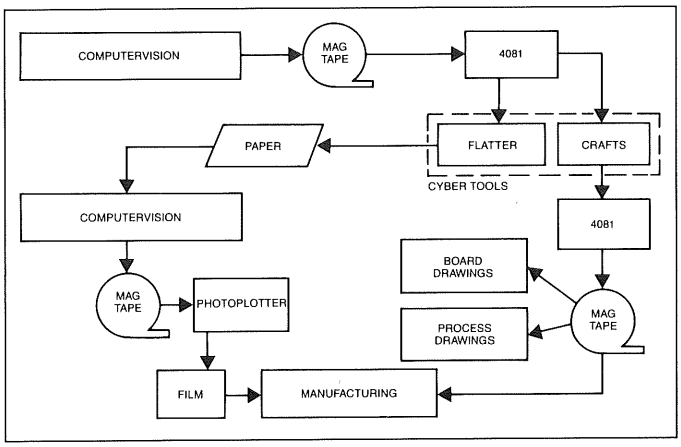


Figure 8. The tooling stream that was. Using this system required parallel paths to the tools and to the fabrication facility. (See figure 9.)

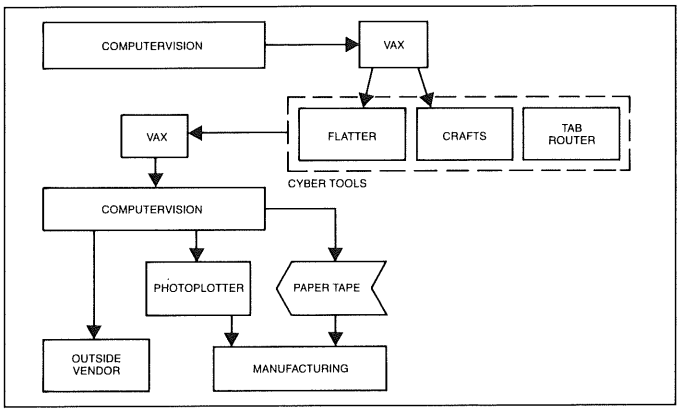


Figure 9. The tool path now is straightforward, eliminating the inefficient processes inherent in the old process.

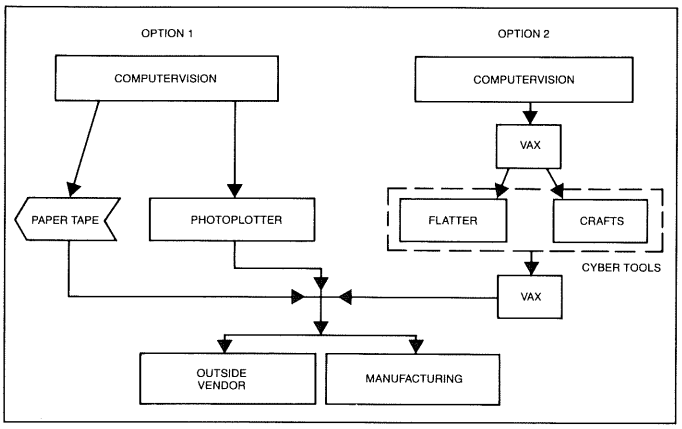


Figure 10. IG's future board tooling stream will offer the designer two ways to provide data to board manufacturers.

Currently, we rely on the CYBER's CAD tools to produce our drill tapes and flat layouts. Figure 9 shows the linkage between the CV and CYBER through the VAX to CYBER and back again.

As we develop more tools on our CV system, we will be able to skip the Cyber step and feed data directly to the manufacturing areas. Figure 10 shows both options.

For More Information

For more information, call Garey Fouts 627-4269, 47-578. □

CALL FOR PAPERS

OFS '85

San Diego, California
February 13-14, 1985

Co-sponsored by the Quantum Electronics and Applications Society of the Institute of Electrical and Electronics Engineers and the Optical Society of America.

Abstract deadline: September 14, 1984

This conference will cover all aspects of optical fiber and integrated optic sensors, including analysis, research design, performance evaluation, and application; as well as fundamental and technical problems, performance limitations due to nonreciprocal effects, backscatter, and coherence.

Subject Category

Sensors for temperature, acoustic wave strain, pressure, acceleration, current, voltage, magnetic field, flow, level, scatter, radiation, rotation.

Optical fiber gyroscope systems — fundamental and technical problems, performance limitations due to nonreciprocal effects, backscatter, and coherence.

Optical sensing techniques — interferometry, coherant and incoherent systems, ending loss systems, time domain reflectometry, polarimetry, remote spectrometry, polarization based systems, sensor multiplexing.

Optical components with particular reference of sensors — sources, detection schemes and componentry, specialized fibers — polarization isolators, couplers, multiplexers, polarizers, integrated optic devices.

Application of optical fiber sensors — instrumentation, metrology, process control, incipient machine failure analysis, navigation aids, sonar.

SOUTHCON/85

Do YOU have an idea for a professional program session for Southcon/85?

The Southcon/85 Professional Program Committee invites you to propose a SESSION to be presented at Southcon/85, March 5-7, 1985, in Atlanta, Georgia.

A Session consists of four to five papers. Session possibilities include, but are not limited to:

Artificial intelligence, automation, CAD/CAM/CAE/CAT, communications, components and devices, computer aided design, computer architecture, computer graphics, computer hardware, software, and peripherals.

Data base management, data communications, data storage, design technology and methodology, displays and display systems, digital signal processing, electro optical systems, EMI/RFI, entrepreneurship and venture capital, fiber and integrated optics, and field programmable ICs.

Human factors, image processing and pattern recognition, instrumentation and measurement, local area networks, management, marketing and careers, manufacturing methods and technology, and memories.

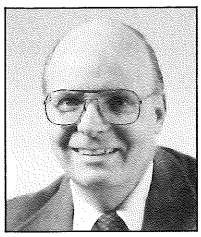
Microprocessor technology and applications, packaging technology, productivity, reliability and quality assurance, RF, microwave, and millimeter technology, semiconductor technology, sensors, and transducers.

Signal processing, simulation, software, software operating systems, surface mounted technology, testing, VLSI/VHSIC, and workstations.

For More Information

For More Information, call Art Andersen 642-8934 (53-077). □

CAE TOOLS FOR EEs AT IDG



Dick Preiss is the manager of the IC/Hybrid Design Center, part of Graphic Systems Products (GSP) in IDG. Dick joined Tek in 1969 after developing computer-controlled instruments at Beckman Instruments. He received his BSEE from California State Polytechnic University in 1964.

This article will cover the evolution of the EE-CAD tools in IDG, looking first at work-flow requirements and then computer networking considerations. IDG in Wilsonville is a remote site. Historically, this remoteness has made it difficult to communicate with project groups in Beaverton. Some of those problems have been overcome, but many still remain. This article will look at IDG programs which address the remoteness barrier and at specific enhancements planned for CAE over the next two years.

In IDG, we have tool clusters to support four major phases of product development:

- Product planning
- The design process
- Physical layout
- Prototyping

Product Planning

Product planning tends to be division-staff intensive, often consuming excessive engineering time. In this phase, CAE supports four tasks:

- System specification: alphanumeric editors, EMACS and VI
- Technology selection: component libraries
- Partitioning: behavior simulators (future)
- Project planning: PERT, GANTT, and others

Technology choices and partitioning depend on what technologies are available. The major tools used are component libraries in the form of TTL, HMOS, bipolar, and simulation models.

