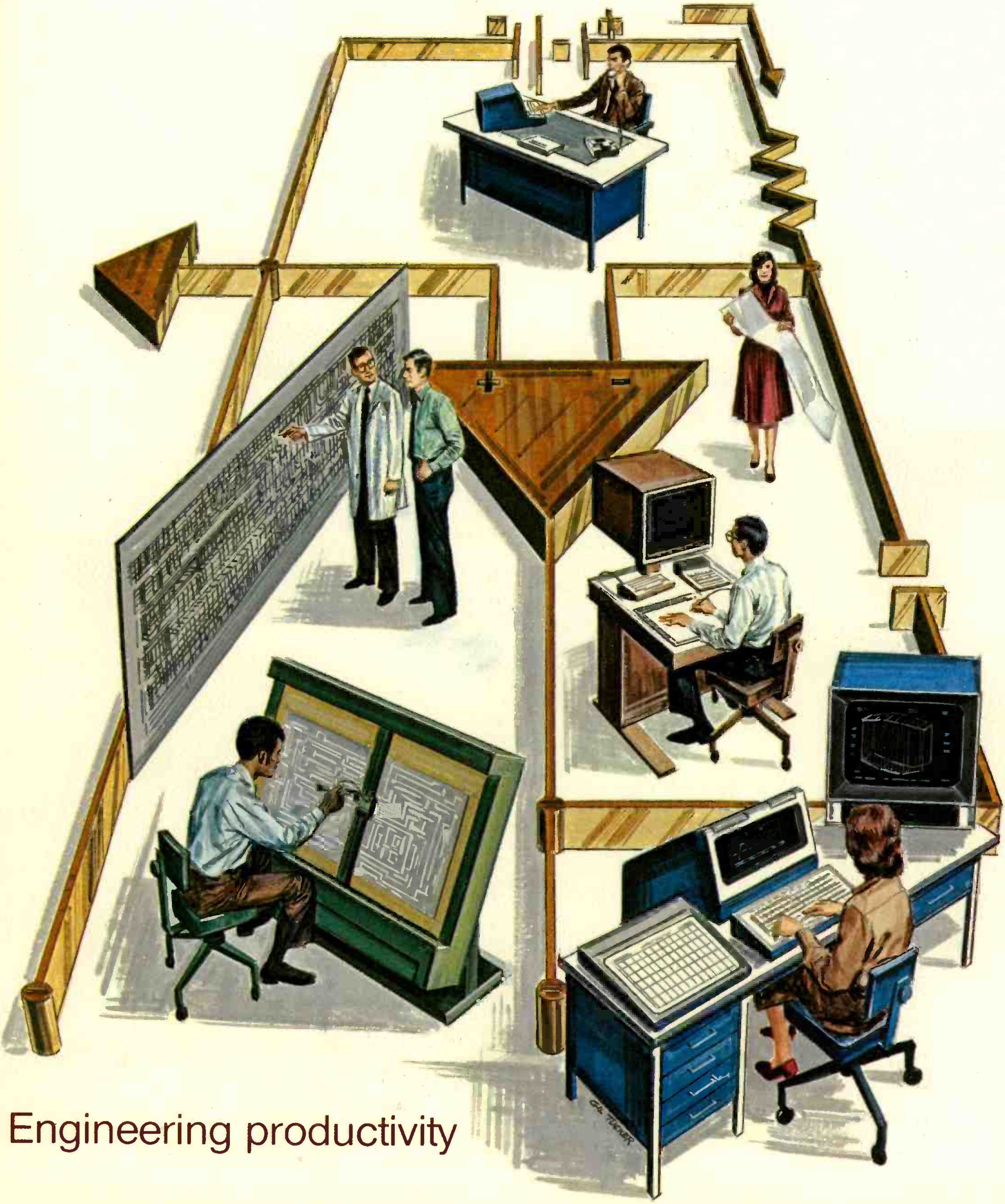
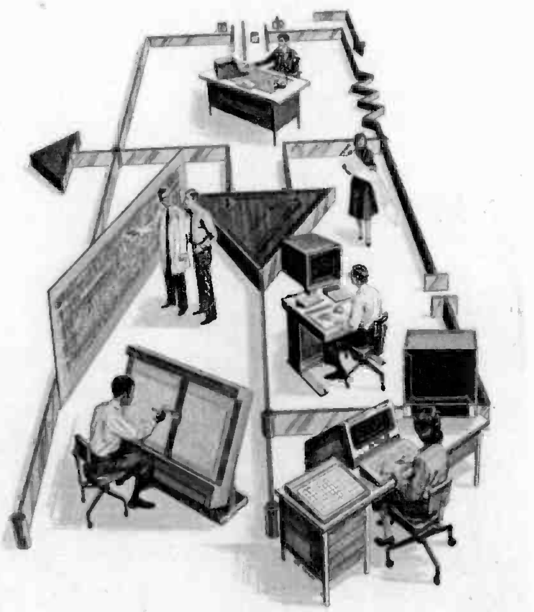
RCA Engineer

Vol. 28 No. 1 Jan./Feb. 1983



Engineering productivity

Cover illustration: Gil Tucker, Missile and Surface Radar



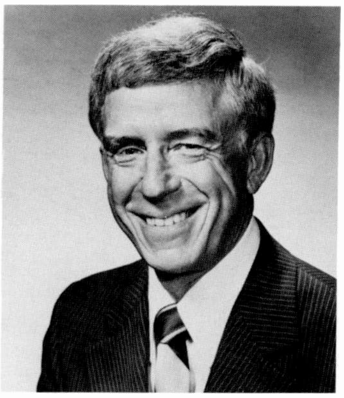
Amplification of engineering skills—that's what our cover and contents demonstrate in this issue. Our authors focus on the tools and techniques that lead to greater engineering productivity, company-wide. New computer and robotics technologies, new software, new methods for organizing engineering resources to achieve better producibility, and new uses for statistics in manufacturing take center stage. Engineers throughout the company are integrating their skills with these new-found assistants. They're making the amplifications "operational."

—*MRS*

RCA Engineer

A technical journal published by
RCA Research and Engineering
Bldg. 204-2
Cherry Hill, NJ 08358
TACNET: 222-4254 (609-338-4254)

● To disseminate to RCA engineers technical information of professional value ● To publish in an appropriate manner important technical developments at RCA, and the role of the engineer ● To serve as a medium of interchange of technical information between various groups at RCA ● To create a community of engineering interest within the company by stressing the interrelated nature of all technical contributions ● To help publicize engineering achievements in a manner that will promote the interests and reputation of RCA in the engineering field ● To provide a convenient means by which the RCA engineer may review his professional work before associates and engineering management ● To announce outstanding and unusual achievements of RCA engineers in a manner most likely to enhance their prestige and professional status.



James Vollmer

Productivity: Where it begins and ends

Most people attribute the need for increased productivity to the Japanese but that is not so—the requirement was there before the Japanese competitive threat. Neither did it originate within our corporations; it is not traceable to our boards of directors, or our supervisors or any of us. *It is our customer who imposes the productivity improvement requirement.* He imposes it by continuously expecting lower prices, or improved performance, or more often, both. Historically, he has always managed to get what he wants. I believe he will continue to do so, because we operate in a competitive society.

Let's explore this idea a little further. It is axiomatic that if we don't meet the customer's needs, someone else will. But the only way that we can cut prices to our customer, and still maintain the quality necessary to stay in business, is to cut our costs. It is important to note that such cost reduction should not be considered the burden of any single group. It is everyone's responsibility. Furthermore, we must take cost out of every element in the process . . . be it by spending less time to produce . . . or by testing more effectively with fewer tests. . . or by requiring less overhead to manage. This is a difficult challenge, made more so by a simple reality—everyone involved expects increased pay as his experience increases, a condition which drives the costs up, not down.

For us to realize both expectations—the customer's and our own—each of us must not only find a way to be truly worth more, but that extra worth must show up in the product. That's the only place we can generate the rewards that we all want.

Thus, the need for productivity improvement comes down to each of us in a very personal way. *We must each establish a program to improve our performance, to measure that improvement, and to make sure that it shows up in cost reductions in the product.*

But the customer's role does not end with his impact on costs. In addition to imposing the productivity improvement requirement, he's the ultimate judge of our quality system—he literally applies the final quality test when he decides if our product does what he wants it to do. Accordingly, we must correctly anticipate this—*the design of our quality system must begin with the customer and his expectations, then work its way back through the salesman, through manufacturing, into engineering and to the inventor.*

Recognition of the cited roles of the customer leads us to his primary role, that of the Product Specifier. He says what he wants and *he always gets what he wants.* Rarely does he say it by speaking, however; instead, he chooses from current availables. When a newer product or service comes closer to his needs, he restates his choice by choosing again, then again, and again. Thus, product success is fleeting. *Clearly, we must make every effort possible to find out what the customer wants, and we must do it continuously and endlessly.*

Where does all this lead us? We conclude that *the customer demands on cost, on quality, and on product, are inescapable*—we must respond to them, or we most certainly will fail. But responding alone will not assure our success . . . we must couple these necessary but insufficient efforts with our individual and our collective imagination, creativity, job commitment, and energy. Only our best will be good enough. It is safe to say the opportunity now is greater than ever. The future is there to be taken—all we have to do is take it.

James Vollmer
Group Vice-President

RCA Engineer

Vol. 28|No. 1 Jan.|Feb. 1983

productivity applications

4 Enhancing engineering productivity in GSD
E.E. Moore

11 The computer: An engineer's toolbox
R.P. Dreher

18 APL and engineering productivity
J.G. Solomon

25 Producibility—The road to increased productivity
R.F. Gimigliano

29 CAD/CAM at RCA Laboratories—
Tools and applications, present and future
K.S. Reid-Green|W.A. Hatter|D.C. Quardt

37 Using software tools to enhance engineering productivity
J.S. Heffernan|G.E. Munk

43 A computerized system for gun parts measurement
H.O. Hook|J.P. Beltz|D.P. Bortfeld|P. Kuczer|W.C. Saunders|A. Mavrodin

49 Process control charts to monitor and guide device production
R.F. Morey|R. Kleppinger

56 Steady-state NOdal Thermal Analysis for printed wiring assemblies
T.N. Nguyen|D.B. Wolfe

62 Advanced Technology Laboratories' Image Processing Facility
W.B. Schaming|J.S. Zapriala

general interest

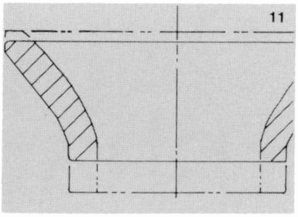
72 The CDP1800 family in intelligent telephone applications
J. Nizet|D. Shand

76 Adherent dust particles
R. Williams|R.W. Nosker

departments

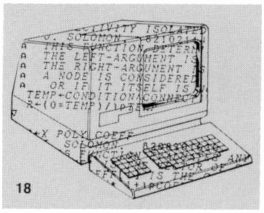
Pen and Podium, **83**|News and Highlights, **85**

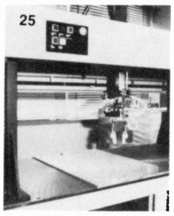
in this issue ...
engineering productivity: tools and techniques



■**Moore:** *"It is hoped that the applications strike a responsive chord in other engineering activities and will be a catalyst for similar improvements."*

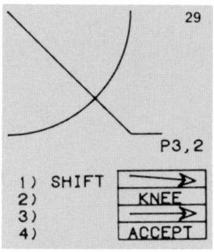
■**Dreher:** *"The software that will be discussed throughout this paper resides on the corporate computer center computers."*

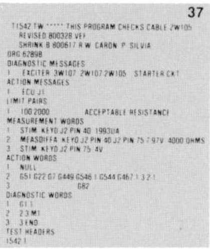




■**Solomon:** *"APL allows the user to accomplish in a few lines of instructions—even in one line—calculations that in other languages would require dozens of statements."*

■**Gimigliano:** *"This team concept has subsequently been extended and applied on all major programs."*



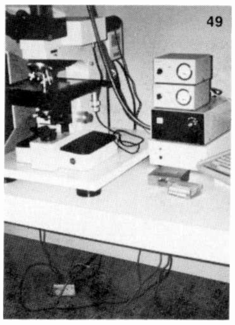


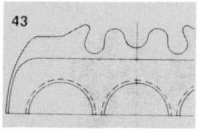
```
 T1542 TW ***** THIS PROGRAM CHECKS CABLE 2W105
   REVISED 800328 VEF
   SHRINK B 800617 R W CARON P SILVIA
ORG 62898
DIAGNOSTIC MESSAGES
1   EXCITER 3W107 2W107 2W105   STARTER CKT
ACTION MESSAGES
1   ECU J1
LIMIT PAIRS
1   100 2000           ACCEPTABLE RESISTANCE
MEASUREMENT WORDS
1   STIM  KEYO J2 PIN 40  1993DA
2   MEASDIFFA  KEYO J2 PIN 40 J2 PIN 75 7 97V  4000 OHMS
3   STIM  KEYO J2 PIN 75  4V
ACTION WORDS
1   NULL
2   G51 G22 G7 G449 G546 1 G544 G467 1 3 2 1
3                     G82
DIAGNOSTIC WORDS
1   G1 1
2   2 3 M1
3   3 END
TEST HEADERS
1542 1
```

■**Reid-Green/Hatter/Quardt:** *"Using the appropriate computing tools, a designer can try more than one idea without resorting to having each idea reduced to a physical part or subassembly."*

■**Heffernan/Munk:** *"In our situation, we developed a compiler, together with associated support software, that is memory efficient and that maps the structure of diagnostic flowcharts directly."*

■**Hook, et al.:** *"We describe here a computerized system designed particularly to measure the surface contour of a metal part to high accuracy (~0.1 mil) at moderate speed."*

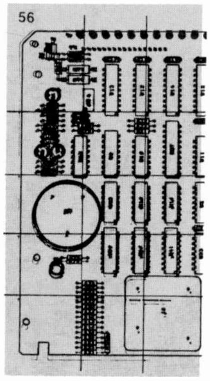


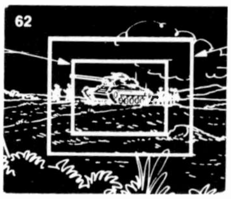


■**Morey/Kleppinger:** *"Although this statistical method of charting has been used in production for less than a year and has gone through several changes, the benefits appear to more than compensate for the effort required."*

■**Nguyen/Wolfe:** *"To maximize engineering productivity by a CAD system, engineers must use the stored database to its fullest capacity."*

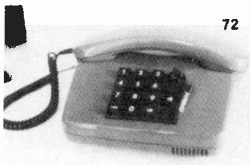
■**Schaming/Zapriala:** *"This facility has been useful for various applications including target detection and tracing, fasimile bandwidth compression, and optical disk demonstrations."*





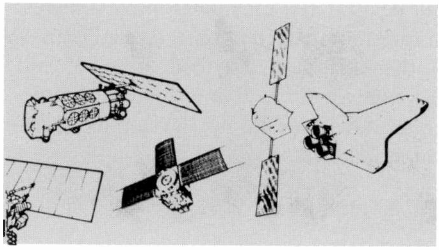
■**Nizet/Shand:** *"The versatility of the CDP1800 family has sold a number of telecommunication companies in Europe on the CMOS microprocessor-controlled telephone design."*

■**Williams/Nosker:** *"Many of these particles are so small that they remain suspended in air for days, but a single one of them, landing on an electronic device during manufacture, can cause its malfunction."*



in future issues...

space technology,
technology transfer/energy,
RCA technology guide



E.E. Moore

Enhancing engineering productivity in GSD

Government Systems Division uses computers to free engineers for more meaningful tasks.

Abstract: *A collection of examples from the four business units at Government Systems Division make up this package on computer aids for the engineer. Short descriptions of recently developed systems and programs are accompanied by a name and phone number, so that readers can obtain more information.*

Examples here from the four business units of Government Systems Division (GSD) illustrate the wide-ranging application of computers to engineering methods and problems, to improve engineering productivity. These examples have all been drawn from experiences during the last two years. Several of them remind us that very mundane engineering functions benefit from the computerized approach. It is hoped that the applications strike a responsive chord in other engineering activities and will be a catalyst for similar improvements. In case more information is desired on any of the examples described, a name and telephone number follows each example. These individuals did not develop the application in all cases.

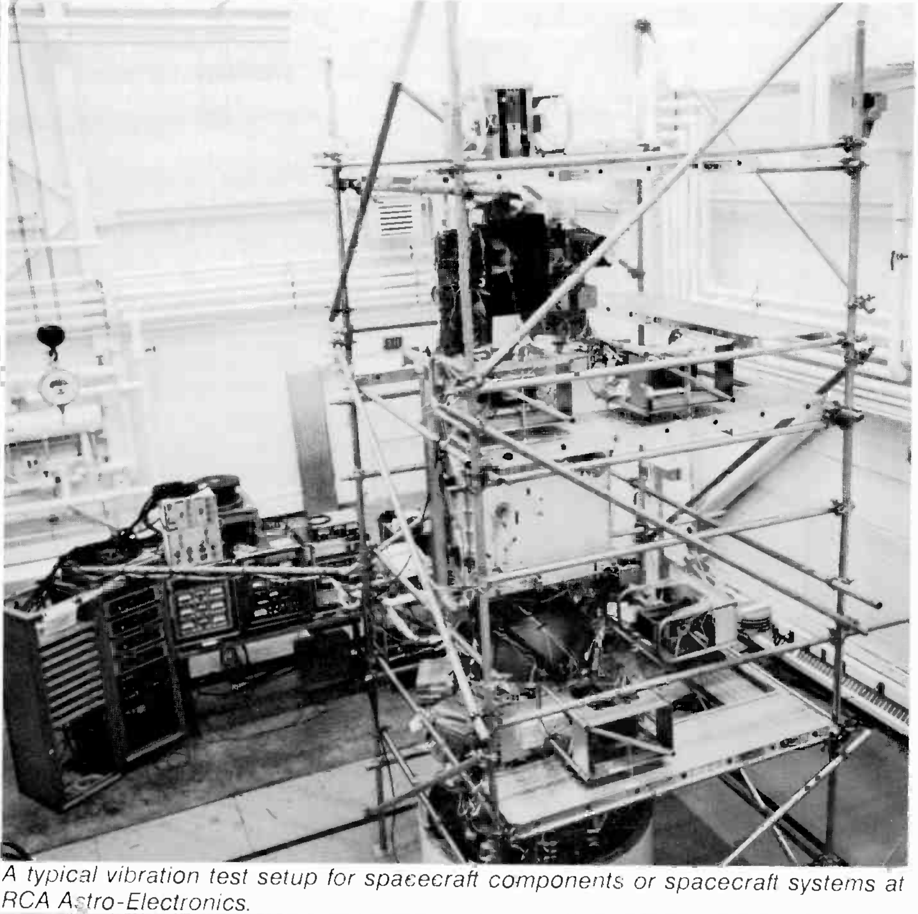
© 1983 RCA Corporation
Final manuscript received November 18, 1982
Reprint RE-28-1-1

Mike Johnson, Manager, Products Engineering, Automated Systems, stands beside the the NOVA 840 used for processing failure data.

Failure data collection

All test events on all products in the factory at Automated Systems in Burlington, Massachusetts, are recorded on Unit Under Test (UUT) forms that are then transcribed into a magnetic record on the Software Center NOVA 840 computer. About 500 entries per day are made. The data can be merged, sorted, and otherwise manipulated to provide insight into failure trends and other production and reliability problems. The database is currently on the order of 50 megabytes. The effort takes one engineer and two clerks. A nonautomated system required two engineers and six clerks, and the data access was far less immediate.

(M.L. Johnson, 326-3745)



A typical vibration test setup for spacecraft components or spacecraft systems at RCA Astro-Electronics.

Signal analysis

An analog signal-analysis system has been replaced in the environmental testing area at Astro-Electronics. The previous four-channel analog system consisted of the necessary electronics to process vibration test data and to plot it on four x-y recorders. The setup and processing/plotting of four channels of data took approximately forty minutes. Hence, the processing of fifty channels of data from a typical three-axis spacecraft vibration test would take approximately 100 hours or 14 shifts. In addition, the manual setup and operator interaction was prone to the usual human errors, and engineers would not receive the test data until at least a week after the test. In 1981, the analog system was replaced with a 16-channel digital system based on a DEC PDP 11/34 minicomputer.

This system parallel processes 16 channels of data and plots serially to a line printer/plotter. Setup and processing of 16 channels now takes approximately one-half hour and the time for data processing from a typical spacecraft test now takes less than one shift. Engineers can examine test data in nearly real time as it is processed, greatly improving efficiency and, more important, enabling the evaluation of test data between each phase of the test.

(J.M. McClanahan, 229-2392)

Carl Voorhees, Senior Member Technical Staff at Astro, checks the setup for modal testing of a K-band reflector.

Modal testing of structures

At Astro-Electronics, computers have improved efficiency in structural analysis through modal testing. Modal testing is a process in which the fundamental governing dynamic characteristics of a structure are experimentally determined. The process involves the acquisition, complex manipulation, and analysis of large quantities of response data acquired during artificial excitation of the structure. In the past, the cost of modal testing was prohibitive due to the time required to perform and analyze data from such a test. Modal tests were therefore, at best, limited only to large developmental programs, primarily in the aeronautical areas. But with the advent of small microcomputer systems, the techniques for modal testing and data processing—and the speed with which the tests can be performed—have made great advances.

As a result, modal testing of satellite structures and substructures has become a feasible and cost-effective tool used during the development cycle for understanding and improving the dynamic performance of lightweight spacecraft structures. The identification of structural resonances and the understanding of a dynamics structure is key to solving many design problems efficiently and cost-effectively.

(C.R. Voorhees, 229-3494)

Functional simulation of a new computer

Implementing a complex architecture design using VLSI technology can be very time consuming and expensive. Any design errors at the functional level, if not discovered early, will be extremely costly at the later development stages. Therefore, the design must be verified at the functional level before entering the equipment fabrication and test phases. One alternative is to breadboard the design using off-the-shelf components. This is expensive and time consuming. For more complicated architecture designs, functional simulation seems to be the most efficient and most practical approach.

Missile and Surface Radar has implemented a microprogrammable CPU architecture, executing the MILSTD-1862A (Nebula) instruction set under the Military Computer Family (MCF) project. The architecture employs a horizontal microcode structure, driving numerous parallel hardware functions to achieve a throughput performance approaching 3 millions of instructions per second. The architecture design consists of the detail specification of the CPU hardware, and the implementation of the firmware needed for executing the target instruction set on the machine. On the hardware side, the architecture design is mapped into five unique VLSI chip types, each of which is specified in detail at the functional level. The firmware implementation uses 4K words × 128 bits of microprogram to execute the standard Nebula instruction set.

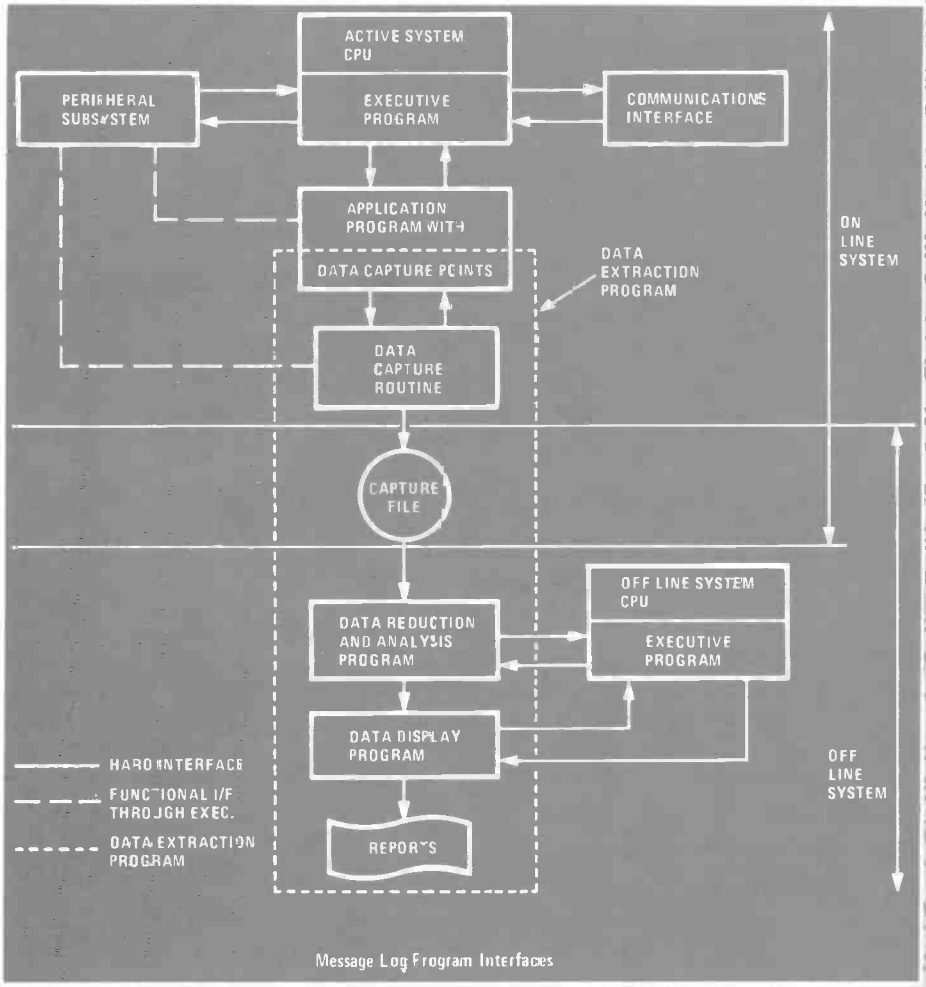
In our design and development process, we found that the network microprocessor (N. mPc) functional simulator and the associated instruction set processor' (ISP') hardware descriptive language (developed by Case Western Reserve University) was an excellent design aid. The figure on the next page shows the top-level block diagram of the N.mPc system. The register-transfer-level (RTL) descriptions of the computer architecture, using the ISP' language augmented with timing specifications, serve as a complete functional specification of the hardware design. The hardware implementation, firmware generation and verification via simulation can therefore proceed in parallel with the detailed hardware design and development based on these specifications. The architecture RTL modeling, detailed microcode development, and functional system validation required only three man years of development for the MCF project using this methodology.

The effectiveness of this methodology has been clearly demonstrated on the MCF project by the fact that several errors in the functional design were uncovered early in the simulation effort and were transmitted to the hardware designers in time to circumvent costly wire changes. Also, the debugging of the firmware proved invalu-

Message logs

In some complex real-time message systems, manual logs have been kept to determine how many messages have been handled, a gross time-in/time-out log has been kept, and because of the impact on occupying operators, most other records have been minimal. Government Communications Systems has developed a data-capture system. It is a software probe that collects data as a message flows through the system. This system is much more complete than the manual system because, without affecting the operator, it collects data on entry/exit, on delays between various operations, and on corrections or rejections due to traffic errors; it categorizes various message types; and it prepares a summary report for off-line analysis. This system, which was developed for Bulldog, an ELINT system, not only reduces time and effort by the operator, but allows us to get much more insight into system bottlenecks, with little (less than 1 percent) impact on throughput.

(I.N. Susskind, 222-3986)

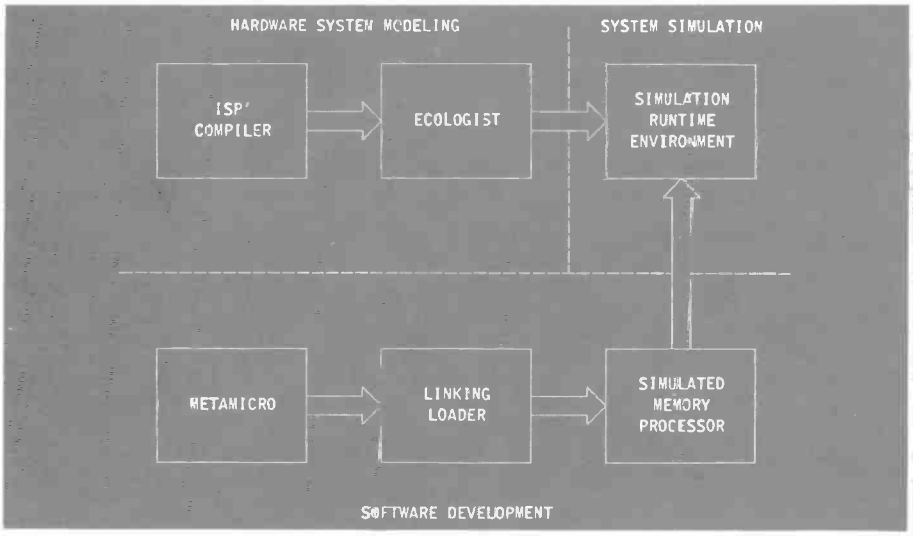


This flowchart shows the interface of the message log system with the overall message system.

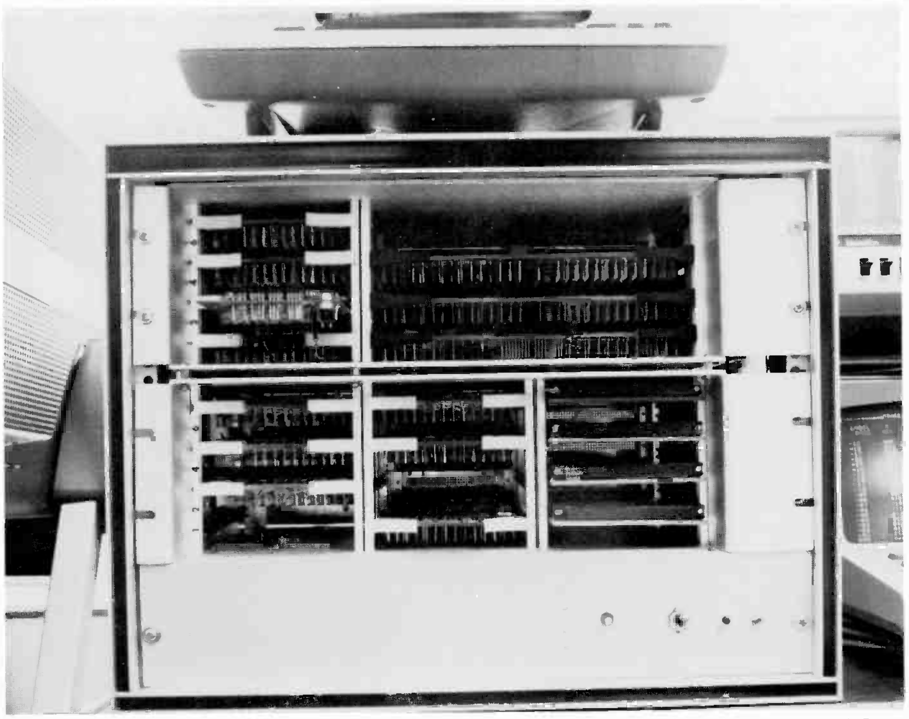
able because numerous errors were uncovered during the simulation phase by running the test programs on the simulator. The resulting code was virtually errorless when installed on the advanced development model of the CPU. For example, only two additional firmware errors were uncovered during the basic arithmetic and logic instruction-set diagnostic program, which test in excess of 1000 lines of microcode using over 1500 lines of Nebula test software.

The current N.mPc system operates under the UNIX operating system and is installed on a PDP-11/70 minicomputer. As indicated above, this system has proven to be an excellent design aid. However, as greater portions of systems are committed to custom VLSI designs, the N.mPc design approach will become an essential part of our design methodology. Recognizing this requirement, Missile and Surface Radar is funding Case Western Reserve University to significantly enhance N.mPc. The most significant enhancements will include converting the system to operate under the VMS operating system, on a VAX-11/780. This conversion will significantly improve functional simulation capability by providing reduced computer-execution time during simulation and greater modeling capacity.

(R. Cheng, 224-2752)



A top-level block diagram of the Case Western Reserve University N.mPc functional simulation system.



A high-speed, 16 bit, RCA Control Processor (RCP) developed at Missile and Surface Radar using the DOD standard instruction set.

Verification of new computer design

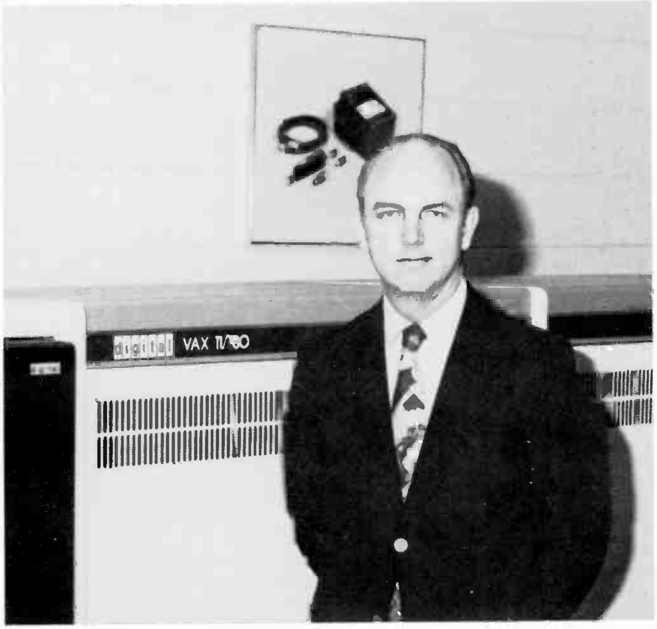
At Missile and Surface Radar many of the systems developed require embedded computers. In response to this requirement, a high-speed, 16-bit computer, known as RCP (RCA Control Processor), has been developed. In the development, the choice of an instruction set must be made. This choice is essentially between specifying a unique instruction set for the computer or designing the computer to execute an existing set. In either case, the designer must be concerned with the problem of verification of the computer's ability to execute the instructions exactly as specified. This is no small problem since all possible results of all possible instructions must be tested.

Fortunately, the Department of Defense has solved this problem for the designer of military computers by specifying a standard instruction set and providing a test program that exhaustively tests such computers. This is contained in MIL-STD-1750A.

MSR has chosen to design the RCP to execute this standard instruction set at least in part using the supplied acceptance test program (ATP), which contains over 125,000 lines of code. The ATP can be loaded into the RCP via a PDP-11 and used to verify the performance of the RCP. The ATP contains a small executive allowing the operator to control the test via an interactive terminal connected to the RCP, allowing selection of the instructions to be tested. Detailed error messages are generated. These messages tell the designer exactly where the design deviates from the correct instruction execution.

By electing to design its RCP to the MIL Standard, MSR has been able to exhaustively test over 220 instructions using the government-supplied software and to avoid the costly and time-consuming problems that occur when hardware designers turn over a computer with hardware errors to the applications programmers.

(T.J. Forquer, 224-3059)



Ted Kupfrian, Senior Project Member at Automated Systems, explains the Logistics Support Analysis (LSA) system, which operates on a VAX 11/780.

Logistic support

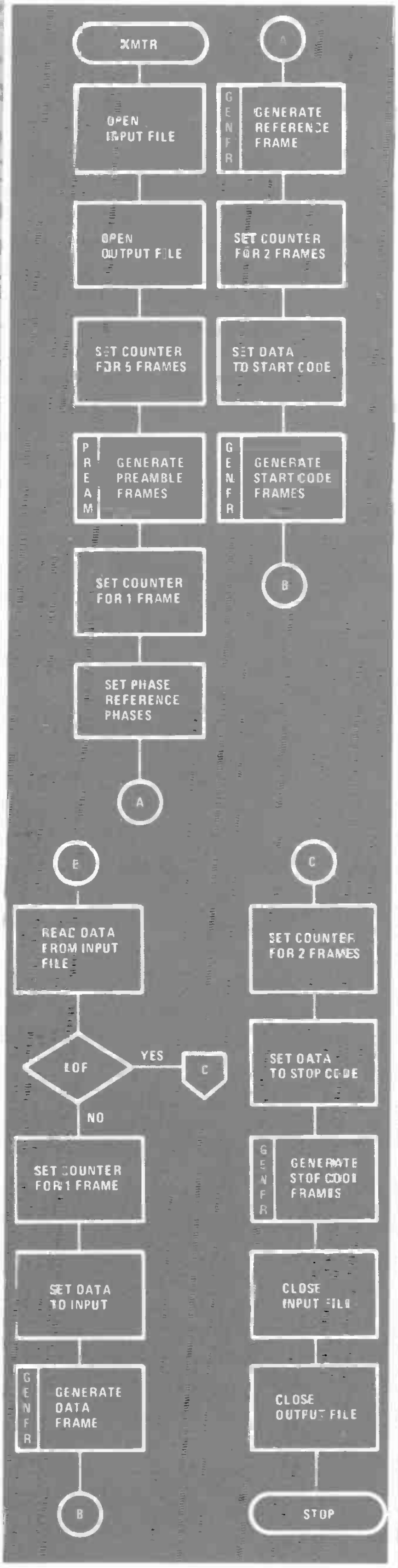
A computerized Logistic Support Analysis system has been developed and placed in operation at Automated Systems. It serves to keep track of thousands of drawings, parts manuals, and other maintenance data items. For Automated Systems it provides logistic support on major programs for the Army, such as EQUATE USM-410 and the STE products, and it complies with Specification MIL-STD-1388. An advantage of the RCA-developed system is the flexibility provided to the engineer. Via a display-terminal interface, and the RCA-developed software, the engineer is able to enter data updates immediately, including changes to maintenance instructions. Previously this was not possible with keypunched computer programs or other procedures.

The system provides the following advantages of:

- A two-to-one cost saving when running in accordance with the Government specification.
- A very large improvement in error rate.
- Lease of data pickup/transfer from program to program.

The system runs on a VAX 11/780. It is in the forefront of industry's drive to automate logistic support analysis and has gained recognition, with the users, for its efficiency and dependability.

(M.L. Johnson, 326-3745)



The flowchart of the transmitter portion of the modem simulation program. A similar flowchart exists for the receiver portion.

Modem simulation

Government Communications Systems has been under contract to the Navy for over a year for modeling and simulating details of existing Link 11 (RF) modems; these are the AN/USQ-74 and AN/USQ-76. This entailed modeling these designs by determining the performance of important elements in the modem and then generating meaningful stimulus algorithms. The results have compared favorably with actual field results. This now means that further tests will not require the expense of running tests on aircraft and ships (although some verification testing in the field is still essential). In addition, design elements in the modem circuitry can be changed within the simulation to quickly determine their effect on modem performance, prior to physical modification. This allows us, and the customer, to consider using the simulation as a modem "test bed," and it therefore becomes a valuable "tuning" tool.

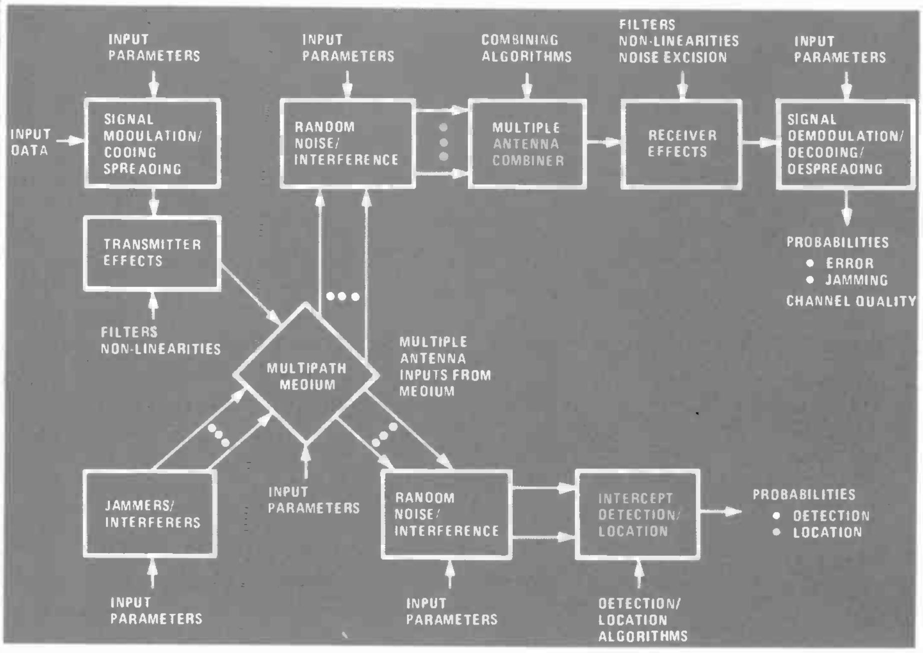
(I.N. Susskind, 222-3986)

RF link

End-to-end RF-link analysis has historically been done manually in Government Communications Systems. The work is time consuming. Either the results are parametric and therefore difficult to use when trying to relate to multiple conditions (propagation, dynamic noise, antenna performance, and so on), or they reflect single-point performance. Through the use of computer-based modeling, we are now able to incorporate algorithms which, although not executed in real time, allow examination of RF-link performance with complex, multiple-occurrence, real-time-related, stresses.

Therefore, the use of computer programs (some internally designed and some purchased) greatly magnifies our ability to handle complex models. Through judicious use of run-time to real-time scaling, programs help us to relate performance to time-related perturbations in the RF media, input signals, modem performance, antenna and coupler response, and so on. It also means that once such a model and simulation environment has been created, the time consumed by engineers is reduced dramatically for future tests.

(I.N. Susskind, 222-3986)



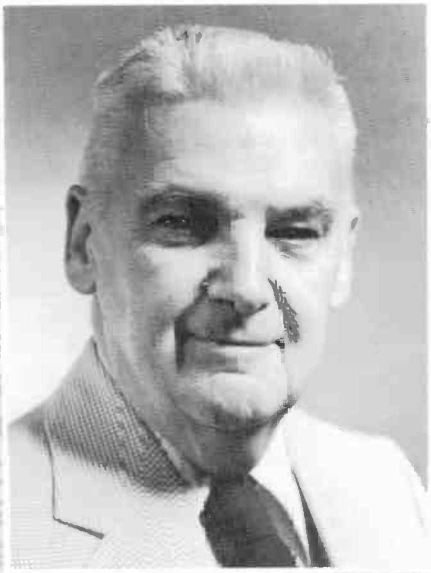
The block diagram illustrates RF Link simulation being used by Government Communications Systems.

Kathleen Cosgrove, Logistics Support Analyst, Automated Systems, inputs data for software problem control.

Software problem control

Automated Systems has developed and implemented a computer-aided software problem reporting-and-control procedure for the Technical Control Analysis Center project. A database is maintained that provides for entry of software problems identified anywhere (in the field or in the plant), problem tracking, problem fixing, and weekly updated status lists. A typical week's activity shows over 200 problems reported, 50 software fixes awaiting management approval, 50 problems waiting for software changes, and over 100 software actions closed. This system has proven invaluable in maintaining control of an enormous amount of newly developed software on this large Army program. In addition to the increased efficiency at RCA, it has been invaluable in keeping the customer current on project status and assures that all parties are working from "one set of books."