In project planing, some IDG managers use PERT- and GANTT-chart techniques. Apple's Lisa-project is the primary tool used in IDG. Soon, many engineers will have the MAC-project to use as well. Cyber-based planning tools are not used because access is difficult and user experience is lacking.

The Design Process

Design is the engineering-intensive, time-consuming phase of the product-development process. Computer support requires only moderate expenditures. These are the steps and tools:

- Schematic Capture: editor
- State design: SMS, Netlist Merge
- Simulation: CYTLOGS, SPICE
- Test generation: editor, TLOGS (fault coverage)

Schematic capture – Today, engineers do schematic capture manually with alphanumeric editors to develop netlists. We have a great need for graphical schematic-data capture with automatic netlist generation.

State design – Engineers do a minor amount of design using SMS. 4100-Series engineers wrote a netlist-conversion program so that the output of SMS can be merged with an existing netlist for simulation as a single design.

Simulation – IDG uses two simulators: SPICE, for analog simulation, and CYTLOGS (Cyber TLOGS). CYTLOGS integrates the VAX system in Wilsonville with the Cyber in Beaverton. This integration enables designers to enter design data on the VAX UNIX system.

Test generation, in IDG, is a manual editing task. Test support tools exist within the TLOGS simulator and make test generation anywhere from an easy task – if you accept very poor fault coverage – to laborious for complete fault coverage. The Component Development Group (CDG) provides training that gives designers the skills necessary to perform testability analysis at both the design and simulation levels.

CYTLOGS automatically sends the netlist over the Hyperchannel where it is submitted as a batch job to the Cyber. Results are returned by electronic mail to the VAX. Users do not have to learn all the intricacies of Scribe and the Cyber operating system. Instead, they can learn all the intricacies of EMACS, UNIX, and VI; these are part of the IDG computing environment.

Layout Phase

Layout is something that engineers in IDG consider a service. This is true for both etched circuit boards and integrated circuits, and hybrids.

Steps and tools used for layout include:

- Place and route – MP2D, PIRATE, NOMAD, HCAD
- Netlist extraction and and verification – TCAD, ENAP
- Resimulation – ADDCAP, TLOGS
- Checkplots – CIFPLOT, NOMAD, HCAD

Traditionally, engineering schematics are turned over to service groups for layout and fabrication. However, layout does not have to be service-group intensive; a designer can do it with a suite of tools that reside in his or her division. But unfortunately, there are problems: such tools tend to fail, or cell-library releases are inconsistent. These and other problems make it beneficial to put layout tools in the hands of a layout specialist.

Today, the IC CAD group runs MP2D as an IC-design service IDG engineers work with building 59 to ensure that IC layout runs are successfully completed. The layout group in Wilsonville does the same for engineers using PIRATE or NOMAD to design ECBs. HCAD, the layout service for hybrid circuits, is also available in Wilsonville.

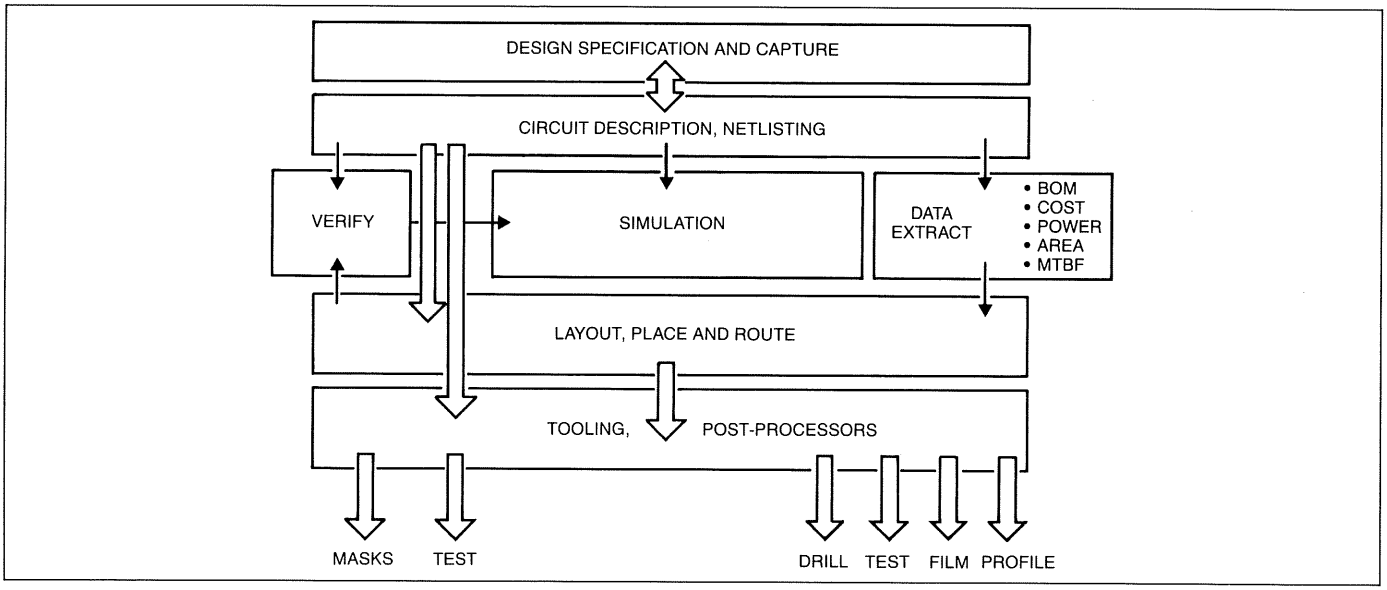


Figure 1. The product design flow. Within this flow, figures 2 and 3 detail the roles of CAE tools in helping engineers develop ICs and ECBs.

When you have completed placement and routing, and used the various layout-verification programs that are part of TCAD's suite of tools, it is time for resimulation.

Resimulation is a form of design verification to extract capacitances after running a trial physical layout. These capacitances are used to modify the simulation delays so that a more accurate simulation will result.

Check plots are another service that the design group provides to designers. These plots assure that layouts conform to the specifications. In other words, will it actually fit the available space, and can it be fabricated? What will it cost? Is the electrical connectivity correct?

Prototyping Phase for ICs

In the prototyping phase, IDG engineers are involved in these functions:

- Digitizing: Calma
- Mask-making: Calma, David Mann
- Fabrication
- Wafer sorting
- Packaging
- Testing: DAS, S-3260
- Evaluation

The prototyping phase involves the Component Development group. We, in IDG, work with CDG for digitizing and make masks on the Calma system, fabricating, sorting wafers, packaging, and testing. We have tools to drive DAS and S-3260 Testers with simulation test-vector files. Then, as a final step, the engineer evaluates a prototype of his or her IC.

Let's look at figure 1 and review the process we have just covered, starting at the top with the IC-design specification. Today, we capture the design with an alphanumeric editor. Eventually it will be captured schematically. The capture process produces a circuit description (netlist). The circuit description is critical to other important processes, for example, driving placing and routing simulation programs and various tooling postprocessors.

The output of place and route verifies that the logic description matches the simulated physical layout. The output is also used to verify that the layout capacitances extracted by the program will resimulate delays appropriately. Finally, post processors convert the physical data into drill tapes, test files, profiles, data, and masks; these are used to fabricate actual parts.

CAE Brings Expert Knowledge

How does CAE fit into the design flow in Wilsonville? Let's take a look at the integrated circuits design flow in figure 2.

In this issue, Glen Widener talks about expert systems bringing knowledge and technology into the hands of the business-unit (division) designers. (See *Introduction to Computer Aided Electrical Engineering.*) The four "knowledge"drums in figure 2 represent this concept. The drums are hierarchial libraries containing "expert knowledge." One drum holds the logic-symbol library; the other drums hold logic models, shapes, and patterns, respectively.

The drums are relational. For each logic symbol there is a corresponding logic model that instructs the simulator as to what the logical function of that logic symbol is and what delays the implemented device will have. For each symbol there also is a corresponding cell shape – a cell shape is an outline of the cell – and a description of connection points. A connection point is analogous to a dolly in ECB layouts.

The final IC-design step, producing masks for making ICs, is accomplished on the Calma system. To produce a mask, cell patterns are added inside the cell shapes. Cell shapes and patterns, as part of the mask, are use to make actual devices on silicon.

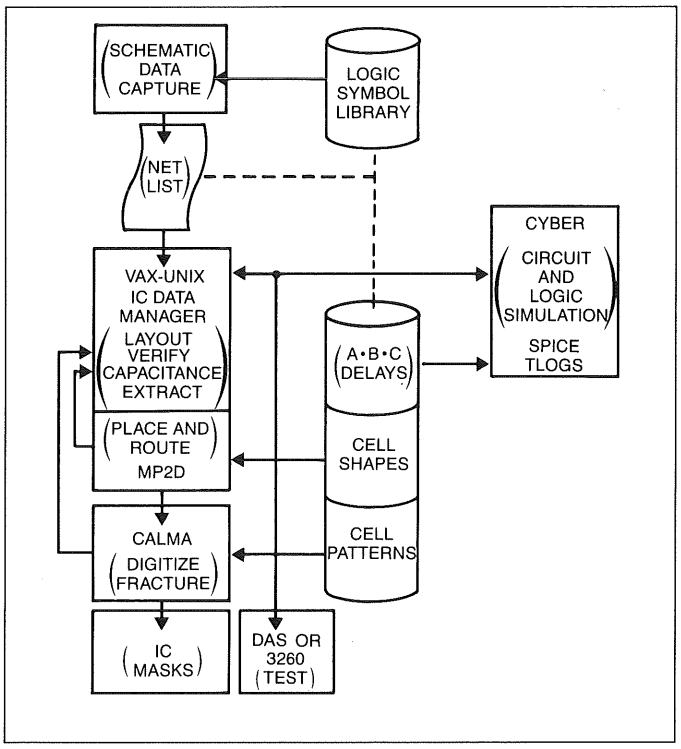


Figure 2. Embodying the expert system's concept of putting the technology into the hands of the design engineer, four "knowledge" drums hold accessible know-how for use in making a well-designed IC.

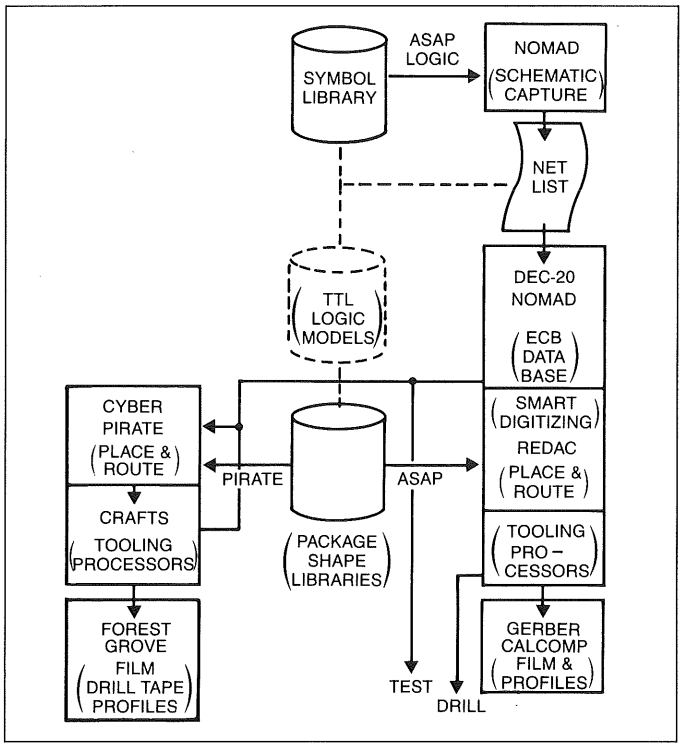


Figure 3. In ECB design too, "knowledge" drums store accessible technology know-how. The TTL "drum" does not exist in IDG; IDG engineers cannot simulate TTL on the Cyber when doing ECBs.

The library hierarchy table (see table) helps describe the level, function, and application of various design libraries. The proper setup and updating of design libraries are important steps in providing "expert" systems to users. Libraries available in IDG use presently encompass the gate through pattern levels. Future enhancements will include the block through function levels.

Turning next to etched circuit boards, we can see the similarity to the IC-design flow in figure 2. Because the drum for TTL logic does not exist – but it is being worked on – TTL cannot be yet simulated on the Cyber.

In circuit-board design, the major job has always been the implementation of engineering schematics into layouts. There are two ways to do this in IDG. One is with the ASAP/REDAC/NOMAD systems. Another is via an EDF link to the Cyber system in Beaverton. From Cyber, Pirate and Crafts interface the circuit board plant in Forest Grove.

In a parallel manner, the tooling path through the Nomad system yields Gerber plots, drill information, film and profile information that can be sent to Forest Grove or to some other vendor for fabrication.

LIBRARY HIERARCHY

LEVEL	Analog IC	Gate Array	ECB	Description
Block	electrical domain	X	X	Architecture level
Package			X	Planning level
Function		X	X	Schematic level
Gate		X		Engineering level
Circuit	X		X	Technology level
Swap	physical domain		X	Placement level
Dolly/Shape		X	X	Layout level
Pattern	X	X		Fabrication level

ATTRIBUTES (Applied to the planning level except as noted)

Power	Delay (Gate engineering level)
Mass	Logic model (Gate engineering level)
Area	Functional model (Schematic level)
Reliability	Thermal model
Cost	