(H.L. Fischer, 326-2138)



Ed Moore joined RCA in 1942 and after five years in manufacturing, transferred to engineering where he has held a variety of managerial and administrative posts. He is currently the Administrator, Engineering Staff Operations in Government Systems Division, a position he has held since 1970.

Contact him at:
Government Systems Division
Cherry Hill, N.J.
TACNET: 222-5833

R.P. Dreher

The computer: An engineer's toolbox

Direct conversations with the corporate-computer-center users form the basis for this survey of software availability and applications.

Abstract: *A computer alone does not provide any inherent benefit.A computer is simply a machine that is used to store and execute software. This software is developed to help solve an assortment of problems that are not easily solved manually. This article portrays the computer as a toolbox that can only provide assistance when it contains the proper tools (software).*

The age of the information explosion is upon us. The engineering community is affected by this explosion as much, if not more than, the business community. Technological advances have allowed our scientists and engineers to gather more and more information from the experiments they perform. The task of sorting through the data becomes harder as time passes. The computer has both added to the problem and provided the basic tool to help solve the problem.

The problem of voluminous output is only one of the problems that face the engineer or scientist. Because of technological changes and the desire of everyone to design and produce more efficient products that add to the users' productivity, solutions that were accepted yesterday no longer satisfy today's demands. Old methods, although they may still work, soon become outdated and obsolete in today's society. The computer can be used to help solve these types of problems. This article portrays the computer as the toolbox of the engineers and scientists. The software that resides on the computer provides the engineers and scientists with the tools they need to write programs and design applications that will *help* solve their day-to-day problems. I stress the word *help* because the computer is not the answer to all problems. The software that will be discussed throughout this paper resides on the corporate computer center computers. These computers are at Cherry Hill and Somerville, New Jersey. The philosophy presented here applies to microcomputers and minicomputers as much as it does to the large IBM mainframes that are an integral part of corporate computer center. For more information on the corporate computer center you may refer to the *RCA Engineer* (Vol. 26, No. 5).

Computers are means to ends

The concept of the toolbox and its tools resulted from my conversations with several engineers and scientists. In addition, because much of my time at RCA has been spent in helping the users of the corporate computer center in the sometimes frustrating battle between themselves and the software residing on the computer, I have gathered information over the years. I have come to one basic conclusion about the use of the computer by noncomputer scientists; that is, the computer is only a means to an end. This philosophy was stated most clearly in a conversation I had with Roger Crane of the Applied Mathematics and Physical Sciences Research group at Princeton Laboratories. Dr. Crane described three levels in the interaction of engineers with the computer. First, and most important, we have the real-world problem, the problem the engineer has been designated to solve. Second, is the application program which an engineer may use to help solve the problem. Third, is the computer software (the "tools" as they will be referred to in this article) that are used to develop the applications.

The software

I spoke to several users of various software packages. Some of the software mentioned may be unfamiliar, so a brief description of each of the software tools mentioned in this article is given in the sidebar, page 12. In general, the computer is used to solve a wide assortment of problems that fall into the following general categories: data gathering, data storage, data manipulation, data analysis and data display.

The answers to what software is being used are as diverse as the answers to why a particular tool is chosen. The whys range from familiarity to the idealistic reason—appropriateness. Both of these reasons are valid. A primary example is FORTRAN. FORTRAN is often used to solve problems that are more efficiently solved by other computer languages. Because FORTRAN is the language of choice of a vast majority of engineers, those applications developed in FORTRAN become more portable and modifiable. However, the software of choice may have some inherent inefficiencies that require alternate means to be researched. It is, therefore, important to have a wide range of tools accessible as the problems get more sophisticated and require more accurate answers. I expected to hear about tools such as FORTRAN, SAS (a statistical analysis package),

Final manuscript received November 9, 1982
Reprint RE-28-1-2

FOCUS (a database package) and DISSPLA (a series of graphical subroutines), and I did, but I heard much more. Conversations sometimes strayed to less common tools such as FLECS (a FORTRAN preprocessor), SPITBOL (a string processing language) and NAG (a series of mathematical subroutines). I also had a long coversation about the benfits of APL, a language which is the topic of another article in this publication. These are just a few of the tools that are being used throughout the corporation; it is the availability of such tools that adds to the productivity of both the writers of the applications and also the users of those applications.

Software tools used at RCA

APL: A Programming Language is an extremely powerful, advanced language that uses a concise notation to define and execute complex procedures that would entail a much greater programming effort in other languages. APL uses common mathematical symbols with which the user can imitate and evaluate expressions in "calculator mode." APL is particularly well suited to matrix and array operations, but also has file-handling capabilities. The system includes a series of "public libraries" for carrying out a wide variety of applications.

DISSPLA: DISSPLA is a series of FORTRAN subroutines that can be used to create professional quality graphics through calls from a user's application program. DISSPLA will generate output on a great number of graphics devices, including monochrome and color CRTs, pen plotters, electrostatic printer/plotters and Computer Output Microfilm (COM) recorders.

FLECS: FLECS, the FORTRAN Language with Extended Control Structure, is both an extension to FORTRAN and a preprocessor to implement the extension. The extension allows the user to do structural coding that is not available in FORTRAN IV.

FOCUS: FOCUS is a comprehensive information management system that affords fast and easy creation and maintenance of, as well as reporting from, databases. Its language is mostly nonprocedural. FOCUS features an English-like query language, file structures, data encryption, sophisticated security, terminal and plotter graphics.

FORTRAN: FORTRAN (short for FORmula TRANslator) is a high-

Conversations with users

My belief that the availability of the proper tools adds to the productivity of those who both indirectly and directly use them can best be explained by relating a few of the conversations I had with users of the corporate computer center and its associated software. My first conversation was with Roger Crane. Dr. Crane makes use of several of the software tools residing at the corporate center. We mainly discussed the use of FORTRAN, FLECS, IMSL, NAG and DISSPLA.

Problems people can solve

Our conversation digressed from the specific uses for each of the software products to the reasons why these tools are needed. The reason is relatively simple—people formulate problems they feel they can solve. Thus the types of problems are limited by the solution techniques available or perceived to be available. This is not to say that problems do not exist that are not solvable with today's technology. The key here is that technological problems being resolved today would not have been investigated yesterday, simply because there were no feasible solution techniques. Products such as IMSL and NAG have made feasible the investigation of problems that a few years ago either would not get solved in a timely fashion or for which the best solution obtainable with more primitive tools would be inadequate. Examples were discussed in the area of optimization. A generalized optimization routine, if written to solve a particular problem, might take upwards of a year to write. A project that would require such a routine may never get undertaken. Tools such as IMSL and NAG provide optimization routines and thus allow the problem to be investigated.

Much of Dr. Crane's work is done in FORTRAN, but even that tool is sometimes not enough. Many times it is too difficult or too time consuming or too frustrating to write an application in the language of choice and a decision must be made between either learning another language, not writing the application, or finding something that will make the language of choice a more suitable tool. For Dr. Crane, FLECS (a FORTRAN preprocessor) provides a tool which allows him to continue to develop his applications in FORTRAN and to do them in the style and time frame required.

Graphics is a hot topic

Graphics, especially the area of computer-generated graphics, is a very hot topic these days. The reasons are simple: The cost of graphics hardware has come down tenfold in the past decade; a one-page graph can display as much information as a multiple-page tabular report; and graphics software has become more sophisticated and easier to use. In the past, computer graphics was not used to present data for much the same reason that optimization problems were not undertaken—there was too much effort required to get a reasonable payback on investment. Packages such as DISSPLA have made feasible the use of graphics in many applications. Graphics are now being used in engineering in much the same way as they are being used in business—to replace tables of numbers. Information feedback is the key. Dr. Crane has made heavy use of graphics in the application programs he has written.

level programming language particularly adaptable for scientific and engineering applications.

IMSL: IMSL, the International Mathematical and Statistical Libraries, is a set of callable subroutines written in FORTRAN. IMSL provides routines for a variety of mathematical and statistical analysis including differentiation, interpolation and smoothing, regression analysis, nonparametric statistics, generation and testing of random numbers, eigensystem analysis and linear programming.

NAG: The NAG (Numerical Algorithms Group) Library is a comprehensive collection of algorithms used for solving numerical problems on computers. These algorithms have been incorporated into a series of FORTRAN subroutines devoted to several branches of numerical analysis that are based on the Association for Computing Machinery (ACM) modified SHARE classification index.

SAS: SAS, the Statistical Analysis System, is a high-level language for data manipulation, data description and display, descriptive and inferential analysis, graphic presentation, and report writing. Additional procedures cover the areas of nonparametric statistics, econometrics and time-series analysis.

SPITBOL: SPITBOL is a compiler version of the SNOLBOL-4 language. SPITBOL is a high-level language that is particularly useful for string and character manipulation, offering an extraordinary number of built-in functions for these purposes. SPITBOL is extremely useful for large-scale file editing and may be used for more unusual applications such as compiler design.

The tools and topics we have discussed so far have been related to mathematical problem solving and data display. Two sidebars (pages 14-16) show how the computer, and specifically a tool called SAS, can be used to solve a variety of data-analysis type problems.

Data management

One of the biggest if not the biggst area for which computers are used is data management. Data management is most common to business organizations, but engineers and especially engineering management have as much a need to have good data management tools available to them as any business-management information-services group (MIS). I visited Linda Bowker of RCA Astro-Electronics to inquire about the tools that Astro is using to manage its engineering projects. Our conversation again turned away from the specific software tools to center on how software tools can make the difference in the feasibility of completing projects. Ms. Bowker strongly believes in using the tools that are best suited for the project. However, if an application is going to be transferred from person to person, then the best suited tool may be one that is known by all those involved. Although some other languages may have some capabilities that FORTRAN does not have, FORTRAN is known by a vast majority of engineers.

Our conversation centered on the RCA Astro-Electronics Engineering Load Program. This is a system that is used by engineering management to keep track of engineering projects and the time spent by the engineers on those projects. The Engineering Load Program uses FOCUS as its basic tool. Ms. Bowker noted that the project would not have been attempted without a database-management package. FOCUS was chosen as the database-management system because of its compatibility with CMS (the resident time-sharing system at the corporate computer center) and its relatively easy initialization and report generation. Although FOCUS is an integral part of the program, we spent more time discussing a little-known software language called SPITBOL. Ms. Bowker stated that SPITBOL is used as a general-purpose preprocessing language. Such a tool is required because data hardly ever resides in the form that is needed. SPITBOL's strong string manipulation capability allows you to extract the required information and tranforms it into the desired format.

APL is a powerful language

Most of what has been said about data and numerical analysis has all been based on FORTRAN and FORTRAN-related tools. FORTRAN is not the only way an engineer can accomplish his or her work. In fact, the existence of other means to solve the problems has made tasks of many much easier. I visited John Grosh in Lancaster to talk to him about his use of APL in an engineering environment. John makes extensive use of APL in developing quality and reliability systems for Solid State Division. John is an avid proponent of APL—he "doesn't do FORTRAN or windows."

John states that APL is a conceptual language. By this he means that when you think about a problem you think of things such as "add all of these to all of those." APL has operators that will perform that exact operation, whereas most other languages do not. As I heard in several of my other interviews, John started using APL because it allowed him to do things that he would not have been able to do in any other language. The main application we discussed was MAPS (Manufacturing Attribute Performance Summary), an application written solely in APL. John stated that he would not do it in any other language. He did not say it could not be done using other tools, but the feasibility would be extremely reduced. More information on the MAPS application can be obtained by referring to the April/May 1980 issue of the *RCA Engineer.*

(Continued on page 17)

SAS used as a design aid for picture tube yokes

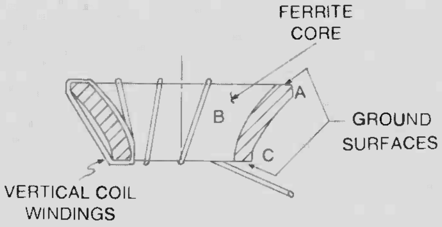
User comments by Paul Bruey

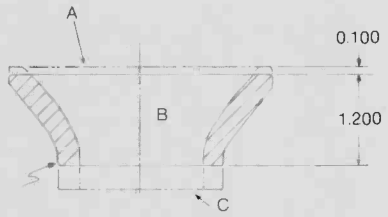
The picture tube deflection yoke basically consists of two coils of wire and a ferrite core to position the three electron beams coincidently on the phosphor screen surface via the magnetic field generated by the coils. One coil deflects the three beams horizontally and the other deflects them vertically. The vertical deflection coil is wound about the ferrite core in a specific fashion with wire placement a function of core geometry.

It was hypothesized that irregularities of surfaces A, B, and C in production of the cores would result in improper alignment of electron beams on screen surfaces. To test this theory, 200 cores were produced and divided into 4 categories of 50 each. Category 1 had the upper edge (surface A) ground smooth with a diamond bit, category 2 had surface B ground smooth, category 3 had surface C ground, and the fourth category remained unground.

Coils were wound on each of these cores and beam separation at specific points was measured. It was found that the beam triad convergence significantly improved when the ground cores were used. This finding prompted a decision to produce cores with surfaces A and C ground. Today, however, all these surfaces are ground to within design specification before vertical deflection coil winding is undertaken.

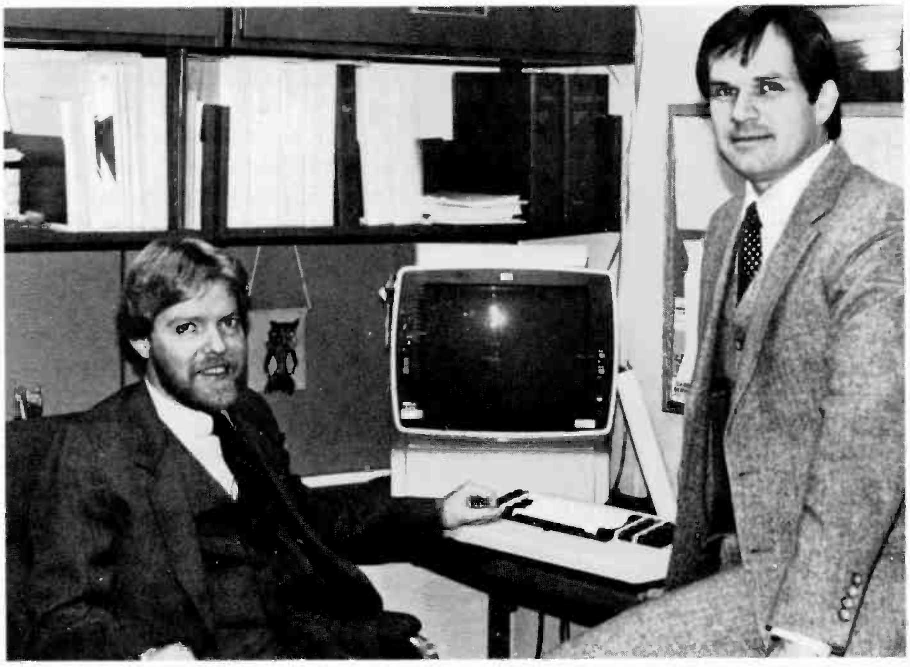
SAS was used because of its sophisticated data-input and manipulation techniques as well as its effective procedures to complete the analysis. Mr. Bruey feels that without the advantages offered by SAS, he would not have been able to complete the project.





Core grinding experiment. An experiment investigating the effects of ferrite core grinding upon the precision of toroid wound coils extensively employed SAS routines for data compilation. Vertical deflection coils are a major component of a television receiver deflection system. The vertical coils are toroid wound by machine on a ferrite core. On the same winding machines, samples of end ground, and unground ferrite cores were used in the manufacture of vertical coil windings. The vertical coils were then tested on a CRT to determine their distribution of raster convergence errors. Use of SAS routines resulted in the production of mean, standard deviation, variance, data plots and group frequency distributions of each set. A review of the SAS data reduction showed that a lowering of the manufacturing reject rate was achieved by end grinding the ferrite core before winding the coils. The SAS routines used significantly reduced compilation time and limited computational errors to the data input phase. Recommendations based upon the ground ferrite core experiment proved to be cost effective in actual manufacture.

—Produced by Charles Sockwell



Author Dreher (left) with Charles Sockwell.

Roger Dreher is Manager, Product Support, Corporate Information Systems and Services (CISS). His group provides end-user support for the software residing in the computers at the corporate computer centers. He joined RCA in 1978 as an instructor in the CISS Education and Publications group. He has an MS in Computer Science from the University of Rhode Island.

Contact him at:
RCA Corporate Information Systems and Services
Cherry Hill, N.J.
TACNET: 222-4264

SAS used for VideoDisc test instrument quality control

User comments by Karen Pitts

In the ongoing effort to improve disc quality and performance, a variety of materials and production techniques are tested to determine their characteristics. Among the sources of error present in testing new discs is that associated with the test equipment itself. It is possible, for example, for this equipment to malfunction and, if undetected, to be attributed to disc problems. For this reason, test equipment is checked daily, the results of several months of data are plotted, and the plots are visually examined to determine if equipment performance has perceivably changed. The plots have time-dimensioned x-axis and signal strength measured along the y-axis. The data mean and three standard deviations above and below the mean are calculated and plotted along with the data.

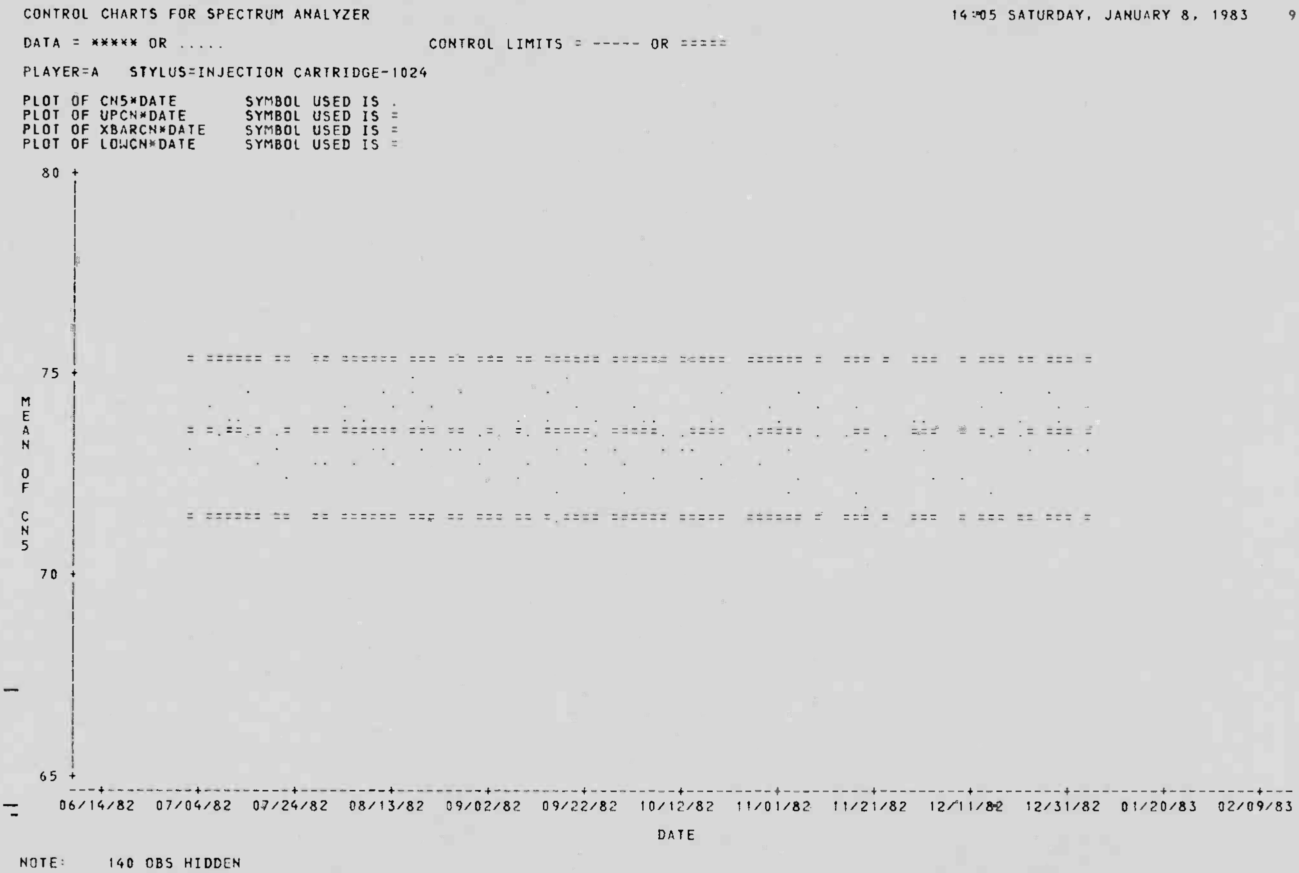
To illustrate, some hypothetical data were run through the program, giving the results below. Plot 1 shows data from equipment that is rather steady over the testing period. Plots 2 and 3 show data that seem to be behaving more erratically during the end of the testing period.

We cannot determine what effects these test equipment anomalies may have had on disc testing. The researcher may, however, choose to exclude results from the equipment used in plots 2 and 3 from his/her experiment.

Originally, this type of analysis was done using programs written in APL. In this instance it was felt that using SAS has saved considerable programming effort.

—Produced by Charles Sockwell

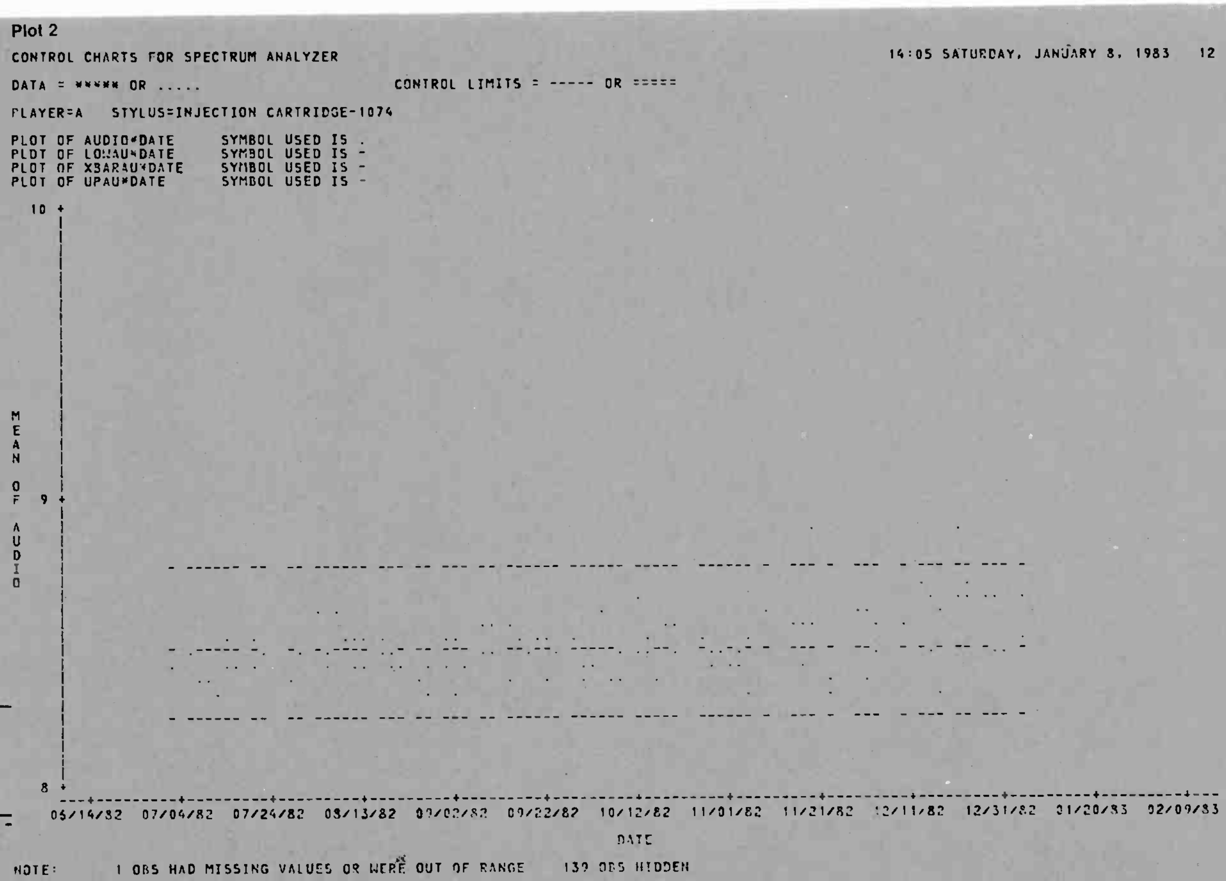
Plot 1



```
CONTROL CHARTS FOR SPECTRUM ANALYZER                                              14:05 SATURDAY, JANUARY 8, 1983    9
DATA = ***** OR .....                 CONTROL LIMITS = ----- OR =====
PLAYER=A   STYLUS=INJECTION CARTRIDGE-1024

PLOT OF CN5*DATE       SYMBOL USED IS .
PLOT OF UPCN*DATE      SYMBOL USED IS =
PLOT OF XBARCN*DATE    SYMBOL USED IS =
PLOT OF LOWCN*DATE     SYMBOL USED IS =

NOTE:    140 OBS HIDDEN
```

Testing equipment yields consistent results.

Plot 2

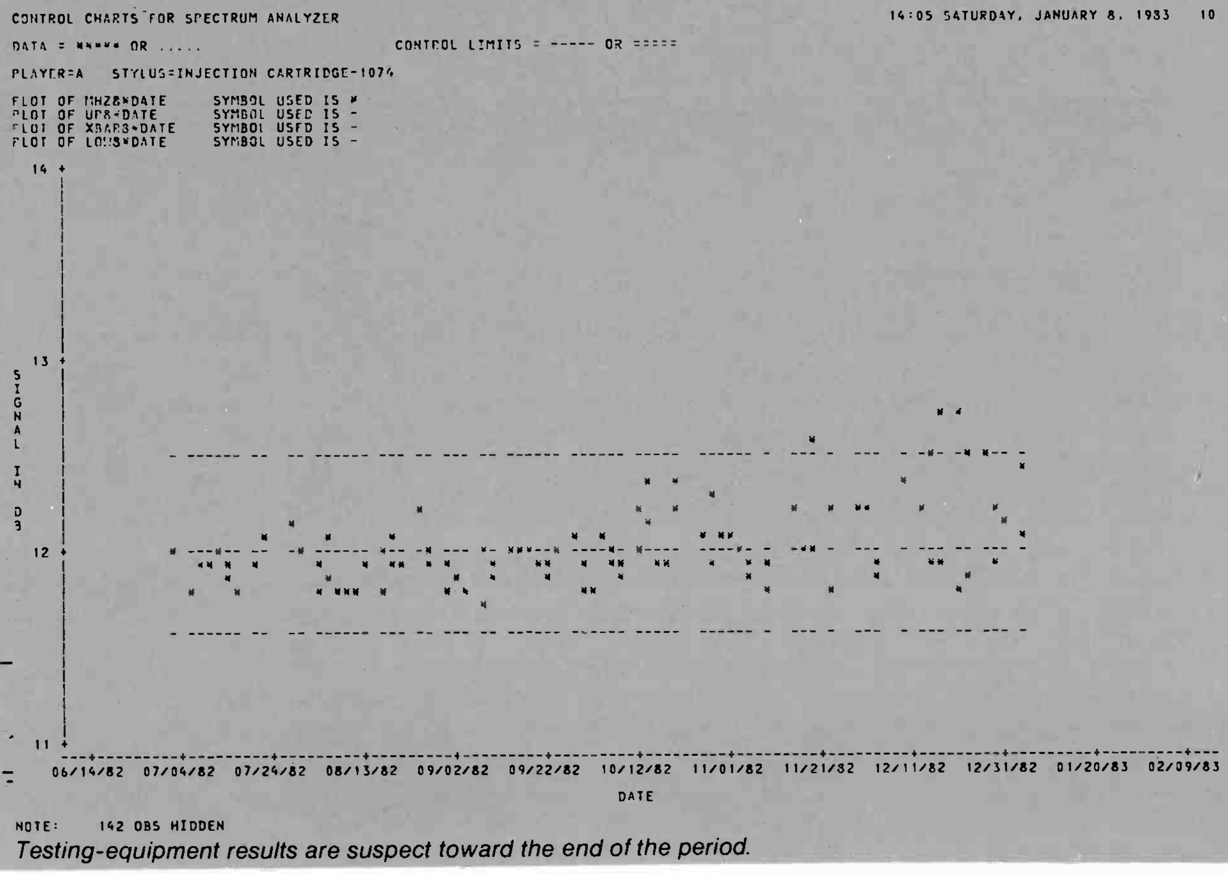


```
CONTROL CHARTS FOR SPECTRUM ANALYZER                                14:05 SATURDAY, JANUARY 8, 1983   12

DATA = ***** OR .....                     CONTROL LIMITS = ----- OR =====

PLAYER=A   STYLUS=INJECTION CARTRIDGE-1074

PLOT OF AUDIO*DATE      SYMBOL USED IS .
PLOT OF LOWAU*DATE      SYMBOL USED IS -
PLOT OF XBARAU*DATE     SYMBOL USED IS -
PLOT OF UPAU*DATE       SYMBOL USED IS -
```

```
NOTE:     1 OBS HAD MISSING VALUES OR WERE OUT OF RANGE     139 OBS HIDDEN
```

Testing-equipment results are suspect toward the end of this period.

Plot 3



```
CONTROL CHARTS FOR SPECTRUM ANALYZER                                14:05 SATURDAY, JANUARY 8, 1983   10

DATA = ***** OR .....                     CONTROL LIMITS = ----- OR =====

PLAYER=A   STYLUS=INJECTION CARTRIDGE-1074

PLOT OF MHZ8*DATE       SYMBOL USED IS *
PLOT OF UP8*DATE        SYMBOL USED IS -
PLOT OF XBAR8*DATE      SYMBOL USED IS -
PLOT OF LOW8*DATE       SYMBOL USED IS -
```

```
NOTE:     142 OBS HIDDEN
```

Testing-equipment results are suspect toward the end of the period.

Specialized engineering software on corporate computer

AFTER helps generate functional test patterns for testing Large Scale Integration (LSI) or Automatic Test Equipment (ATE).

DYNA-3 is a large finite element program suitable for analyzing any type of structure.

LOGSIM aids the logic design engineer in checking and providing design results.

MIMIC is a logic simulation program.

MOTHA is a general thermal analysis program that permits solution of steady-state and transient problem formulations.

NASTRAN is a general-purpose finite-element computer program for structural analysis.

R-CAP is a circuit simulation program which can analyze either the DC operating points, or the transient or small signal responses of bipolar and metal oxide semiconductor (MOS) transistor circuits.

SPICE2 is a circuit analysis program designed primarily for the simulation of integrated circuits, both bipolar and MOS.

SUPER-COMPACT is an advanced, general purpose, high-frequency, microwave circuit design program.

TESTGEN is a test generation program to be used in conjunction with sequential digital logic.

TDAS collects and stores voluminous data generated during the test cycle and produces specialized reports.

(Cont. from p. 13)

Conclusions

The availability of the proper software is the underlying theme of this article. The choice of the proper tools depends on several factors: The type of problem that is to be solved; the type of solution presentation that is desired; and the availability of tools that the problem solver knows how to use or can easily learn to use. Proper use of the computer can be shown to increase the productivity of the users. This productivity gain may show up directly in the time required to solve a problem or indirectly by the solving of problems that previously were thought to be unresolvable.

The use of tools in combination appears to be the best way to achieve the highest rate of return from the smallest investment of time. Historically, most applications were written in a single language such as FORTRAN, and if FORTRAN could not handle the task, the task was either modified or circumvented. FORTRAN is still the language of choice but it is now supplemented by various tools, some of which have been mentioned in this article. The bottom line is that a job cannot be completed satisfactorily if the proper equipment is not available. Quality and productivity suffers when the right tools are not available. This is true if you are building a house or designing a satellite. A computer that does not have the software required by the engineer is of no use to the engineer.

This article has not delved into a very important area with respect to the future use of the computer, large or small, by all of us; that is , the area of specialized software application packages. These packages can be used directly to solve many of the real-life problems if they fit the available information. Corporate Information Systems and Services attempts, whenever possible, to provide information about existing packages either on the system at the corporate computer center or in the marketplace. A sidebar (this page) lists some of the specialized software tools that reside at the corporate computer center. In addition, much specialized software is running in minicomputers and microcomputers throughout the corporation. Much of the information on these products is transmitted through publications such as the *RCA Engineer* and *TREND*.

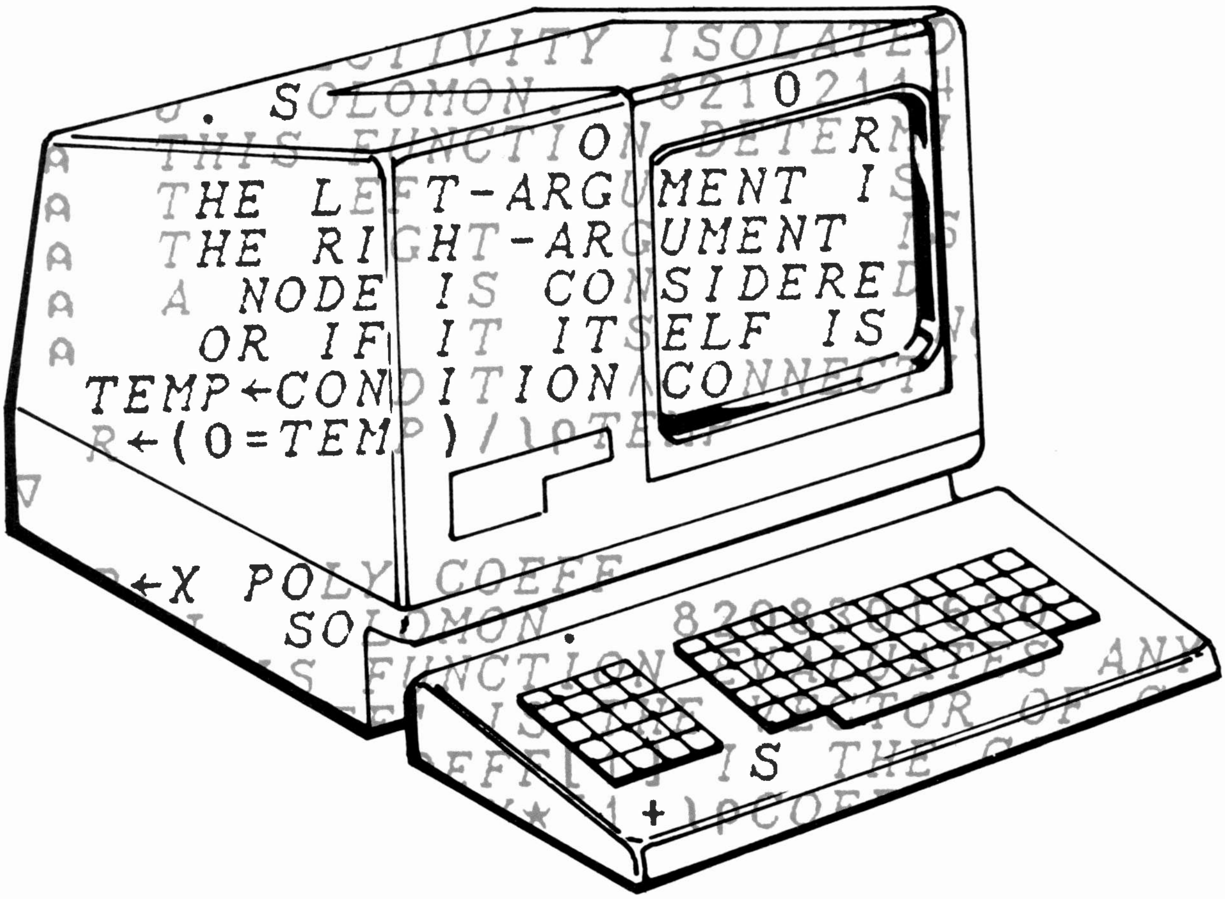
Acknowledgment

I would like to thank Paul Bruey, Roger Crane, and Karen Pitts from Princeton Laboratories; Linda Bowker from Astro-Electronics; and John Grosh from Solid State Division. These people provided the necessary information to write this article. I would especially like to thank Charles Sockwell of Corporate Information Systems and Services in Cherry Hill, for gathering the information and writing the sidebars on the use of SAS for data analysis.

J.G. Solomon

APL and engineering productivity

A basic step-by-step introduction to APL shows how the language can save you programming time, effort, and headaches—even if you've never seen a computer.



Abstract: *APL, a powerful and advanced programming language, uses a unique and concise notation that simplifies programming effort. It is particularly able to handle matrix operations, but has other strengths as well. The author introduces the language and its capabilities, particularly with reference to FORTRAN, using a step-by-step approach and plenty of examples—curve-fitting, string manipulation, etc. Sources of further information round out the article.*

If productivity is the issue, then APL must take the spotlight. APL is an amazingly powerful computer programming language. It allows the user to accomplish in a few lines of instructions—even in one line—calculations that in other languages would require dozens of statements. Besides that, it's fun to use. I have programmed (and taught) FORTRAN, COBOL, BASIC, BAL, PL/I, . . . but I *enjoy* APL.

Dr. Kenneth E. Iverson, the inventor of APL, claims that the ideal audience for APL is people who have never seen a computer, but who know that they have a problem that a computer could solve: engineers, accountants, businessmen, and so on. Being thus motivated—and knowing already what it is they want to do—they quickly learn to use APL. Thus, instead of turning over their problem to the programming staff, and waiting for it to arrive at the top of the backlog stack, they can do it themselves. You can learn, too. You can get information on APL at RCA in the last section of this article.

Final manuscript received November 5, 1982.
Reprint RE-28-1-3

APL is a higher-level programming language, so you don't need to know anything about the computer you're using. Because it is an interactive language as well, you get answers more easily, and you get them immediately. Furthermore, APL can let you easily answer "what if" questions. That is, you can not only write an implementation in APL much more quickly than in other languages, but also you can frequently make drastic changes in an APL program, successfully and in a remarkably short time. Drastic changes in a specification very often require only minimal changes in the APL implementation. You will see examples later. In general, my experience with APL has been that, if you find yourself saying, "Wouldn't it be nice if I could . . . ," you can. Easily.

APL's ability to solve your problems quickly derives from several sources. For one thing, those operations that are built into the language (the so-called "primitive scalar dyadic functions") can generally be used for scalars, vectors, matrices, or variables of even-higher dimension, with no change in syntax: A + B means add A and B (if it's possible), regardless of their size or shape—or contents. In only one statement, APL can perform the same operation on each element of an array—in FORTRAN this would require a DO-loop (or several nested loops). This fact alone tends to make APL programs much shorter than programs written in other languages. There are no declarations of variables, either—the bane of COBOL programmers. APL automatically allots for each variable the amount of space needed for the way you are using that variable.

APL is an interpretive language. This means that there is no compiler to translate the entire program once and generate a permanent "object module," which can be run whenever necessary. Instead, each statement must be read and interpreted each time it is executed. That sounds terribly inefficient but, because there are far fewer statements to interpret, you still come out ahead as compared to executing many statements of code previously compiled in another language. And, as noted, most APL statements are executed only once, instead of being part of a loop that is cycled through many times.

It's so easy!

Some examples at this point may help to explain my enthusiasm—and why I believe that you will become equally enthusiastic.

Suppose that you would like to add up a list of numbers. For our convenience, let us suppose that the numbers are stored in a variable named LIST. In most programming languages, the user must identify by name those variables that have some kind of structure to them, describe the structure, and tell the compiler how many numbers to reserve space for. In FORTRAN, for example, one might write:

```
      DIMENSION LIST(100)
                  .
                  .
      SUMLIST = 0.0
      DO 78 I = 1,100
      SUMLIST = SUMLIST + LIST(I)
   78 CONTINUE
```

Needless to say, it's easier in APL:

```
      SUMLIST←+/LIST
```

pronounced, "SUMLIST is assigned the plus-reduction of LIST." The APL syntax will be explained later.

Note, however, that the APL statement does not include any mention of how many elements there are in LIST: The language-processor knows how big the variable is—why should I have to tell it? The significance of this apparently trivial favor becomes clear when you learn that APL allows you to change the size of a variable dynamically. In other words, LIST may have 100 elements in it now, but I can change that to be 427, if I want. And if I want to add up all the 427 elements, I still write:

```
SUMLIST←+/LIST
```

In FORTRAN, on the other hand, I would have had to make at least one change in the program to add the elements.

Other languages have helpful functions built into them; in particular, several have a SUM function that does the same thing as the above DO-loop. One would write, then, in place of the above FORTRAN DO-loop:

```
SUMLIST = SUM(LIST)
```

and it may appear that the advantage of APL has been overcome. No. Suppose, for example, that I want to add up only the positive numbers in LIST. A head-on comparison of FORTRAN and APL is revealing:

FORTRAN

```
      DIMENSION LIST(100)
                .
                .
      SUMLIST = 0.0
      DO 78 I = 1,100
      IF (LIST(I).LE.0) GO TO 78
      SUMLIST = SUMLIST + LIST(I)
   78 CONTINUE
```

APL

```
      SUMLIST←+/(LIST>0)/LIST
```

Please note that productivity gains like this one are realized in APL every time you write a single statement that in other languages would require an entire loop.

The power and flexibility inherent in APL make it easy to select items from a collection, and to process only those items that match the selection criteria. In addition, "primitive" functions—functions built into the design of the APL language—are not restricted to operating only on scalar variables (that is, those variables that do not appear in a FORTRAN DIMENSION statement), but can also be used, for example, for lists and tables (that is, those variables defined in FORTRAN as being one- or two-dimensional, respectively).

As an example, if I want to double each number in LIST, I write in APL:

```
2×LIST
```

Let us suppose that the variable named LIST is actually a 27-dimensional variable (APL allows more!). How would you multiply each element in LIST by 2? Of course:

```
2×LIST
```

As a more-interesting example, suppose I want to multiply each element of LIST by a different number. Let these multipliers be stored in a list named COEFF. Again, we compare FORTRAN and APL:

FORTRAN

```
        DIMENSION LIST(100),
        COEFF(100),PROD(100)


        DO 78 I = 1,100
        PROD(I) = COEFF(I)*LIST(I)
     78 CONTINUE
```

APL

```
        PROD←COEFF×LIST
```

Without being given special instructions, APL will multiply each element in LIST by the corresponding element in COEFF. Please note that the above APL statement will work as long as both LIST and COEFF are the same length—whatever that length is. Note also that the length can be changed many times during execution of the program; the statement will still work, as long as the changing length of LIST stays the same as the changing length of COEFF. As a matter of fact, APL also allows you to change dynamically the *shape* of a variable. If a design change suddenly makes LIST and COEFF three-dimensional variables,

```
                    COEFF×LIST
```

will still multiply together corresponding elements of LIST and COEFF, as long as they are the same size and shape. If they are of different size or shape, then it is not clear what you want APL to do—differently shaped variables don't have "corresponding" elements.

Let's get practical: Suppose I have a list of quantities ordered, conveniently stored in a variable named QUANT, and a list of prices for the corresponding items, conveniently stored in a variable named PRICES. To find the total cost for each item, I would multiply the quantity of that item times the price of that item. In APL, that is

```
                    PRICES×QUANT
```

and the printed result would be the total cost of each item. If I would like the total price of the order, I write:

```
                    +/PRICES×QUANT.
```

It is interesting to note that the above APL statement to determine the total price of the order can be described in engineering terms as taking the dot-product of the two vectors, PRICES and QUANT.

As with any language, APL allows the user to define his own functions to accomplish some task not built into the language. However, APL is so powerful that frequently it is not worth going to the trouble. For example, it would be silly to write a function to compute the dot-product of two vectors—because, as shown above, all the operations necessary can be stated in three character-strokes.

A convincing example

At this point, a user-defined function may be the best way of indicating the power of APL. One problem I assign to my students of any programming language is the following: Write a program, function or subroutine that has as inputs a short string and a long string, and finds all occurrences of the short string in the long string. In PL/I, this can be done in 15 to 20 statements—although a student of mine once did it in 55 statements. In APL, it can be done in 20 characters. Instead of just blasting you with the syntax, however, I will explain it as we go:

First, invent a short string and a long string.

```
      S1←'THE'
      S2←'WHAT IS THERE AT THE
              ATHENAEUM THEATER?'
```

Just out of curiosity, how long are the strings?

```
      ρS1
3
      ρS2
39
```

The "outer product" applies the specified function between each element of the left argument and each element of the right argument. The dimension of the result is the sum of the dimensions of the arguments.

```
      ρQ←S1∘.=S2
3  39
      Q
000100001000000101000010000000010001000
010000000100000000100001000000001000000
000000000010100000010000100100000100100
```

The result has a 1 where the elements are equal, a 0 where they are not. In this two-dimensional array, a "hit" is signalled by a diagonal of ls. Now if only there were some way I could slide the second row one character to the left, and slide the third row two characters to the left . . . .

```
      ρX←(¯1+ιρS1)⌽Q
3  39
      X
000100001000000101000010000000010001000
100000001000000001000010000000010000000
000000001010000001000010010000010010000
```

In its dyadic form, ⌽ is called "rotation," and it does exactly what we need here. Now, we will "And" the rows together; if all elements in a column are ls, we have found an occurrence of the shorter string.

```
      ρY←∧/[1] X
39
      Y
000000001000000001000010000000010000000
```

Lastly, we use the above vector to compress the numbers 1,2,3, . ., and find out exactly where each occurrence of the shorter string begins.

```
      ρZ←Y/ιρS2
4
      Z
9  18  23  32
```

Now, we put all that syntax into one statement, and don't save any of the intermediate results.

```
      (∧/[1](¯1+ιρS1)⌽S1∘.=S2)/ιρS2
9  18  23  32
```

Space constraints forced us to "fold" some of the statement lines. In these cases we have not inserted a continuation character.

Of course, you are never forced to put everything in one line—the point is, in APL, you can.

At last, we are ready to write a function to do all this:

```
[0]   R←SHORTER FIND LONGER
[1]   ⍝  J. SOLOMON.  8208301630.
[2]   ⍝  THIS FUNCTION FINDS ALL
         OCCURRENCES OF THE SHORTER STRING
[3]   ⍝   IN THE LONGER STRING.
[4]   R←(∧/[1](¯1+⍳⍴SHORTER)⌽SHORTER∘.=
         LONGER)/⍳⍴LONGER
      ∇
9 18 23 32
```

The routine works (of course). The only problem is that it finds even those occurrences of the shorter string that are embedded within words of the longer string. If that's what we want, we're done. However, if we insist on treating the shorter string as a word, then we have to do something that will ignore occurrences of the shorter string in the longer string unless they are separate words.

```
[0]   R←SHORTER FINDW LONGER;LEFT;RIGHT
[1]   ⍝  J. SOLOMON.  8208301630.
[2]   ⍝  THIS FUNCTION FINDS ALL
         OCCURRENCES OF THE SHORTER WORD
[3]   ⍝   IN THE LONGER STRING.
[4]   ⍝  PUT A BLANK BEFORE AND AFTER THE
         SHORTER STRING--MAKE IT A WORD.
[5]   LEFT←' ',SHORTER,' '
[6]   ⍝  JUST IN CASE, PUT A BLANK AT
         EACH END OF THE LONGER STRING, TOO.
[7]   RIGHT←' ',LONGER,' '
[8]   ⍝
[9]   R←(∧/[1](¯1+⍳⍴LEFT)⌽LEFT∘.=RIGHT)
         /⍳⍴RIGHT
      ∇
      S1 FINDW S2
18
```

A little more convincing

Even those people who know almost nothing about APL know that "APL is good for processing arrays." It certainly is—in fact, one major purpose of this article is to show that APL is great for other uses, too. But let us proceed with APL and arrays. Suppose that M1 and M2 are three-by-three arrays. The preceding discussion should have convinced you that M1 + M2 will add corresponding elements, and M1 × M2 will multiply corresponding elements. Then how does one multiply M1 and M2 in APL according to the rules of matrix algebra? Easily:

```
M1+.×M2
```

In mathematics, the operation represented by the combination of the three symbols "+.×" is called the "inner product"; in APL it is a special case of the "generalized inner product," because the "+" and "×" can be replaced by any primitive functions. There are over 20 primitive scalar dyadic functions in APL; therefore, there are over 400 different ways of writing a generalized inner product.

Another application where APL makes life easy is in curve-fitting problems. We will consider here only polynomial curve-fitting, because it's the type most often used, but other methods are available and are also relatively painless.

Let us suppose that you have a collection of data points, with values of x and y stored in the vectors **X** and **Y**, respectively, and you would like to pass the best straight line through them—"best" in the least-squares sense, as usual.

```
Y⌹X∘.*0 1
```

will give you the intercept and slope, regardless of the number of datapoints. You don't need to write a complicated subroutine—you don't even have to CALL one. You say your problem isn't linear, but quadratic? Then you need a second-degree polynomial:

```
Y⌹X∘.*0 1 2
```

will give you the least-squares-best quadratic coefficients. Would you like a fifth-degree polynomial?

```
Y⌹X∘.*0 1 2 3 4 5
```

How about an *N*th-degree polynomial?

```
Y⌹X∘.*0,⍳N
```

Imagine what you would have to do to a PL/I program to implement those changes! I like APL.

A small amount of APL syntax

The distinctive rules of APL are simple, and there are no exceptions:

1. If a function requires two arguments, they are specified by having one argument to the left of the function name and one argument to the right. If a function requires one argument, that argument appears to the right of the function name.
2. There is no hierarchy of operations.
3. All expressions are evaluated from right to left.
4. Except where otherwise indicated by parentheses, the right-argument of a function is everything to the right of the function.

Although these rules are simple, they do take a little getting used to. For example, while 4+3−5 is still 2, 4−3+5 is not 6 but ¯4. (See below for explanation of ¯4.) By Rules 3 and 4, the evaluation is done as if the expression had been written 4−(3+5). In "ordinary" algebra, 4−3+5 is (4−3)+5. With a little care, however, this should cause no trouble.

In constructing the non-English symbols of the APL character-set, Iverson used symbols that were already available to publishing houses—generally, symbols from mathematics or the Greek alphabet. Addition and subtraction, at least, use familiar symbols: + and −. However, APL retains a distinction between certain concepts that are usually coalesced in other languages. In particular, the symbol for the minus sign that is the algebraic sign of a number (that is, not indicating the operation of subtraction) is the "high minus": not −4 but ¯4. Multiplication uses × and division uses ÷ (not the slash, which has a totally different meaning in APL). The single asterisk is used for exponentiation: 3*4 means 3^4, or 81.

"Monadic" and "dyadic" symbols

Most of the symbols in APL are used in two distinct contexts: requiring one argument ("monadic"), and requiring two arguments ("dyadic"). For example, the monadic ρ ("rho") is the "shape" operator: Tell me the shape of whatever is to the right. For example, ρ LIST might return the value 100, which would mean that LIST is a vector containing 100 elements. Dyadic ρ, however, is the "reshape" operator. It means: Build, out of whatever is on the right, something whose shape is described on the left. The result of executing

```
T←100ρ0
```

would be to build a vector of 100 elements, each equal to zero, and store it in the variable named T. The result of executing

```
TEST4←3 5ρι15
```

would be a matrix containing 3 rows and 5 columns, and whose elements were the numbers 1 through 15. If I then executed

```
ρTEST4
```

the result would be

```
3 5
```

telling me that TEST4 is indeed a 3 x 5 matrix. One can do both in one statement:

```
ρTEST4← 3 5ρι15
```

will generate TEST4 and then tell me its shape. Note how easy it is in APL to create variables of arbitrary size, shape, and contents.

Assignment operator

The assignment operator in APL is not the equal sign, but the left-pointing arrow. If the final result of all the calculations in an APL statement is not to be stored in a variable—that is, if you don't assign the result to a variable—APL assumes that you want the result printed at the terminal. In particular, to check whether TEST4 really does contain the integers 1 to 15, type out TEST4 and see for yourself.

"Take" and "drop" functions

As mentioned earlier, APL makes it easy to select certain elements from a collection. Two related functions for selecting elements are "take" and "drop," represented by ↑ and ↓ respectively. If I would like to select the first 37 elements of LIST, I need only use the syntax 37↑LIST. Presumably, this would be part of a longer expression, although it is executable as it is. To add up the first 37 elements of LIST, then, I would write:

```
+/37↑LIST.
```

To add up all except the first 37 elements of LIST, I would write:

```
+/37↓LIST.
```

If I want to add up all except the last 9 elements of LIST:

```
+/¯9↓LIST.
```

"Take" and "drop" can be combined. For example, to select elements 7 through 20 of a one-dimensional variable named BILL, I could write

```
6↓20↑BILL
```

and this would take the first 20 elements of BILL, and then drop the first 6, thus leaving me with the desired elements 7 through 20. Note that use of the take and drop operators does not change the variable in any way; it only selects certain elements of the variable, and makes their values available for use elsewhere.

Take and drop can also be used on variables of higher dimension. If I would like to select the first 3 rows and the first 7 columns of a two-dimensioned variable named CHECK, and store them in an array named BLOCK, I write

```
BLOCK←3  7↑CHECK
```

To take all the rows, but ignore the last column, of CHECK, I write

```
0 ¯1↓CHECK
```

(Taking all the rows is the same as dropping none of them.)

Trigonometric functions

Trigonometric functions are built into APL. Would you like to build a table of sines? How about a vector of 100 entries, representing angles from Pi/200 to Pi/2 in steps of Pi/200?

```
TRIG←1○(○÷200)×ι100
```

will do it for you. If you would like to print all 100 values, along with the values of the arguments, type

```
( (○÷200)×ι100),[1.4] TRIG
```

Pseudo-random numbers

The question mark is used to generate pseudo-random numbers. For example,

```
?100
```

will generate a pseudo-random number between 1 and 100. They can also be generated in bunches:

```
?3  7ρ125
```

will generate a 3-by-7 matrix containing 21 pseudo-random numbers between 1 and 125. Some types of simulation require the use of pseudorandom numbers. The ease with which they can be produced in APL—in whatever quantity and range desired—makes simulation an easier task. While you can, of course, generate a set of such numbers once, and save them for repeated future use, it is not necessary, inasmuch as APL will generate the same sequence every time unless you instruct it otherwise.

Polynomials

Evaluation of polynomials is interesting. Viewed from a certain perspective, a polynomial can be considered to be a dot-product of two vectors: powers of *x*, and coefficients of those powers. Well, we know about dot-products in APL:

```
FOFX←+/COEFF×1,X,(X*2),(X*3),...
```

where COEFF has the coefficients for the lowest to the highest power of *x*. But how many powers of *x* do we need? Well, we have one coefficient for each power of *x*, and one for the constant term; in other words, the highest power of *x* we need is one less than the number of coefficients:

```
FOFX←+/COEFF×X*¯1+ιρCOEFF
```

Again, let me point out that the above statement contains no explicit reference to the degree of the polynomial, nor do you have to know the length of the vector of coefficients: It will evaluate *any* polynomial of *any* degree. So, let's look at a user-defined APL function to evaluate polynomials. To print a listing of a user-defined APL function, type ∇FN-NAME[⎕]∇, so:

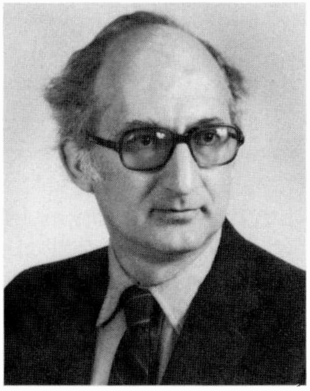
```
[0]  R←X POLY COEFF
[1]  ⍝  J. SOLOMON.  8208301630.
[2]  ⍝  THIS FUNCTION EVALUATES
        ANY POLYNOMIAL IN X.
[3]  ⍝  'COEFF' IS THE VECTOR OF
        COEFFICIENTS,
[4]  ⍝   AND COEFF[1] IS THE CONSTANT TERM.
[5]  R←+/COEFF×X*¯1+⍳⍴COEFF
     ∇
```

The function is named "POLY," and it is dyadic. There are five statements, and four of them are comments. To invoke a function, use its name in an appropriate context:

```
      ⍝ TRY 1+   (2×X) + X*2
      6 POLY 1 2 1
49
      ⍝ TRY 3    + (7×X) +   (4×X*2)
        + S  2×X*3    :
      5 POLY 3 7 4 ¯2
¯112
```

One easy way to check that the evaluation is correct is to write coefficients that are all positive integers less than 10, and set $x = 10$:

```
       10 POLY 3 6 1 9
9163
       10 POLY ⍳7
7654321
       10 POLY ⌽⍳7
1234567
```



Josef Solomon was awarded a BS degree in mathematics, from the University of Oklahoma, while on active duty with the United States Air Force. He first joined RCA in 1959, working on BMEWS at the Missile and Surface Radar Division in Moorestown. He was awarded the MS degree in physics from Temple University in 1963. After two years of work in high-energy physics at the University of Pennsylvania, he returned to RCA in 1966, joining the Systems Programming group of the information Systems Division in Cherry Hill.

When RCA left the mainframe business at the end of 1971, Mr. Solomon became an independent software consultant. His projects included an APL-based logic simulator for a 370-compatible microprogrammed minicomputer, for Advanced Technology Laboratories in Camden, and work in quantum statistical mechanics, for Temple University. He also implemented for Mathematica an interface between APL and RAMIS, Mathematica's database language. Mr. Solomon joined the Systems Engineering group of Government Communcations System in March of 1982, and has worked in digital signal processing, propagation of measurement errors, and network vulnerability. Mr. Solomon has recently transferred to Astro-Electronics in Princeton.

Contact him at:
RCA Astro-Electronics
Princeton, N.J.
TACNET: 229-2451

The more you learn about APL, the more you will be intrigued by its power. There is a function called "base value" or "decode," which can be used to convert numbers from one number-system to another. For example, 2 ⊥ 1 0 0 1 1 0 would regard the right-argument as the binary number 100110, and convert it to a decimal number. Positional notation is a shorthand way of indicating how many to add of each power of the base: To the base 2, 100110 represents

$$0 + 1\times2^1 + 1\times2^2 + 0\times2^3 + 0\times2^4 + 1\times2^5$$

or, in the decimal system, 37. That may be understandable more easily if we reverse the order of the terms:

$$1\times2^5 + 0\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0$$

Now the terms are in the same sequence as the digits in the original number. Does that look like the evaluation of a polynomial? That's what number-systems are, using the digits from left to right as coefficients of successively lower powers of the base. Well, then, it should be possible to use "decode" to evaluate a polynomial. The only problem is that now the vector of coefficients has to go in the opposite sequence. If only there were some way to reverse a vector . . . .

```
C←⌽COEFF
```

called, of course, the "reversal" of COEFF, is exactly that. The shortest statement I know of to evaluate a polynomial is

```
X⊥C
```

How could it be much shorter? The operative part is one keystroke!

APL at RCA

APL is available at RCA on the CISS computer at Cherry Hill, and, on computers at other RCA sites, as well. Computer Services at Cherry Hill teaches APL classes twice a year. For further information, see the systems programming staff at your location, or call "the sixes"—TACNET: 222-6666. There is considerable use of APL within RCA, and a user group is being formed. If you are interested, get in touch with Jon Strong, RCA Corporate Information Systems and Services, Bldg. 204-2, Route 38, Cherry Hill, NJ 08358. His telephone number is 609-338-6007; his TACNET is 222-6007; and his VM Memo address is JON. I'll be happy to talk or listen to anyone on the subject of APL. My address and telephone number are listed with the biographical data alongside this article.

As I have mentioned, and tried to demonstrate here, APL not only increases productivity, but it is fun to use. For those reasons, you may wish to write your own functions in APL. For those very same reasons, however, other people have already written libraries of APL functions to do a large assortment of tasks. For example, CISS has available APL functions that perform statistical analyses; design RLC ladder filters; calculate the rate of return on investments; plot data; lead you through a financial reporting system; and so on.

Many time-sharing companies offer APL in addition to the better-known languages, and several even specialize in APL. There is within the Association for Computing Machinery a Special Interest Group on APL ("SIGAPL"), making APL the only language to have its own SIG—other languages fall within SIGPLAN, the Special Interest Group on Programming Languages: APL interpreters have been implemented for large computers and small, from huge mainframes to the newest micros. APL interpreters for the larger computers tend to be pretty much the same—namely, complete. Some micros are too small to support implementation of full APL; for information on a specific micro, consult the vendor of that micro. As far as I know, there does not exist a compiler for APL—to compile object code is to lose all the benefits of the interactive interpreter. Nonetheless, some people are working on APL compilers, and compilers may be available by the time this is published.

How do I join the faithful?

I hope that you are sufficiently interested to explore the feasibility of using APL. The most-popular text is

APL: An Interactive Approach
Leonard Gilman & Allen J. Rose
John Wiley & Sons, Inc.
New York, New York
Second Edition, revised reprinting, 1976

Ken Iverson has written several interesting pamphlets on APL: "APL in Exposition," "Introducing APL to Teachers," and "An Introduction to APL for Scientists and Engineers," to name the most prominent. He has also written two APL-based algebra books: *Algebra, an Algorithmic Treatment,* and *Elementary Analysis.* All of these are available from APL Press, 220 California Ave., Palo Alto, CA 94306. And, of course, Iverson wrote the "Bible":

A Programming Language
Kenneth E. Iverson
John Wiley & Sons, Inc.
New York, New York,1962

Automated aids for the author

Engineers in ever increasing numbers are using their terminals to accomplish their writing tasks. And the compelling reason for this, we are told, is convenience. Authors who have used word processors (or word processing programs on computers) find that they write faster, but spend more time on the critically important rewriting, because it is so convenient to add, delete or move words on the screen.

At the *RCA Engineer,* we received over one third of the articles published in the last four issues as machine-readable, word processing files. The time saved by not rekeyboarding entire manuscripts allowed the staff to give more attention to other aspects of the production process—which has given us more interesting articles.

Most of the WP files came to us as Wang floppy disks and double-spaced printouts of the files. In the *RCA Engineer* office we have a Wang telecommunicating workstation wired to an AM Varityper typesetter. This arrangement allows us to amend the Wang files either before or after transmitting them to the typesetter. A telecommunications board in the typesetter converts the word processor's code set to the modified-ASCII code set used for photocomposition. Only editorial changes, corrections, type specifications, math and tabular information are entered at the typesetter keyboard. Also at the typesetter we have a Racal-Vadic 212LC modem, permitting us to receive files over the telephone from computers and word processors equipped with asynchronous 212A-compatible modems. Users of the corporate computer system have sent copies of their files to the *RCA Engineer* ID on CMS.

First, try out word processing to prepare your article, and then consider using these valuable telecommunications links to transmit your article to us.

Follow these guidelines when you decide to send us a file. Keep it simple. Eliminate underscoring and page breaks. Use a line length of between 60 and 80 characters. Avoid end-of-line hyphenations, other than those you want in the published article. In addition, avoid indenting material (at least, in the version of the file sent to us). Tabular material and mathematical equations will be reset in our office, so don't worry about how they are coded in the file. *Do* send us a *double-spaced* printout of the article along with the Wang disk or telecommunicated file. Carefully mark unusual characters, mathematics or underscores on this printout.

As of January 1983, we have received files from authors at the following locations:

Astro Electronics, Princeton (Wang OIS 140)

Automated Systems, Burlington (Wang OIS 140)

Corporate Computer Services (CMS files from the IBM mainframe)

Missile & Surface Radar (Wang files from various input devices in MSR's text management system)

RCA Laboratories, Princeton (DEC VAX 11/780 and Lab/Net, their local area network)

Solid State Division, Somerville (Wang OIS 140 and CMS files)

Outside RCA (Televideo computer, TS-802)

Please call us for more information at TACNET 222-4254.

—TEK

R.F. Gimigliano

Producibility—The road to increased productivity

The basic economics of producing goods have made producibility a focus of concern even in essentially one-of-a-kind manufacturing operations.

Abstract: *Producibility, a cornerstone of efficient long-run production for many years, has become an important factor in one-of-a-kind high-technology development as well. This paper describes the application of producibility concepts to advanced radar system development at MSR in Moorestown. Three specific areas treated are (1) formation of interdisciplinary teams to plan and implement producibility approaches, (2) use of work centers in Manufacturing Engineering to promote producibility, and (3) application of mockups and mechanized assembly techniques to the production process.*

Producibility, a product characteristic denoting ease and economy of manufacture, has been a prime objective of mass production since that type of manfacturing came into being. In recent years, the basic economics of producing goods have made producibility a focus of concern even in essentially one-of-a-kind manufacturing operations.

Cost drivers that create the demand for producibility in such operations include high cost of materials, high interest rates, and the scarcity of skilled labor. Investments in new products and facilities are closely scrutinized for the best return on investment; this means that product materials must be of optimal quality with minimal cost and that production facilities must find ways to reduce their dependence on skilled labor. Producibility planning represents an approach to such problems. Success in implementing a producibility plan requires the early involvement of both design engineers and manufacturing operations engineers to maintain continuous monitoring of a new design from concept to delivery.

Planning for producibility is nowhere more important than at the very beginning of a project, in the early design stages. At this point a team approach, involving engineering, management, manufacturing, and purchasing personnel, can assure that designs require the least costly materials and techniques without sacrificing quality and product performance.

Manufacturing operations must be flexible enough to incorporate a variety of producibility approaches during the production phase. The development of effective approaches involves considering all available parts, processes, facilities, production technologies, and types of personnel required for the project. All phases of the manufacturing process must be considered, from initial planning to shipment of the product, to ensure that fabrication and assembly methods meet producibility requirements.

The application of producibility techniques to one-of-a-kind manufacturing operations is receiving a great deal of attention from government agencies as they discover the benefits derived from cost reduction and increased product quality. In fact, producibility efforts are being invoked contractually on more and more government-sponsored projects. At Moorestown, as an example, producibility has become a prime consideration in manufacturing operations involving development of advanced-design radar systems. Producibility at Moorestown is addressed in terms of team planning, special organizational approaches, and applications of special techniques such as mockups and mechanized assembly.

Planning for producibility

A basic element in Moorestown's producibility planning process was the formation of a producibility team, to concentrate in this area on an advanced radar system development program. This team concept has subsequently been extended and applied on all major programs. The team includes participants from Project Management, Design Engineering, Production Engineering, Quality Assurance, and Materials Management (Purchasing), each of whom has unique responsibilities.

Representatives of Project Management monitor the overall producibility plan to ensure its effectiveness for the project. They participate in tradeoff evaluations and are responsible for obtaining consultants. They also specify cost allowances in the project plan to cover producibility efforts during design and manufacture. Representatives of Materials Management monitor acquisition of parts and equipment at Moorestown and suppliers' manufacturing sites. They also check to see that producibility

Final manuscript received December 10, 1982
Reprint RE-28-1-4

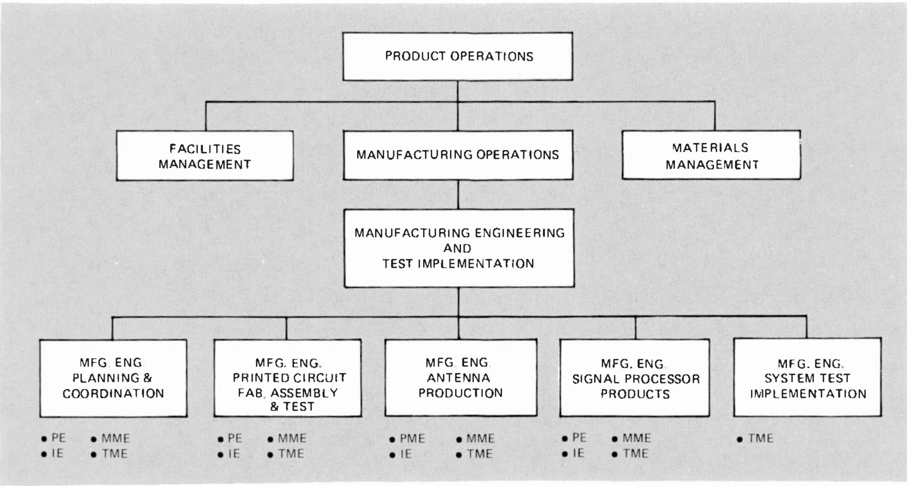

PRODUCT OPERATIONS
FACILITIES MANAGEMENT
MANUFACTURING OPERATIONS
MATERIALS MANAGEMENT
MANUFACTURING ENGINEERING AND TEST IMPLEMENTATION
MFG. ENG. PLANNING & COORDINATION
MFG. ENG. PRINTED CIRCUIT FAB. ASSEMBLY & TEST
MFG. ENG. ANTENNA PRODUCTION
MFG. ENG. SIGNAL PROCESSOR PRODUCTS
MFG. ENG. SYSTEM TEST IMPLEMENTATION
• PE • MME • IE • TME
• PE • MME • IE • TME
• PME • MME • IE • TME
• PE • MME • IE • TME
• TME

Fig. 1. *In Manufacturing Operations, various types of engineers responsible for producibility are assigned to particular work centers. The first four groups each have teams of Production Engineers (PE), Industrial Engineers (IE), Manufacturing Methods Engineers (MME), and Test Methods Engineers (TME). The last group requires only Test Methods Engineers.*

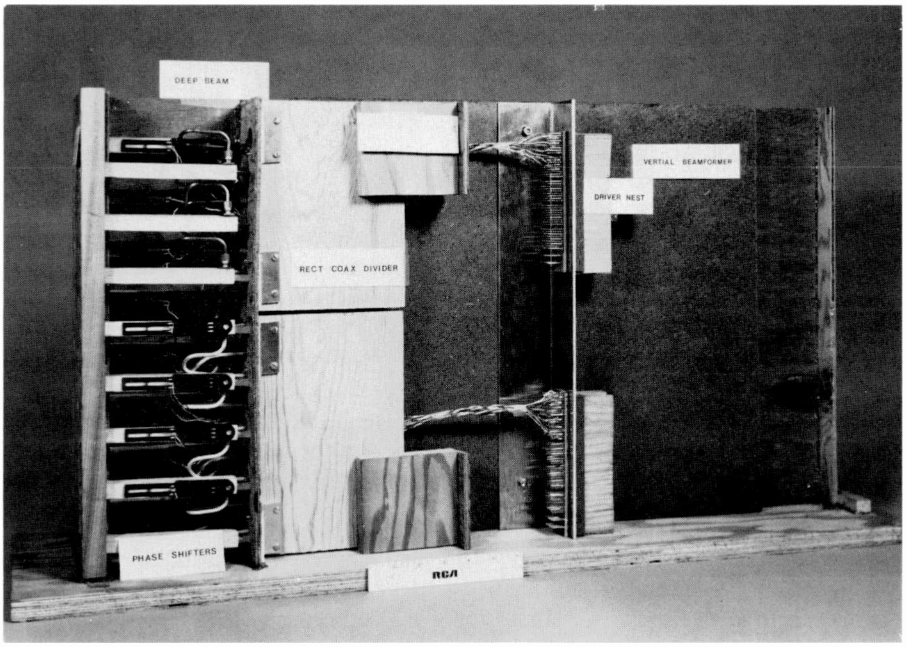


Fig. 2. *A section of a mockup of a column beamformer assembly is used to solve problems in assembly and design configuration.*

implementations are on schedule in the suppliers' plants.

Design Engineering group members ensure that producibility standards are integrated into the design. In addition to their standard responsibilities during a project (designing equipment and specifying hardware), they interact with Production Engineering representatives during design reviews, design release, and engineering change notices. They participate in tradeoff studies with engineers from Manufacturing Operations and develop guidelines to document critical or special constraints that may arise during the project. They then follow production through to the first piece of equipment manufactured.

Those producibility representatives from Production Engineering review whether and how the design can effectively be developed into manufactured hardware. These engineers follow the design from concept through layout and detail phases. They establish producibility checklists, provide cost estimates, and recommend areas for tradeoffs and cost reduction. This group not only presents producibility data at all conceptual and major design reviews, but also prepares producibility review reports and documents supporting the tradeoff and cost reduction studies. Quality Assurance supports the reviews of indicated design changes and processes, tooling, and test plans to ensure that product quality is not compromised.

Guidelines for producibility during a project evolve as members of the producibility team perform their individual functions, interacting with each other. Establishment of target costs signals the beginning of tradeoff studies triggered by design variations for improving producibility; the product then may be formally reviewed for manufacturability. Planning for the actual build cycle also evolves from this stage.

Getting producibility into manufacturing

One of the principal approaches to producibility during the manufacturing process at Moorestown is the work center. The work center concept was selected as a potentially powerful approach to supporting production of major system components by all the disciplines required to carry out the operation. As an example, the Printed Circuit Fabrication area has a work center involving representatives from Production, Production Control, Cost Control, Quality Assurance, and Manufacturing Engineering.

The same work center concept is also applied in the Manufacturing Engineering organization (Fig. 1). The technical support for major work centers varies according to the need; in fact, a properly structured manufacturing engineering group is the key element to most efficiently introducing a product into the manufacturing cycle.

The Planning and Coordination group shown in the figure ensures that the proper manufacturing procedures are established. This group is also responsible for coordinating producibility aspects of new designs and for ensuring a smooth transition of new products to manufacturing. A Production Engineer from the Planning and Coordination Group works closely with design engineers during the conceptual phase, guiding design decisions so that a producible design results that is within specifications and cost constraints. As the product design matures, the Production Engineer interfaces with a counterpart in a product line group (such as Printed Circuit Fabrication) to ensure that effective technology is developed and implemented. Each of the product line groups contains representatives of Manufacturing Engineering disciplines required for total technical support and documentation of a project.

The concept of work centers organized within Manufacturing Engineering is relatively recent at Moorestown. The first work center was established in 1981, and the concept is being tested on a variety of radar production programs. To date, each work center established has proved successful, both operationally and in support of producibility goals.

Engineering aids for producibility

In addition to the organizational approaches described above, a variety of engineering tools are used to supplement producibility during design and manufacture. At Moorestown, two approaches to improving producibility involve the use of mockups to expedite problem solving prior to manufacture, and the application of robotics to reduce costs during manufacture.

Mockups have multiple uses

A well-designed mockup can be invaluable in the development of product design, in facilitating the development of assembly techniques, and in highlighting problem areas. For example, a column beamformer mockup (Fig. 2), constructed in one project to highlight a scheme for assembling phase shifter elements to the antenna structure, is also used to investigate associated wire routings, printed circuit card nests, and waveguide assemblies. A full-scale wooden mockup of an antenna structure (Fig. 3) is used to determine cable lengths and their routings, define handling fixture needs, develop configurations for waveguides and air flow ducts, and analyze subassembly placements.

Robots are here to stay

A mockup of the antenna front face was used as a means of determining how robotics can be used during assembly of the antenna. Robotics are generally considered for assembly tasks that are routine and tedious, yet highly specialized. In the case of this antenna, more than 4000 alumina windows (Fig. 4) must be emplaced on the front face and held in place by an adhesive, which must be accurately applied. A producibility tradeoff study resulted in the decision to use automated assembly equipment to insert these windows and apply the adhesive. The approach for the window insertion system by automated assembly was finalized after meetings involving producibility representatives from Design Engineering, Manufacturing Engineering, Materials Management, and the supplier.

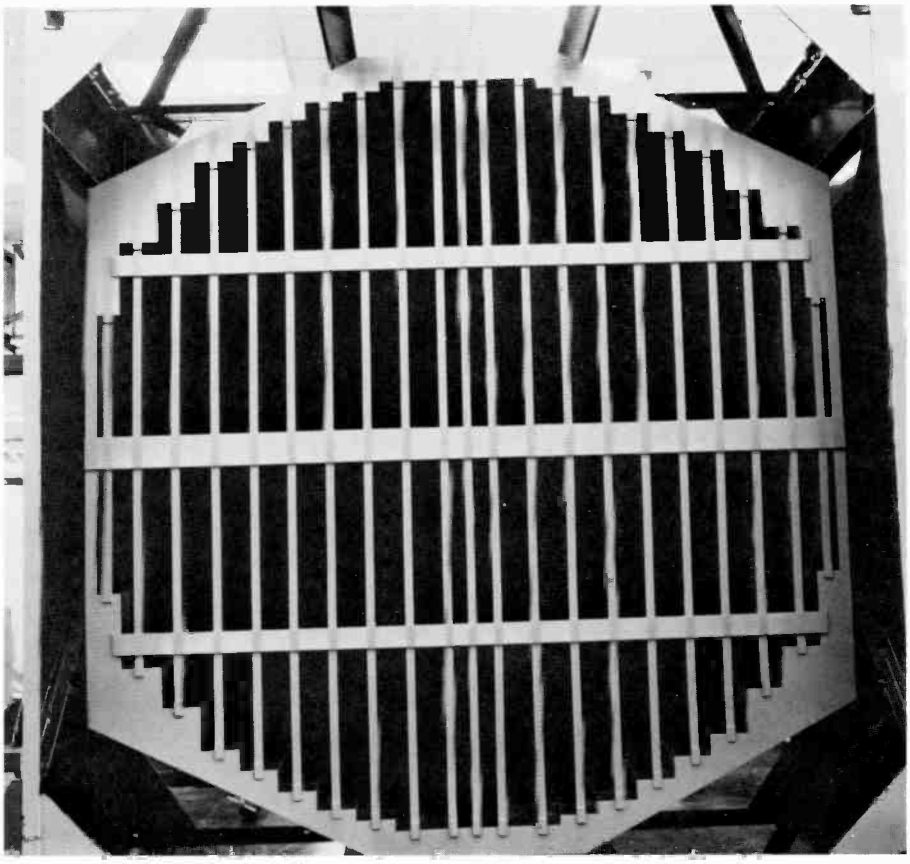


Fig. 3. *A wooden mockup of an antenna structure serves to highlight potential producibility problems.*

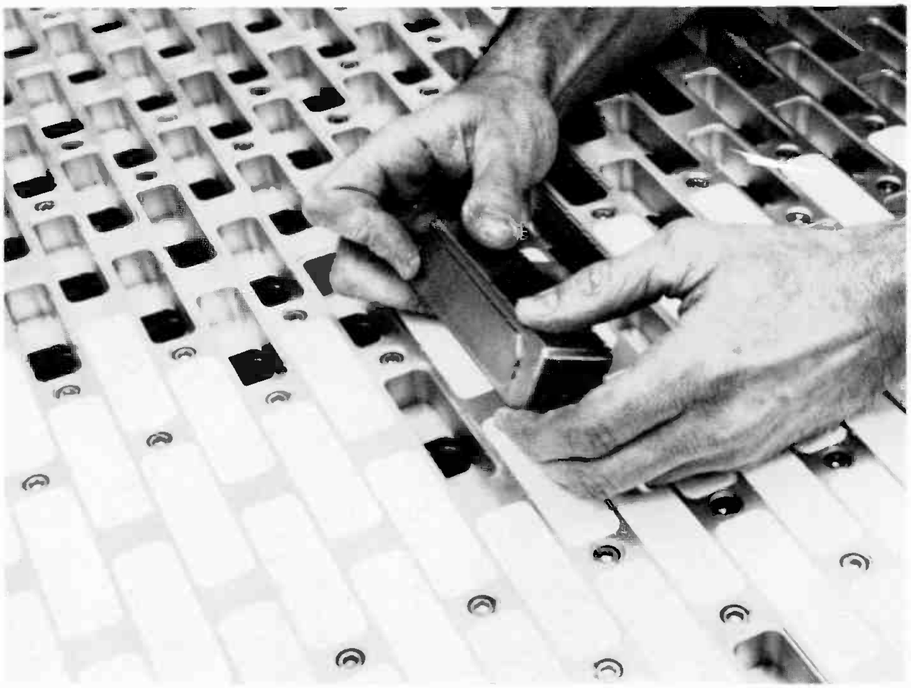


Fig. 4. *Alumina windows are carefully placed by hand in the face of a metallic antenna mockup. Robotics are planned for the actual assembly.*

This antenna face also must be accurately coated with a certain paint, applied to exact thickness. Controls must be pro-

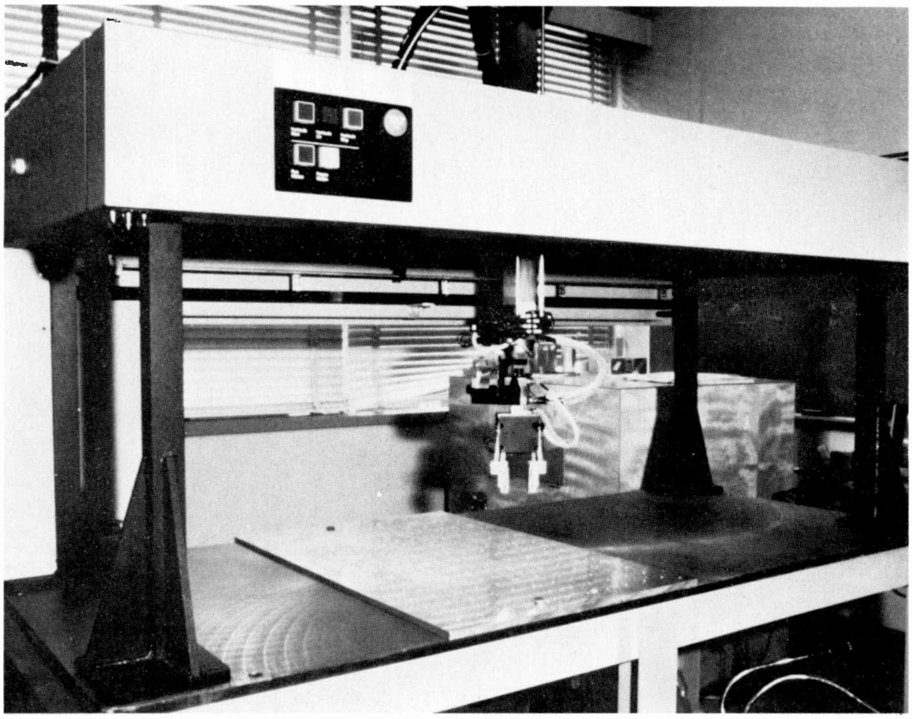


Fig. 5. *The IBM RS-1 robot, introduced experimentally for phase shifter manufacture, offers the speed, accuracy, and tactile abilities considered necessary for cost-effective production.*

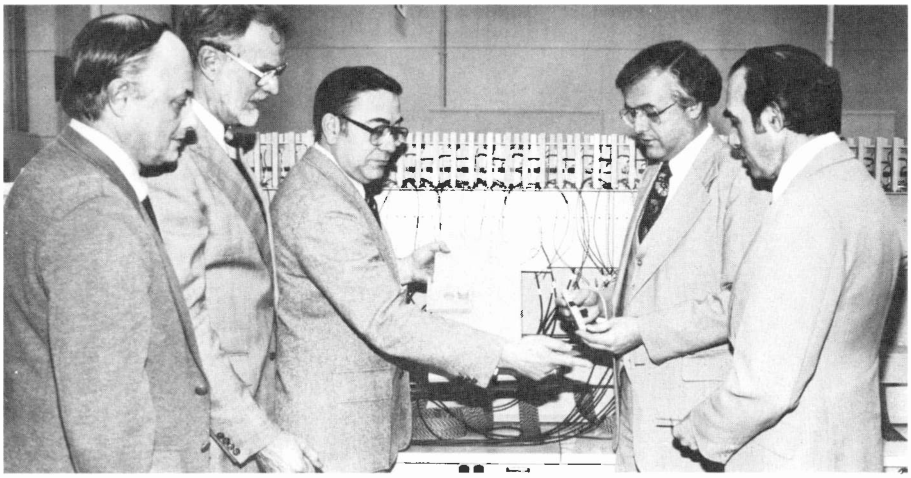
vided that allow for variations in material viscosity and slumping characteristics. A series of tests established the processes for material applications and suitable equipment manufacture; the desired results were obtained by employing a mechanized spraying system.

Phase shifter elements contain critical electrical components whose function depends on accurate mechanical tolerancing and uniform assembly techniques. These elements are prime candidates for mechanized assembly, since more than 300 discrete motions are necessary to assemble each phase shifter. Evaluation of available robotic systems resulted in the selection of an IBM RS-1 robot (Fig. 5) as the most versatile for these tasks. The combination of speed, accuracy, and tactile sensing that this robot exhibits was considered advantageous for performing the various assembly steps successfully. A series of tests has been performed in cooperation with Manufacturing Systems and Technology Research at RCA Laboratories utilizing their IBM robot. As a result, procurement of an RS-1 robot for the Manufacturing Engineering activity at Moorestown is planned for 1983 as a means of further investigating techniques and design enhancements required for best use of robotics in phase shifter manufacture.

Preliminary evaluation has shown that mechanized assembly with robotics is cost effective, greatly enhancing producibility and product reliability. Assembly units using robots prove to be useful in achieving the desired improvement in producibility and subsequently in return on investment.