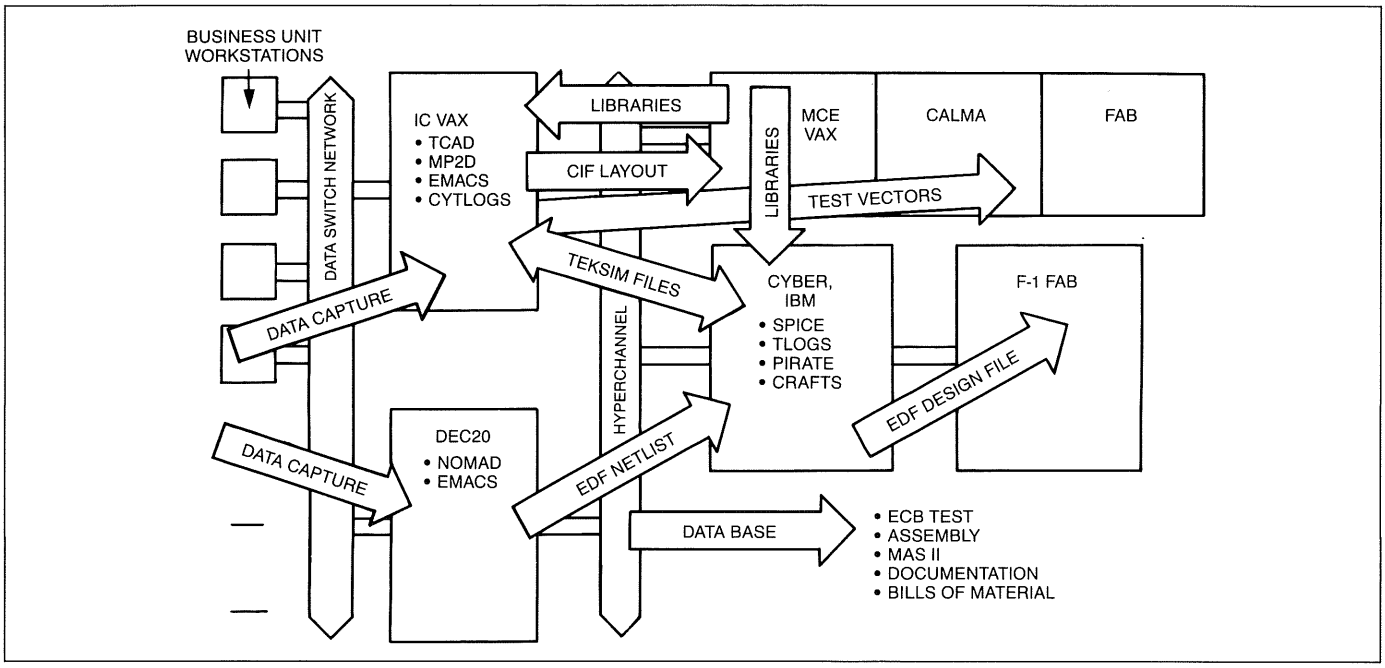


Figure 4. Tektronix Network Data Flow.

Figure 4 shows the whole EE CAD system put together, showing what physically resides in the engineering network.

The Wilsonville data-switch network is being expanded to include a SYTEK broadband data network. Hyperchannel connects IDG to the computers outside Wilsonville.

80% of the data flow paths in figure 4 are working today. Networking various sites together has been exciting to watch develop; it is making Tek's design efforts effective, even as more businesses move to remote sites.

In Conclusion, Why CAD?

Some may still be asking WHY CAD? Speakers at the recent EAC seminars on EE-CAE tools answered this question:

1. *For design verification* – connectivity, design rules, simulation
2. *For a knowledge base* – design libraries, attribute data, automatic functions
3. *For communication* – networking, documents, visibility, ME/EE coupling
4. *To impose discipline* – rules, procedures, engineering changes
5. *To improve productivity* – Productivity is a product of the preceding four benefits of CAE as well as the automation of schematic capture, place and route, and design verification.

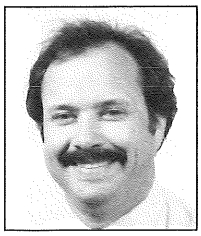
When setting CAD objectives and priorities, it is more important to concentrate on the five reasons for CAD listed above than on the idealized model – the integrated CAD system. Attack the most important problems first, and then work toward an integrated chain of tools that do not require hand intervention somewhere in the middle of the chain.

It is also important to keep tools' development in perspective. Tools are but one leg of a three-legged stool. The other legs are technology (TTL, CMOS, materials) and manufacturing capabilities. Together, the three legs support the application, the end product of the development process. Don't forget applications – and the engineers who develop them. They are the ones who ordered the CAE stool in the first place.

For More Information

For more information, call Dick Preiss 685-3647 (63-296). □

THE DIRECTION OF CAE AT TEK



Jack Hurt is the CAE manager in the Computer Science Center. Jack joined Tek in 1974 after working on electronic switching at Bell Labs. His ten years at Tek have been spent in various aspects of electrical and mechanical CAE. Jack was the principal developer of HCAD, the computer aided design system for hybrid circuits. He received his BSE and MSE from Oregon State University.

We have been hearing a lot about CAE development. Why all the interest? Why is Tektronix so involved in this area? The simplest answer: We have to:

- Design engineers have been in short supply for some years. This situation will continue.
- In the future, product obsolescence will occur even more rapidly.
- Products are becoming more complex.
- Technology is changing rapidly.
- The demand for quality remains high.

These trends lead us to increased usage of CAE tools, with their promise to help us remain competitive.

According to a recent article (*IEEE CG&A,* Dec. 1983), the market life of products has been declining, while product development time has been increasing (figure 1). Just a few years ago, an electronic product had a market life of six years, and it took only two years to develop it. By 1987, according to the same article, development and market life will be of equal length, about 3.5 years. I disagree with half of this gloomy projection. CAE development and usage has to push the slope of the product development curve down so that total development time, not just the design engineering period is reduced.

We are not just doing design engineering. A recent HP study showed the typical engineer spends only 30% of his or her time designing. Technical discussion and documentation each consume as much time as design (see figure 2). The set of tools Tek provides to our engineers must take this non-design effort into account – Computer Aided Engineering is only the first step.

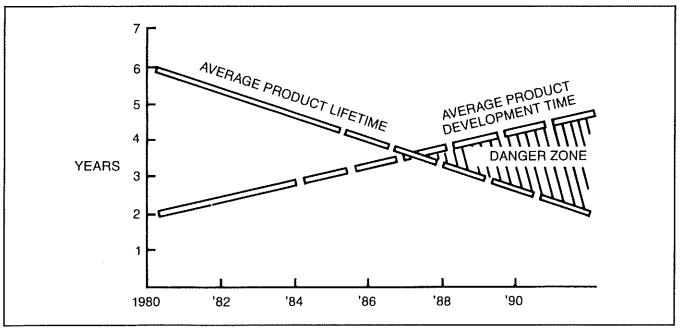


Figure 1. CAE promises to dispel the pessimistic prospects in this graph. Product development time should not continue to increase at the rate predicted here.

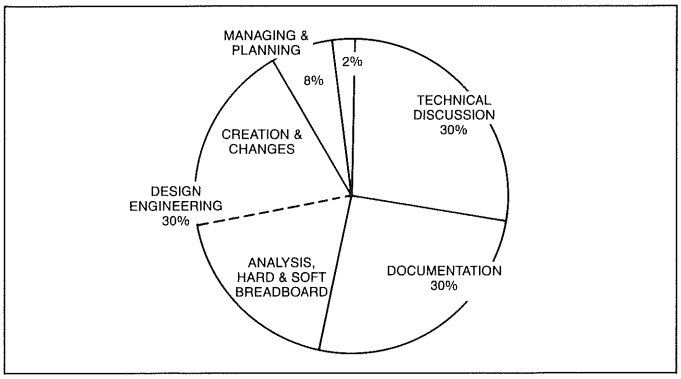


Figure 2. How do you spend your time? If you are a design engineer, you are doing nondesign tasks 70% of the time according to a Hewlett Packard survey.