Conclusion

Production capability has traditionally relied on efficiency derived in part from the application of advanced manufacturing technology. Efficiency is a recognizable element of producibility that is becoming critical to all categories of manufacturing organizations, from massive automated production complexes to the model shops in a one-of-a-kind assembly operation. First-class products can be produced by considering cost-effective producibility, with an eye toward quality—beginning in initial engineering design and continuing throughout all phases of the process. It is important to recognize that increasing productivity using producibility techniques is not restricted to mass manufacturers; these techniques are just as valid and critical in the area of high-technology products produced in very small numbers, such as the advanced radar systems built at Moorestown.



Author Gimigliano (center) explains the planned application of mockups for members of a producibility team representing Project Management, Materials Management, Design Engineering, and Production Engineering.

Dick Gimigliano has been involved in technical support of manufacturing operations since joining RCA in 1953. He is currently Manager, Planning and Coordination, in MSR's Manufacturing Engineering organization, responsible for establishment and coordination of producibility plans for all major programs.
Contact him at:
Missile and Surface Radar
Moorestown, N.J.
TACNET: 224-2924

K.S. Reid-Green | W.A. Hatter | D.C. Quardt

CAD/CAM at RCA Laboratories—Tools and applications, present and future

This three-part report glimpses the future of computer-aided design and manufacturing, including robotics, while showing some of the equipment and techniques available today at RCA Laboratories.

When asked if human spacecraft pilots would soon be unnecessary, an astronaut replied: "Where else can you get a highly adaptable nonlinear servomechanism that weighs only 150 pounds and that can replicate itself almost automatically?" The astronaut nearly defined the ultimate goal of designers of CAD/CAM and robotics equipment—to build machines that can completely take over jobs that once were the province of human workers.

The principal reason for trying to use machines to build other machines is one of economy. In the past thirty years or so, the cost in salary and benefits of a human worker has continually increased while the size, cost and availability of adaptable machines (computers and computer-controlled machinery), have decreased dramatically. Given the tireless consistency, versatility, economy, and general good health of an assembly-line robot, it is obvious that in the near future no human beings will be employed on assembly lines. For the time being, however, it doesn't look as if robots are likely to take over the job of replicating the human species.

Most people in industry recognize that the acronym CAD/CAM stands for Computer-Aided Design and Computer-Aided Manufacturing. The word *robotics* has a much more esoteric ring, conjuring up images of anthropomorphic robots such as the well-known gold-anodized C3P0 character in *Star Wars.* I am a computer programmer by trade, and to me a robot is an output device. To be sure, it is of a different class of output device from a printer or paper tape punch, but in the sense that a computer controls the actions of a robot, an output device it is.

Abstract: *A combination of computers, robots, and automatic machinery is used in CAD/CAM technology. Computer-Aided Design and Manufacturing has shown significant productivity increases in repetitive operations such as assembly-line production. It can also improve productivity in the specialized or one-of-a-kind production found in a research environment, beginning with design aids and ending with automatically produced prototypes.*

Final manuscript received January 10, 1983.
Reprint RE-28-1-5

Integrating CAD and CAM

There is a tendency to think of computer-aided design and manufacturing as separate entities. Although it is possible to set up a CAD facility without CAM (in architectural design, for example), and conversely (say, in a job-shop manufacturing environment), there are places where the two components should be integrated. A good example of such a place is the Manufacturing Systems and Technology Research Laboratory at RCA's David Sarnoff Research Center, where prototype machines are designed, built, and tested. It is not difficult to imagine why computer-aided design would be useful, but is it reasonable to include computer-aided manufacturing in a place where only a very few identical parts are to be made? Most people tend to confuse CAM with automation—the purpose of CAM is not to facilitate the production of large quantities of identical parts, but to provide the capability of completing parts accurately, quickly, and, therefore, economically.

Numerical control

Computer-aided manufacturing requires numerically controlled machinery. Numerical control (NC) of tools-is a lot like computer programming in the sense that the tool is designed to be of general utility and, to be useful, needs detailed guidance in the form of coded instructions. The earliest numerical machine was probably the one built at MIT in 1952 under contract to the U.S. Air Force. Within five years, large numerical milling machines were in use for building aircraft and spacecraft structures. It was clear by that time that computers were necessary to convert a geometric description of a part into the numerical codes used by the milling machine.

A three-axis mill can move the cutting bit up and down (Z-axis), left and right (X-axis), and forward and backward (Y-axis), either by moving the spindle or the bed on which the stock to be machined is mounted. Most machines can simultaneously move in two or more axes. To control repeatability, stepping motors are used to move the tool. Each step taken by a motor moves the tool a small distance, perhaps 0.001 inches, in a straight line. This means that to cut around a circular arc it is necessary to compromise a little and imagine that an arc is actually a regular polygon made up of as many short straight lines as is necessary to maintain accuracy within a specified tolerance.

The tool-path programmer should not be required to compute the endpoints of all these lines, but the milling machine expects them as input. This is where the computer comes in—the tool programmer tells a computer to cut an arc whose center and radius are specified, by using a tool of a certain radius, to a given tolerance. The computer calculates all the straight-line motions of the tool, offsetting the tool from the actual arc to allow for the fact that the edge of the tool cuts on the arc, and minimizing the number of cuts depending on the allowed tolerance.

Traditionally, the medium of input to a numerical machine has been punched paper tape, so the computer, having completed the conversion of the tool programmer's instructions into machine codes, causes those codes to be punched for subsequent input into the numerical tool. A more modern and much faster practice is to use either magnetic tape or direct transmission down a wire, but paper tape is still a significant medium in this field.

Computerized numerical control

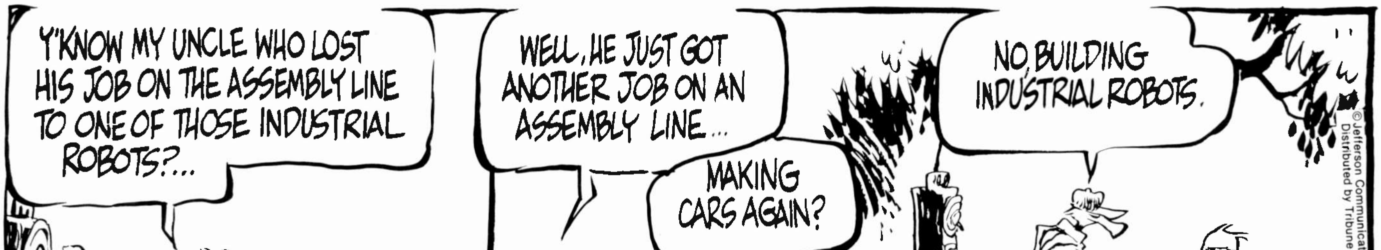
As computers became cheaper and smaller, numerical control equipment has been improved by the addition of a controlling computer on the milling machine. Computerized Numerical Control (CNC) increases the local intelligence of the tool—for example, the arc described above can be expressed as an arc to the tool rather than as a series of straight lines. The tool will compute the most accurate arc it can cut. Features of CNC include automatic pocketing, peck drilling (retracting the tool periodically while drilling in order to clear chips from the flutes of the drill), and other well-defined processes.

In fact, the availability of high-speed, low-cost computing makes CAD/CAM and robotics possible. Using the appropriate computing tools, a designer can try more than one idea without converting each idea to a physical part or subassembly. Having selected a design, the designer can instruct the computer to produce an engineering drawing and a control program for a numerically controlled lathe or mill, which will produce the finished part exactly to specifications. Since the advent of relatively inexpensive computer graphics, a designer can see on a display screen exactly what the finished part will look like. It is not necessary to build anything except a mathematical model to see how a part will look and how it will behave when installed in a machine.

Rapid changes and robotics

A manager is faced with a very difficult job if he has to build up a computer-aided manufacturing facility from scratch. The field is changing rapidly at present because the economics of human versus automated processes are superimposed on rapidly improving computer technology. Turnkey systems (packaged hardware-software combinations) hold out the hope of instant access to CAD, but they tend to be very expensive. Frequently, a large system requires a large investment in training and a long learning curve before new users exceed their former productivity. On the other hand, home-grown software takes time to develop and will be difficult to control in terms of cost, but it offers the opportunity to provide exactly for the needs of the user. Similarly, the robotics field is changing continually. There is clear pressure to automate assembly lines, and in fact, progress has been made in this direction, especially in hazardous environments. Manipulators, which were used to handle radioactive materials, represent an example of an early development in robotics. Although they were not computerized (they required a human operator), work was done to transmit "feel" to the operator so that he could detect in his own fingers that the manipulator had a firm but not crushing grip on, for example, a glass beaker.

A robot to be used in welding an automobile door frame may be wired to do this one job for an entire model year. A com-


Y'KNOW MY UNCLE WHO LOST HIS JOB ON THE ASSEMBLY LINE TO ONE OF THOSE INDUSTRIAL ROBOTS?...
WELL, HE JUST GOT ANOTHER JOB ON AN ASSEMBLY LINE...
MAKING CARS AGAIN?
NO, BUILDING INDUSTRIAL ROBOTS.
MACNELLY
© Jefferson Communications, Inc. 1983
Distributed by Tribune Company Syndicate, Inc.
1/17

(Reprinted by permission of Tribune Company Syndicate, Inc. Rights granted for one-time use only.)

puter-controlled robot, on the other hand, may be used for many kinds of activity. As soon as the robot has been made to do a job successfully, it can be reprogrammed for a new job with no changes to the physical device at all. If the new program is simulated in advance, on computing equipment, the robot can be ready for a different job once the new program is loaded.

The design of functions that a robot can do depends to a great extent on the capability of the robot hardware. Imagine a human arm with the normal shoulder, elbow, and wrist. Below the wrist, the arm is not human, having only an index finger and a thumb. These digits have no joints, but can separate and come together, sensing pressure when they touch an object. An emitter on the finger gives off infrared radiation and, opposite it, on the thumb, is a detector. This rudimentary eye can detect an object that is between the thumb and finger.

Robots

Given the fact that this robot also has a brain in the form of a computer, what can it do? It can play chess because all it needs to know is the approximate height and location of the pieces and the size and location of the chessboard. Using the robot arm to make chess moves is much easier than writing a good chess-playing computer program, and the latter has certainly been done.

But although it is possible to use a robot to put screws in tapped holes, the robot cannot do this inexpensively. When a human being does this job, he must take a screw lying in a random position and position it so that the point is extending away from the fingers, which are gripping the head of the screw. When he lets it go, it will probably tilt off the vertical axis, so the human then has to use the left hand to straighten the screw while the right hand guides the screwdriver into the slot and then twists it, repeating the process until the screw is tight. Screws are good fasteners, but a lot of hand-eye coordination is needed to use them. Try to drive a screw with your eyes closed, using only the thumb and forefinger of one hand. If you succeed, it is because your processor is more flexible than the robot's.

Some robots have better eyes than the one described above, and a controlling computer program that can recognize shapes. At present, such robots are too expensive to compete with human workers. The state of robot assembly today requires that the designer adapt the design to the capability of the assembly robot. As it turns out, the adaptations usually make assembly easier for humans as well.

Most robots today don't move around or carry on a meaningful dialogue with human beings. They are supposed to stay in one place and to do boring or dangerous jobs very quickly, or at least inexpensively, without complaining or asking for a raise. But, they don't always do so. At a recent exposition, a robot was demonstrating its ability to hang up a series of lettered tiles to spell out *Robots Spell Productivity.* It had just reached the point where most of the audience recognized the last word when it went berserk and wallloped the back of the sign it was building, knocking most of the tiles to the floor. Nobody expected it to recover sufficiently to pick up the tiles and start again. Maybe in another thirty years . . . .

—Keith Reid-Green

CAD/CAM: A productivity tool at RCA Laboratories

Dennis Quardt

The model shop at the David Sarnoff Research Center has two Spindle Wizard (Spindle Wizard is a trademark of Lesnor-Maehr Manufacturing Corporation, Farmingdale, New York) computerized numerical-control milling machines. These machines were acquired to increase shop productivity and to make prototypes used by the technical staff. A computer-support system has been developed to greatly assist this process, including facilities for original design, design modification, drafting, and machining.

System components

An analysis of the needs of the computer-support system indicated that all functions could be performed efficiently by three types of operations: manual, generalized software, and in-house software. When the milling machines were first installed, all designs were done by hand—the machinists wrote all the code to drive the numerical mill. Shortly thereafter, a Computervision system was acquired. It is a generalized system that requires skilled operators and a computer professional. A Tektronix 4052 graphics system was purchased for in-house software development.

In-house software systems

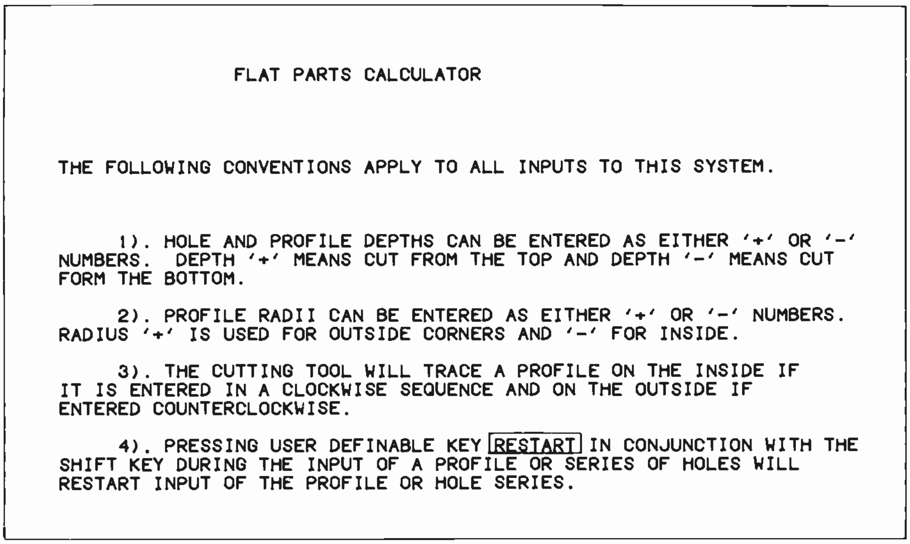
Since all parts can be designed and cut manually or by the Computervision system, in-house software must have specific advantages over either of the other operations to justify the cost of development. The following requirements were developed for in-house software:

1. All software must be very easy to use. The average user should understand each program after less than one hour of training.
2. The software should help the user find errors in design. Overhead in producing parts includes costs of tool setup, and costs associated with debugging the design, and with writing and debugging the milling machine program. If many copies of the same part are produced, the percentage of overhead to total job cost is lower than if only a few parts are cut. Researchers usually require only a few copies of each part, making it essential to keep overhead down. The best way to do this is to get it right the first time.
3. All programs within a system will be integrated. Systems will be constructed to perform all phases of a process. Data will be passed from phase to phase without reentry.

4. The software must be free from errors.

5. No effort will be expended to optimize tool movement by the milling machine.

So few copies of each part are produced that the time saved at the mill by rapid movement and shortest path operations is insignificant.



```
                    FLAT PARTS CALCULATOR


THE FOLLOWING CONVENTIONS APPLY TO ALL INPUTS TO THIS SYSTEM.


     1). HOLE AND PROFILE DEPTHS CAN BE ENTERED AS EITHER '+' OR '-'
NUMBERS.  DEPTH '+' MEANS CUT FROM THE TOP AND DEPTH '-' MEANS CUT
FORM THE BOTTOM.

     2). PROFILE RADII CAN BE ENTERED AS EITHER '+' OR '-' NUMBERS.
RADIUS '+' IS USED FOR OUTSIDE CORNERS AND '-' FOR INSIDE.

     3). THE CUTTING TOOL WILL TRACE A PROFILE ON THE INSIDE IF
IT IS ENTERED IN A CLOCKWISE SEQUENCE AND ON THE OUTSIDE IF
ENTERED COUNTERCLOCKWISE.

     4). PRESSING USER DEFINABLE KEY [RESTART] IN CONJUNCTION WITH THE
SHIFT KEY DURING THE INPUT OF A PROFILE OR SERIES OF HOLES WILL
RESTART INPUT OF THE PROFILE OR HOLE SERIES.
```

Fig. 1. *Preamble to the design and modification phases of the Flat Parts System.*

The in-house package

Over 80 percent of the software provided by large generalized systems is used less than 20 percent of the time. The larger the system, the more dramatic these numbers. For the purpose of in-house software, usefulness is defined in terms of two factors. The first is the time that can be saved by using the software on a single job. All other things being equal, systems that save

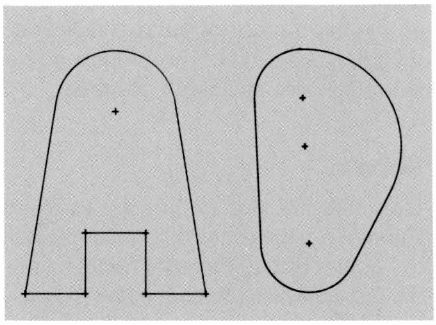
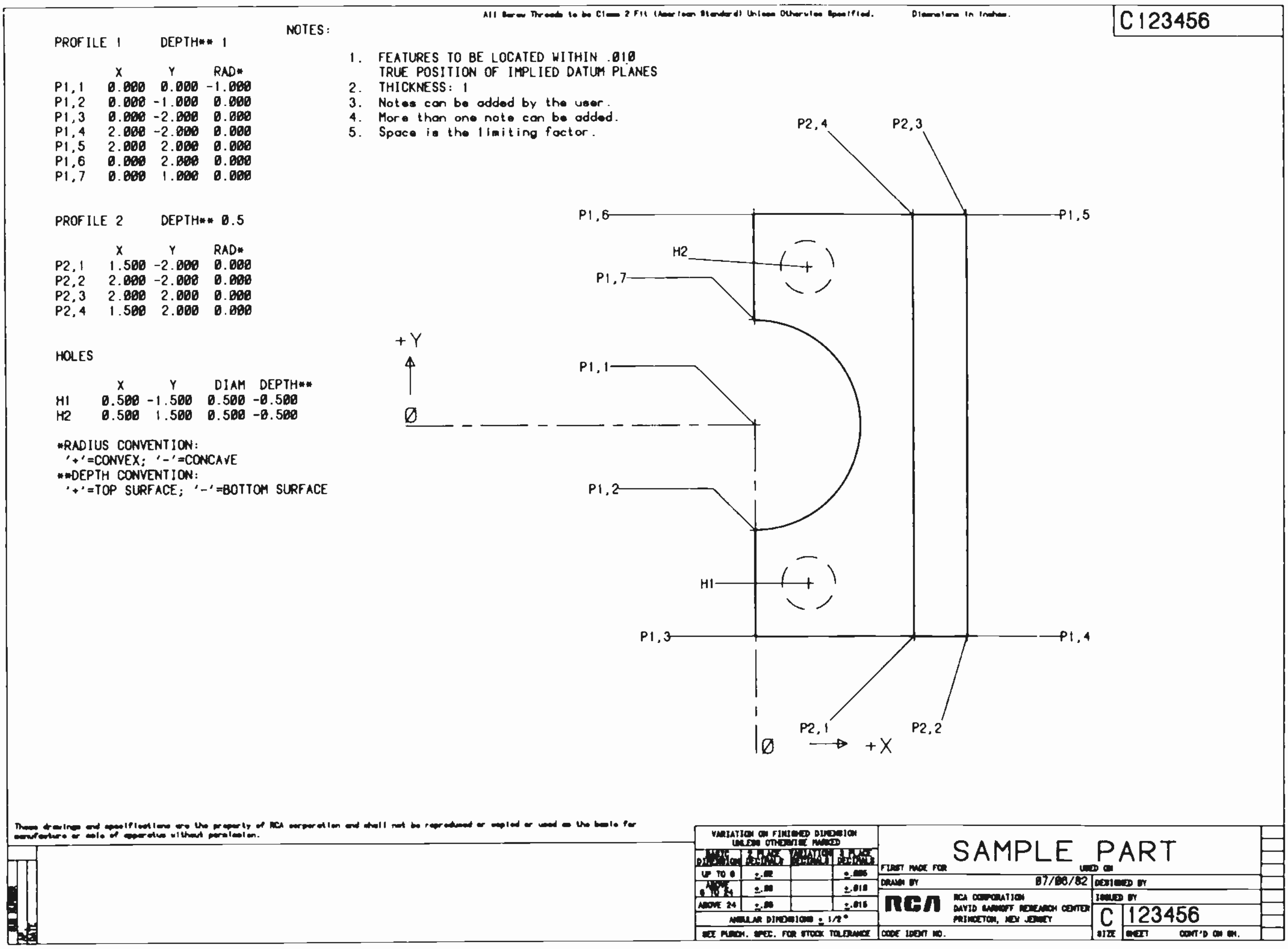


Fig. 2. *Two profiles defined by the Flat Parts System.*



```
PROFILE 1     DEPTH** 1

        X      Y      RAD*
P1,1   0.000  0.000 -1.000
P1,2   0.000 -1.000  0.000
P1,3   0.000 -2.000  0.000
P1,4   2.000 -2.000  0.000
P1,5   2.000  2.000  0.000
P1,6   0.000  2.000  0.000
P1,7   0.000  1.000  0.000

PROFILE 2     DEPTH** 0.5

        X      Y      RAD*
P2,1   1.500 -2.000  0.000
P2,2   2.000 -2.000  0.000
P2,3   2.000  2.000  0.000
P2,4   1.500  2.000  0.000

HOLES

        X      Y      DIAM  DEPTH**
H1    0.500 -1.500  0.500 -0.500
H2    0.500  1.500  0.500 -0.500

*RADIUS CONVENTION:
 '+'=CONVEX; '-'=CONCAVE
**DEPTH CONVENTION:
 '+'=TOP SURFACE; '-'=BOTTOM SURFACE
```

NOTES:

1. FEATURES TO BE LOCATED WITHIN .010 TRUE POSITION OF IMPLIED DATUM PLANES
2. THICKNESS: 1
3. Notes can be added by the user.
4. More than one note can be added.
5. Space is the limiting factor.

All Screw Threads to be Class 2 Fit (American Standard) Unless Otherwise Specified.
Dimensions in Inches.
C123456
P2,4
P2,3
P1,6
P1,5
H2
P1,7
+Y
P1,1
Ø
P1,2
H1
P1,3
P1,4
P2,1
P2,2
Ø
+X
These drawings and specifications are the property of RCA corporation and shall not be reproduced or copied or used as the basis for manufacture or sale of apparatus without permission.
VARIATION ON FINISHED DIMENSION UNLESS OTHERWISE MARKED
UP TO 6
ABOVE 6 TO 24
ABOVE 24
ANGULAR DIMENSIONS ± 1/2°
SEE PURCH. SPEC. FOR STOCK TOLERANCE
SAMPLE PART
FIRST MADE FOR
USED ON
DRAWN BY
07/06/82
DESIGNED BY
ISSUED BY
RCA
RCA CORPORATION
DAVID SARNOFF RESEARCH CENTER
PRINCETON, NEW JERSEY
C
123456
CODE IDENT NO.
SIZE
SHEET
CONT'D ON SH.

Fig. 3. *Drawing produced in the plotting phase of the Flat Parts program. Two levels of cuts are illustrated along with holes cut from the bottom surface.*

greater amounts of time are more useful. This saving usually occurs if the system handles large volumes of data or performs long calculations.

The second factor is how frequently the software can be applied. Usually, software used on a daily basis is more useful than that run only monthly. Other factors, such as development and hardware costs, affect decisions to write programs in-house or to omit them. The most significant factor is usefulness.

Flat Parts System

One of the packages developed in-house that is very useful is the Flat Parts System, developed to satisfy all the criteria for in-house software. The system is menu-driven. The user is prompted for each input by a descriptive message. Three of the four phases of the system have preambles that describe the conventions in use within the phase (Fig. 1).

Perhaps the most significant aspect of the program is that the user sees graphical displays of all design work. These displays uncover many design errors and make mis-cut parts at the mill nearly nonexistent. Productivity gains produced by this system primarily result when it detects errors before they get to the milling machine.

Flat Parts definitions

Within the Flat Parts System, a profile is defined as a closed path consisting of line segments and circular arcs connected at their endpoints. All segments and arcs in a profile must be in the same plane, perpendicular to the axis of the tool (Fig. 2). Parts that consist of profiles and vertical holes can be designed and machined with the Flat Parts program. Profiles and holes can be cut from either the top or the bottom of the material (Fig. 3). Many profiles at different depths can be cut into the same part.

Program phases

The four phases of processing in the Flat Parts System are as follows:

1. Design a part;
2. Change the design of an existing part;
3. Draw the part on the plotter; and
4. Produce a milling program.

Holes are cut in a part by identifying the X and Y coordinates of their centers, diameters, and depths. Negative depths imply cutting from the bottom.

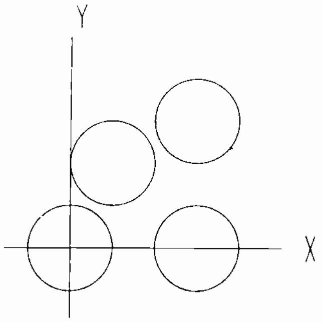


Fig. 4a. *Circles defining the part.*

Circle	*X*	*Y*	*Radius*
1	*0*	*0*	*1*
2	*3*	*0*	*1*
3	*3*	*3*	*1*
4	*1*	*2*	*1*

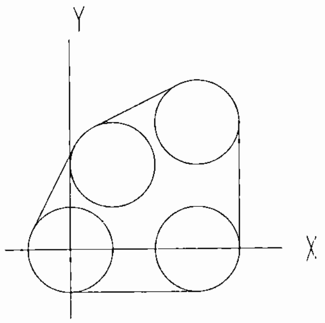


Fig. 4b. *Software-introduced segments defining profile.*

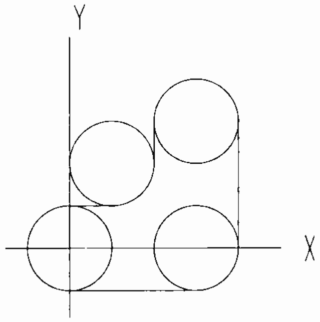


Fig. 4c. *Radius of circle 4 changed to -1.*

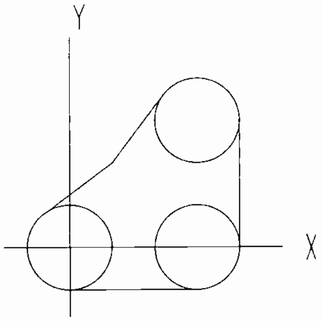


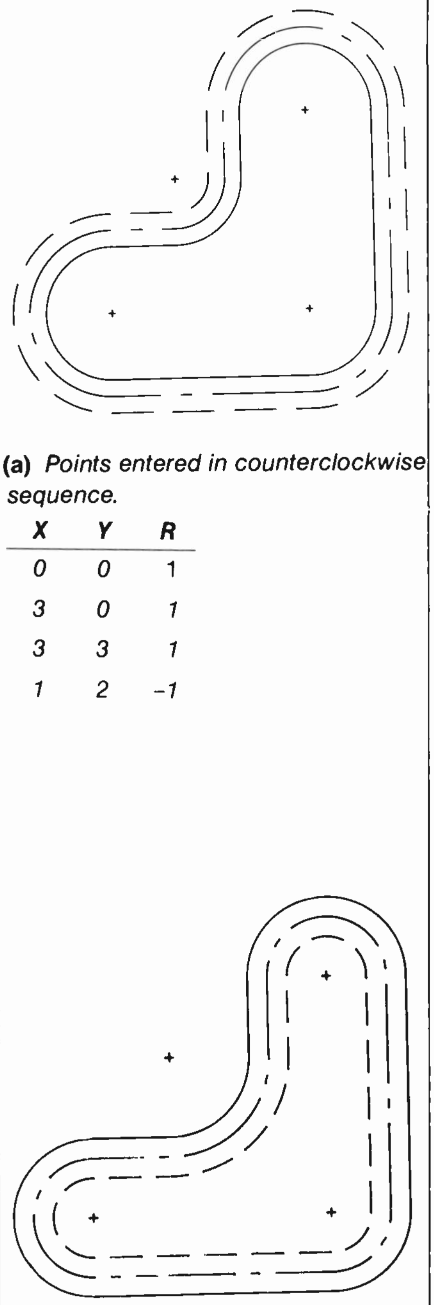
Fig. 4d. *Radius of circle 4 changed to 0.*

Profiles are defined by depth and the position and size of each arc in sequence. The depth of the profile is negative if cut from the bottom, and positive if cut from the top. Arcs defined with positive radii are convex when viewed from the center of the tool. Arcs defined with negative radii are concave (Fig. 4). The position of the center of the tool, relative to the profile, is determined by the order of input. If the arcs of a profile are entered in a counterclockwise sequence, the tool will cut outside the profile. Interior cutting occurs when the arcs are entered in clockwise sequence (Fig. 5). The software will insert segments tangent to consecutive arcs if needed. This relieves the designer or machinist of the need to calculate coordinates for the points of tangency. Arcs of zero radius are acceptable and are used to generate two consecutive line segments. Parts are designed and modified by entering, deleting, and changing the profiles and holes. If an operator can define a hole and a profile, he can define a part.

All data required for the drawing are taken directly from the design except the drawing number, date, notes, and labels. The drawing number, date, and notes are entered through the keyboard. Labels, which define dimensions, are entered through the user-definable keys (Fig. 6). Figure 3 is an example of a drawing produced by the system.

Like the drawing phase, all data needed for machining are taken from the design except the diameters and lengths of the tools, and the material type. These data are supplied by the machinist. The Flat Parts System calculates spindle speeds and feed rates, and it generates a program that is sent directly to the milling machine through a connecting wire.

In conclusion, the Flat Parts System is easy to use, reliable and inexpensive. Alternative designs can be visualized without recourse to drafting or manufacturing. Steps in the development of the part are automated wherever possible, feedback to the designer and machinist provides for modification during design and manufacturing, the possibilities for human error are minimized, and users never have to say, "Back to the drawing board."



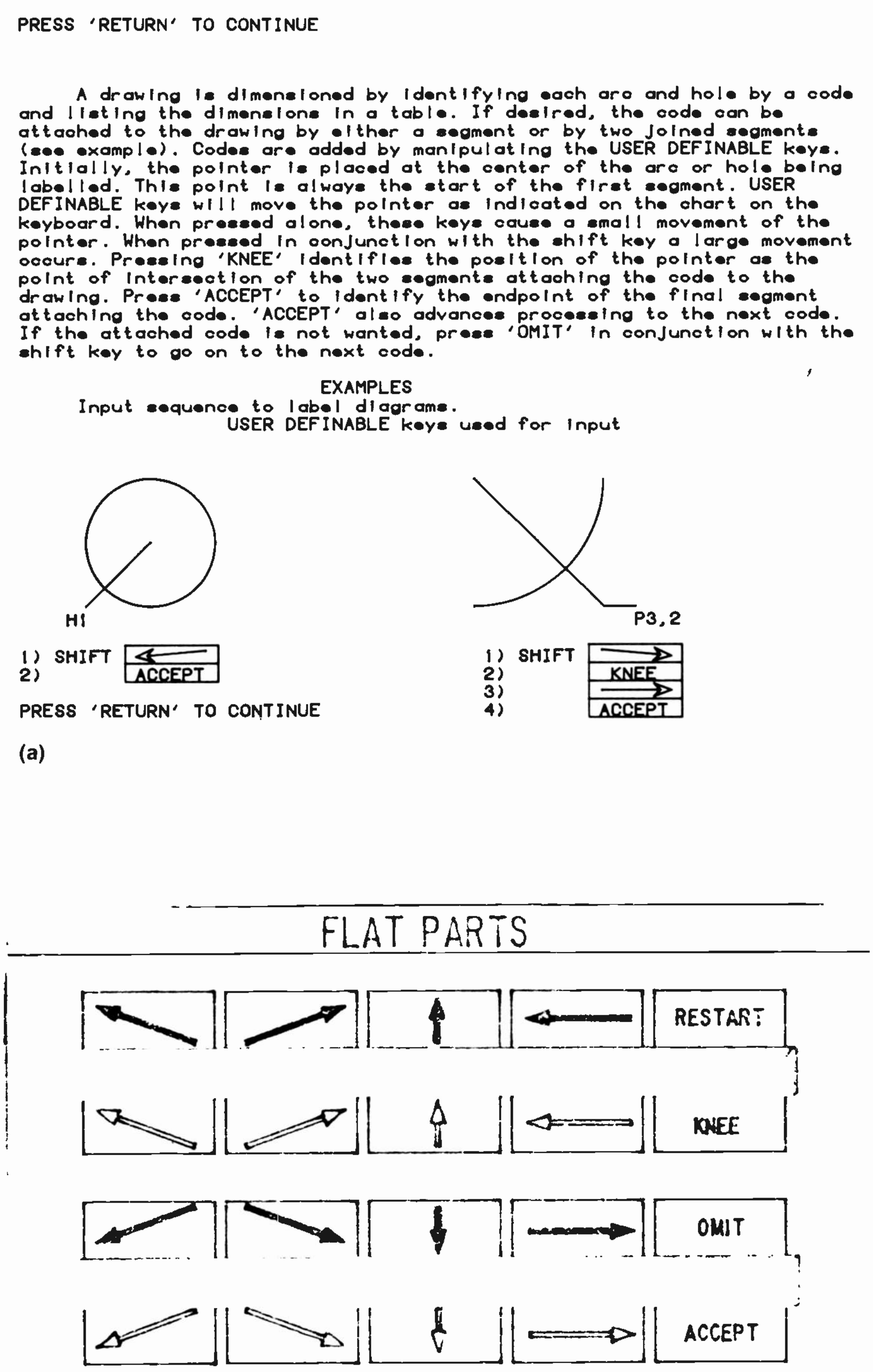
(a) *Points entered in counterclockwise sequence.*

X	*Y*	*R*
0	0	1
3	0	1
3	3	1
1	2	−1

(b) *Points entered in clockwise sequence.*

X	*Y*	*R*
0	0	−1
1	2	1
3	3	−1
3	0	−1

Fig. 5. *The solid line indicates the part outline, the line with two sizes of dashes indicates the center of the tool, and the dashed line indicates the outside of the tool.*



```
PRESS 'RETURN' TO CONTINUE


     A drawing is dimensioned by identifying each arc and hole by a code
and listing the dimensions in a table. If desired, the code can be
attached to the drawing by either a segment or by two joined segments
(see example). Codes are added by manipulating the USER DEFINABLE keys.
Initially, the pointer is placed at the center of the arc or hole being
labelled. This point is always the start of the first segment. USER
DEFINABLE keys will move the pointer as indicated on the chart on the
keyboard. When pressed alone, these keys cause a small movement of the
pointer. When pressed in conjunction with the shift key a large movement
occurs. Pressing 'KNEE' identifies the position of the pointer as the
point of intersection of the two segments attaching the code to the
drawing. Press 'ACCEPT' to identify the endpoint of the final segment
attaching the code. 'ACCEPT' also advances processing to the next code.
If the attached code is not wanted, press 'OMIT' in conjunction with the
shift key to go on to the next code.

                          EXAMPLES
     Input sequence to label diagrams.
                 USER DEFINABLE keys used for input

     H1                                                    P3,2

1) SHIFT                                   1) SHIFT
2)        ACCEPT                           2)           KNEE
                                           3)
PRESS 'RETURN' TO CONTINUE                 4)           ACCEPT
```

(a)

(b)

Fig. 6. *Preamble* **(a)** *displayed before labeling, and* **(b)** *user-definable key chart for the Flat Parts System.*

Computervision equipment and numerical control

William Hatter

Numerical control, well suited to small-lot production, has gained its reputation by substantially reducing setup time, which consumes unproductive man-hours. In some instances, it has even been more economical to manufacture one part by means of numerical control (NC) than by conventional methods. To be cost-effective, numerical control implies the use of computers to convert the visualization of the part into a milling-machine program. The generation of a correct numerical description of an object is essential. Dimensions must be exact, since all NC programming will be developed from information contained in this description. An operator, therefore, begins the NC process by converting engineering sketches into a three-dimensional computer model. He is aided in this step by a software package for mechanical drafting.

Having developed the model and verified its correctness with the designers, the operator consults with a machinist to establish raw stock (Fig. 1), toolpaths, and the sequence in which the part will be made. He is then ready to generate those paths on the computer and is guided by an interactive routine that asks for information about the type of tool, the number of base and side rough cuts, surfaces to be cut, tolerance, and so on. Entities are then selected in the proper sequence to define the toolpath (entities are the lines, arcs and

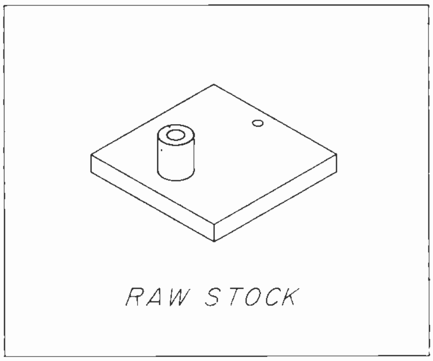


Fig. 1. *The shape of the raw stock is determined by the model maker. The one shown above was cut from brass alloy to 3.00 in. by 2.50 in. by 1.375 in. and then put into a lathe. There it was faced and the hub was turned to 0.625-in. diameter. The entire job could have been cut from a solid block; this is the model maker's decision.*

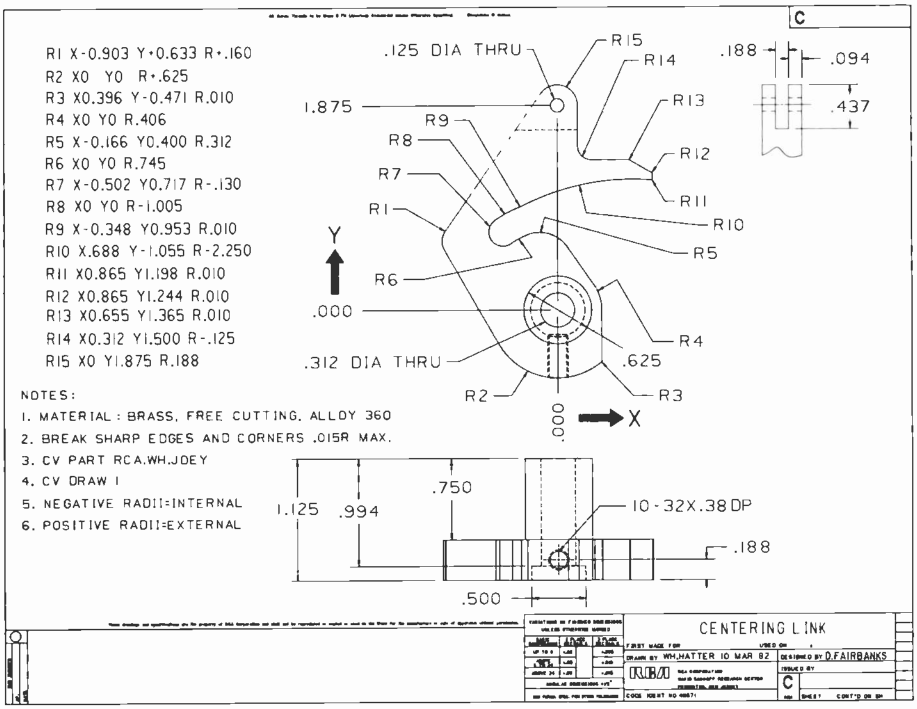
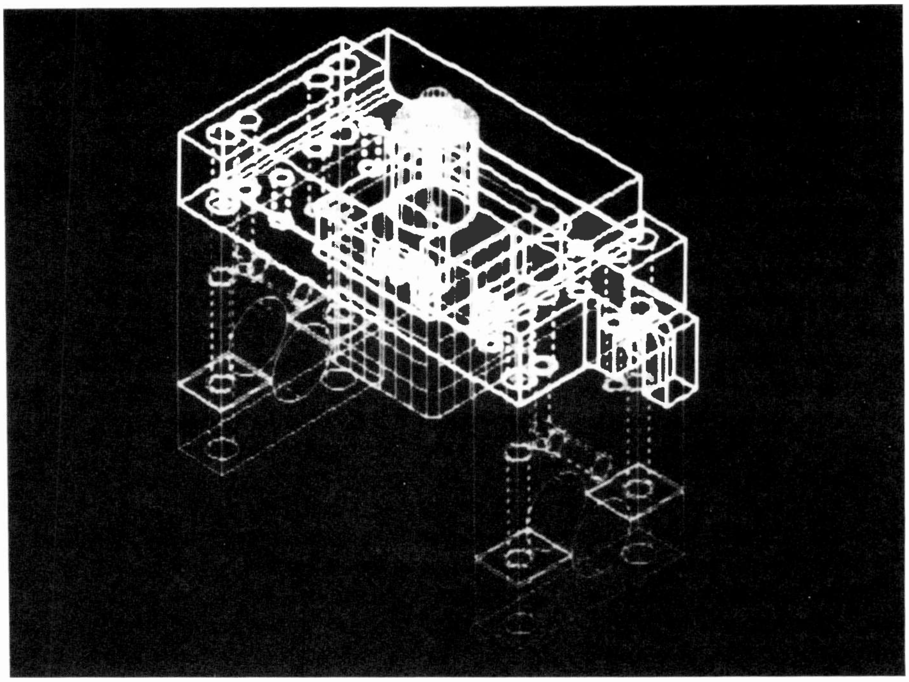


Fig. 2. *The centering link operates a slide on the diamond insertion fixture, used in the assembly of VideoDisc cartridges. In addition to being the main operating handle, it also is used to center the diamond electrode facet, before it is inserted into the stylus arm.*



A three-dimensional model can be built on the Computervision equipment, which can aid in the generation of toolpaths.

circles that were used to create the part.) Once all entities are selected, the system generates a numerical description of the toolpath that conforms to the defined geometry.