Integration is the Key

You've heard this over and over, but it bears repeating: "Integration is a key to successful use of CAE tools." Integration is the means to full-cycle data transfer. It means passing data electronically from one tool to another. It also means feedback through the entire product design cycle – product design through manufacturing through field service.

Integration also means a systems approach so the tools can efficiently communicate with each other, with a consistent user interface. In other articles in this issue, tools are described which are not well integrated. Unfortunately this is fairly typical of the state of CAE systems in the industry today.

ICAX

ICAX is a major program at Tek, not just an acronym standing for Integrated Computer Aided Everything. ICAX certainly stands for a broader spectrum than just design and engineering. ICAX also encompasses manufacturing, documentation, test, and service. The total scope of ICAX includes all portions of the product-development cycle, including electrical, mechanical and software. The focus of this article is the electrical portion of ICAX. However many of the concepts we will discuss also apply for mechanical and software development.

Let's take a closer look at the ICAX program. It's a set of core utilities and tools to achieve the functional integration of Tek's distributed and dissimilar (heterogeneous) CAE/CAM systems. The major components and objectives of the tools are:

- Electronic data links
- Communication capability between dissimiliar systems
- Supervision and control of product design data
- Control and dissemination of engineering changes
- Autonomous (non-centralized) development of applications

A key concept shown in figure 3 is that of a closed feedback system. Information management utilities pass design data from engineering to manufacturing in the conventional feed-forward direction. These utilities (see figure 4) must also feed-back information to the design related to design fitness, manufacturing capability, component availability, buildability, and cost. Today, this feedback loop from manufacturing to design is missing or is quite informal at Tek.

Let's discuss the major applications and data components of figure 3, starting with CAE applications on the left-hand side of the figure. These represent the software traditionally used to design a product or component. Some examples are graphical design capture, simulation, layout, design verification, design-rule check, and artwork generation. These CAE applications require design library data to run (technology design rules, component libraries, etc.). A present and long-term goal at Tek is to strive for integration (electronic data transfer and consistent user interface) between related CAE applications. The primary need is for integration within each technology (IC, hybrid, ECB, etc.) . However, we are also striving for commonality and reusability of the tools across the technologies, where appropriate.

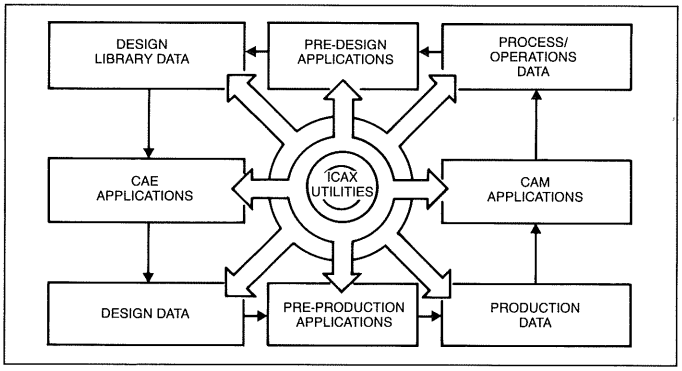


Figure 3. If integration is the key to CAD-tool success, what is Tek doing? Part of the answer is in the ICAX Program, which encompasses all parts of the product-development cycle with a set of core utilities. These utilites will enable a closed feedback system to pass product-development information in both directions. This information will assure, for example, that a product is buildable and that manufacturing has the tools to build it.

ICAX will produce design data

Design data is the end product of the CAE applications – for example the layout of a circuit, the profile of a board, selected components, the connect list of a circuit, test vectors, and performance specs.

Design data is passed into preproduction (or prototype) applications where it is used to create production data required to manufacture the component – for example pick-and-place-data generation, automatic wire-bond data, automatic flat layout, profiling-and-drill-data generation.

CAM applications are then used in making the component – for example, MRP links, generative-process planning, distributed numerical control, and automatic testing.

This would complete the normal feed-forward design process. A feedback loop is needed so subsequent designs can benefit from knowledge gained about the manufacturing process. Equipment capabilities and limitations must be fed back through predesign applications software to up-date design-library data for use in new designs.

The ICAX program started when Jerry Sullivan joined Tek about three years ago. (Jerry now manages the Logic Design Systems (LDS) division.) That's when we started gearing up to accomplish the above objectives. Some have been met; some are in the process of being met; and some have still to be addressed.

Data links and applications

The electronic data links are a key piece of the ICAX program because they will allow dissimilar systems to communicate directly with each other. These links will allow engineering managers to exercise just the right levels of design supervision; working with on-line data, they can control engineering changes anywhere in the process – concept to PSR.

Then there is another key piece: autonomous and decentralized development of the actual applications. This part of the program has created some of the diversity in Tek programs. We in the Computer Science Center have not attempted to develop all CAX applications at a corporate level because we feel that the areas where the need exists know the needs and details best. Applications development and acquisition has been going on for several years by groups throughout Tek.

Let's break the ICAX program overview into two portions, covering first utilities and then applications and tools.

ICAX Utilities

The intent behind ICAX Utilities is to allow engineers and managers to manage data during the entire product design cycle – starting with the CAE application. (A breakout of the utilities portion of figure 3 is shown in figure 4.) Our aim is to use three types of computer systems. First, Tek will use *locally owned machines* to obtain interactive response. By the end of FY400 there will be approximately 75 VAXs and 300 personal computers in the company. ECS workstations will be moving in during FY500 and FY600. We will continue to use mainframe computers for computer-intensive CAE applications as the second type of computer system. Finally, we expect special-purpose hardware to off load simulation/fault coverage and design verification from some of these mainframes in FY600 and FY700.

Let's see how much computing power will be available in Tek by AP413. Let's use MIPS (one-million instructions per second) as our unit of computing power. The application program will determine whether or not these "computing-power ratings" are valid.

We have about 75 MIPS of VAX power and around 50–75 MIPS of personal-computer power in the company now. Whether that will double in FY500 and FY600 is a little questionable, depending on how many workstations we utilize throughout the company. But I do see up to 500 to 600 MIPS of Tek-wide interactive power by the end of FY600 (see figure 5).

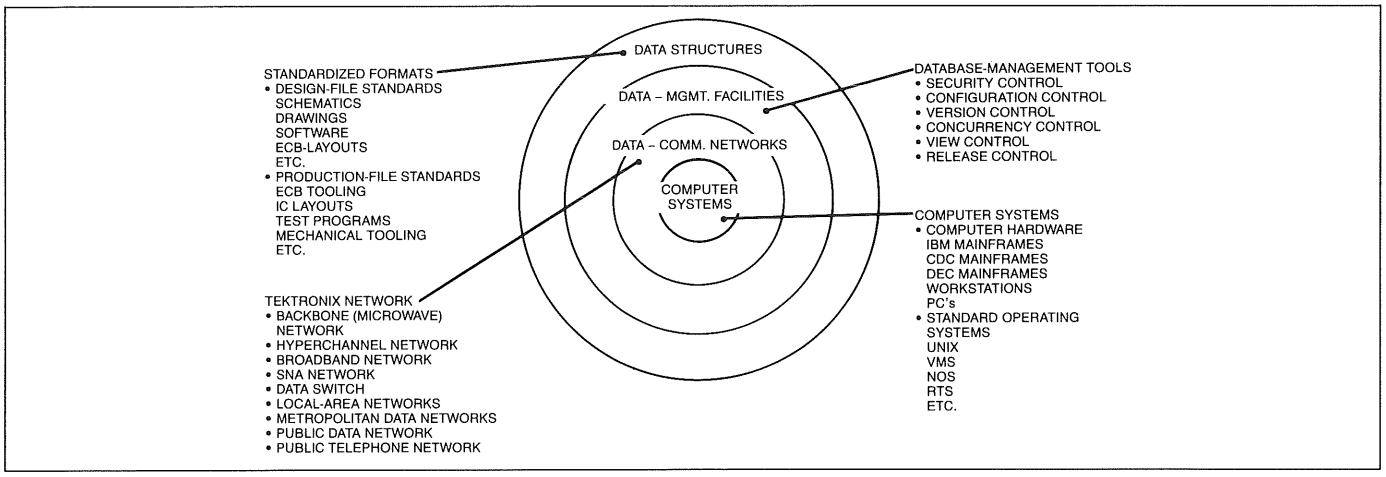


Figure 4. These ICAX utilities support, or will support, the successful integration of CAD tools.