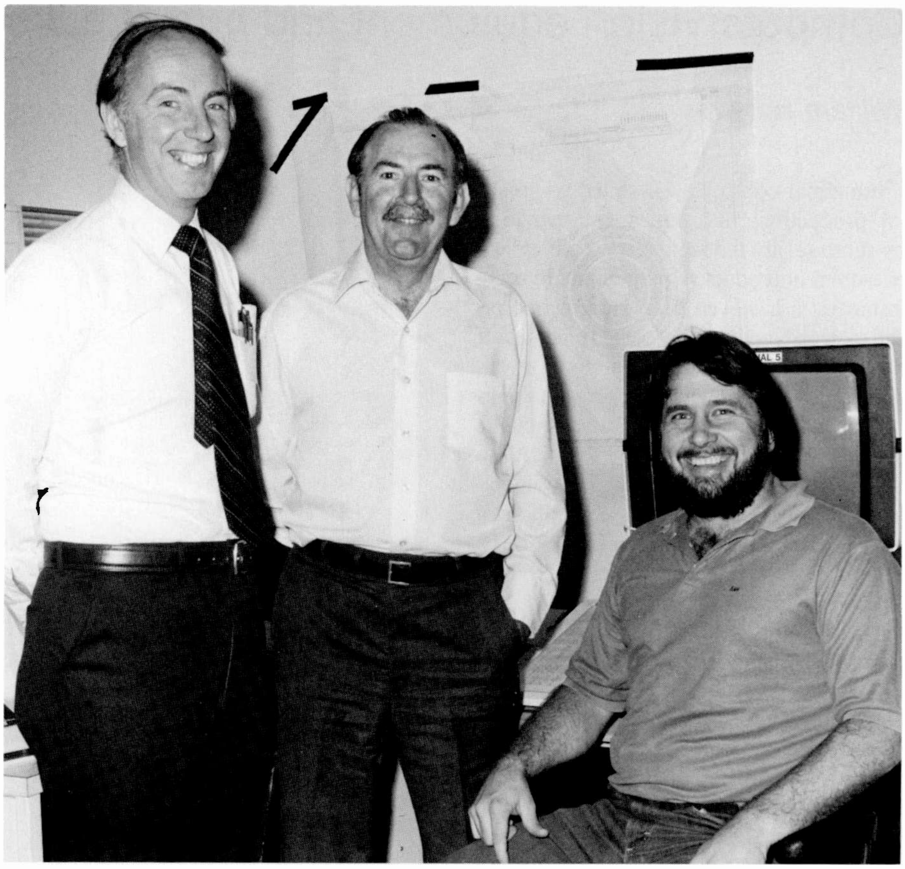
In the case of the part shown here (Fig. 2), after the toolpaths were completed and the part was ready for cutting, we encountered the toolmaker's nightmare, a design change. The engineer wanted to change radius 7 (Fig. 2) from −0.165 to −0.130. This meant the tool we were planning on using, a 0.375-in. diameter end mill, would be too large for the radius. Normally if this change were to occur on milling machines without computer assistance, the model maker would have to rekey his entire program manually after recomputing the path. This process, being lengthy, is subject to errors and is very time-consuming.

One of the features of the Computervision software was used to generate a new milling-machine program by specifying a new tool of 0.250-in. diameter and instructing the computer to regenerate the toolpath—a process taking only a few minutes.

After all toolpaths have been established, the order in which they are to be cut is selected. The milling machine will follow this sequence, and the machinist will be fully aware of his task because, in addition to his having made all machining decisions, the system will remind him of tools, feed rates, spindle speeds and other machining information at the time the part is to be cut.

The last step in the process is conversion of the Computervision toolpath description into milling-machine code, which is done in the model shop by a Tektronix 4052, a desk-top computer with graphics capability. Before loading the milling machine, the 4052 draws each toolpath on its display screen and prints a list of the tools to be used. If the toolpath exceeds the storage capacity of the milling machine's controller, the 4052 will act as controller and send one instruction at a time to the mill.

The process as established at RCA Laboratories leaves intact the benefits accruing from the skills of the engineer, the computer operator and the machinist, and allows accurate, top-quality parts to be produced in a minimal time.



Authors (left to right) Quardt, Reid-Green, and Hatter.

Keith Reid-Green is a Member of the Technical Staff in Electromechanical Systems group. He joined RCA Laboratories in December, 1978, in the Microsystems Research department. He has 25 years of experience in computer programming and analysis, with research interests in computer graphics and its application to personal computing and computer-aided design and manufacturing.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2180

Dennis Quardt is a Member of Technical Staff in the Electromechanical Systems group. He has worked at RCA Laboratories since September 1981. Research interests include applied mathematics and computer-aided instruction.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-3211

William Hatter is a Senior Technical Associate in the Electromechanical Systems group. He joined RCA Laboratories in October 1979. He has three years of experience as a CAD/CAM operator.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2685

J.S. Heffernan | G.E. Munk

Using software tools to enhance engineering productivity

In the future, structured test languages and computer-aided software engineering will be two advanced tools for enhancing productivity. Here's what Automated Systems software engineers are working on today.

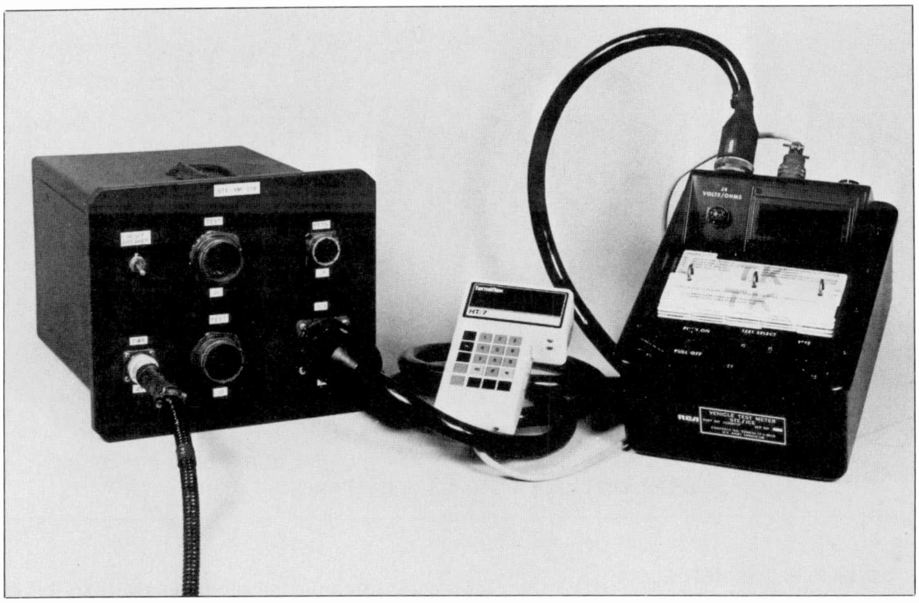
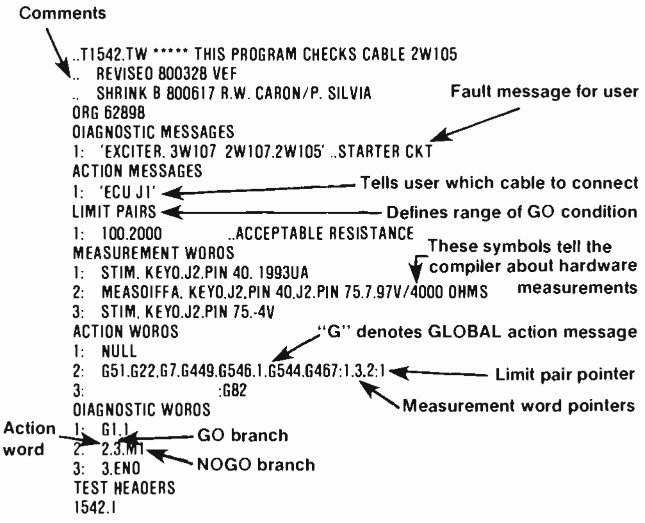


Fig. 1. Simplified Test Equipment for the Army's Abrams M1 tank and Bradley Fighting Vehicle System (STE-M1/FVS). The STE-M1/FVS is three separate units—the VTM (right), the SETCOM (middle), and the CIB (left). The VTM, Vehicle Test Meter, the main controller of the system, handles input from and output to the user and controls the analog measurement system. The SETCOM, set communicator, is a small hand-held terminal with 18 input keys and two 20-character output lines. This is used to enter the test number, guide the user through a diagnostic test, output fault messages, and to query the user for information. The CIB, Controllable Interface Box, provides the VTM with a 256-channel analog measurement system, which contains its own computer. The VTM communicates through a serial link to the CIB and commands the CIB to make a measurement and return the results.

In designing and supporting portable, microprocessor-based test sets for military vehicles, we have found that eighty percent of the development effort consists of software design, debug and test. This software work encompasses the development and maintenance of operating systems for hardware products, and the creation of applications software to test and troubleshoot vehicles such as the Abrams M1 tank and the Bradley Fighting Vehicle. With small sys-

Abstract: *This paper details the techniques used to improve engineering productivity during the creation of software packages for portable, microprocessor-controlled vehicle test systems. A test language called Diagnostic Flowchart Language (DFL) was created to reduce memory needs and avoid problems associated with assembly language coding. The authors discuss solutions to problems encountered when creating a language. Finally, the authors examine areas where productivity could be increased, through software tools, in the future.*

Final manuscript received December 14, 1982
Reprint RE-28-1-6



```
..T1542.TW ***** THIS PROGRAM CHECKS CABLE 2W105
..   REVISE0 800328 VEF
..   SHRINK B 800617 R.W. CARON/P. SILVIA
ORG 62898
OIAGNOSTIC MESSAGES
1:   'EXCITER. 3W107  2W107.2W105' ..STARTER CKT
ACTION MESSAGES
1:   'ECU J1'
LIMIT PAIRS
1:   100.2000          ..ACCEPTABLE RESISTANCE
MEASUREMENT WOROS
1:   STIM. KEY0.J2.PIN 40. 1993UA
2:   MEASOIFFA. KEY0.J2.PIN 40.J2.PIN 75.7.97V/4000 OHMS
3:   STIM. KEY0.J2.PIN 75.-4V
ACTION WOROS
1:   NULL
2:   G51.G22.G7.G449.G546.1.G544.G467:1.3.2:1
3:                      :G82
OIAGNOSTIC WOROS
1:   G1.1
2:   2.3.M1
3:   3.EN0
TEST HEAOERS
1542.1
```

Comments
Fault message for user
Tells user which cable to connect
Defines range of GO condition
These symbols tell the compiler about hardware measurements
"G" denotes GLOBAL action message
Limit pair pointer
Measurement word pointers
Action word
GO branch
NOGO branch

COMMENTS ON THE FLOWCHART ARE OELIMITEO BY A LEFT BRACKET.
THE SYMBOL IN THE FLOWCHART MAP TO THE FOLLOWING OFL CONSTRUCTS.

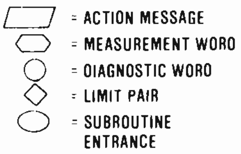


Fig. 2a. This is an example of a short DFL test's source code. The code corresponds to the diagnostic flowchart of Fig. 2b. Diagnostic messages are displayed to the user when the fault is found. Action messages consist of text that directs the user to perform operations, for example, to connect cables. Limit pairs denote the acceptable range for a GO or PASS condition. Measurement words define the hardware channels to measure and the type of measurement to take. An action word defines a composite of action messages, measurements to take, and measurement limits. Finally, diagnostic words control the flow of the test.

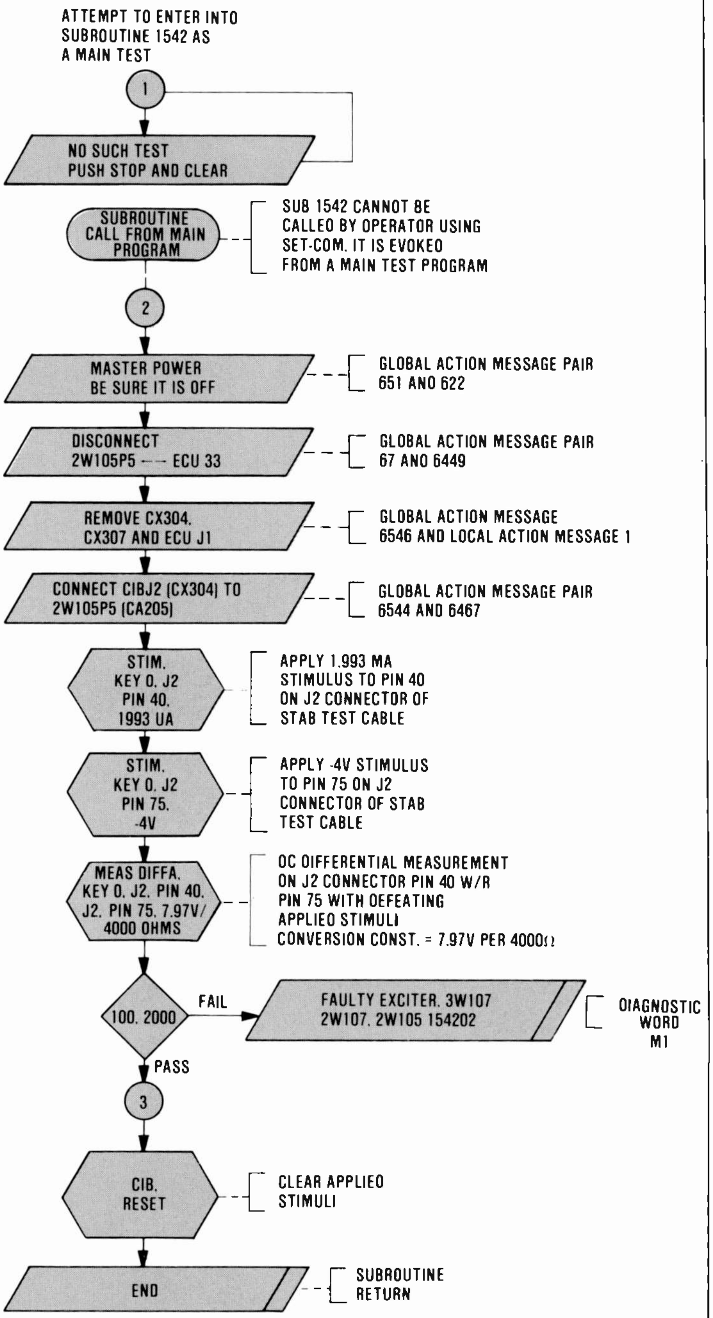

ATTEMPT TO ENTER INTO SUBROUTINE 1542 AS A MAIN TEST
1
NO SUCH TEST PUSH STOP AND CLEAR
SUBROUTINE CALL FROM MAIN PROGRAM
SUB 1542 CANNOT BE CALLEO BY OPERATOR USING SET-COM. IT IS EVOKEO FROM A MAIN TEST PROGRAM
2
MASTER POWER BE SURE IT IS OFF
GLOBAL ACTION MESSAGE PAIR 651 ANO 622
DISCONNECT 2W105P5 — ECU 33
GLOBAL ACTION MESSAGE PAIR 67 ANO 6449
REMOVE CX304. CX307 AND ECU J1
GLOBAL ACTION MESSAGE 6546 ANO LOCAL ACTION MESSAGE 1
CONNECT CIBJ2 (CX304) TO 2W105P5 (CA205)
GLOBAL ACTION MESSAGE PAIR 6544 AND 6467
STIM. KEY 0. J2 PIN 40. 1993 UA
APPLY 1.993 MA STIMULUS TO PIN 40 ON J2 CONNECTOR OF STAB TEST CABLE
STIM. KEY 0. J2 PIN 75. -4V
APPLY -4V STIMULUS TO PIN 75 ON J2 CONNECTOR OF STAB TEST CABLE
MEAS DIFFA. KEY 0. J2. PIN 40. J2. PIN 75. 7.97V/ 4000 OHMS
OC OIFFERENTIAL MEASUREMENT ON J2 CONNECTOR PIN 40 W/R PIN 75 WITH OEFEATING APPLIEO STIMULI CONVERSION CONST. = 7.97V PER 4000Ω
100. 2000
FAIL
FAULTY EXCITER. 3W107 2W107. 2W105 154202
OIAGNOSTIC WORD M1
PASS
3
CIB. RESET
CLEAR APPLIEO STIMULI
END
SUBROUTINE RETURN

Fig. 2b. Typical diagnostic flowchart comments are shown within brackets to the left of the flowchart map.

tems, we found that programming in assembly language with no advanced software aids was practical. However, with large systems, we found it necessary to develop a test language compiler.

In the future, we see two ways of creating advanced software tools that realize large gains in productivity. They are the development of a structured test language and the development of computer-aided software engineering. The latter route produces machine readable code from a graphics file that represents a flowchart. This scenario contrasts sharply with that of engineers "hand assembling" code only ten years ago.

Software problems

Originally, the test systems developed for military vehicles were coded entirely in RCA 1802 assembly language. The only convenience used by programmers was a pseudo-instruction interpreter. The major problem with programming in assembly language is the time it takes to produce a "bug-free" operating system program. It is estimated that it takes from two to ten times longer to code an equivalent program in an assembly language than it does to code in a Higher Level Language (HLL). Assembly language programs are difficult to read and modifying the software is time consuming, even for the original programmer. Furthermore, assembly langauge coding requires intimate knowledge of the machine architecture and the input/output subsystem. These problems become intractable as a system grows in size. Additionally, the 1802 cross assembler is written in unstructured FORTRAN IV, making it difficult to support and modify.

Diagnostic Flowchart Language

The Army's M1 tank contains numerous electrical subsystems that are tested on the STE-M1/FVS test set, seen in Fig. 1. The major subsystems of the tank are: the turret and hull, laser rangefinders, ballistics computers, and a night vision system. If assembly language coding is used to program the test set for this vehicle, the memory requirements would be close to one megabyte. The high cost of producing and building a test set containing a megabyte of EPROM (erasable programmable read only memory), and then managing the assembly language code, led to a method of programming the tests in a more memory-efficient manner. This effort resulted in the development of a new application test language. The language is called Diagnostic Flowchart Language (DFL). DFL's object code is interpreted in the target test-set system. Its use has reduced the memory required for application code by a factor of five to one over assembly language and has resulted in excellent memory efficiency when compared to equivalent assembly language applications programs. Moreover, with DFL applications, programmers no longer need to have intimate knowledge of the measurement system hardware and the operating system code of the test set. This decreases the applications programmer's time spent on creating or modifying a test. Applications tests are defined using a standard flowcharting method. The constructs of DFL follow the structure of the flowchart in a one-to-one manner. Therefore, once a flowchart test has been drawn, the coding of the test becomes a simple task. The correlation between a diagnostic flowchart and DFL is illustrated in Fig. 2. A sample DFL object record is shown in Fig. 3. DFL has been proven to decrease memory size and programming time, while increasing program structure, ease of program modification, and program-software-configuration management.

Test word generator and linker

The creation of a relocatable compiler and linker for DFL results in a sevenfold improvement in turnaround time to build systems and less rote work for the project librarian. While the DFL language increased the productivity of application programmers, the fact that it was translated to absolute addresses posed problems. The DFL language requires a compiler and linker. The test word generator (TWG) compiler was originally written using absolute addresses, which start applications tests at a fixed point in the target system memory. A linker had to be written to join the tests together in a binary tree by test number. When a test was modified and required more memory, all the tests that followed the modified test had to be re-compiled with new starting addresses to accommodate the new test. This required vast amounts of computer time to reconfigure a total test system. To correct the shortcomings of the absolute TWG and compiler, a new TWG and linker were created to allow the application test code to be relocatable. Both of these have been developed by RCA at private expense. The immediate advantage is that any test can be modified, and become shorter or longer, without requiring any other test to be re-compiled. The linker also became much more automated, providing information to the librarian that previously had been generated by hand. An example of the relocatable linker's output is shown in Fig. 4.

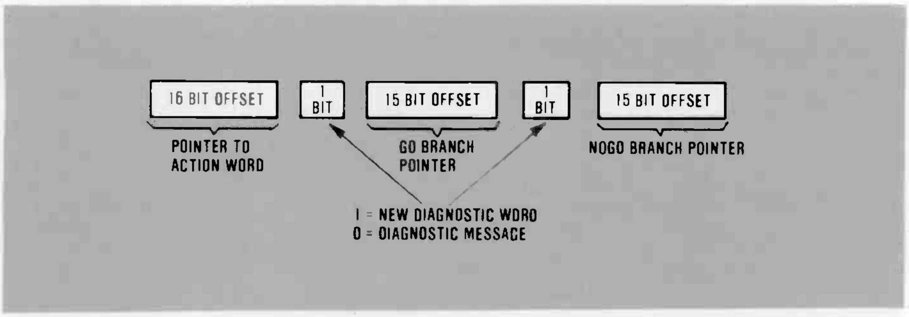
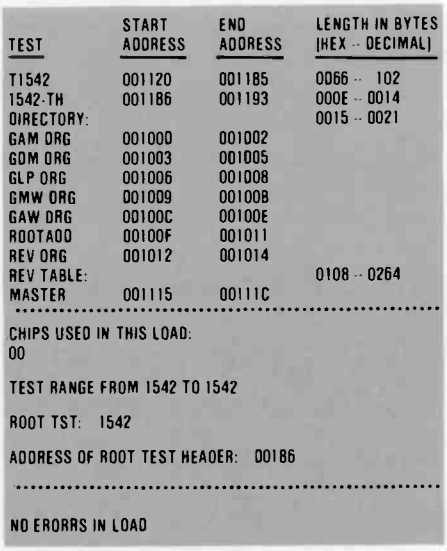


Fig. 3. Diagnostic Word Object Code Format. The diagnostic word is the basic test step of the Diagnostic Flowchart Language. The object code is in bit-packed form. The first record of the object code is a 16-bit pointer to an Action Word, which describes the user output, the measurements to be taken, and a limit pair for comparison. The comparison results determine whether a GO or NOGO branch will be executed. The single-bit records determine if the branch will be to a new Diagnostic Word or to output a Diagnostic Message. The 15-bit records contain a pointer to either the new Diagnostic Word to be executed, or to the Diagnostic Message to be displayed. The interpretation of the branch pointer is dependent upon the previous 1-bit record.

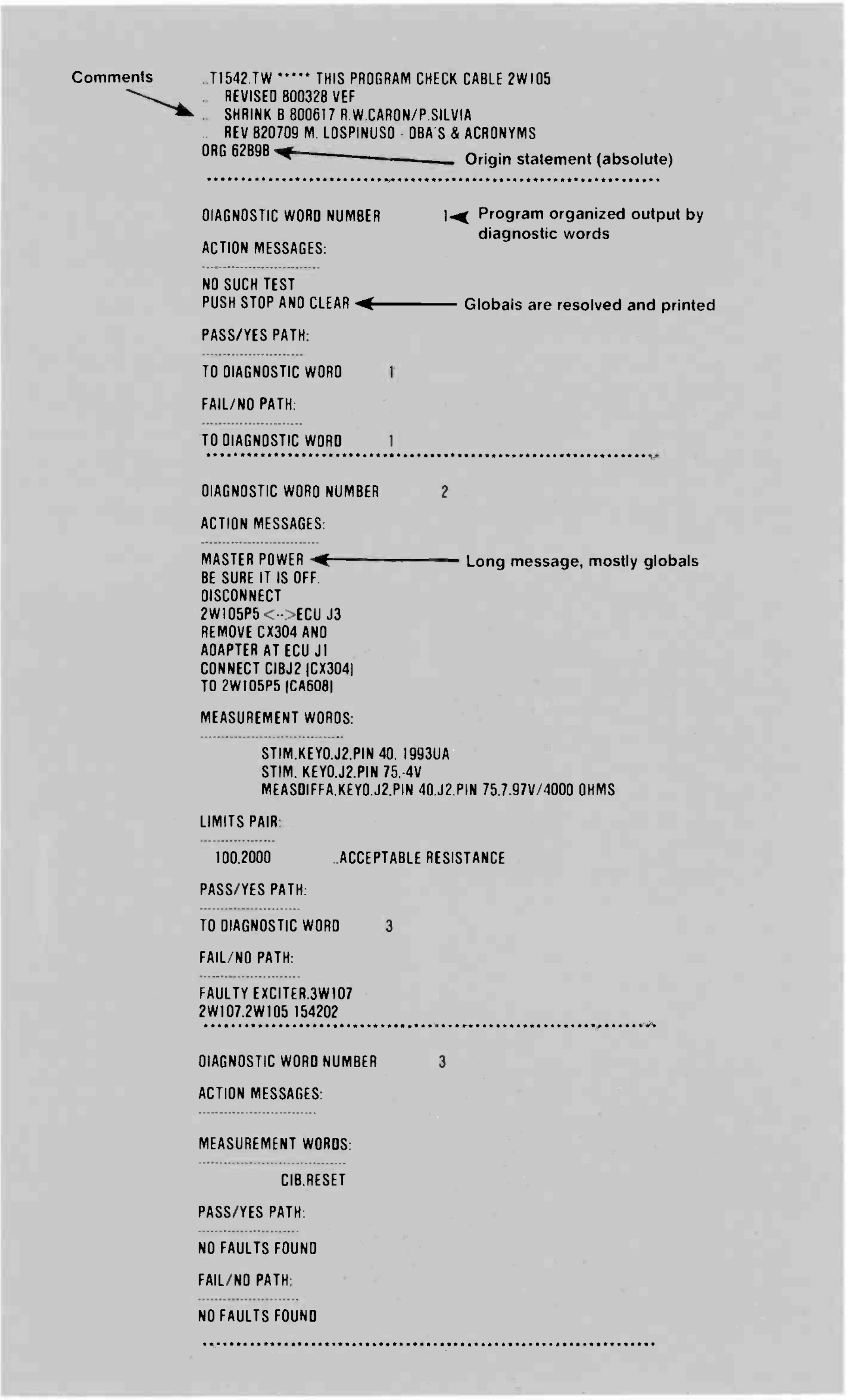


```
               START      END        LENGTH IN BYTES
TEST           ADDRESS    ADDRESS    (HEX -- DECIMAL)

T1542          001120     001185     0066 --  102
1542-TH        001186     001193     000E -- 0014
DIRECTORY:                           0015 -- 0021
GAM ORG        001000     001002
GDM ORG        001003     001005
GLP ORG        001006     001008
GMW ORG        001009     00100B
GAW DRG        00100C     00100E
ROOTADD        00100F     001011
REV ORG        001012     001014
REV TABLE:                           0108 -- 0264
MASTER         001115     00111C
.....................................................
CHIPS USED IN THIS LOAD:
00

TEST RANGE FROM 1542 TO 1542

ROOT TST:   1542

ADDRESS OF ROOT TEST HEADER:   00186

.....................................................

NO ERORRS IN LOAD
```

Fig. 4. DFL improved linker output. The original DFL linker provided the user with a table of the chips used by the link and a map of the open link pointers. This required the user to hand-link separate links. The improved linker provides automatic linking of separate links. The improved linker also outputs information previously generated by hand, including: test numbers, starting and ending addresses, length of the test in hex and decimal, range of tests, root test number and root test start address.

Special tests

DFL has allowed vast improvements over assembly languages in several areas, but there are occasions when it does not support a special algorithm needed for testing, or does not have the speed necessary for implementing a test on a particular piece of hardware. In these cases, special tests written in assembly language, but callable by DFL, are used. To support these assembly language special tests, a pre-assembler for 1802 assembly code was created, called PMAS. This pre-assembler does processing on lines within the assembly code that were marked for the PMAS to handle. These lines of tagged code contain parameters specified in standard DFL notation, such as limits and the channels used for the measurement. This allows a special test programmer to define parameters to different system-supplied routines in stan-



```
..T1542.TW ***** THIS PROGRAM CHECK CABLE 2W105
..   REVISED 800328 VEF
..   SHRINK B 800617 R.W.CARON/P.SILVIA
..   REV 820709 M. LOSPINUSO - DBA'S & ACRONYMS
ORG 62B9B
*************************************************************

DIAGNOSTIC WORD NUMBER          1

ACTION MESSAGES:
.........................
NO SUCH TEST
PUSH STOP AND CLEAR

PASS/YES PATH:
.........................
TO DIAGNOSTIC WORD        1

FAIL/NO PATH:
.........................
TO DIAGNOSTIC WORD        1
*************************************************************

DIAGNOSTIC WORD NUMBER          2

ACTION MESSAGES:
.........................
MASTER POWER
BE SURE IT IS OFF.
DISCONNECT
2W105P5<-->ECU J3
REMOVE CX304 AND
ADAPTER AT ECU J1
CONNECT CIBJ2 (CX304)
TO 2W105P5 (CA608)

MEASUREMENT WORDS:
...............................
        STIM.KEYO.J2.PIN 40. 1993UA
        STIM. KEYO.J2.PIN 75.-4V
        MEASDIFFA.KEYO.J2.PIN 40.J2.PIN 75.7.97V/4000 OHMS

LIMITS PAIR:
................
  100.2000          ..ACCEPTABLE RESISTANCE

PASS/YES PATH:
.........................
TO DIAGNOSTIC WORD        3

FAIL/NO PATH:
.........................
FAULTY EXCITER.3W107
2W107.2W105 154202
*************************************************************

DIAGNOSTIC WORD NUMBER          3

ACTION MESSAGES:
.........................

MEASUREMENT WORDS:
...............................
           CIB.RESET

PASS/YES PATH:
.........................
NO FAULTS FOUND

FAIL/NO PATH:
.........................
NO FAULTS FOUND
*************************************************************
```

Comments

Origin statement (absolute)

Program organized output by diagnostic words

Globals are resolved and printed

Long message, mostly globals

Fig. 5. To aid in the validation and reading of tests, the Munkchart program was created. The output seen above displays the action that each test step will perform and the next paths that the test step may take. This allows the test programmer to quickly check test flow and proper actions. This output is much more readable than the source code of Fig. 2, from which it was generated.

dard DFL format using symbolic values as opposed to binary numbers. The in-line conversion eliminates the need of the special test programmer to have knowledge of what preset values the hardware requires to set up measurement channels and of the binary format of values returned from the hardware. This decreases the coding time of the special tests, and reduces errors associated with hardware-dependent values.

Munkchart Generator Program

The DFL language is difficult to read and debug, so a program called the Munkchart Generator was created. This program is designed to supply the DFL programmer with a simulation of the test on paper, giving an easy-to-read representation of the source code as seen in Fig. 5. The Munkchart Generator accepts an application test written in DFL as input, and generates an output that describes the actions that each test step will enact. The first use of the Munkchart Generator is to reduce the comparison time between the source code and the flowchart by two thirds. Previous to the Munkchart Generator program, the DFL programmer had to have seven documents to check the code for accuracy: the test source code, the test flowchart, and one copy of each of the five global DFL source files. To check if a global reference in the source code was correct, the appropriate global file had to be searched manually for the reference. The Munkchart Generator provides the DFL programmer with all the information in one document that can be compared directly with the flowchart, eliminating the need for the global source files. The second use of the Munkchart Generator has reduced the validation time of tests on the vehicle by an average of 25 percent. When a test is validated on a vehicle, it is compiled and linked into an applications system which is then programmed into EPROMs. These EPROMs are then placed in a test system (that is, the STE/M1) and the test is run on the vehicle it is intended to test. Faults in the test program are then discovered, corrected, and run through the vehicle validation cycle again. The output from the Munkchart Generator is used to validate minor changes to a test, without running the test on the vehicle, and is now an accepted standard method of validation.

Flowchart Generator Program

All application software needs a flowchart describing each test. A change in a flowchart requires a change in the source code, and vice-versa. A large amount of programmer time is involved in maintaining this one-to-one correspondence between the code and the flowchart. So, a program called the Flowchart Generator (privately developed by RCA) was designed. This program takes the DFL source code as an input, and draws the flowchart that reflects the code. This sophisticated program draws a flowchart using the same symbols that the application programmer uses for each part. It also creates a representation of the paths that may be followed through the test. The program generates a flowchart of the applications code in a fraction of the time it takes to create the same flowchart using current drafting methods. This allows the applications programmer to have rough sketches of the flowchart, code the test, and create a final version of the flowchart once the code has been debugged. An actual computer-generated flowchart is shown in Fig. 6. The Flowchart Generator program also allows fast updating of existing code and flowcharts.

Future improvements

The test word generator (TWG) has proved difficult to maintain because commonly changed parameters are embedded in the source code. Each hardware change results in a new version of the test word generator. Furthermore, the development of a new TWG typically takes two man-years. This is a process that could be automated. There is an ongoing effort to do just this, and the test word generator for the next generation of vehicle systems will be generated by a program. The specification for a test word generator will be found in a file that is written in STE Meta Language (SML). This is a formal language created at Automated Systems that will specify a target language source code and object code format (specifically, a test word generator). The output of the SML program generator will be a PASCAL program. When compiled and linked on the VAX, the program will be a runnable test word generator. This program will significantly reduce the programmer overhead associated with the maintenance of test word generators. In the future, the costly development of new test word generators will be eliminated.

Expandability of DFL

To expand DFL so that special tests do not have to be programmed in assembly language, a number of extensions that allow new constructs within the DFL language have been created that provide a more structured language and greater measurement flexibility. These constructs include: sitting in a tight loop until a measurement is above or below a programmed value, creating arrays of measurements for further analysis, measuring two channels simultaneously, and triggering a measurement of one channel depending upon the value measured on another channel. Another feature to be implemented will be time-stamped measurements.

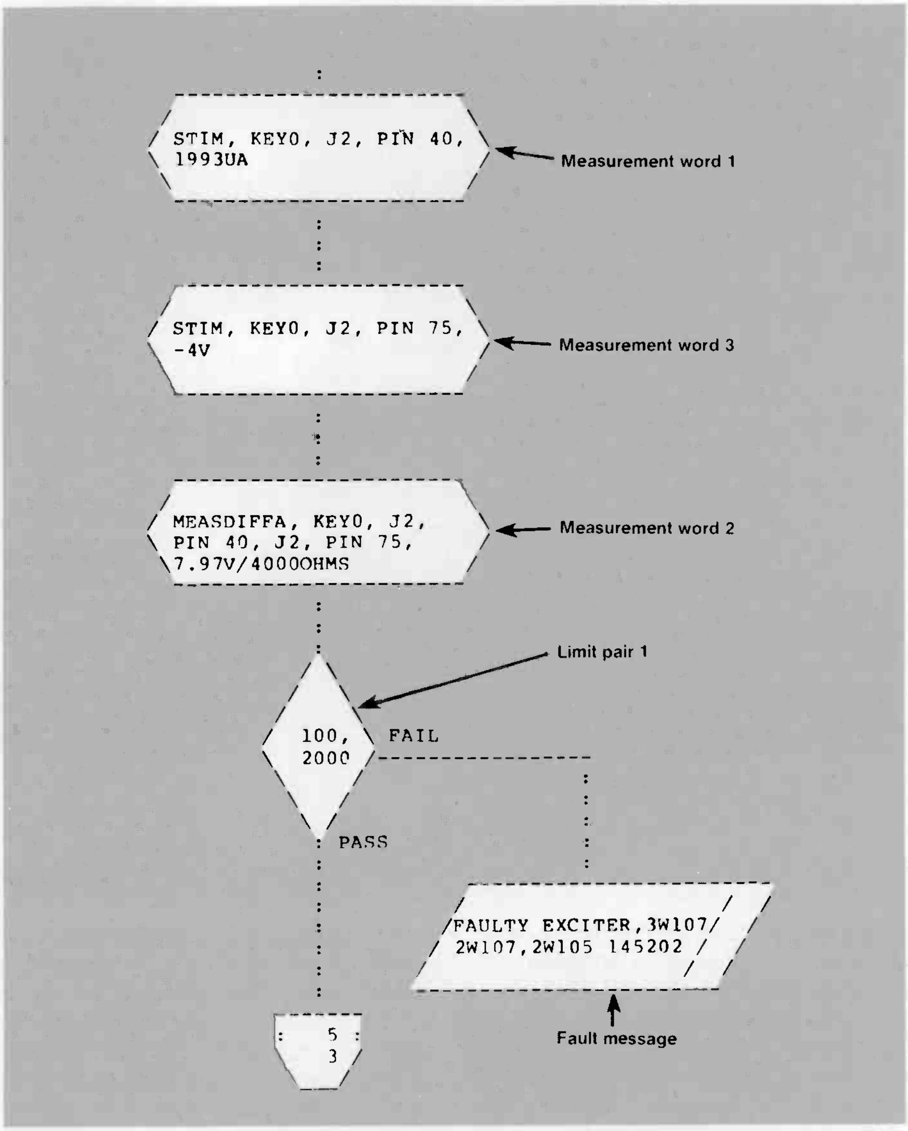
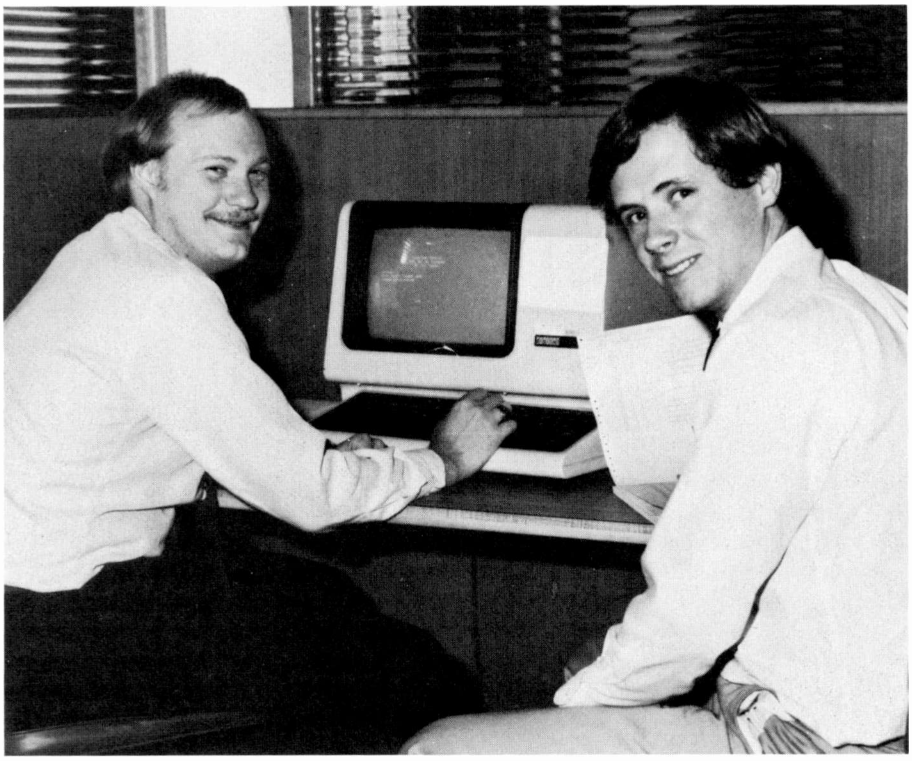


Fig. 6. Portion of a computer-generated flowchart. The program looks at the DFL source code and constructs the print file from information contained in the source file. This program enables the engineer to easily check that the source code and flowchart match. This represents part of the test shown in Fig. 2.

Future trends

As a long-term goal, we envision using a computer-aided software engineering system to build applications programs. Currently, the applications database for a complex vehicle requires large numbers of diagnostic flowcharts (that is, eight-hundred pages for the M1). The flowcharts are released to the users as documentation for the software product. These tedious tasks presently being done by engineers and draftsmen could be done by computer. One such task is the actual pen-and-pencil creation and correction of flowcharts. When a change is called for, the flowchart must be pulled and manually corrected. This process could be done by a flowchart editing program. This program could work using Precision Visual's DI3000 device-independent graphics package with VT100s, VT125s, or raster graphics stations. If creating a new flowchart, it would present a series of options to the user (for example, "draw diamond"). The user could then use the cursor to move around and put words and symbolic information on the screen and into a file. Syntax checking could be done at this stage. The program would output a file that would contain a complete description of the flowchart in a device-independent manner. In this way, the tedious tasks of creating and modifying flowcharts could be automated. Furthermore, it would then be possible to inte-



Authors (left) Munk and Heffernan.

Gregory Munk is a member of the technical staff of Vehicle Test Systems Engineering at RCA Automated Systems. He is a graduate of the State University of New York at Buffalo where he received a BSEE. Greg joined RCA in June of 1979 where he has worked on many hardware and software projects.
Contact him at:
RCA Automated Systems
Burlington, Mass.
TACNET: 326-2421

John Heffernan is a member of the technical staff of Vehicle Test Systems Engineering at RCA Automated Systems. He has worked on software, hardware and computer-support projects since joining RCA in 1981. John received a BSEE from Tufts University in 1981.
Contact him at:
RCA Automated Systems
Burlington, Mass.
TACNET: 326-2870

grate these flowcharts into a database, making retrieval of information or resources of existing flowcharts easy.

Currently, after flowcharts are manually created, they must be translated into Diagnostic Flowchart Language (DFL). However, since the flowchart could be completely descripted in a file created by the flowchart editor, it would be possible to bypass the manual translation into DFL source code and have the computer do it at less expense with better accuracy. This step would also eliminate problems that arise when flowcharts and DFL source code do not match and discrepancies arise. In the far future, simulation of tank electronics might be accomplished via a raster graphics system, and the time-consuming task of manual fault insertion, in order to verify diagnostics, would not have to be done on the vehicle itself. In summary, graphics hardware and software will be used to further enhance engineering productivity.

Conclusions

The major lesson we have learned over the years in producing large software systems is that support software must grow and expand along with the size and complexity of the system under test. Clearly, assembly language coding techniques become impractical for systems with over a thousand lines of code. In our situation, we developed a compiler, together with associated support software, that is memory efficient and that maps the structure of diagnostic flowcharts directly. While such a move increased productivity substantially, computer-aided software engineering, structured programming and automatic code generators will change the whole approach to programming now employed. These exciting new fields provide a vast potential for further decreasing software development time.

H.O. Hook | J.P. Beltz | D.P. Bortfeld
P. Kuczer | W.C. Saunders | A. Mavrodin

A computerized system for gun parts measurement

Accurate knowledge of the shape of the focussing lens parts of the electron gun is vital to the design and production of improved color picture tubes. RCA researchers found a way to measure these parts.

Abstract: *The part-height measuring system can measure parts within a cube 1.8 inches on a side to an accuracy of approximately 0.0001 inch. Under computer control, parts are measured in one-tenth of the time taken by the larger commercially available parts-measurement machines. The software written for this machine customizes it for measurement of G3 and G4 parts for color picture tube guns. The hardware can measure the height of any portion of a part accessible from the top. For a typical part requiring measurement at 800 to 1500 points, the measurement time is 5 to 8 minutes.*

All color-picture-tube electron guns contain stamped sheet-metal parts whose surface contours and apertures shape the electric fields so as to form the electron beam on the screen. Constant demand for improved resolution at high brightness over the entire picture area has led to a generation of weak, low-aberration lens designs in which the surface contour plays an increasingly important role. We describe here a computerized system designed particularly to measure the surface contour of a metal part to high accuracy (~0.1 mil) at moderate speed. We will discuss its use to measure the blend radius of a standard precision in-line (PI) gun part (see sidebar, page 45) although more complicated experimental parts have been measured with the system.

Present production-measuring techniques use dial-indicator gauges to measure local surface deviations and optical comparators on which the blend radius can be estimated by depth-of-field methods. Besides having limited accuracy, these gauging techniques do not define the surface profile unambiguously, because the two systems lack a common reference plane. The new system overcomes both of these limitations.

Gun parts can also be measured by means of standard coordinate-measuring machines—sophisticated, computerized three-axis measurement machines capable of very high accuracy. However, such systems can take from 30 minutes to an hour to measure the same number of X-Y-Z locations that the parts measurement system does in five to eight minutes.

Finally, although the machine could measure many kinds of parts, the present software customizes the system for use with gun parts. This graphics-based software is specific to the needs of electron-gun design engineers. The software package not only simplifies data input, but it aids in the analysis of the thousand or more data points measured on a typical part surface.

In what follows, we first describe the mechanical and electronic design of the system. We then show how the custom software aids our use of the system, give some of the rationale for the data reduction, and finally show some of the results.

General description

Figure 1 is a photograph of the engineering system built for Mount Development Engineering in Lancaster. The massive granite base was chosen to provide an accurate and stable support for the motor-driven stages. The electronic drive and interface equipment is mounted beneath the granite base.

Figure 2 represents the equipment schematically. The X-, Y- and Z-stages all have 0.1-mil incremental motion when their stepping motors are driven in the "half-step" mode. The measurements are made by moving the part to the desired location using the X- and Y-stages; the part height is then measured by lowering the linear variable differential transformer (LVDT) probe (see sidebar, page 45) and recording the result. The Z-stage moves the LVDT so it will be in its linear range when the probe touches the part. The system can measure any surface that can be seen by looking straight down on the part.

If the surface measured is not horizontal, a side thrust is put on the probe, which can lead to errors. The LVDT we use (Pneumo Precision) has very precise and stiff air bearings. This relatively short, stiff

Final manuscript received October 21, 1982.
Reprint RE-28-1-7

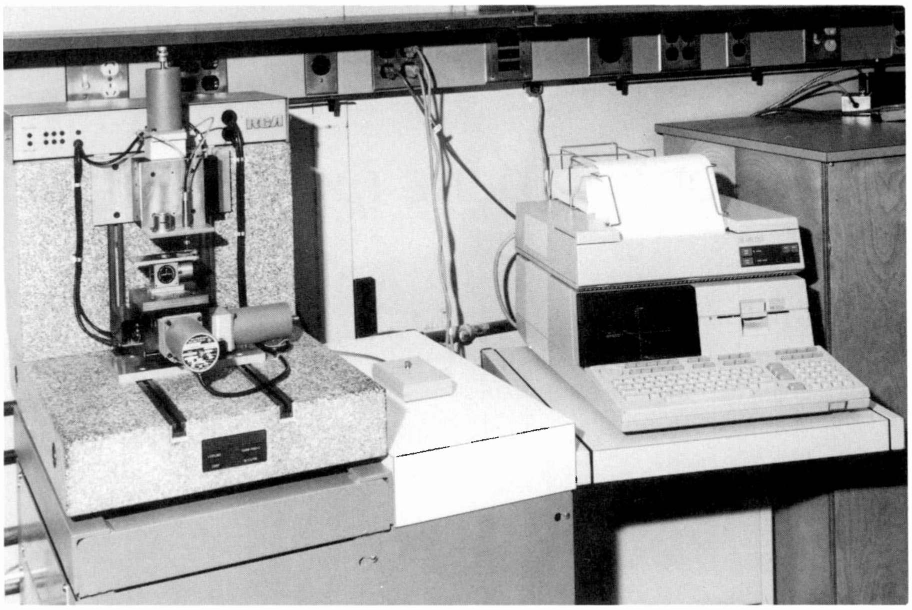


Fig. 1. *A heavy granite base for the measuring tables provides good stability, accuracy and repeatability. The desk-top computer is powerful enough to control the measurement process and reduce the data at the same time.*

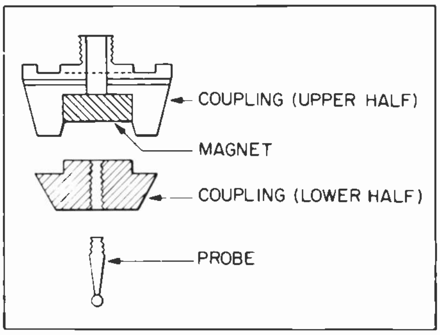


Fig. 3. *A coupling using a rare earth cobalt magnet is rigid enough to hold the measuring tip accurately, but will break away before the tip breaks.*

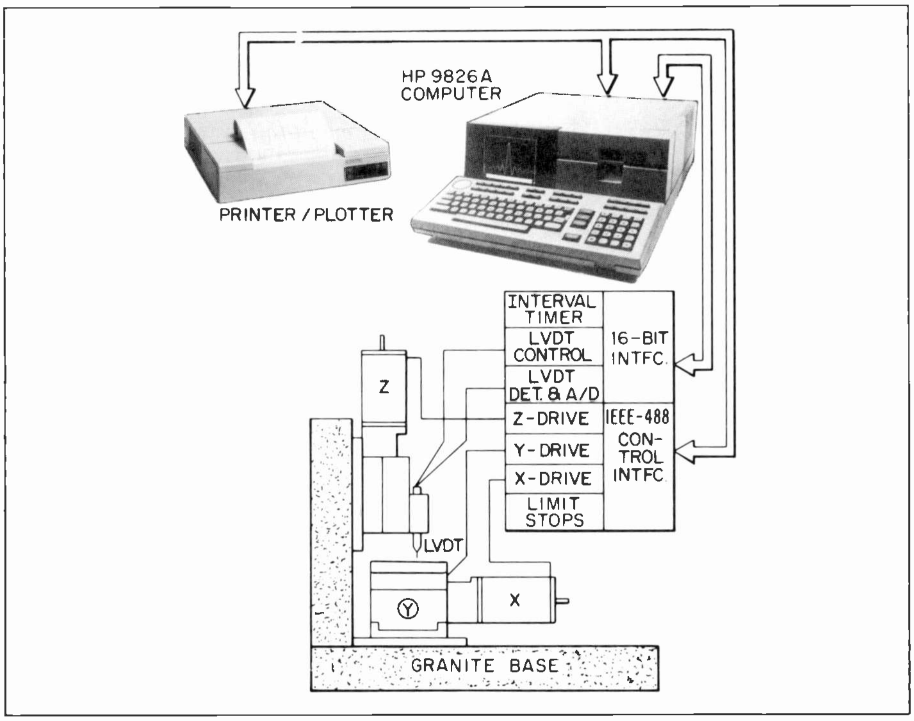


Fig. 2. *Two data busses handle the information flow between the computer and the control interfaces. The IEEE-488 bus also is the communication channel to the printer-plotter.*

probe allows us to measure surfaces tilted >80° from horizontal.

Practical measurement requires a measuring probe that is small compared to the features to be measured. The 15-mil ball-tip probe is easily broken or bent; therefore, some protective arrangement is needed. The electronic limit stops described below protect against excessive end thrust. Figure 3 is a cross-sectional view of the magnetic holding device. It is stiff enough to resist the side thrust of measuring steep slopes, but will break away before the tip is damaged.

The surface of the X-Y stage has three pins accurately placed to locate interchangeable parts holders. Each part will have its own holder machined to place it in the desired position (usually centered on the stage) for measurement.

The controller is an HP9826 computer with built-in graphics capability, a range of I/O options, and sufficient computer power to do the data reduction in a reasonable amount of time, all in a convenient desk-top package. The associated printer-plotter provides hard copy as required.

Computer interface

The computer controls and monitors the measurement process through two communication channels. The IEEE-488 bus communicates with the Modulynx® motor controllers, the limit stops, and the printer-plotter. The 16-bit interface communicates with the LVDT and the custom interval timer.

Modulynx® control

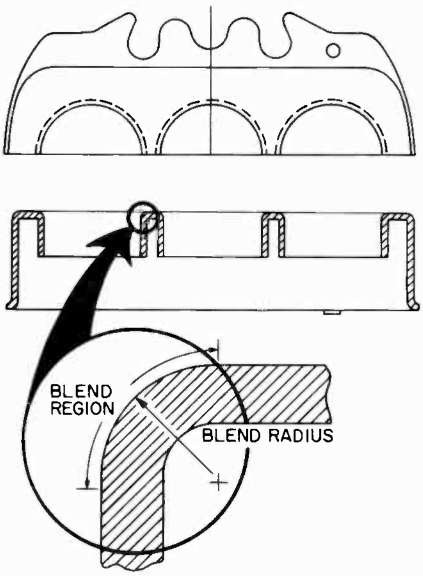
During the initialization of the measurement program, the predetermined motor parameters (low speed, top speed, and ramp time) are sent to the Modulynx and executed. Then a sequence of moves are made to the negative "soft" limit stops for each axis; this point is the system (0,0,0) point. The stages are then moved as needed to make the measurements. If the stage reaches the outermost, "hard" limit stops, hits the top limit of the LVDT, or suffers vacuum or air-pressure failure, then a direct signal sent to the Modulynx stops all motion quickly and sends an interrupt to the computer.

Interval timer

The crystal-controlled clock for the custom IEEE-488 interface is used as the time base for an interval timer. It stores

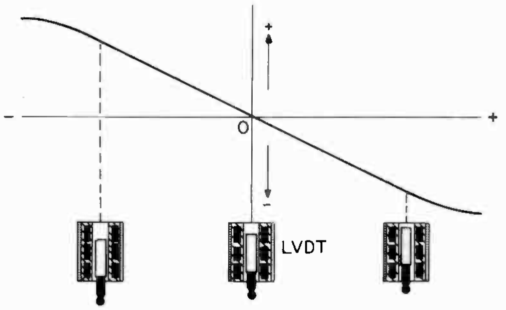
®Modulynx is a trademark of Superior Electric Company

Precision in-line color-tube guns

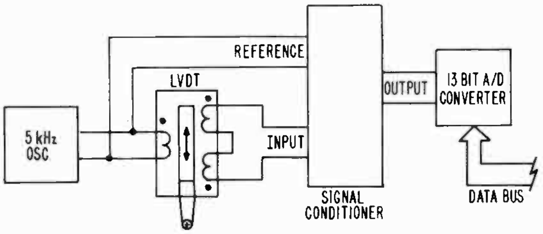


Precision in-line color-tube guns have focusing electrodes with large openings. The shape of the upper part of the blend region is important to the proper focus of the electron beam.

The linear variable differential transformer (LVDT)



The LVDT is an electromechanical device that produces an output voltage that is proportional to the displacement of a separate movable core. The above plot of output voltage versus core displacement is essentially a straight line within the specified range of core motion. Beyond the nominal linear range, the output begins to deviate from a straight line in a gentle curve. Since an LVDT operates on the principle of magnetic coupling, its output is stepless; resolution is limited only by the signal measurement electronics.



A 5-kHz sinewave oscillator with good amplitude stability drives the primary of the LVDT. A signal conditioner converts the ac from the secondary to a dc voltage that is proportional to core position. An A/D converter provides 13 bits of data to the computer.

numbers sent to it in a time counter. After a "go" signal, it produces an interrupt via the 16-bit interface that many milliseconds later.

"Soft" limit stops

The "soft" limit stops are activated when a vane attached to a stage interrupts the beam of an optical interrupter. During initialization, the computer uses these stops to find the 0,0,0 locations. While measuring, the computer produces a warning when it detects a soft limit stop. The IEEE-488 bus is used for communication.

LVDT system

The lift and lower commands to the LVDT, the control of the A/D converter, the data from the A/D, and the control signals are communicated by the 16-bit interface. The LVDT vacuum lift is activated for all long motions and any motion where the part is higher than the starting point. The A/D converter completes its conversion 45 ms after a convert command is issued. The delay may be accommodated by using the timer or a wait instruction in the program.

The use of the timer allows the computer to do some computing during the conversion time.

Two comparators provide out-of-range signals from the LVDT. The signal at the compressed end is used as an emergency stop signal. The other limit may be used as a dropped-off-the-edge signal when moving the *X-Y* stage with the probe lowered.

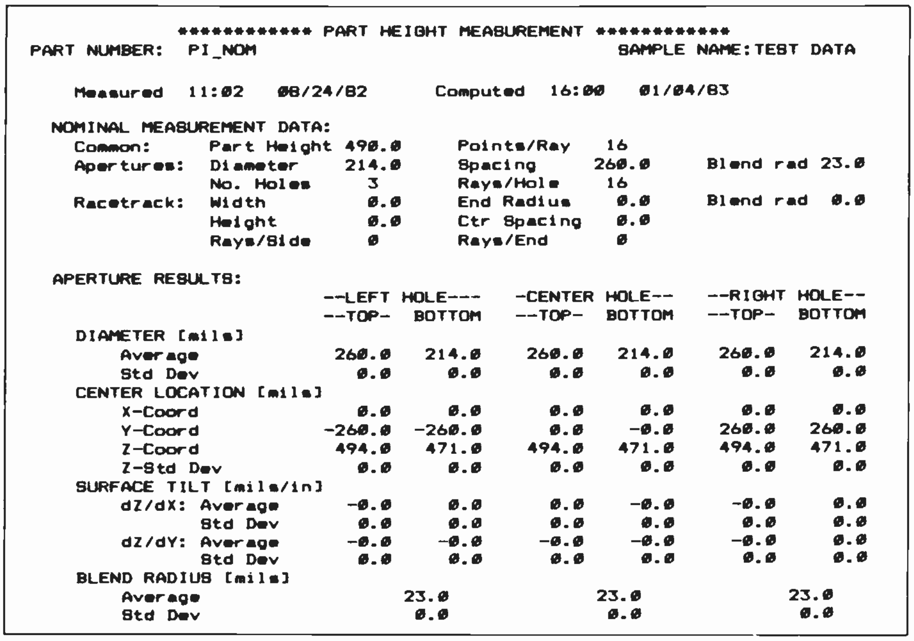
Computer programs

The computer handles three major tasks in the system: (1) measurement control and data taking; (2) data reduction; and (3) result presentation. The computer software customizes the system for a particular measuring task. Although many widely useful subprograms are contained in the software, no attempt has been made to create a universal package. The top levels of the software are very specifically tailored to measure color-picture-tube gun parts.

Measurement control and data taking

Data describing the nominal part to be measured is stored in a data file named with the part number. This file is prepared in a uniform format by a computer program, NEWPART, using dimensions from the part drawing. When a part is to be measured, the operator responds to a computer query by giving the part number. Based on the part data a default program is offered, which may be accepted or modified by the operator.

In the following, a default program for a precision in-line (PI) part is assumed. Each of three holes will be measured along 16 equally spaced radii. The measurements will be made at 16 points on each radius which, on a part with nominal dimensions, would be equally spaced along the surface of the blend arc. The start of each blend will be determined and stored as the zeroth value in the array of Z-values.



```
            ************ PART HEIGHT MEASUREMENT ************
PART NUMBER:  PI_NOM                                 SAMPLE NAME:TEST DATA

   Measured  11:Ø2   Ø8/24/82      Computed  16:ØØ   Ø1/Ø4/83

  NOMINAL MEASUREMENT DATA:
   Common:     Part Height 49Ø.Ø    Points/Ray   16
   Apertures:  Diameter    214.Ø    Spacing     26Ø.Ø    Blend rad 23.Ø
               No. Holes     3      Rays/Hole    16
   Racetrack:  Width         Ø.Ø    End Radius    Ø.Ø    Blend rad  Ø.Ø
               Height        Ø.Ø    Ctr Spacing   Ø.Ø
               Rays/Side     Ø      Rays/End      Ø

  APERTURE RESULTS:
                           --LEFT HOLE---   -CENTER HOLE--   --RIGHT HOLE--
                           --TOP-  BOTTOM   --TOP-  BOTTOM   --TOP-  BOTTOM
   DIAMETER [mils]
      Average               26Ø.Ø   214.Ø    26Ø.Ø   214.Ø    26Ø.Ø   214.Ø
      Std Dev                 Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø
   CENTER LOCATION [mils]
      X-Coord                 Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø
      Y-Coord              -26Ø.Ø  -26Ø.Ø      Ø.Ø    -Ø.Ø    26Ø.Ø   26Ø.Ø
      Z-Coord               494.Ø   471.Ø    494.Ø   471.Ø    494.Ø   471.Ø
      Z-Std Dev               Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø
   SURFACE TILT [mils/in]
      dZ/dX: Average         -Ø.Ø     Ø.Ø      Ø.Ø    -Ø.Ø     -Ø.Ø     Ø.Ø
             Std Dev          Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø
      dZ/dY: Average         -Ø.Ø    -Ø.Ø     -Ø.Ø    -Ø.Ø     -Ø.Ø     Ø.Ø
             Std Dev          Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø      Ø.Ø     Ø.Ø
   BLEND RADIUS [mils]
      Average                    23.Ø             23.Ø             23.Ø
      Std Dev                     Ø.Ø              Ø.Ø              Ø.Ø
```

Fig. 4. *Some 3000 numbers in the data taken are reduced to less than 100 critical dimensions for the use of the electron-gun designer or parts maker.*

The measuring peripherals sequence the measurement program through the use of interrupts. Whenever a time-consuming operation is started, control is returned to the computation program; upon completion of the operation, the hardware interrupts the computation program to start the next measurement step. The two programs are synchronized after the 16 points along each radius are measured.

Along with the controlling of the motions and capture of the data, the computer will be applying calibration corrections and preparing the data for storage in a disk file in standard units. After measuring each part, the computer stores the data for that part, the date, and the time of measurement.

Data reduction

For a measurement system to be useful, one must be confident that the results are correct. Many measurement systems simply assert their correctness, expecting the user to believe the assertion. However, if one

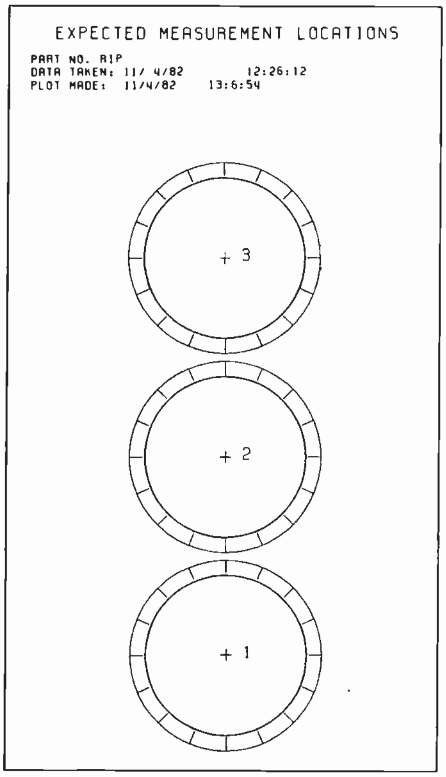

EXPECTED MEASUREMENT LOCATIONS
PART NO. RIP
DATA TAKEN: 11/ 4/82 12:26:12
PLOT MADE: 11/4/82 13:6:54
+ 3
+ 2
+ 1

Fig. 5. *The radial lines show the paths along which measurements were made. This drawing provides a quick check of the measurement program.*

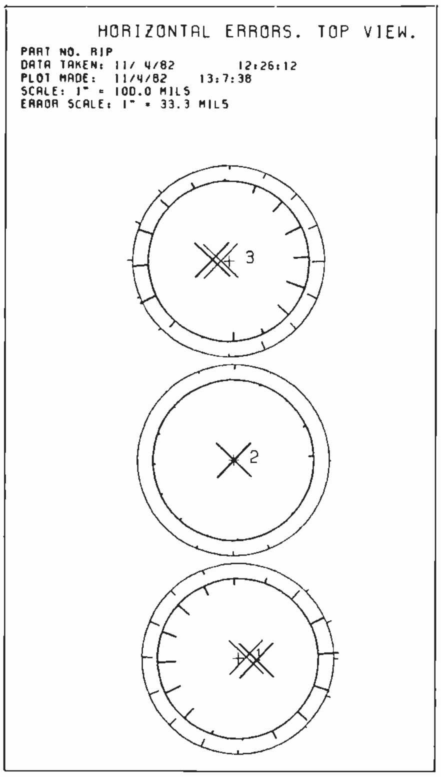

HORIZONTAL ERRORS. TOP VIEW.
PART NO. RIP
DATA TAKEN: 11/ 4/82 12:26:12
PLOT MADE: 11/4/82 13:7:38
SCALE: 1" = 100.0 MILS
ERROR SCALE: 1" = 33.3 MILS
3
2
1

Fig. 6. *A top view of the part has the errors shown by the length of the radial lines (normal to the lines representing the design dimensions of the part).*

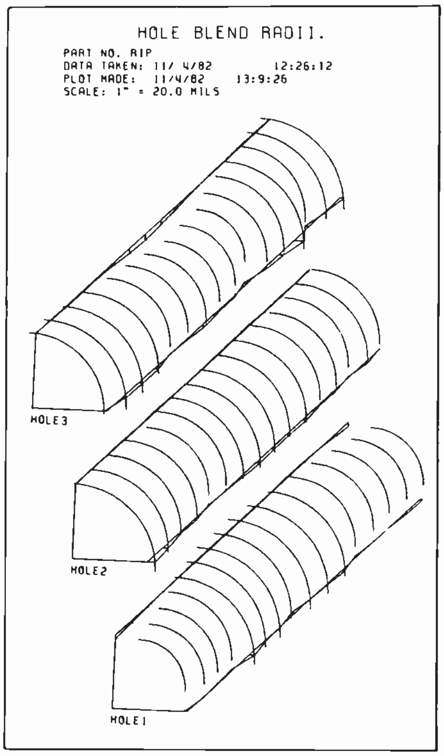


Fig. 7. *A dimetric view of the blend arcs helps the viewer to visualize the errors in the part. The circular hole is "unwound" into a straight line for this drawing.*

wishes, a computer-controlled system can do a certain amount of self-checking. Also, if enough measurements are made, averaging can improve the accuracy, and the confidence level of the results can be computed during data reduction.

Each time a model is fitted to the data (for example, an arc of a circle is the model of a line on the blend region), the standard deviation of the sample from the model is computed. These numbers provide the user with an estimate of how well the system is measuring and modeling the part.

Curve fitting

Least-squares fitting is used for all models except averages. All points definitely proved bad (as when the probe missed the part) are removed from the data before fitting is done.

To fit straight-line models, the classical linear-regression methods are used. The "textbook" shortcut and one-pass methods are usually avoided since they sometimes give poor results due to truncation errors in computer arithmetic. Since each model is fitted to no more than 16 points, the extra computation of using one pass to find the mean and a second pass (using the deviations from the mean) to do the regression is acceptable.

The full circles representing holes in the part are fitted to 16 points evenly spaced around the circumference. The X- and Y-coordinates of the center and the radius are the required outputs. By computing the square of the radius, the problem may be linearized for easier computation.

When data are available over less than 90 degrees of a circle, the straightforward fitting of the best circle to the data often does very poorly in fitting the real part that was measured. Constraining the fit by specifying a tangent line (in our case, the top of the part) substantially improves the results.

Presentation of results

No one is likely to be happy trying to determine the quality of a part by examining a listing of 750 to 1400 X,Y,Z coordinates. The measurements of a part should be presented so the user has only the information he needs and understands. Although graphical representation is good for giving an image of what has been found, certain numerical values are still needed. The choice of what to present depends on a user's time and experience with the use of

Authors (left to right) Mavrodin, Kuczer, Bortfeld, Hook, Beltz, and Saunders.

Dave Bortfeld has been Head, Manufacturing Systems and Process Control, since 1978. Starting with RCA in 1960, he spent two years in the Semiconductor and Materials Division. In 1967, he joined RCA Labs in Zurich working on electroluminescence and electro-optical applications. After transferring to Princeton in 1974, he concentrated on electron-gun design, and on gun and picture-tube manufacturing technology. In addition, he is involved in robotics applications and process control projects.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2515

John Beltz joined RCA EDP Division in June 1958 with the peripheral engineering group, and later became group leader of the Optical Character Recognition group. In 1968, John joined the Graphic System Division and was named group leader of the Digital Electronics activity. He has been an MTS at Princeton Laboratories since 1977, working on noncontact inspection systems, digital circuit design, and microprocessor-based control systems for manufacturing applications in several divisions.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2934

Harvey Hook joined RCA Laboratories in 1950 as a Member of the Technical Staff. He has worked on color picture tubes, display storage tubes, opto-electronic displays and components, thin-film optical and electro-optical devices, semiconductor devices including making masks for their fabrication, and instrumentation for laboratory experiments. He is presently working on measurement and fabrication problems involved in the production of improved color-tube guns.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2369

Paul Kuczer joined RCA Laboratories in 1962 in the Engineering Service Department where he worked on the development of electronic instruments. Since 1978, he has been involved in various projects for Manufacturing Systems and Process Control.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2579

Al Mavrodin received the MSc degree from Massachusetts Institute of Technology in 1976. As a Member of the Technical Staff with RCA Laboratories since 1978, he has worked on automatic assembly and testing equipment and is currently involved with the flat-panel TV project.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2247

Bill Saunders joined RCA Laboratories in 1964. He received his BSEE from Monmouth College in 1977. While at the Labs, he has worked on cryogenic and optical memories, VideoDisc, and systems for use in other RCA divisions. Since 1981, he has been involved in the development of the gun-parts measuring system.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2863

the results as well as the particular application. Figure 4 is the summary of measurements made on a gun part for a (PI) series color kinescope. It probably has items on it which will be eliminated as we gain experience.

How one uses graphics depends to a great extent on how easy they are to use. The prototype system being built for Lancaster uses an HP9826 computer with fairly easy-to-use graphics for either the CRT screen or the printer-plotter. Three different plots are now available:

1. Figure 5 is a plan view of the nominal part with the expected measurement locations shown.
2. Figure 6 is a plan view of the part with the results of the measurements shown as errors from the nominal locations. The magnification of the measurements is selectable.
3. Figure 7 is a false-perspective (dimetric) representation of the blend radii at the edges of the holes. It is presented as if the loci of the blend radii centers were straight lines.

Summary

A measurement system has been built to measure the height of small parts rapidly and accurately. Custom software, written for the computer/controller, tailors the system to the measurement of parts for color-picture-tube gun parts. The same system with other software could measure any part that fits within a 2-in. cube.

Acknowledgment

Joe Pierro designed the console arrangement. R.L. Leigh and M.K. Brown of Mount Development Engineering (Lancaster) continuously encouraged us by their interest and eagerness to get parts measured. Carl Lundvall of M.K. Brown's group has helped by supplying parts for measurement as well as prints of the parts. We are grateful for their help.

Productivity in Engineering Documentation

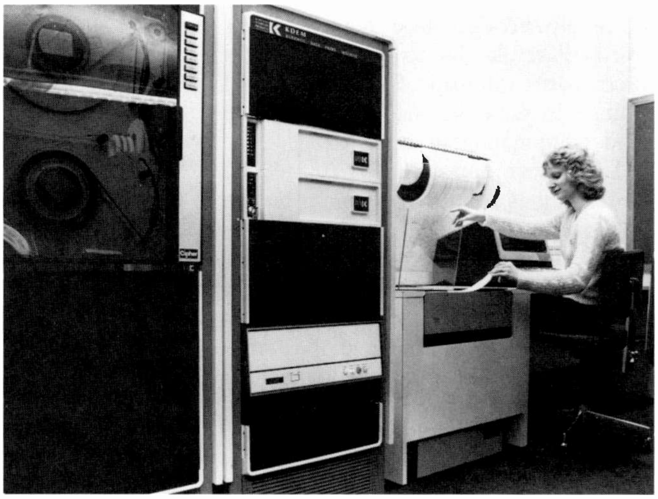
The Kurzweil Data Entry Machine breaks the manual entry bottleneck.

Documentation is an area of engineering productivity that has received increasing attention in recent years. The expanding availability of word-processing equipment offers the potential for important reductions in time and effort devoted to engineering documentation. But word processors and the larger, centralized text management systems share a common problem that prevents them from reaching their full potential for productivity. This problem is the cost of manually converting *existing* documents to digital form; high labor costs and the relatively slow speeds of keyboarding usually mean that only the most urgently needed documents are converted.

All printed materials are machine-readable

Now, however, there is a way around this manual entry bottleneck. At Moorestown, Missile and Surface Radar (MSR) is using a Kurzweil Data Entry Machine (KDEM) to convert printed documents to digital form for further revision and reissue on MSR's Text Management System. Unlike conventional optical character readers, the KDEM scans and recognizes virtually *any* type face, either typewritten or typeset. Because it digitizes printed material at rates much faster than those attainable by human keyboard operators, the KDEM makes conversion of existing documents to digital form a practical, cost-effective element of the documentation process. As an example, the KDEM was used to convert more than 50,000 pages of previously published system specifications to digital form for revision and updating—at a per-page cost well under half that of standard rates for typing and proofing.

The KDEM uses a proprietary omni-font character recognition technology to process printed material into electronic or magnetic form. The system undergoes a brief "training session" for each typeface to be scanned, then stores up to 25 of these training sets on line for immediate reference. After the document is scanned, it is sent to the host system over asynchronous or bisynchronous lines, or by industry-standard nine-track magnetic tape (800 or 1600 characters per inch).



In addition to these output formats, the MSR facility is also able to output material from the KDEM onto floppy disks suitable for direct processing on either Wang or MICOM word processors. The services are available, at cost, to all RCA operating units.

For additional information contact:

Rich Camlin
Missile and Surface Radar
Moorestown, N.J.
TACNET: 224-2708

R.F. Morey | R. Kleppinger

Process control charts to monitor and guide device production

Process control charts provide manufacturing with a method of gauging whether the product is meeting specifications.

Abstract: *Statistical process control charts now establish control limits for each process on the production line, and determine reject limits that assure that the product being produced meets specifications. These charts have increased the sensitivity of production personnel to the importance of maintaining process averages at the center of the specification, and not just inside the limits.*

Process control charts are used at RCA Palm Beach Gardens to monitor a number of production processes and their results. The introduction of the statistical process-control-charting techniques described in this article has improved the adherence of product to specification, and has assured that production processes are actually being controlled. The charts have increased the sensitivity of operators, supervisors, and engineers to the need to keep the process average at the center of the specification instead of just inside the limits. In addition, the relationship between sample averages and control of product distributions is better understood.

The charts are also being used to evaluate new process equipment and changes to current processes. Although this statistical method of charting has been used in production for less than a year and has gone through several learning changes, the benefits appear to more than compensate for the effort required.

Previous method—Incentive for change

Until the development of the chart described in this paper, plots governing most of the critical wafer-processing operations at Palm Beach were constructed using engineering knowledge to specify minimum/maximum limits based on design and process needs. These earlier plots were often diagrams of individual sample readings with upper and lower specification limits and "action" lines inside the limits that were to call engineering attention to measurement values approaching the specification limits. Normally these limits would be applied strictly by production to control product-lot acceptance.

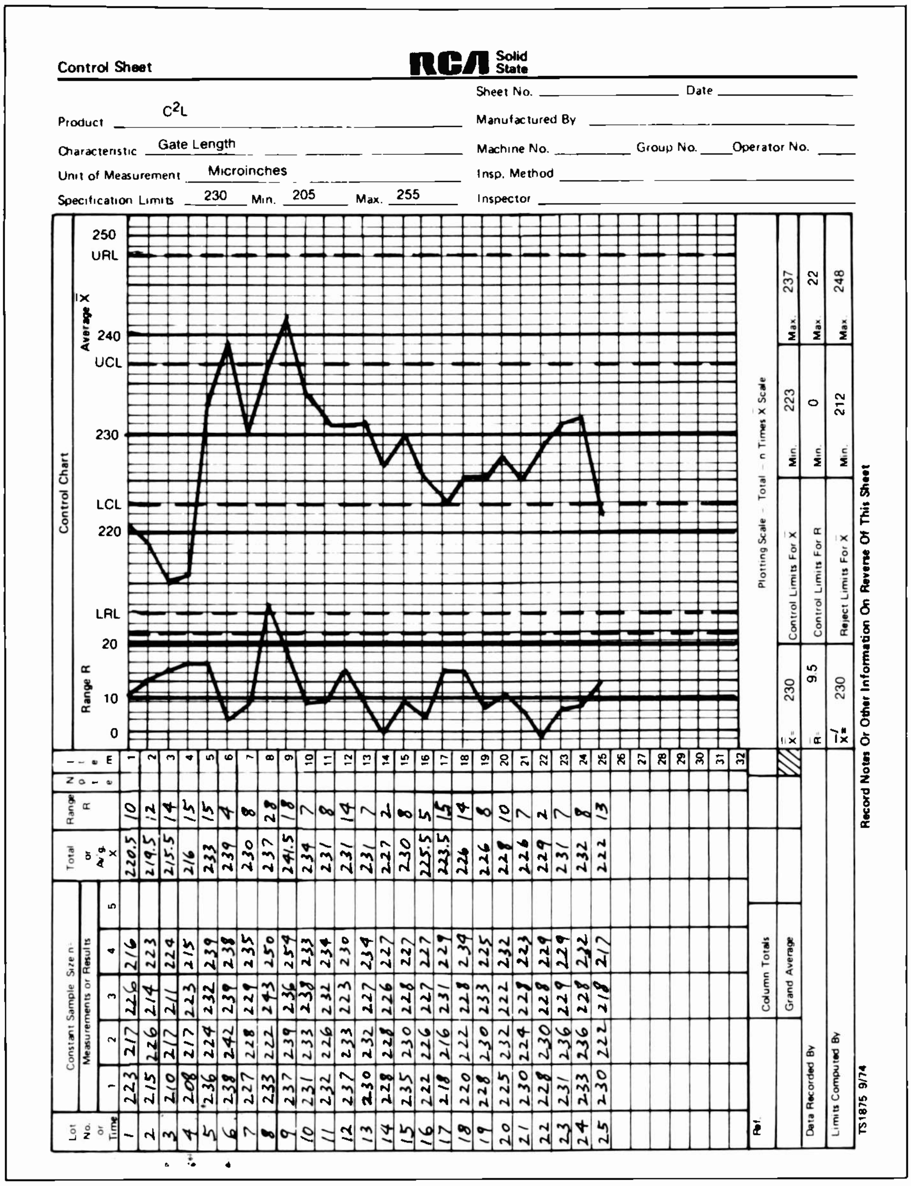
In other cases, two or more readings of samples from a process run were averaged and plotted. Although average readings of monitor wafers from each production lot were used to determine lot acceptance, the relationship of these averages to specification limits was not considered. Process engineers had responsibility for a process or group of processes and the control plots that were used for them.

The change—Impetus and ground rules

An RCA in-house Process Quality Control course precipitated the decision to develop statistical process control charts for variables in accordance with the concepts developed some years ago.[1]

The ground rules agreed upon for the development of the new control charts were explicit in that minimum disruption of the wafer fabrication operation was to be allowed. Accordingly, although during development of the technique the placement of control wafers and measurement practices were modified to provide improved control, the number of the sample measurements being made was not increased, nor were specification limits tightened solely to assure that the process was operating under control. The closed-cell logic (C^2L) LSI wafer-fabrication process

Final manuscript received December 22, 1982.
Reprint RE-28-1-8



Control Sheet RCA Solid State

Product C2L

Characteristic Gate Length

Unit of Measurement Microinches

Specification Limits 230 Min. 205 Max. 255

Sheet No. ________ Date ________

Manufactured By ________

Machine No. ________ Group No. ________ Operator No. ________

Insp. Method ________

Inspector ________

Plotting Scale - Total - n Times X̄ Scale

	X̄ =	Min.	Max.
Control Limits For X̄	230	223	237
Control Limits For R	R̄ = 9.5	0	22
Reject Limits For X̄	X̄ = 230	212	248

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Note																																
Range R	10	12	14	15	15	4	8	28	18	7	8	14	7	2	8	5	15	14	8	10	7	2	7	8	13							
Total or Avg. X	220.5	219.5	215.5	216	233	239	230	237	241.5	234	231	231	231	227	230	225.5	223.5	226	226	228	226	229	231	232	222							
5																																
4	216	223	224	215	239	238	235	250	254	233	234	230	234	227	227	227	229	239	225	232	223	229	229	232	217							
3	226	214	211	223	232	239	229	243	236	238	232	223	227	226	228	227	231	228	233	222	228	228	229	228	218							
2	217	226	217	217	224	242	228	222	239	233	226	233	232	228	230	226	216	222	230	232	224	230	236	236	222							
1	223	215	210	208	236	238	227	233	237	231	232	237	230	228	235	222	218	220	228	225	230	228	231	233	230							
Lot No. or Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25							

Constant Sample Size n- Measurements or Results

Column Totals

Grand Average

Ref.

Data Recorded By

Limits Computed By

Record Notes Or Other Information On Reverse Of This Sheet

TS1875 9/74

Fig. 1. Typical statistical process-control chart used at RCA Palm Beach Gardens to control transistor gate length of a C^{2}L device. This chart shows the control and reject limits imposed on the process operation and provides insight to the factory as to the degree to which product is meeting specification. The definition of the symbols and acronyms used in the chart is given in Table I.

was converted first, since it was a matured production line.

The C^{2}L wafer-fabrication or "fab" line operates on a three-shift basis and involves thirteen different process operations, some requiring the use of several machines. It was impossible to do a classical process-capability study on each operation in a reasonable time. Therefore, the most recent control-plot data for each process was analyzed to establish the trial limits for the development of the statistical control charts. The data from each different machine used for a given operation was analyzed separately to allow comparisons between machines.

A statistical process-control chart

There are three basic general steps in the construction of a statistical control chart such as the one shown in Fig. 1:

1. Determine and standardize the number of sample measurements.
2. Establish control limits for both the average measurement and average range of measurement for each process.
3. Establish reject limits for the average value of the sample measurement for a run so that the wafers in the run are within specification limits.

The mathematics for establishing required limits is straightforward.

The characteristic to be controlled through the use of the sample chart of Fig. 1 is the transistor gate length of a C^{2}L device (this is the dimensional measurement obtained after plasma etching of the polysilicon layer, see Fig. 2). The symbols and acronyms used in the chart are defined in Table I. More detailed information concerning calculation of the values represented by these symbols and acronoyms is given in the sidebar, page 52, and in Table II. Study the chart at each step for the best understanding of the operations involved in its construction.

The first step in control-chart construction is the determination and standardization of the number of measurements to be made for each run. A "run" is a wafer lot or batch of 48 wafers processed simultaneously. Two wafers are measured for gate length in each of two locations, providing a sample size, *n*, of 4.

Specification limits are developed to meet finished electrical-device requirements. Consequently, this limit applies to each individual structure (wafer or die). The specification limits for the sample control chart are 230 ± 25 microinches, or minimum, 205, and maximum, 255 microinches.

Measured values of gate length are taken for 25 lots, the minimum number of runs needed to obtain an accurate average range, $\bar{R}$. The range is the difference between the highest and lowest measured sample values for each run. The average range for the 25 runs is calculated:

$$\bar{R} = 10.28.$$

R_{max}, the upper-control limit for *R*, is then calculated as

$$R_{max} = D_4 \times \bar{R}$$

where D_4 is obtained from Table II for *n* of 4. Plugging in our numbers, we see that

$$R_{max} = 2.28 \times 10.28 = 23.4$$

If R_{max} is exceeded by a particular run, that run has a wider product distribution than that predicted by the standard range, which is to say that it has a standard deviation that is almost certainly larger than the other runs, and is considered "out of control." Such runs are eliminated from the data and $\bar{R}$ is recalculated.

Reviewing the data, note that one lot, No. 8, has exceeded the upper control limit for range, R_{max}, with a value of 28. Eliminating this lot and recalculating:

$$\bar{R} = 9.54$$

$$R_{max} = 21.75$$

(A further discussion of standard devia-

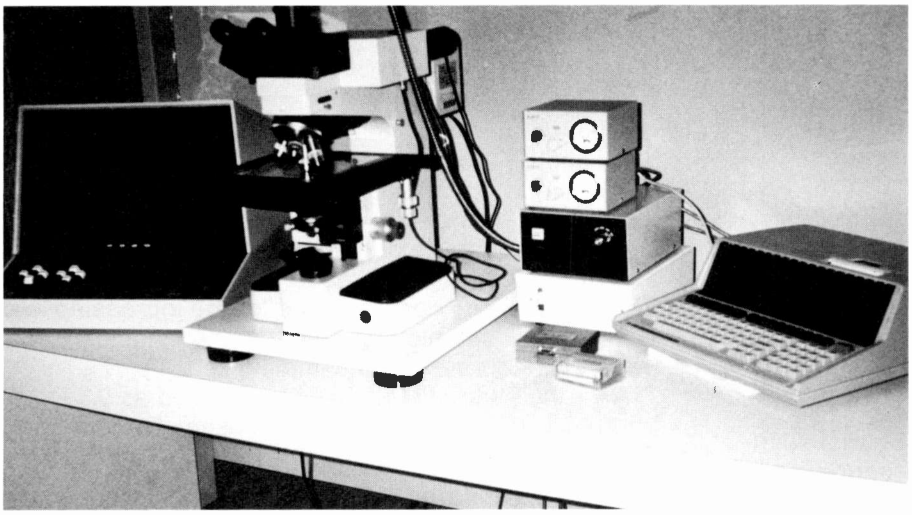


Fig. 2. The Leitz linewidth measuring system includes: Leitz optics; an AMS 100 microprocessor; and an HP85 computer. The system is used for ADI, AEI Gate-1 length measurements.

tion as it applies here is presented in the sidebar, part I, page 52.)

Since no remaining lots exceed the new upper-control limit for range, R_{max}, proceed to the next step. In actual production, any lots that fail the maximum-range upper control limit, R_{max}, are screened 100 percent; that is, each wafer is measured, and only those that fail the specification limit are rejected.

Calculate the average measurement $\overline{X}$. The same 25 runs used to calculate the average range, $\overline{R}$, are used in this computation. All data points plotted are an average of the number of samples measured per run. Calculation of either the average measurement of each run or the average of all the sample values will yield the same results. However, at this point a decision must be made as to whether this calculated average or the desired average measurement will be used. The desired target averages are usually the center of the specified minimum/maximum limits for the process, unless another different target value has been explicitly stated. Since there is no wish to alter the specifications, the desired average measurement, $\overline{X}$ = 230 microinches, is used as the centerline on the sample control chart.

Calculation to find the control limits for $\overline{X}$ is done by using the expression:

$$A_2\overline{R}$$

where A_2 is obtained from Table II for n of 4, or calculated as explained in the sidebar, page 52, Part II.

$$A_2\overline{R} = 0.73 \times 9.54 \cong 7$$

Applying this expression to the desired $\overline{X}$ = 230 microinches yields the following:

Upper control limit, UCL = 237

Lower control limit, LCL = 223

$\overline{X}$ values that exceed these limits are from lots whose average is presumed to have shifted from the desired target of 230 microinches. These lots are considered "not in control" for $\overline{X}$.

Lot numbers 6 and 9 exceed the upper control limit for the average. The process must be checked to determine that all process conditions and settings are correct and have not changed. However, even though the process average has shifted upward, the lots may still contain product that is within specification limits. This is determined by calculating the upper and lower reject limits for the average.