At the same time we see mainframe power, in terms of MIPS, tracking that at about 1/10th the total CAX power in the company. Presently the Scientific Computer Center operates the CDC 760 Cyber. They have recently installed two new machines, the CDC 855 and IBM 3083B, to off load the 760. And then the next-generation machine of around 40 MIPS is planned for the end of FY600 (figure 6). We think that mainframe machines will continue to be needed because your local station, with one to five MIPS available to you, will not have enough power for the complexity of future circuits. The extra power will be particularly needed in testability-analysis design and in some layout programs.

I think engineers will use the mainframe computer power in a batch mode under local control from their local workstation. Thus the trend will be away from direct login to mainframe computers, and in fact, the mainframe will be transparent to the user.

Data communications

On-site data communications for the next few years will be similar to what is now coming on line: front-end data switches and broad band for terminal access into the local machines – Dick Preiss describes this in *CAE Tools for EEs at IDG* – and then, higher-speed links for transferring data to and from the mainframe computers using Hyperchannel and local-area networks. Home access will increase, continuing to depend on telephone links. Field office and customer access will also be into the network by the telephone line.

Figure 7 shows the engineering network today: microwave transmission to the various sites and local-area networks to transmit information between machines at each site. My understanding is that Merlo Road and Forest Grove will probably be linked to this network through cable TV, Redmond and Grass Valley by leased line.

Another project is underway to link the engineering networks and a business network. This link would provide an electronic mail system that will reach everyone in the company. The intent is to have something at least as reliable as interoffice mail. Additional capabilities for appointment and conference scheduling are planned.

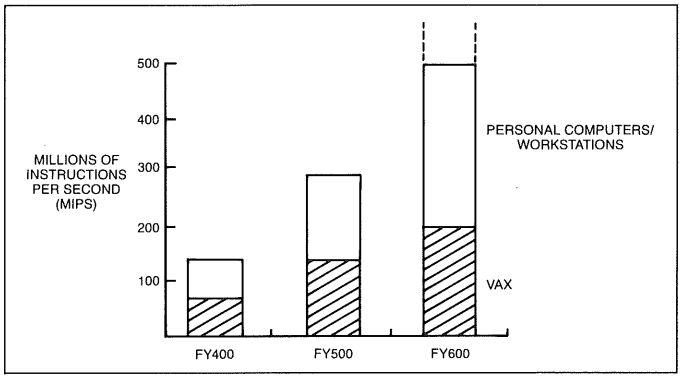


Figure 5. Tek should have at least 500 MIPS of Tek-wide interactive computing power by FY600.

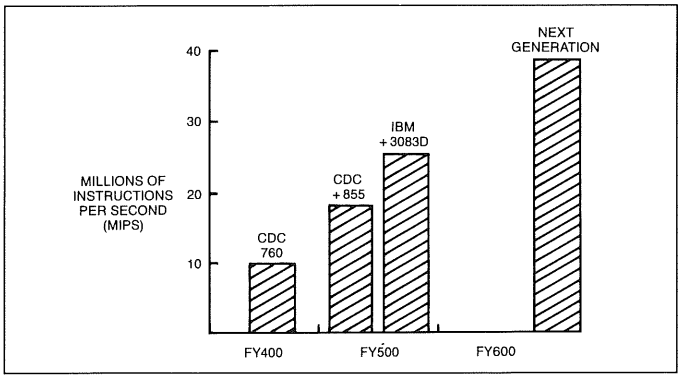


Figure 6. Mainframe power, controlled from local workstations will be necessary for testability analysis and some layout programs. Mainframe usage from the workstation will be transparent to the design engineer.

Data management

Data management facilities are a key portion of the ICAX program because we want to allow people to manage data reliably and consistently. Distributed management for CAX files is being worked on now – due to be completed in 500. This would allow version control, access control, and data transfer between sites. We are

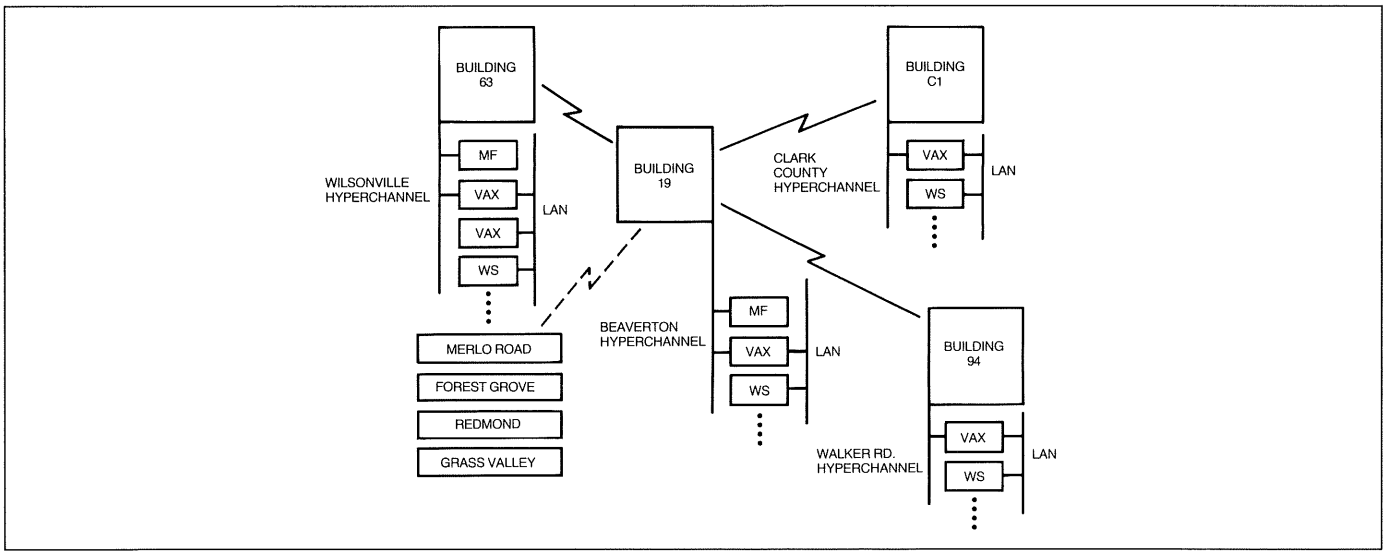


Figure 7. In the Tektronix Engineering Network, sites are microwave linked. Ethernet local area networks link machines at each site.

staffing up towards implementing an ICAX data management system which will link into component libraries, business applications and MRP applications. Most of our work will be on engineering-data management for product design.