Since specification limits apply to each individual in a population and since control charts plot averages, it is necessary to establish reject limits for the average that correspond to the specification limits for individuals. The quantity that must be added to the minimum specification limit and subtracted from the maximum specification to arrive at the reject limits for the average is defined by the expression $V\overline{R}/d_2$. Determining the values of V and d_2 from Table II (or the sidebar, part I):

$$V\overline{R}/d_2 = 1.50 \times 9.59/2.059 = 7$$

Table I. *Symbols and acronyms used in the statistical control chart.*

USL	= Upper specification limit
LSL	= Lower specification limit determined from final electrical device properties
$\overline{X}$	= Center of the specification limits
$\overline{R}$	= Average range of 25 or more runs
n	= Number samples per run
UCL	= Upper control limit ($\overline{X} + A_2\overline{R}$)[1]
LCL	= Lower control limit ($\overline{X} - A_2\overline{R}$)[1]
URL	= Upper reject limit (USL $- V\overline{R}/d_2$)[2]
LRL	= Lower reject limit (LSL $+ V\overline{R}/d_2$)[2]
R_{max}	= Upper control limit for $\overline{R}$ ($D_4\overline{R}$)

1. See Sidebar, Part II, page 52; Table II; or ref. 2 for information on A_2.
2. See Sidebar, Part I, page 52; Table II; or ref. 2 for information on V.
3. See text, Table II, or ref. 2 for information on D_4.

Table II. *Parameters required in calculation of control-chart values. For a more elaborate description of parameters,* A_2, D_4, V *and* d_2, *see ref. 2.*

n	A_2	D_4	d_2	V
2	1.88	3.27	1.128	0.88
3	1.02	2.57	1.693	1.27
4	0.73	2.28	2.059	1.50
5	0.58	2.11	2.326	1.66

Note: The parameter designations are those applicable to the construction of control charts using range, R.

Calculations in control-chart construction

I. Minimum and maximum values.

The estimated typical 25-run standard deviation, σ', is:

$$\sigma' = \bar{R}/d_2$$

where d_2 is obtained from Table II and n is the sample size of a run. Increasing the number of runs averaged will increase the accuracy of this estimate, but 25 data points (runs) is considered sufficient.

Statistical theory states that for a normal distribution, Fig. A, 99.7 percent of the measured values of the population are included inside of $\pm 3\sigma'$ about the average measured value $\bar{X}$ ($\bar{X}$ is discussed in detail in the text). Hence, $\bar{X} + 3\sigma'$ describes the minimum and maximum values of the population in a typical run.

The observed range, R, is a function of the number of samples n that are drawn from the population. For $n = 2$, the range is small, since the probability is high that the samples drawn will be near $\bar{X}$. As n is increased, the range increases, since the probability of obtaining values farther away from $\bar{X}$ becomes greater. The empirically derived factor d_2 compensates for the increasing $\bar{R}$ with larger sample sizes, so that the estimate from $\bar{R}$ is accurate for all sample sizes.

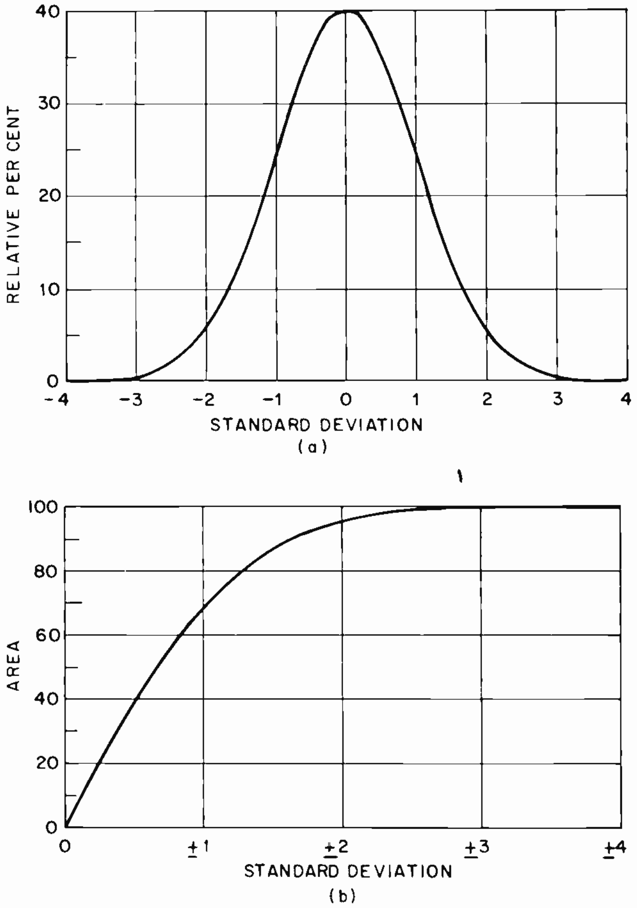


Fig. A. Graphs showing standard normal deviation **(a)** and area under the normal curve **(b)**. According to statistical theory, for a normal distribution, 99.7 percent of the measured values of a population are included inside a range of $\pm 3\sigma'$ about the average measured value.

$\bar{R}$ for the example has been calculated as 9.54. Therefore, the standard deviation estimate σ', for the lots is, again:

$$\sigma' = \bar{R}/d_2 = 9.54/2.059 = 4.63$$

$$3\sigma' = 13.9 = 14$$

Thus, the typical run used in constructing the sample chart contains individual gate length values ranging from 14 microinches below to 14 microinches above the average. This range is well within the allowed specification tolerance of ±25 microinches.

II. Calculation of A_2.

Since the control chart plots the average value $\bar{X}$ for each run, the limits of the variation of $\bar{X}$ must be known. These limits are defined by:

$$3\sigma'/\sqrt{n} = 3\bar{R}/d_2\sqrt{n} = A_2\bar{R}$$

The Central Limit Theorem of Statistics states that, when samples are taken from a population and the values averaged, the standard deviation of the averages is less than the standard deviation of the individual values. In fact, regardless of whether the population is a normal distribution or not, the standard deviation of the averages is related to the standard deviation of the individual value by $\sqrt{n}$, where n is the number of samples being averaged. For example, for $n = 4$, the averages have a spread of half that of the population of individuals.

III. Calculation of reject limits.

The method used to calculate reject limits is found in the references and bibliography and is applicable when range $\bar{R}$ is in control and $3\sigma'$ limits for individuals in a process run are within the specification limits. This condition permits a degree of shifting of the average $\bar{X}$ beyond the strict control limits while keeping all the individual values within the specification limits.

How far the average can shift and still produce wafers that are within specification limits can be determined by considering the situation in Fig. B. The typical distribution of individuals in a run is shown: (1) In a center position (at the target); (2) in a worst-case upper position with the $3\sigma' = 3\bar{R}/d_2$

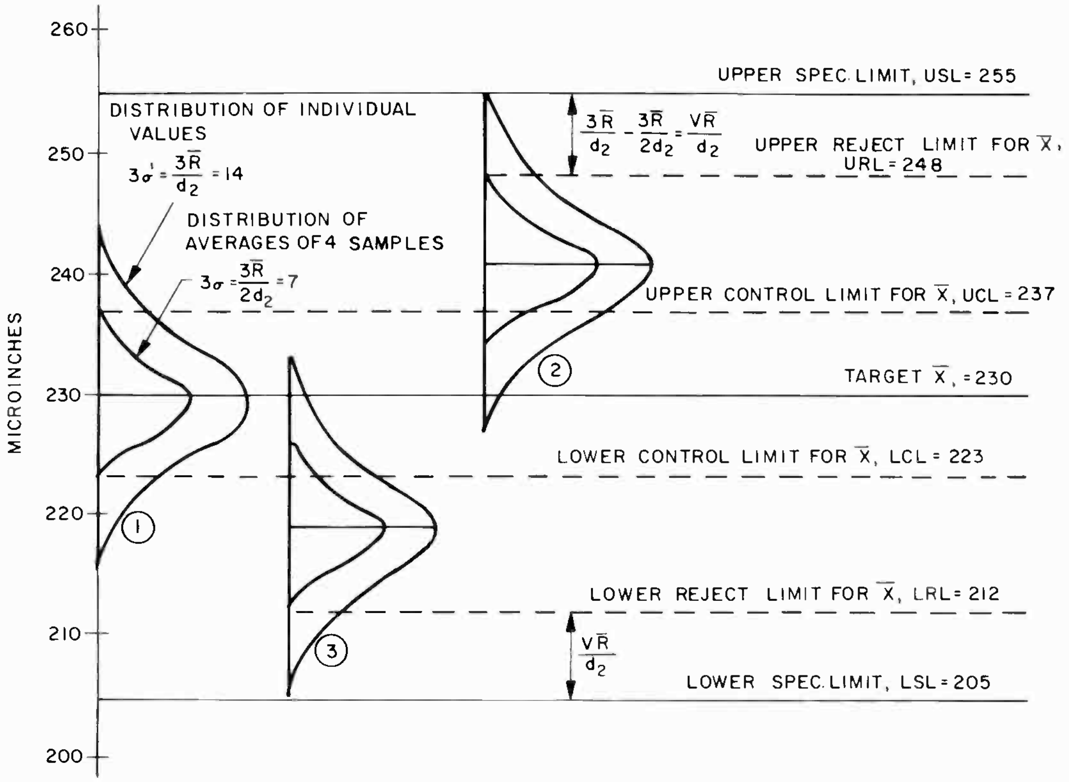

260
250
240
230
220
210
200
MICROINCHES
DISTRIBUTION OF INDIVIDUAL VALUES
$3\sigma' = \frac{3\bar{R}}{d_2} = 14$
DISTRIBUTION OF AVERAGES OF 4 SAMPLES
$3\sigma = \frac{3\bar{R}}{2d_2} = 7$
UPPER SPEC. LIMIT, USL = 255
$\frac{3\bar{R}}{d_2} - \frac{3\bar{R}}{2d_2} = \frac{V\bar{R}}{d_2}$
UPPER REJECT LIMIT FOR $\bar{X}$, URL = 248
UPPER CONTROL LIMIT FOR $\bar{X}$, UCL = 237
TARGET $\bar{X}$, = 230
LOWER CONTROL LIMIT FOR $\bar{X}$, LCL = 223
LOWER REJECT LIMIT FOR $\bar{X}$, LRL = 212
$\frac{V\bar{R}}{d_2}$
LOWER SPEC. LIMIT, LSL = 205
1
2
3

Fig. B. How far the average measured value, $\bar{X}$, can shift and still produce wafers that are within specification limits can be determined by examination of this figure. The typical distribution of individuals within a run is shown in a center position $\bar{X}$ (at the target), in a worst-case upper position with the upper limit at the upper specification limit, and in a worst-case lower position with the lower limit at the lower specification limit.

upper limit at the upper specification limit; and (3) in a worst-case lower position.

Shown inside of each distribution of individuals is the distribution of the averages (four in this case). The difference between highest individual value in the upper distribution and the highest average value is:

$$(3\bar{R}/d_2) - (3\bar{R}/\sqrt{4}d_2) =$$
$$(3 - 3/2)\ \bar{R}/d_2 = V\bar{R}/d_2 = 1.5\ \bar{R}/d_2$$

Values of V for sample sizes from 2 to 5 are shown in Table II or can be calculated from the expression:

$$V = 3 - 3/\ \sqrt{n}$$

Adding $V\bar{R}/d_2$ to the lower specification limit and subtracting it from the upper specification establishes a reject limit for the average that guards against the specification limits being exceeded by any of the individual values. These lines are also shown on the control chart for $\bar{X}$, URL, and LRL.

Upper reject limit,
URL = 225 + 7 = 248 microinches

Lower reject limit,
LRL = 205 + 7 = 212 microinches

Both lots 6 and 9, though requiring process adjustment, are within the reject limits and, therefore, are acceptable. However, Manufacturing must make appropriate adjustments to bring the process back into control. Had they failed the reject limits, the normal disposition would have been to measure each wafer and reject those that fail the *specification* limits.

Results and their interpretation

The statistical method of control charting has been applied to most of the C^2L and SOS processes used in production at Palm Beach Gardens, as shown in Table III. A monthly Process Parameter Summary is published for each operation showing:

1. The process average versus the target.
2. Three standard deviations versus total spread allowed by the reject limits.
3. The number of product runs failing the reject limits.
4. A comparison of data from each piece of equipment used for the same process, Table IV.

This summary helps focus engineering attention on those operations requiring process adjustments.

While introducing the control chart method, a case of over-control at gate oxidation was discovered. Standards required the diffusion operator to alter the deposition time based on the previous run average. Application of statistical control chart techniques revealed that the operator was required to

Table III. *RCA Palm Beach Gardens control chart usage.*

Process type	*Number of different uses*	*Thickness*	*Sheet resistance*	*Etch rate*	*% P EDAX*[1]	*Dimensional*
Thermal Oxides	5	X				
Diffusions	3	X	X			
Implants	2		X			
LPCVD[2]	2	X		X		
Doped CVD	3	X			X	
Dimensions	3					X
Metal	To be developed					

[1] % P EDAX is the percentage of phosphorous in deposited SiO_2films as measured by an energy dispersive X-ray system.

[2] Low-pressure chemical vapor deposition (LPCVD) is a method of depositing layers of SiO_2, SiN_4 (silicon nitride), or polysilicon on integrated circuit wafers.

Table IV. *C^{2}L (July 1982) control-chart summary.*

Process Step	*Samples Per Run*	$\overline{X}$	*Spec Target*	$3\sigma\overline{X}$	*Reject Limits*	*Runs*	*High Runs*	*Low Runs*
P-Well Resistivity								
Machine 1-6	4	2306	2100	± 173.1	± 450	20	0	0
Machine 2-1	4	—	2100	± —	± 450	—	—	—
Machine 2-2	4	2322	2100	± 135.1	± 450	22	0	0
Machine 1-5	4	2264	2100	± 227.3	± 450	21	0	0
Gate Oxide Thickness								
Post anneal	2	1257	1250	± 77	± 80	50	1	0
ADI Gate	4	271.9	270	± 13	± 19	—	—	—
AEI Gate	4	229.2	230	± 14	± 18	122	0	0
Poly Thickness	2	4749	4700	± 912	± 600	29	1	0

make process changes when the averages were *within* the process control limits. These changes caused the variability of the process average to be greater than necessary. The process was stabilized by removing the instantaneous feedback loop.

A controlled process is defined when the $\overline{X}$ and σ' of each individual run is the same as all the others, as is implied in the calculation used in building the control chart of Fig. 1. The variations in $\overline{X}$ and $\overline{R}$ are due only to the randomness of sample selection. When a single point exceeds the UCL or LCL for $\overline{X}$, the theoretical chances for a normal distribution are 997 out of 1000 (99.7 percent) that this is due to an actual shift in the average value of that run. When a single point exceeds R_{max} the theoretical chances are 997 out of 1000 that this is due to an increase in the process spread of that run.

Other rules commonly used as tests to indicate a shift in $\overline{X}$ consider more than one point on the chart. For example, seven points in a row on the same side of the $\overline{X}$ centerline indicates a shift in $\overline{X}$ with the same probability as a single point outside of control limits (UCL or LCL). Four out of five beyond one-third of the control limits (1/3 $A_2\overline{R}$), and two out of three beyond two-thirds of the control limits ($2A_2\overline{R}/2$) are also indications of an $\overline{X}$ shift with approximately the same probability as a single point beyond the control limits. Examination of Fig. 1 reveals that the process was out of control from lot 16 to 22 because seven sequential points are all below the centerline.

Experience with the first C^{2}L control charts immediately showed that in many cases, the $\overline{X}$ chart was out of control more often than the $\overline{R}$ chart, which only occasionally went out of control. In some cases, the reject limits were inside the control limits, which means that even if the process remains in control, wafers are produced that fail the specification limits. There are two approaches to follow when this happens. One is to review the specification limits for the process step in question and determine if they can be expanded without affecting overall yield. The second approach, if the process step is critical to yield or reliability, is to adhere strictly to the reject limits.

Manufacturing is required to determine the disposition of each lot failing the reject limit and to make process adjustments. Lots may be reworked, scrapped, inspected 100 percent, or continued with proper identification, so that the effect on yield of such lots can be determined.

Summary

The control charts provide product protection through the use of the reject limits. However, equally important is the control of the process provided by analysis of the average and range plotted for each run. Specifically, data points falling outside the control limits are a signal to Manufacturing that the process is out of control and requires the determination of the assignable causes so that adjustments can be made to bring it back into control. Thus, an anticipatory feedback system is built into the control chart to allow Manufacturing to apply preventive adjustments to keep the process in control and the individual product wafers within the desired limits.

References

1. W.A. Shewhart, *Economic Control of Quality of Manufactured Product,* Van Nostrand, Reinhold Co., Inc., Princeton, N.J. (1931).
2. E.L. Grant and R.S. Leavenworth, *Statistical Quality Controls,* 5th edition, McGraw-Hill (1980).
   Note: See Appendices One and Three (Tables B, C, D, E) for definitions and values of d_2, A_1, A_2, D_1, D_2, D_3, D_4. See also pages 306-307 for explanations of V and tables of values for V.

E.R. Ott, *Process Quality Control,* McGraw-Hill (1975).

I.W. Burr, *Elementary Statistical Quality Control,* Marcel Dekker, Inc., New York (1979).

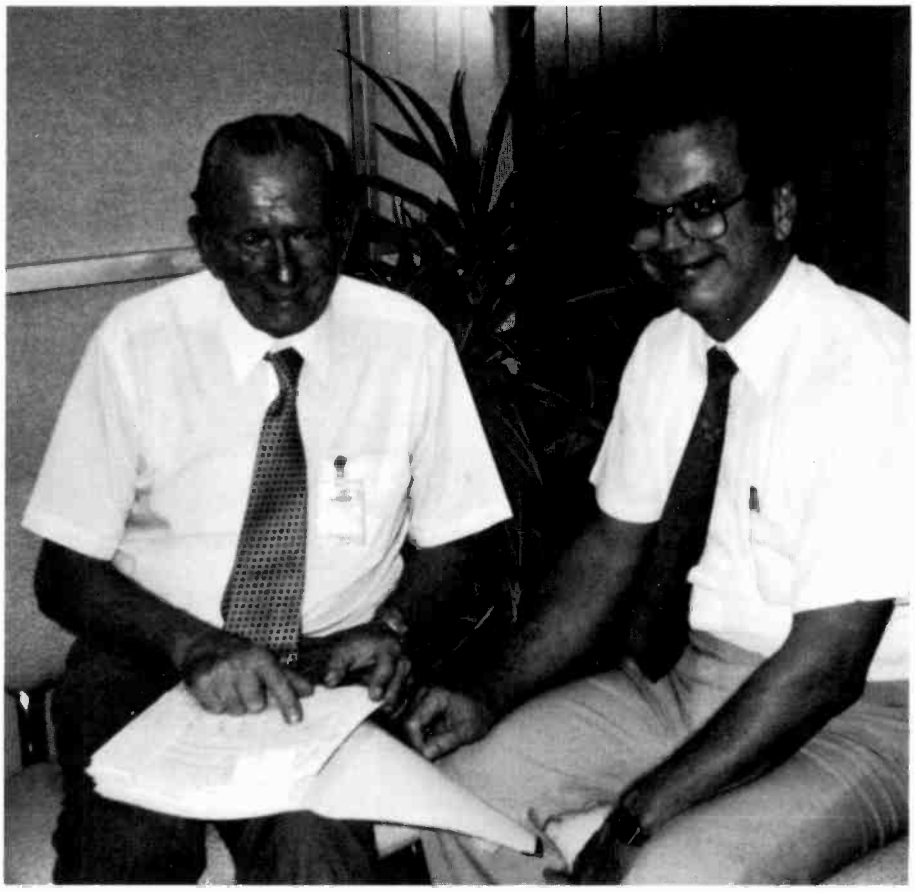
Statistical Quality Control, Western Electric Co., 2nd edition (1958).

Rob Kleppinger is Senior Member of Technical Staff, Yield Improvement Process Engineering at RCA Solid State Division, Palm Beach Gardens, Florida. He is currently responsible for production yield analysis. He joined RCA in 1951 as an engineer in solid-state device design and processing. He has worked in various engineering and leadership positions in discrete-device design, wafer processing, and device packaging groups since then. He holds a B.S. in Engineering and an M.S. in Mathematics.

Contact him at:
Solid State Division
Palm Beach Gardens, Fla.
TACNET: 722-1343

Dick Morey is Quality Engineer at RCA Solid State Division, Palm Beach Gardens, Florida, with responsibility for in-process quality control for the LSI wafer-fabrication operation, and for quality assurance for RCA Hi-Rel LSI products. He is also responsible for the Palm Beach Gardens Documentation Center. Dick joined the RCA Florida Quality Department in 1978 after having been with ITT Semiconductors in Florida.

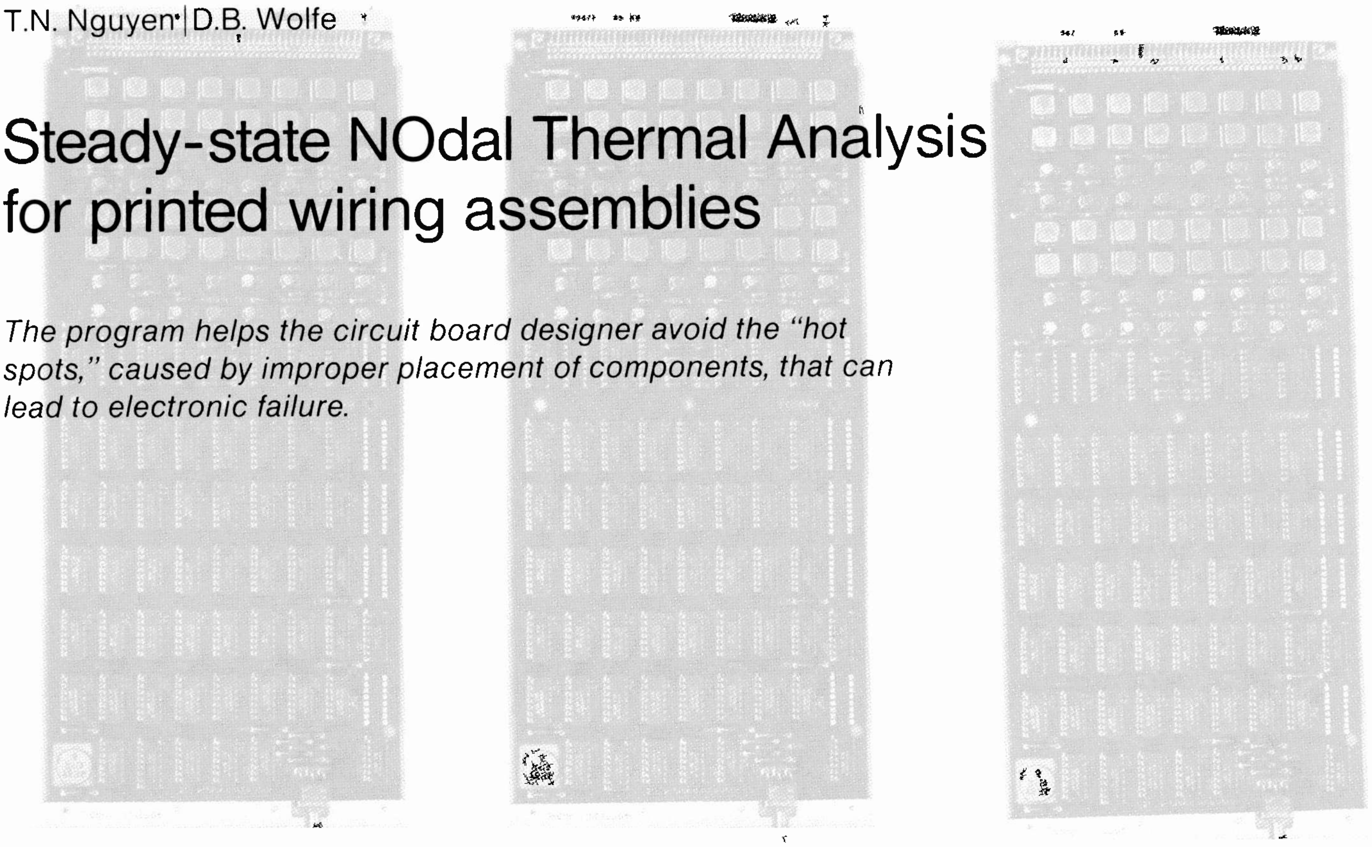
Contact him at:
Solid State Division
Palm Beach Gardens, Fla.
TACNET: 722-1262



Authors (left) Morey and Kleppinger.

T.N. Nguyen | D.B. Wolfe

Steady-state NOdal Thermal Analysis for printed wiring assemblies



The program helps the circuit board designer avoid the "hot spots," caused by improper placement of components, that can lead to electronic failure.

Abstract: *NOTA (NOdal Thermal Analysis) is a GCS-generated FORTRAN program developed primarily to analyze the steady-state thermal characteristics of a printed-circuit-board assembly. Although the program can run on many different machines that support FORTRAN language, it's intended for the Applicon/RSX operating system. Since the program is highly interactive and employs easy-to-understand prompts, both Applicon graphics users and noncomputer personnel can easily access the programs for a quick thermal analysis output. Upon proper instruction, the program will extract board-layout and placement information from an Applicon board-placement data file; then the routine will carry out an analysis, employing the Gaussian elimination method for matrix solution, to arrive at a set of steady-state nodal temperature outputs.*

This program can analyze printed circuit boards having up to 100 nodes. Without a computer, a 100-node analysis is almost impossible. To run a nodal analysis using a general-purpose heat-transfer program, an analyst will spend several days modeling and partitioning the assembly before entering the data into a computer. NOTA, on the other hand, uses the database acquired by the computer when the board is designed. Therefore, a NOTA analysis can be performed in less than a half hour.

One of the most frequent causes of electronic component failure is heat. An examination of MIL-HDBK-217, "The Reliability Stress and Failure Rate Data for Electronic Equipment," shows that the failure rate for electronic components increases approximately exponentially as a function of increasing temperature. Therefore, the mechanical design of a printed circuit board must take into consideration the physical placement of components for thermal management as well as interconnection. Analytical methods can predict component temperatures on a printed-wiring assembly. Many computer programs, such as Missile and Surface Radar's MOTHA (Moorestown Thermal Analysis) and NASA's SINDA (System Improved Numerical Differencing Analyzer), are available for general-purpose heat-transfer analysis. Although these programs are fast and accurate they do require extensive modeling. The "RADC Thermal Guide for Reliability Engineers," published by the Rome Development Center, estimates that a heat-transfer specialist with at least five years of experience would need one week to do a 20-node model of a printed circuit board and two weeks for a 70-node model.* To maximize the full productivity of a computer-aided-design (CAD) system, the database must be used.

The NOTA solution

The NOTA (NOdal Thermal Analysis) program, a Government Communications Systems FORTRAN program, can automatically model, compute, and display or print the steady-state temperatures of a 100-node printed-wiring board in a half hour because it uses the design database of the circuit board. The procedure for designing printed-wiring assemblies, using an Applicon CAD system, starts with the designer entering the logic/schematic diagram into the system. The physical board parameters are also entered into the database. Applicon software is then used to strip the net list from the logic/schematic and to generate component placement via a pairwise interchange algorithm, to provide optimum placement for printed-wiring interconnections. A net is a sequence of two or more connecting points interconnected by a copper

© 1983 RCA Corporation
Final manuscript received November 23, 1982
Reprint RE-28-1-9

* *RADC Thermal Guide for Reliability Engineers*, RADC-TR-82-122, June 1982, p. 58.

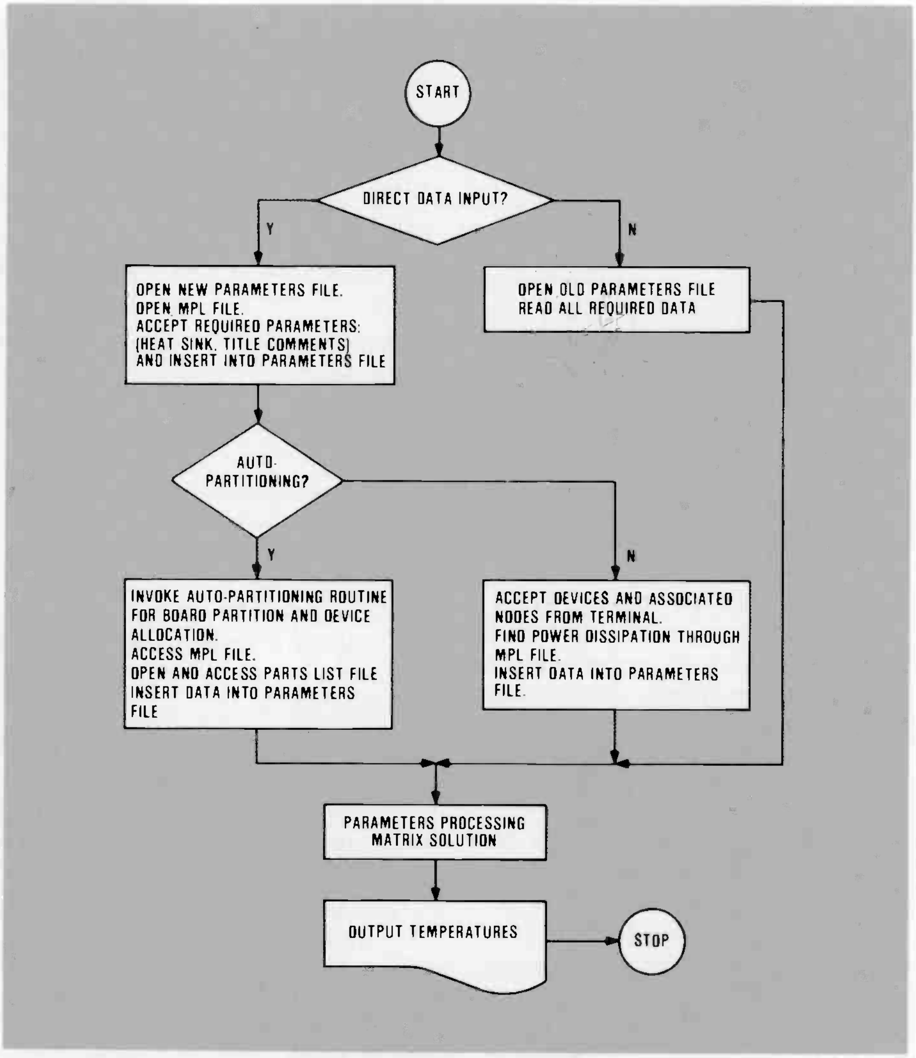

START
DIRECT DATA INPUT?
Y
N
OPEN NEW PARAMETERS FILE.
OPEN MPL FILE.
ACCEPT REQUIRED PARAMETERS:
(HEAT SINK, TITLE COMMENTS)
AND INSERT INTO PARAMETERS FILE
OPEN OLD PARAMETERS FILE
READ ALL REQUIRED DATA
AUTO-PARTITIONING?
Y
N
INVOKE AUTO-PARTITIONING ROUTINE FOR BOARD PARTITION AND DEVICE ALLOCATION.
ACCESS MPL FILE.
OPEN AND ACCESS PARTS LIST FILE
INSERT DATA INTO PARAMETERS FILE
ACCEPT DEVICES AND ASSOCIATED NODES FROM TERMINAL.
FIND POWER DISSIPATION THROUGH MPL FILE.
INSERT DATA INTO PARAMETERS FILE.
PARAMETERS PROCESSING
MATRIX SOLUTION
OUTPUT TEMPERATURES
STOP

Fig. 1. *Functional flow diagram for NOTA shows that the program can run in two different modes—Direct Data Input and Indirect Data Input.*

path. A net list is a listing containing all the nets in the circuit. For example, if Net 1 is U15-Pin 7, U18-Pin 7, CR2-Pin 1, then Pin 7 of the integrated circuit (IC) number 15 is connected to Pin 7 of IC number 18 and is connected to pin number 1 of the diode number 2.

The pairwise interchange algorithm is used to optimize the placement (locations) of components on the printed circuit board to facilitate the printed circuit routing. It does this by finding the minimum overall length of the line-of-sight interconnection for the entire circuit board. The process starts with the basic component placement as entered into the CAD system by the operator and systematically carries out an interchange of pairs of components while continuously calculating the length of the line-of-sight connectivity. The program will terminate when the minimum interconnect length is reached. This procedure could group several high-power dissipating components together, which might cause a localized heat problem.

At this time, the NOTA program can be exercised to predict component temperature. Boundary conditions such as ambient air and heat-sink temperatures, conductive-, convective- and/or radiation-equivalent thermal resistances are interactively entered into the system with the aid of friendly prompts. The automatic partitioning feature of NOTA will model the board into small square isothermal areas. NOTA will then ask if this model is satisfactory; the model can be changed at this time. The program will then automatically partition the portion of each heat-generating component located in each square, with the partitioned power dissipations summed to the center of each square (node).

The partitioning is performed using the drawing files containing the component placement and Master Parts List. This list contains part identification and typical power dissipation stored in the database. NOTA will now indicate components not found in the database. Missing components and/or power dissipations are then added to the database. NOTA will then compute and display the nodal temperatures.

Program description

NOTA, which is written in transportable FORTRAN language and configured to handle a computation of up to 100 partitionings (that is, a printed circuit board can be partitioned into as many as 100 nodes), has the following salient features:

- The program contains a self-explanatory man-machine interface input session.
- User inputs are either in numeric form or in the form of "yes" or "no" answers to computer prompts, so that no special programming is required.
- The program resides on the Applicon system, resulting in easy access to the printed-circuit-board component-placement data file.
- The program can perform automatic board-layout partitioning (via access to the Applicon placement file) and can precisely distribute the devices' power dissipation throughout various nodes within the partitioning limits.

As shown in the functional flowchart, Fig. 1, the program can be run in two different modes—Direct Data Input and Indirect Data Input. When using the Indirect Data Input, the program assumes that there is a data file (available on the system disc) containing all the necessary design parameters required for the thermal analysis of the printed-circuit board. This option is used for rerunning an analysis of an existing design residing in the database.

The Direct Data Input mode is used for constructing a database for running an analysis. When using this feature, the program requires a number of inputs from the user. The program expects to read the following inputs from the user's terminal:

- Conductive resistance;
- Convective and/or radiation resistance;
- Ambient temperature;
- Heat sinks (node sequence and temperature); and
- Optional printed-board-identification number, user's name and comments.

The remaining parameters, depending on the user's command, can be read directly from the user's terminal or extracted from the Applicon drawing-placement file. This option is shown in the functional flowchart at the auto-partitioning step. These parameters are:

- Number of partitioning columns;
- Number of partitioning rows;
- Component part numbers; and
- Component power dissipation.

If the user requests the user-input mode, the program will read-in the devices' identification (ID) numbers and search a database to obtain their associated power dissipations. If the user selects the automatic-partitioning mode, the program will need to access an additional parts-list file for all the devices' ID numbers before it can calculate the power dissipation. This is required because the Applicon drawing-placement file only contains the devices' symbolic numbers, (that is, U6, R20, and so on).

The user-input mode is useful for analyzing circuit boards that were not designed on the Applicon systems, and the board and component placement geometry are not stored in the database. However, when using this mode, the designer must model the circuit board.

In the auto-partitioning mode, the program extracts the following information from an Applicon placement file:

- Printed-circuit-board coordinates and size with respect to some fixed common origin;
- All the devices' coordinates, orientation, and symbolic part numbers;
- The relationship between Applicon Graphic System (AGS) and natural units;
- Natural unit of the printed-circuit board (that is, in., ft., mm, and so on); and
- Type of each device (this information is used by the program to obtain the device's size from an existing table and, in turn, used to determine the device's position within the printed-circuit board).

Solution

The heat-transfer equations are based on an analog of Ohms Law where

$E = IR$

E = Potential (volts)

I = Current or flow (amperes)

R = Resistance (ohms).

The heat transfer analog to $E = IR$ becomes

$\Delta t = QR$

Δt = Temperature potential (°C)

Q = Heat flow (watts)

R = Thermal resistance (°C/W).

Coefficients for the thermal nodal matrix are calculated from the following relationship:

Power dissipation =
Sum of ($(T_m - T_n)$/resistance)

Where T_m and T_n can be one of the following:

- Temperature of a node and its adjacent node;
- Temperature of a node and an associated edge; or
- Temperature of a node and the surrounding air.

The matrix equations contain the boundary condition variables such as heat sinks, ambient temperature, and forced convection.

The procedure for solving a system matrix equation is analogous to that of the Gaussian elimination method. The procedure is as follows:

1. Search for largest coefficient in a column.
2. Move the row that contains the largest coefficient to the top of the matrix if necessary.
3. Divide the entire row by the leading coefficient.
4. Eliminate next variables.

The steps above are repeated until the matrix becomes an upper triangular matrix. Note that if the absolute value of the largest coefficient in any column is equal to zero, the system of matrix equations is singular and no unique solution can be obtained. Finally, the last step, back solution, does the successive substitutions to obtain a unique set of solutions.

Data input

As part of the Nodal Thermal Analysis program, the Man-Machine Interface (MMI) routine is dedicated to handle the Direct Data Input running mode. The MMI interfaces to the user in interactive mode through a CRT or a teletype terminal. The MMI routine asks the user for inputs, then reads and stores them at appropriate memory locations in a file on the system's disk. Later, another routine of the thermal analysis program will retrieve the stored input file, then perform arithmetic calculations to arrive at a set of nodal steady-state temperatures.

The necessary inputs are requested by the routine in the following sequence:

1. Conductive and convective resistances;
2. Ambient temperature;
3. Temperatures at the board's edges (heat sinks);
4. Board configuration (number of columns and rows); and
5. Devices' generic part numbers, their associated nodes, and power dissipations.

The first three items above must be entered into the terminal by the user upon appropriate prompts. However, the user has the option to direct the program to obtain the last two items either from the terminal in user-input mode or from the Applicon file in auto-partitioning mode.

Running the equipment in user-input mode, the user must first enter the number of columns and rows of the partition model, then the part number of each component. During the input session, the MMI routine has access to a sequential file called the Master Parts List (MPL) file, which is stored on the system disc. After reading a device ID from the terminal, the MMI scans the MPL to locate this device. If the device exists in the MPL, the MMI will display its power dissipation (P.D.) value. The MMI also allows the user the option to input a new P.D. value that would supersede the old value. If the device is not already in the MPL, the MMI gives the user the option to enter an associated P.D. value for this device or go on to input a next device, in which case the previous device is ignored. Eventually, all the inputted devices' IDs, their associated nodes, and P.D. values of a particular board will be entered in a sequential parameters file for later reference by the Nodal Thermal Analysis program.

Partitioning

In auto-partitioning mode, the program extracts placement data from an Applicon file to compute the number of columns and rows and to obtain all the devices' symbolic numbers, that is, U3, R12, and so on. When the number of columns and rows are computed, the MMI displays these parameters and allows the user to change them as desired. The user also can change the node width, which defaults to 1.0 in. in the auto-partitioning process, to a number of different sizes if necessary. Note that by changing the partitioning node width the number of columns and rows will change as well. Next, the program uses a parts-list file to obtain all the devices' generic part numbers. Finally, the MPL file will pro-

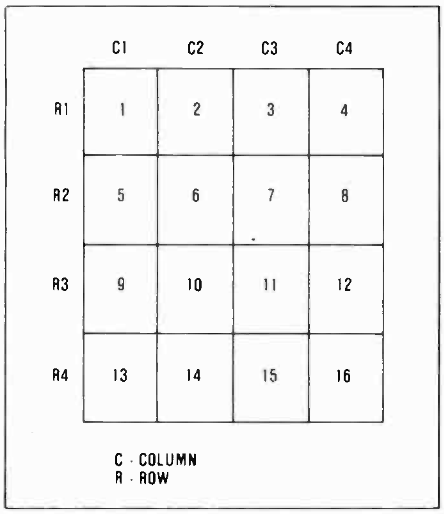


Fig. 2. *Printed-circuit-board node-number sequence shown for a 16-node model.*

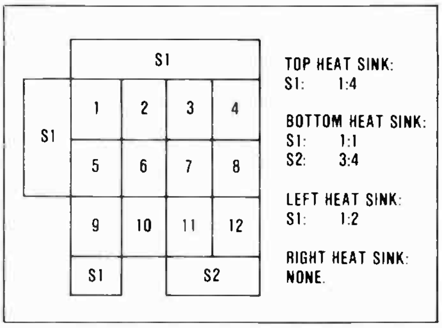


Fig. 3. *This heat-sink node representation shows one sink along the top edge, two sinks along the bottom edge, and one sink along the left edge.*

Table I. *The processor-board-parts list file (portion shown here) contains the reference designators and integrated circuit part number.*

Reference designator	Part number
U10	JM38510/30004BCX
U11	JM38510/30004BCX
U12	HM1-6514B-8
U13	HM1-6514B-8
U14	HM1-6514B-8
U15	HM1-6514B-8
U16	JM38510/30301BCX
U29	JM38510/30301BCX
U17	JM38510/30003BCX
U18	JM38510/30002BCX
U21	JM38510/30002BCX
U22	JM38510/30002BCX
U19	JM38510/30501BCX
U32	JM38510/30501BCX
U20	JM38510/32401BRX
U23	JM38510/30001BCX
U25	JM38510/07101BCX
U26	JM38510/32403BRX
U27	JM38510/32403BRX
U28	JM38510/30701BEX
U30	JM38510/32803BRX
U31	DS1692/833B
U33	JM38510/00801BCX
U34	JM38510/30106BEX
U35	DS78LS120J/833B
U36	Z8442ACMB
U38	JM38510/32004BCX
U39	JM38510/30901BEX
U40	JM38510/30901BEX
U41	JM38510/30107BEX
U42	JM38510/16303BEX
U43	JM38510/11201BEX

vide the program with all the devices' associated power dissipations, neccessary for the calculation of the final nodal temperatures. If the parts list does not contain an associated generic part number, or the MPL does not contain an associated power-dissipation value for a particular device, the program will display appropriate messages and allow the user to enter corresponding inputs as desired. During the auto-partitioning process, the associated power dissipation of any device (or part of the device) that is outside of the partitioning limits is ignored in the temperature-calculation process.

Finally, the user can go off-line and invoke the sequential parameters file through a text editor to make changes if necessary. The thermal analysis program then, upon entering the Indirect Device Input mode, obtains information directly from the sequential parameters file rather than through the MMI session for a nodal thermal analysis. The sequence (updating the sequential parameters file and executing the thermal analysis program) can be repeated until an optimum solution is obtained.

In general, a given printed circuit board can be divided into 100 nodes maximum, and the number of columns (or rows) can be, at most, 20. However, any partitioning scheme satisfying the formulas given below is allowed:

$$1 \leq \text{nodes} \leq 100$$
$$1 \leq \text{columns} \leq 20$$
$$1 \leq \text{rows} \leq 20$$

The number of nodes is simply the product of the values of the number of columns and rows of a board. For example, a board divided into four columns and five rows will have 20 nodes. All nodes must be square for the thermal analysis program to compute output temperatures correctly. By program default, node size is one inch square, but node size can be tailored to a number of different configurations to meet a user's requirements. But we recommend that all nodes in any particular column (or row) have the same width.