Data standards

Data-structures standards are an important part of the ICAX program. Data standards may be Tek developed (e.g., the Electrical Design Format (EDF) (described in Jim Murphy's article *CAX Center Tools for EEs),* de facto industry standards (e.g., CIF), or formal national standards (e.g., IGES). No single standard exists which has provision for all data required to describe a product. Therefore Tek will utilize several, depending on the technology. In most cases it is desirable to have a neutral exchange format.

Neutral so we're not tied to any particular turnkey vendor for product data. Neutrality allows translation of data for any vendor. It also allows us to archive data in a way that doesn't tie us to any particular vendor.

Why are standards important? We just mentioned two reasons: to allow data exchange between dissimilar-systems and to permit organization-independent product archiving. Another place where standards are critical is in the system engineering done at the business-unit (now division) level. With divisionalization, system engineering is done with quite different equipment and data-formats, but the manufacturing of integrated circuits, circuit boards, and hybrids is still centralized outside of the divisions at various sites: Forest Grove, Building 59, Building 13. A standard format allows any division flexibility early in the technology stream because they won't have to negotiate new data formats or adapt to new data-transfer formats as technology changes. This concept is depicted in figure 8.

ICAX Applications and Tools

Tool selection will continue to be a division (business unit) function (this includes turnkey systems and special applications software). As a result systems will continue to be dissimilar – that won't change. Although such things as selection guidelines and precedence will tend to standardize tools across the company, I believe that we must provide utilities which allow dissimilar systems and tools to work together. The benefits of standardization will help move towards homogenaity, but it would be stifling to the divisions to standardize by edict.

Tek CAE products will become important factors in the next few years. The Design Automation Group (DAG) and Logic Design Systems (LDS) plan to solve many of the integration problems design engineers face today. I think LDS will play a key part in developing the applications programs that all Tek divisions will depend on. However, outside vendors and internal tools will still fill many holes. And Tek will continue to develop internal-application programs; there will always be technologies and needs that Tek cannot buy – or that will not have the potential to become Tek products. Internally developed software for these technologies will be needed.

Tool schedule

Figure 9 estimates the time of introduction and Tek-wide use of several CAE tools. During FY400, we saw increased use of simulators, primarily TEKSIM/TLOGS and SPICE. During FY500, SPICE usage will decline in favor of TEKSIM/SPICE: TEKSIM/SPICE has a more powerful user interface; it runs on local machines; it provides better models. We will also see increasing use of higher level design tools from LDS during FY500. These will be aimed at system engineering.

Engineers are now experimenting with graphical schematic capture in place of text-based hardware-description languages. FY500 will see more use of graphics capture. Netlists, from either method, will be used to drive simulators, layout tools, testability tools, and documentation systems. Hardware-description language use will continue. Its usage will depend on designer preference and on circuit architecture.

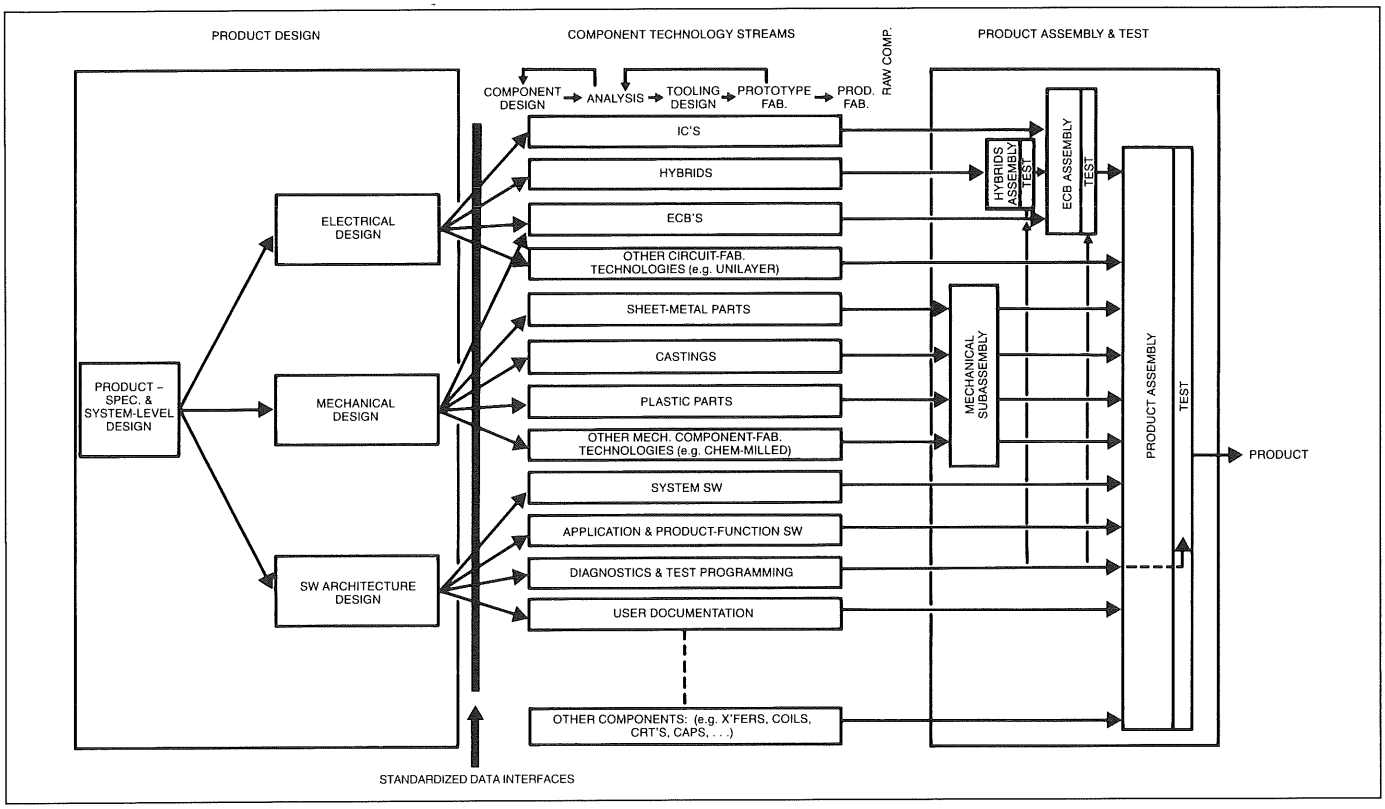


Figure 8. This is how the ICAX project views product flow. Each division, as it does product design, will use CAE systems that may differ from those used by the centralized component-manufacturing organization. Data standards, working through standardized data interfaces, will allow either "half" of this product development process the flexibility to change their internal systems without renegotiating with the other.

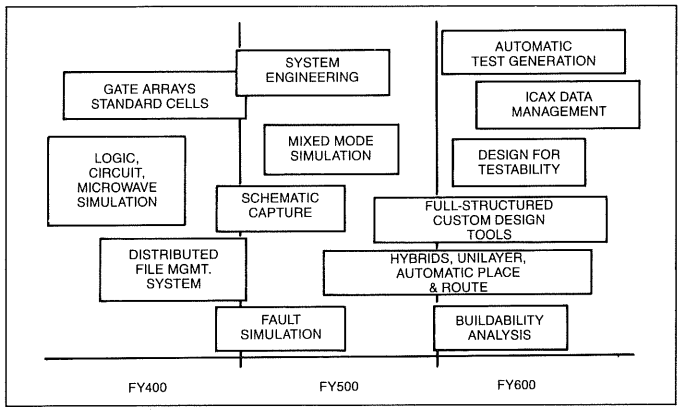


Figure 9. When will you be able to use these tools? Here's the schedule.

By FY600, full-structured custom design tools out of Applied Research should be proven and available to designers. Also, test generation and design for testability systems should be in place.

Buildability and design-fitness feedback for ICs, hybrids, and ECBs should be in general use during FY600, as well as ICAX data management utilities.

The tools shown in figure 9 are examples of what is becoming available. Tool-development is booming outside and LDS will have much competition. The experience gained by developing tools that address immediate Tek needs will be invaluable in meeting this competition. For this reason, the transfer of resources and software from internal groups to LDS will continue to occur.

Summary

Development and integration of CAE tools is an ongoing major program at Tek. Several Tek groups are committed to this effort. To coordinate these efforts, representatives from each division, EMCO, and the Technology Group participate in steering meetings coordinated by the CAX Center. Most Tek CAE-development resources are in the Computer Science Center (CSC) and the Logic Design Systems (LDS) division. Some local-tool development will continue within other divisions. If these tools can be used more broadly, they may be refined and supported by the CSC.

Utilities and standards based on the use of dissimilar tools and systems are being put in place. Although the advantages in start-up and support will always be a powerful incentive to select standardized tools, there will always be exceptions – the tools for Tek-wide data management must be able to handle them.

For More Information

For more information, call Jack Hurt 627-2599 (50-560). □

PAPERS AND PRESENTATIONS

The table below is a list of papers published and presentations given during recent months.

While providing recognition for Tektronix engineers and scientists, the presentation of papers and articles contributes to Tektronix' technological leadership image.

If you plan to submit an abstract, outline, or manuscript to a conference committee or publication editor, take advantage of the services that Technology Communication Support (TCS) offers.

TCS provides editorial and graphic assistance to Tektronix engineers and scientists for papers and articles presented or published outside Tektronix and obtains patents and confidentiality reviews as required.

Call Eleanor McElwee on ext. MR-8924. □

JANUARY

TITLE	AUTHOR	PUBLISHED	PRESENTED
Spectrum Analyzer is Portable Lab Assistant	Dave Barnard	**Microwaves & RF**	
The Optical Time-Domain Reflectometer (OTDR)	Richard Osborn	**Laser Focus**	
Circuit-Board Vibration: A Computer-Aided Design Study	Brian Wood Barry Ratihn	**Machine Design**	
An Experimental Color Monitor for Vision Research	John McCormick Gerald Murch Lyle Leavitt		SPIE Conference, Los Angeles, CA
SQA Contributions to a Quality Software Product	George Tice		Reliability and Maintainability Symposium, San Francisco, CA

FEBRUARY

TITLE	AUTHOR	PUBLISHED	PRESENTED
Obtaining High-Frequency Accuracy in a Portable Spectrum Analyzer	Craig Bryant Dennis Smith	**Microwave Journal**	
New Spectrum Analyzers Help Keep Systems Operational	Dave Barnard	**Defense Systems Review**	
BASIC Language Telecommunications	Bob Broughton	**Dr. Dobb's Journal**	
Process and Device Performance of 1-μm-Channel N-Well CMOS Technology	Tad Yamaguchi G. Kawamoto J.C. DeLacy	**IEEE Transactions on Electron Devices and IEEE Journal of Solid State Circuits: Joint Issue on Custom ICs**	
Process and Device Performance of 1-μm-Channel N-Well CMOS Technology	Tad Yamaguchi Seiichi Morimoto Galen Kawamoto John DeLacy	**IEEE Journal of Solid State Circuits**	
Highly Integrated Processor Eases Design of Low-Cost Display Station	Rebecca Wirfs-Brock Warren Dodge	**Electronics**	
Everything You Always Wanted to Know About EQ	Richard Cabot	**Sound & Video Contractor**	

FEBRUARY

TITLE	AUTHOR	PUBLISHED	PRESENTED
The Architectural Evolution of a High-Performance Graphics Terminal	Doug Doornink John Dalrymple		COMPCON '84, San Francisco, CA
Visual Accommodation and Convergence to Multichromatic Information Displays	Jerry Murch		NATO Workshop on Colour-Coded vs. Monochrome Information Displays, Farnborough, England
Perceived Brightness and Color Contrast of Color Displays	Jerry Murch Michael Cranford Paul McManus		NATO Workshop on Colour-Coded vs. Monochrome Information Displays, Farnborough, England
PSI Generative Planning Process at Tektronix	David Folts		Software Users Conference, Brigham Young University, Salt Lake City, UT

MARCH

TITLE	AUTHOR	PUBLISHED	PRESENTED
Spectrum-Analyzer Reference-Level Control: Simplification Through Internal Intelligence	Morris Engelson	**RF Design**	
Start-up Considerations in Automation	Tom Gifford		Automated Manufacturing Conference (AM '84) South Carolina
Color Graphics Hardcopy: Where Can Ink Jet Win?	Wayne Jaeger		Institute for Graphics Communications (IGC) Amsterdam, Holland
Structure Charts and Program Correctness Proofs	Shang-Cheng Shyou		7th International Conference on Software Engineering, Orlando, FL

Continued

Technology Report

MAILING LIST COUPON

☐ ADD
☐ REMOVE

Not available to field offices or outside the U.S.

MAIL COUPON TO 53-077

Name:_____ D.S.: _____

Payroll Code: _____

(Required for the mailing list)

For change of delivery station, use a directory change form.

APRIL

TITLE	AUTHOR	PUBLISHED	PRESENTED
Using Analog Sampling and Digital Storage to Improve ECL Testing	David White	**Electronics Test**	
Standardizing CRT Measurements	Peter Keller	**Test & Measurement World**	
Physics and Technology of Plasma Processing	David Evans		Oregon State University
High-Speed Analog-to-Digital Conversion	Larry Riley		University of Vermont
Secondary-Ion Mass Spectroscopy	Mary Ryan		American Chemical Society Conference, St. Louis, MO
The Orthogonal-Field Amplifier and Other Applications of Small-Core, Saturable Magnetics to Switch-Mode Converters	Steve Pepper		POWERCON, Dallas, TX
Displays: Market and Technology	Steve Blazo		Pfefferkorn Conference, Ocean City, MD
Progress in the Theory of Electron-Beam Deflection	Ed Ritz		Pfefferkorn Conference, Ocean City, MD
CRT Apertures and Electrostatic Field Patterns	Norm Franzen		Pfefferkorn Conference, Ocean City, MD
Who Needs Hybrids If You Can Put It on Silicon? The Role of Hybrids in 1990 and Beyond	Aris Silzars		ISHM Southern Conference, San Francisco, CA
Viewing a Programming Environment as a Single Tool	Norman Delisle David Menicosy Mayer Schwartz		ACM Software Engineering Conference, Pittsburgh, PA
VLSI Processing Technologies	Jon Schieltz		Iowa State University