Nodes within a board are numbered from left to right, and from top to bottom in ascending order starting at node 1, then followed by node 2, node 3, and so on. Figure 2 illustrates how node numbers must be sequenced.

Heat sinks on the boards's four edges are called *top heat sink(s), bottom heat sink(s), right heat sink(s),* and *left heat sink(s).* If there is more than one heat sink on the top or bottom (right or left) edge, the heat sinks should be numbered from left to right (top to bottom) in ascending order starting at heat-sink 1, then followed by heat-sink 2, and so on. If a particular heat sink spans from node *i* to node *j*, for example, the heat sink is represented by (node *i*:node *j*). Figure 3 is an example of several heat sinks and their nodal representation.

Example

Figure 4 shows an assembly drawing of a processor board partitioned into nine columns and five rows. The multilayer assem-

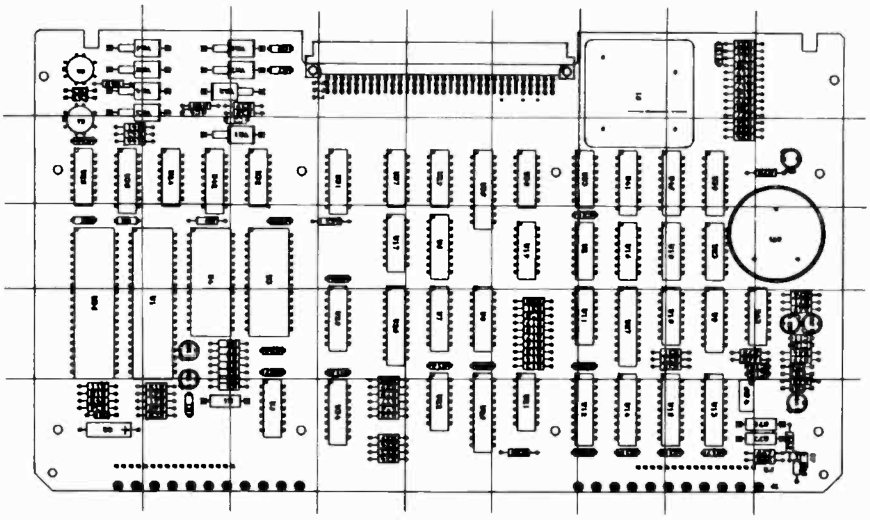
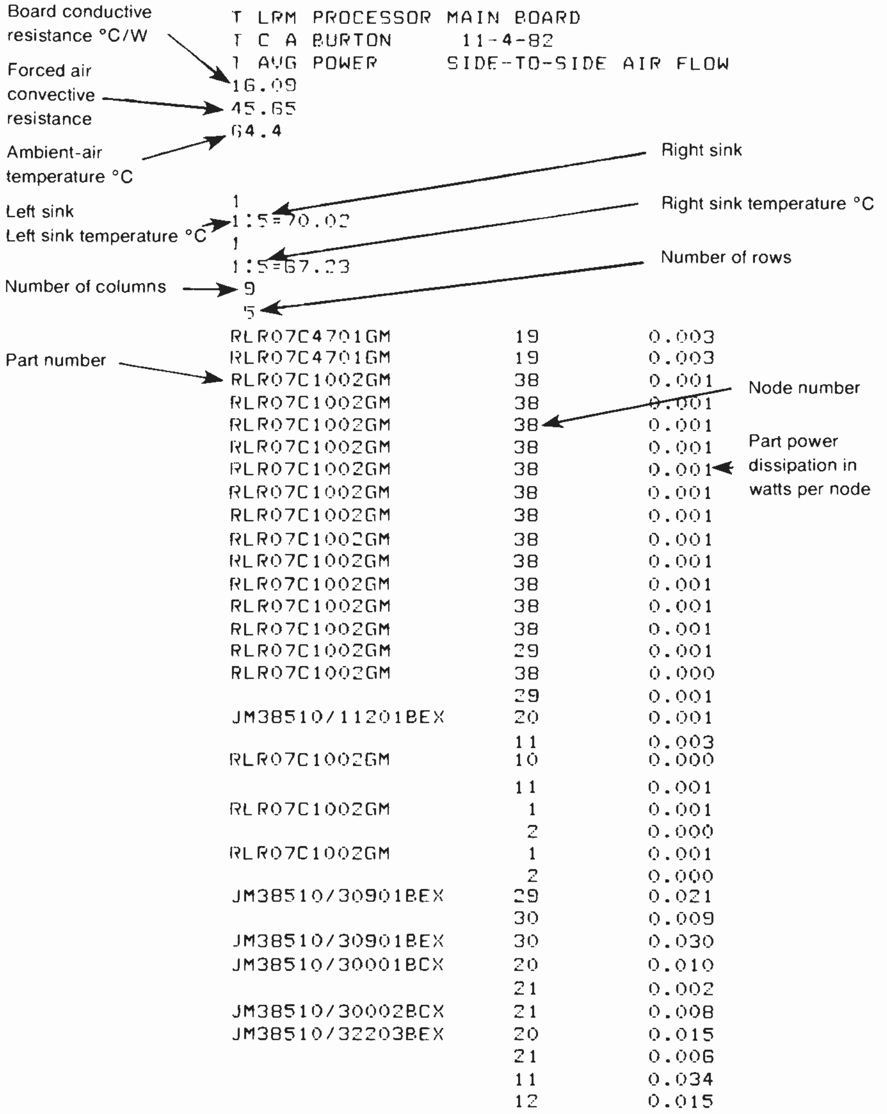


Fig. 4. *This processor board is partitioned into 9 columns and 5 rows.*

Table II. *Master Parts List file (portion) containing the part number and typical power dissipation in watts.*

Part number	Power dissipation
JAN2000	0.20
JAN1000	0.30
RCA07	0.15
M39014	0.20
INTEL 2708	0.90
SN7400	0.09
SN7404	0.17
SN7410	0.09
SN7412	0.08
SN7474	0.12
SN7486	0.15
SN74155	0.13
SN74161	0.33
SN74175	0.23
TR1602	0.32
LM3900	0.50
MC 68000	1.20
AM2904	0.65
RCA 1802	0.95
RCA ATMAC II	1.60
SN74LS74	0.35
DUMMY	5.00
M39003	0.25
Z8400ACMB	0.5
Z8430ACMB	0.3
Z8442ACMB	0.25
HM1-7616-8	0.45
HM1-6514B-8	0.009
MKB4118AP-83	0.175
JM38510/30001BCX	0.012
JM38510/30301BCX	0.014
JM38510/30002BCX	0.008
JM38510/30003BCX	0.018
JM38510/30004BCX	0.009
JM38510/00801BCX	0.16
JM38510/31004BCX	0.022
JM38510/31001BCX	0.017
JM38510/30501BCX	0.025
JM38510/30102BCX	0.020
JM38510/07101BCX	0.15
JM38510/30701BEX	0.032
JM38510/30702BEX	0.034
JM38510/30901BEX	0.030
JM38510/30107BEX	0.055
JM38510/30106BEX	0.080
JM38510/32401BRX	0.145
JM38510/32403BRX	0.160
JM38510/32803BRX	0.32
JM38510/32004BCX	0.045
JM38510/32203BEX	0.070
JM38510/16303BEX	0.325
JM38510/11201BEX	0.004
DS78LS120J/833B	0.050
DS1692/833B	0.090
RLR07C2701GM	0.004
RLR07C4701GM	0.003
RLR07C1002GM	0.001

Table III. *Processor-board file (portion) containing the thermal boundary conditions, node number, part number, and partitioned power dissipations.*


Board conductive resistance °C/W
Forced air convective resistance
Ambient-air temperature °C
Left sink
Left sink temperature °C
Number of columns
Part number
Right sink
Right sink temperature °C
Number of rows
Node number
Part power dissipation in watts per node

```
T LRM PROCESSOR MAIN BOARD
T C A BURTON        11-4-82
1 AVG POWER       SIDE-TO-SIDE AIR FLOW
16.09
45.65
64.4

1
1:5=70.02
1
1:5=67.23
 9
 5
RLR07C4701GM          19       0.003
RLR07C4701GM          19       0.003
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          38       0.001
RLR07C1002GM          29       0.001
RLR07C1002GM          38       0.000
                      29       0.001
JM38510/11201BEX      20       0.001
                      11       0.003
RLR07C1002GM          10       0.000
                      11       0.001
RLR07C1002GM           1       0.001
                       2       0.000
RLR07C1002GM           1       0.001
                       2       0.000
JM38510/30901BEX      29       0.021
                      30       0.009
JM38510/30901BEX      30       0.030
JM38510/30001BCX      20       0.010
                      21       0.002
JM38510/30002BCX      21       0.008
JM38510/32203BEX      20       0.015
                      21       0.006
                      11       0.034
                      12       0.015
```

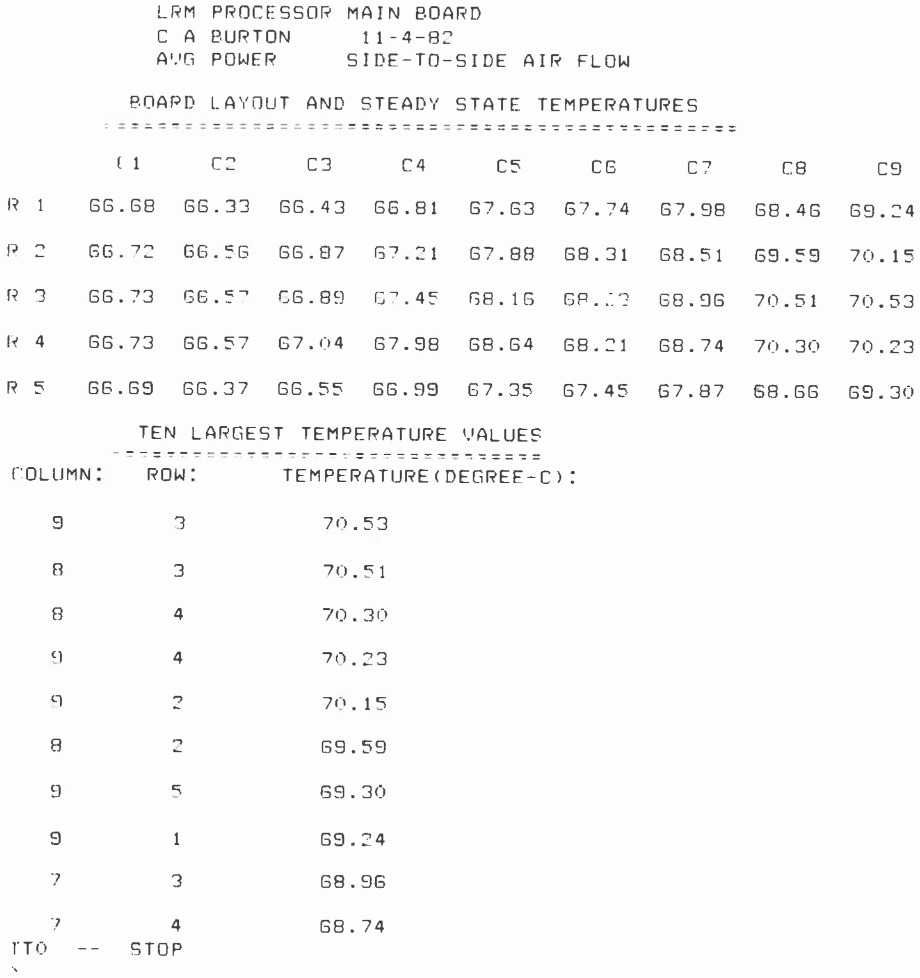
bly contains integrated circuits and discrete components. The printed wiring board was designed on the Applicon system in the Engineering Design Automation Center, Government Communications System. The design procedure followed was as outlined above. This assembly is cooled by forced convection and conduction to heat-sinking board guides located on the left and right edges of the circuit board. Based on its multilayer construction, the circuit board's conductive resistance was calculated to be 16.1°C/W. Convective resistance was calculated to be 45.7°C/W, based on the linear velocity of the cooling air per nodal area. The ambient air temperature was given as 64.4°C and the right and left sink temperatures were 70°C and 67.2°C respectively.

Table I shows a portion of the processor-board parts list, where the reference designators correspond to the generic part identification. Table II shows a portion of the Master Parts List, which contains the various components and their typical power dissipations. Table III shows a portion of the board file containing the partitioning information. Note that this file also contains the conductive and convective resistance, ambient air temperature, right and left sink geometry and temperatures, and the column and row data. The components are shown by type, node location and allocated power dissipation per node. Table IV shows nodal temperatures by row and column, and gives the ten highest temperatures found.

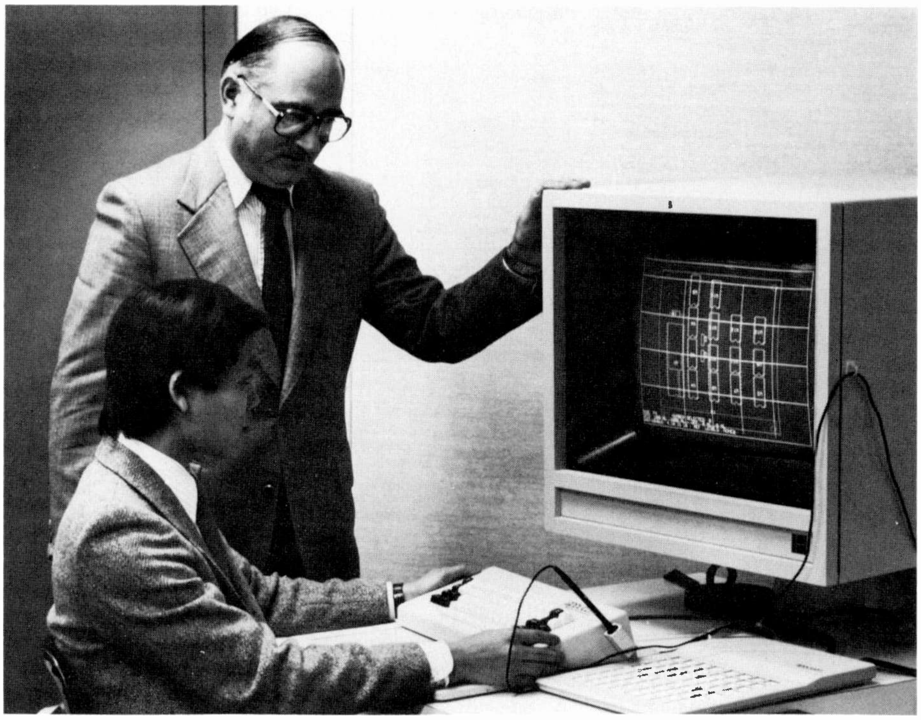
Conclusion

The engineering productivity enhancements that computers provide are well understood. Computers now make it possible to calculate and predict the answers to problems that could only be guessed at in the past. To maximize engineering productivity provided by a CAD system, engineers must use the stored database to its fullest capacity.

Table IV. *Steady-state nodal temperatures in °C. The rows and columns correspond to the processor-board partitioning shown in Fig. 4.*



```
              LRM PROCESSOR MAIN BOARD
              C A BURTON       11-4-82
              AVG POWER        SIDE-TO-SIDE AIR FLOW

            BOARD LAYOUT AND STEADY STATE TEMPERATURES
          ==========================================

            C1      C2      C3      C4      C5      C6      C7      C8      C9

R 1      66.68   66.33   66.43   66.81   67.63   67.74   67.98   68.46   69.24

R 2      66.72   66.56   66.87   67.21   67.88   68.31   68.51   69.59   70.15

R 3      66.73   66.57   66.89   67.45   68.16   68.33   68.96   70.51   70.53

R 4      66.73   66.57   67.04   67.98   68.64   68.21   68.74   70.30   70.23

R 5      66.69   66.37   66.55   66.99   67.35   67.45   67.87   68.66   69.30

             TEN LARGEST TEMPERATURE VALUES
            ============================
COLUMN:    ROW:        TEMPERATURE(DEGREE-C):

   9         3              70.53

   8         3              70.51

   8         4              70.30

   9         4              70.23

   9         2              70.15

   8         2              69.59

   9         5              69.30

   9         1              69.24

   7         3              68.96

   7         4              68.74
TTO   --   STOP
```



Authors Nguyen, seated, and Wolfe, standing, at an Applicon terminal. The display shows a partitioned printed circuit board made ready for a NOTA analysis.

Don Wolfe is Manager of Engineering Design Automation in Government Communications System at Camden, N.J., where he is responsible for engineering CAD/CAM, and the Engineering Design Automation Center. He joined RCA in 1952 and has 30 years of experience in electronic packaging. Mr. Wolfe is a lecturer in the RCA Continuing Education Program where he developed video lecture series on Shock and Vibration and Mechanical Modeling.

Contact him at:
RCA Government
Communications Systems
Camden, N.J.
TACNET: 222-2807

Thach Nguyen graduated with a BSEE degree in 1980 from Pennsylvania State University. In January, 1982, he joined RCA as a Software Development Engineer in the Government Communications Systems Software Engineering activity, Camden, N.J. He is currently developing real-time software for EISN, an Experimental Satellite Integrated Switch Network.

Contact him at:
RCA Government
Communications Systems
Camden, N.J.
TACNET: 222-2357

W.B. Schaming | J.S. Zapriala

Advanced Technology Laboratories' Image Processing Facility

Highly sensitive imagers, coupled with processing equipment and cleverly programmed computers, create entirely new, analytical and complete ways of seeing.

Abstract: *Digital image processing done at the Advanced Technology Laboratories' sophisticated Image Processing Facility, in Camden, has many useful applications. Illustrated examples include target detection and tracking, facsimile bandwidth compression, and optical disk demonstrations. The authors present the facility's equipment and capabilities, together with some principles of operation. The conclusions include a look at future applications for image processing.*

Digital image processing can be viewed as the manipulation of images by computer. Man, long fascinated by visual stimuli of all types, has applied image processing to practically every type of imagery, thus fuelling this interest. It is a multidisciplinary endeavor combining aspects of optics, electronics, mathematics, photography, and computer technology.[1]

Digital image processing can be applied in a plethora of situations, so a facility that is both extremely versatile and able to cater to individuals with various backgrounds and levels of expertise must be available. To meet this real need, the Advanced Technology Laboratories' (ATL) Image Technology Laboratory has assembled the Image Processing Facility (IPF) diagrammed in Fig. 1. This facility has been useful for various applications including target detection and tracking, facsimile bandwidth compression, and optical disk demonstrations.

Facility

The Image Processing Facility is composed of four essential components:

1. An International Imaging Systems Model 70/E Image Computer and Display Terminal.
2. The general-purpose Image Processing Operating System, which creates the "friendly wall" separating the user from the rest of the facility.
3. A Hewlett-Packard 1000 Model Performance 45 16-bit minicomputer, functioning as the image computer's host, that provides both powerful computational capabilities and program-development facilities. Its primary purpose is to act as the all-important "window to the world" for the display terminal.
4. The host's own Operating System, respon-

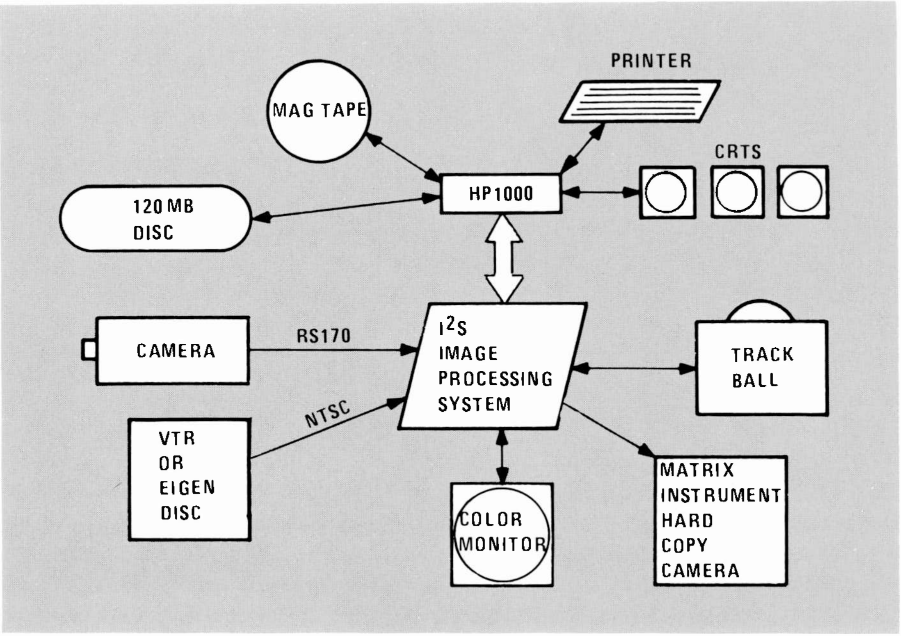


Fig. 1. *Image Processing Facility overview.*

©1983 RCA Corporation
Final manuscript received September 9, 1982
Reprint RE-28-1-10

sible for providing the resource and file management services required by the system.

Note that there is no direct communication between the Image Processing Operating System and the Image Computer. Linkage between these two subparts is accomplished via the host's hardware/software interface.

The Image Computer

The display terminal acts as an extremely powerful peripheral device, the keystone around which the remainder of the system is built. Just as with any other traditional computer, the Model 70 is organized into separate input, output, memory, and arithmetic/control modules.

There are two key elements in the Model 70: the feedback/ALU (arithmetic/logic unit), and the pipeline processors.

The feedback/ALU allows one to retain processed data (appearing at the pipeline output) and return it to refresh memory for later processing in either the display terminal or the host machine. With the feedback/ALU, it is possible to perform recursive procedures such as convolutions in the spatial dimension. The 16-bit data that results from these operations is returned to two 8-bit refresh memory channels that have been "concatenated" to form what is called an accumulator channel. The ALU can perform 48 different logic/arithmetic operations. These ALU operations may be restricted to a specifically defined region-of-interest (ROI), a spatial mask defined by the user in one of the system's graphics channels. Two different operations may be performed concurrently—one inside the ROI, the other outside. The ALU output may be routed to any other refresh memory(s) (including the 16-bit accumulator) or graphics channel.

Processing in each of the pipeline processing channels (one each for red, green, and blue) occurs at real-time video rates. The pipelines are responsible for arithmetic operations as well as generalized combinatorial and switching operations. This means that traditional image transforms can be implemented in (or near to) real time.

The basic pipeline architecture allows for the basic arithmetic operations using the look-up tables (LUTs), adder array, and output function memory (OFM). By using the proper loading of the LUTs and OFMs (with positive, negative, antilog, or negative log mappings), one can perform addition, subtraction, multiplication, and division (or ratio) between any two images in real time.

Each pipeline also provides the circuitry responsible for implementing hardware scroll, zoom, and split screen. The split-screen capability allows rectangular portions of various images currently stored in refresh memory to be simultaneously displayed on a single monitor. This permits the user to perform A-B comparisons of identical portions of different images.

Image Processing Operating System

The System 500 (S500) Image Processing Operating System is usable by individuals with all levels of expertise, including those unfamiliar with computer programming and/or the technical aspects of digital image processing. This arrangement permits even the novice user to quickly begin using even some of the advanced image-processing functions that are available, without having to become familiar with either the host's or image computer's operations.

This complete "turnkey" software system operates as a subtask of the host computer's operating system. The system includes an I/O driver that acts as a device handler for the Model 70. In addition, an image database management subsystem supports data on disc, magnetic tape, and image computer refresh memory. It permits subsectioning, subsampling, and subbanding of images as well as the conversion between byte, integer and real data types. Also, a centralized interpreter responsible for evoking the applications programs in the system, and an independent, non-fatal, error handler are included. A diagnostic interpreter is used for the initial debugging of computational techniques as well as hardware-checkout·functions; small programs can be created and modified by a special internal editor, and then executed, in order to access and control the subunits that comprise the display terminal.

The system also includes a large repertoire of applications programs that provide both elementary and advanced image-processing techniques including:

- Basic image-array arithmetic operations (add, subtract, multiply, divide and ratio) along with the interactive summing of two images under trackball control.
- Histogram calculation done in hardware.
- Profile and points statistics.
- Plotting scattergrams (bivariate frequency distributions) (Fig. 2).
- Both replicative and interpolative zoom.
- Magnification and minification of images.
- Reflection and transposition operations on display-resident images.
- Image rotation.
- Roam and split-screen examination (Fig. 3).

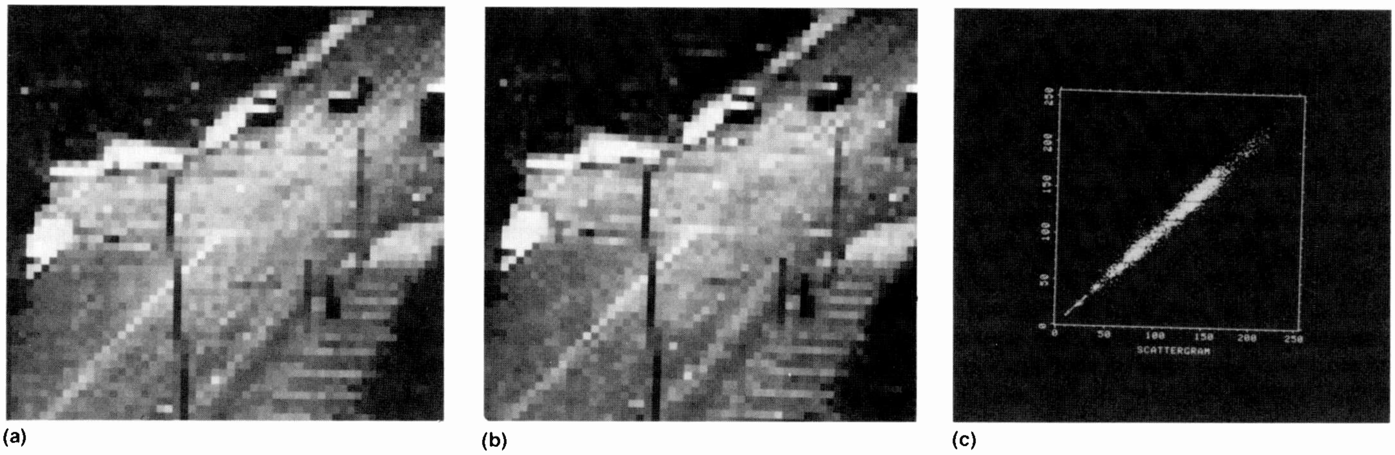


Fig. 2. *Two successive frames* **(a)** *and* **(b)** *are in a sequence used for tracker development, with* **(c)** *showing the resulting scattergram.*

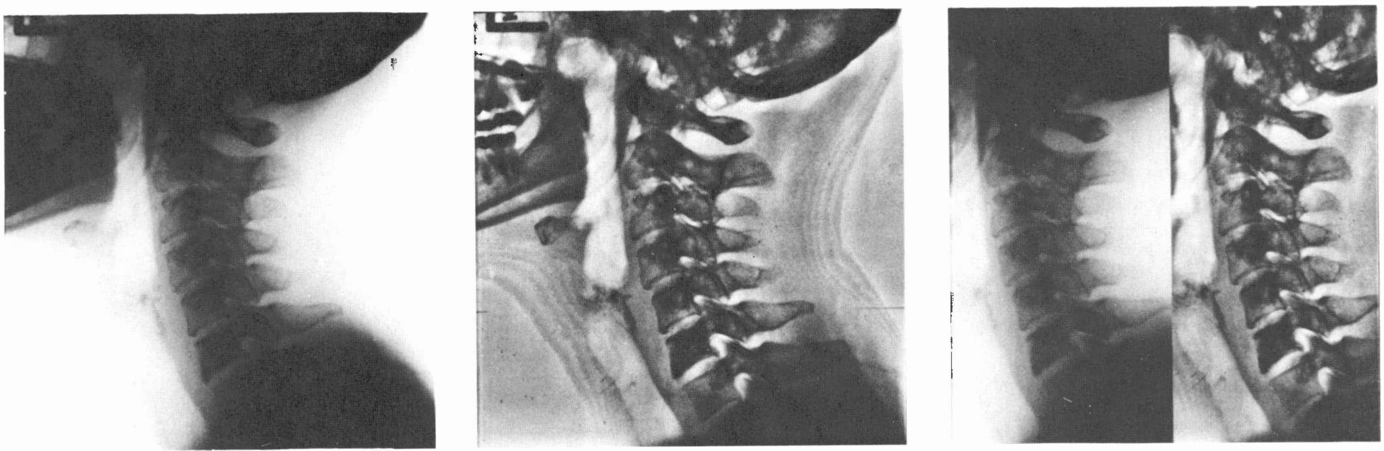

(a) (b) (c)

Fig. 3. *Images* **(a)** *and* **(b)** *can be compared side-by-side using the split-screen function.*

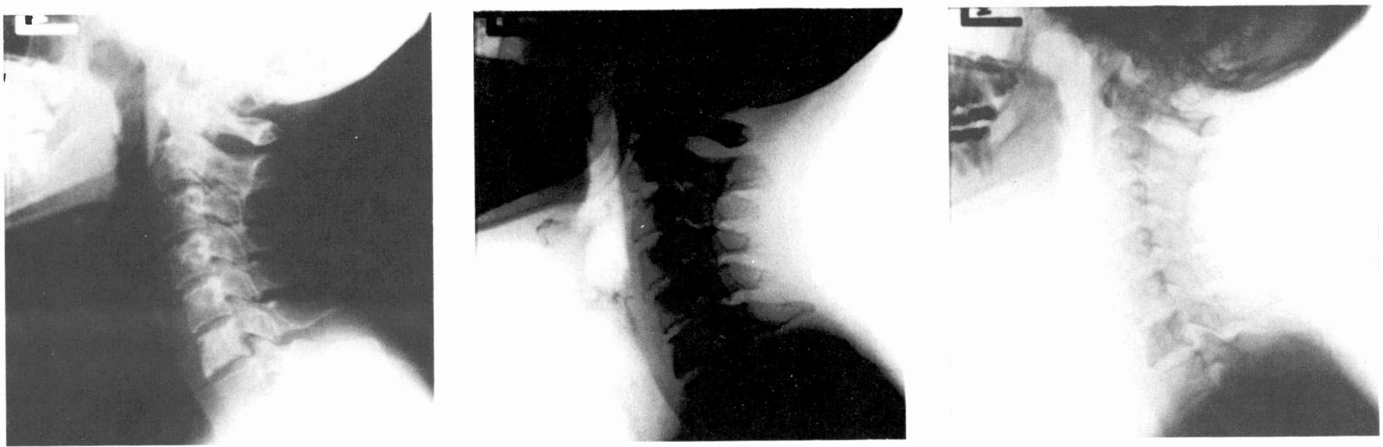

(a) (b) (c)

Fig. 4. *Readily-available transforms include negative transforms* **(a)**, *histogram equalization transforms* **(b)**, *and logarithm transforms* **(c)**.

- Karhunen-Loeve, Hadamard, FFT (one-dimensional: spectral/two-dimensional: spatial), and Slant transforms.
- Scalar, negative, logarithmic, exponential, as well as histogram equalization and normalization transforms (Fig. 4).
- Radiometric transformations of images (to a user-specified mean and standard deviation as well as the ability to have an image altered so that its histogram matches that of another's).
- Local-area enhancement (Fig. 5).
- Deskewing and destriping Landsat and other multispectral scanner images.
- Applying a Gaussian-shaped frequency domain filter to an image.
- One- or two-dimensional Tukey filtering.
- Minimum distance classifier.
- Unsupervised classification.
- Image contouring.
- Piecewise linear intensity mapping (Fig. 6).
- Level slicing (Fig. 7b).
- False (pseudo)-color assignments that can be either automatic or user-specified.
- Trackball-controlled linear mapping (Fig. 7c).
- Spatial convolutions with programmable kernel size and weights.
- Animation of images using "flicker" techniques (video loops).
- Digitization (and frame grab with real-time averaging) for direct-image input.
- Graphics utility functions (Fig. 8).
- Real-time transforms and false-coloring.

Each of the applications programs is a unique subtask, independent of all other applications programs in the system, making each independently modifiable.

The S500 structure was organized to meet the goals of

- Modularity and ease of addition;
- Capability;
- Flexibility and ease of use; and
- Portability.

The system is transportable to various computers. The majority of the code is written in ANSI FORTRAN with the remainder implemented in the host's assembler language. This feature enhances portability because only a lower-level machine-dependent code must be converted when functions are to be moved to another incompatible machine. There are three functional categories of machine-dependent modules: (1) task scheduling and inter-task communication, (2) bit/byte manipulation, and (3) I/O linkage (permitting the initiation of I/O to/from a physical device).

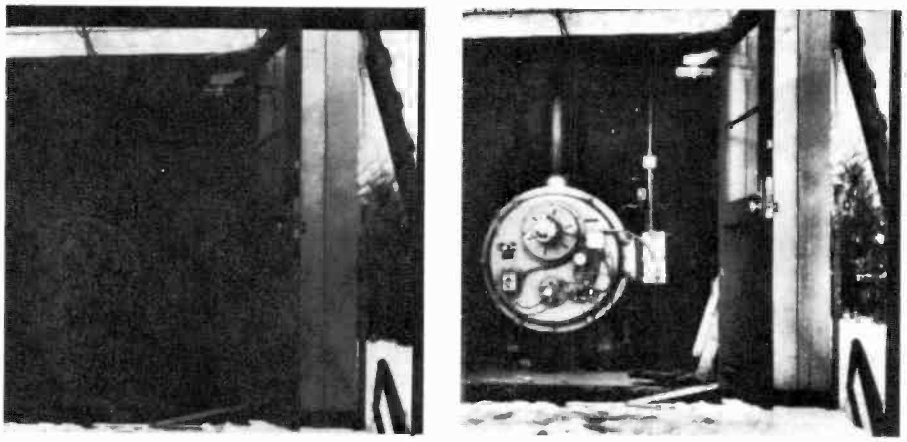
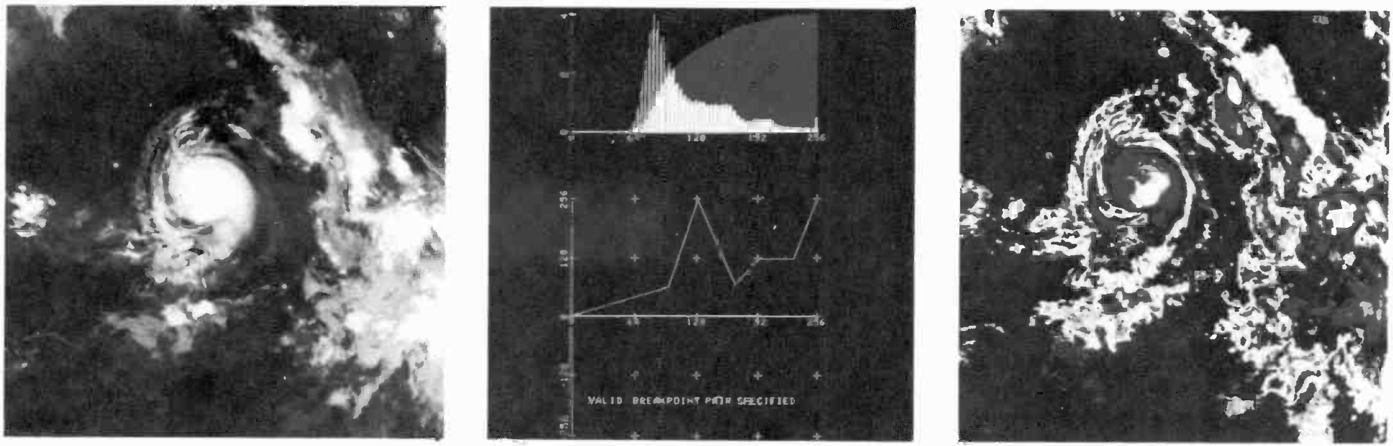
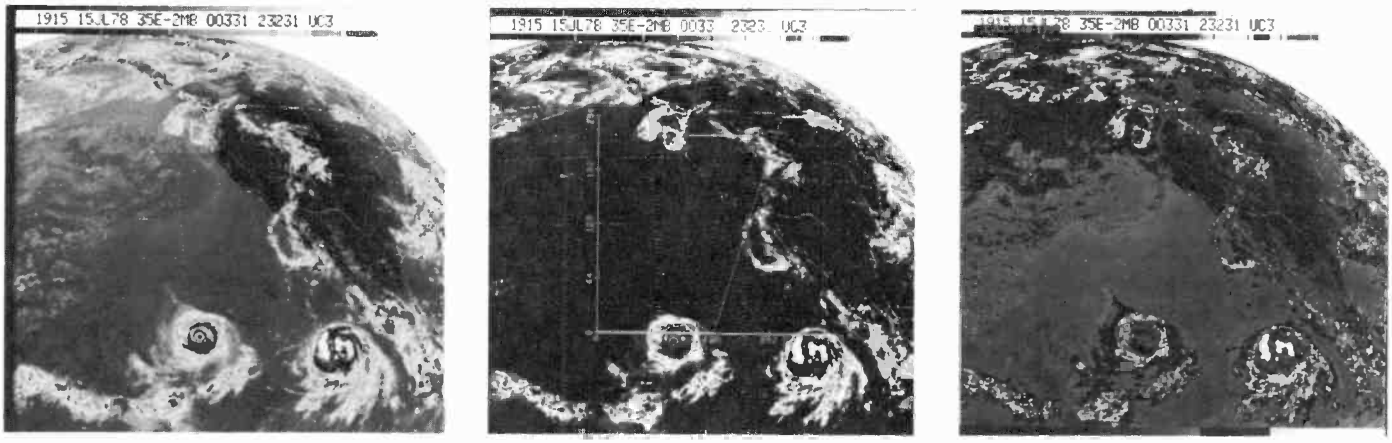


Fig. 5. *Local-area enhancement can be used to "find" visual information not readily apparent at first glance.*



(a) (b) (c)

Fig. 6. *An image can be transformed* **(c)** *by a user-defined intensity mapping in the lower part of* **(b)**.


1915 15JUL78 35E-2MB 00331 23231 UC3
1915 15JUL78 35E-2MB 00331 23231 UC3
1915 15JUL78 35E-2MB 00331 23231 UC3

(a) (b) (c)

Fig. 7. *The trackball is used as an interactive input device in many of the built-in functions, including trackball-controlled linear mapping, where its position determines the breakpoints used in the intensity mapping that has been graphically overlaid in the figure* **(b)**, *or the position and width of the window used in the level-slice function* **(c)**.

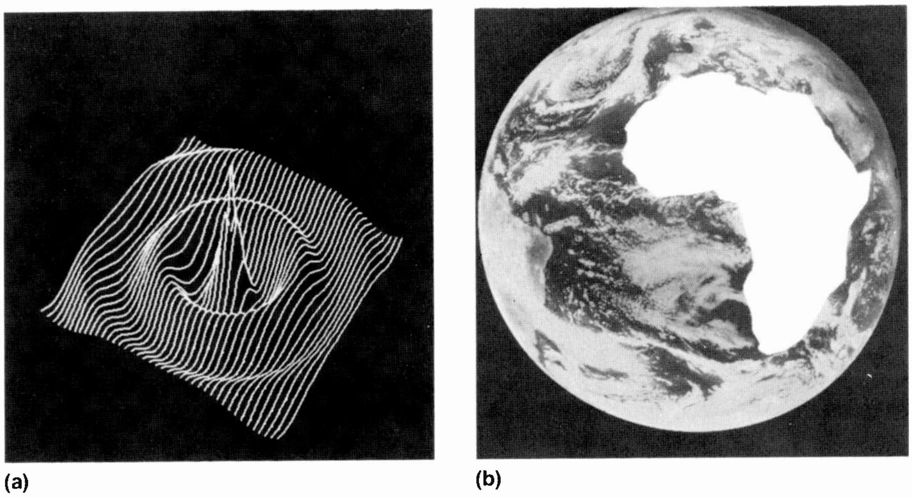


Fig. 8. *The graphics utility functions present in the system include those responsible for generating three-dimensional plots* **(a)** *or regions-of-interest* **(b)**.

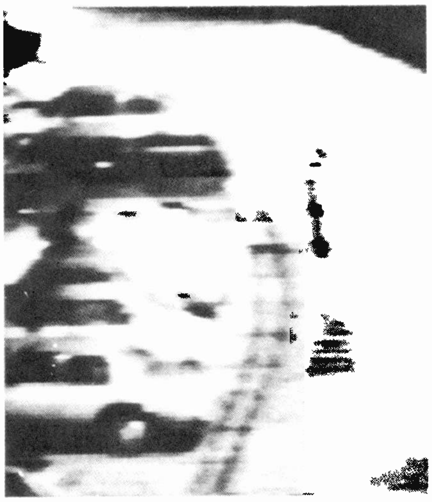


Fig. 9. *64×128 element IRCCD/FPA imagery.*

Simulation imagery

Most of our work uses imagery obtained from two-dimensional sensor systems such as RCA's infrared charge-coupled device focal plane array (IRCCD/FPA). The monolithic silicon Schottky-barrier (Si SB) focal plane array has been under development at RCA Laboratories since 1972. The device, which is operated at 80 Kelvin, is sensitive in the 1.2- to 4.5-μm spectral region.

> "IR radiation is passed through the silicon substrates (present in the device) and received by the Schottky diodes. Interline column CCD registers are used to receive charge from the diodes and to shift information one line at a time into a horizontal shift-register for serial readout. This parallel-to-serial on-chip multiplexing provides a high photon-collection efficiency, no mechanical scanning and a minimum number of off-chip electrical connections."
>
> "Furthermore, because of the excellent array uniformity inherent to Schottky-barrier detectors, spatial uniformity data processing is not needed. The photoemissive photodetection process ensures high speed, high MTF (modulation transfer function), and very low hot-spot bloom."[2]

A 25×50 element array was originally used to develop and demonstrate the technology, with 32×64 and 64×128 (Fig. 9) array imagery already being demonstrated.

Imagery for simulations is obtained with minimal effort. 512×512×8 bit imagery can be input via magnetic disc or tape. External data capture (frame grab) is efficiently performed by the 10-MHz video digitizer, which can digitize an RS-170 signal (for example, from a TV camera, videotape recorder or video disk) in real time. The captured 8-bit 512×480 image is then transferred to magnetic disk (though not in real time).

The digitizer channel appears to be a pseudo-refresh memory channel to allow for the real-time (digital) processing of video input. The overall speed of the system can be greatly enhanced by performing initial scaling tasks in the display terminal's input function memory (IFM), thereby offloading the process of mapping the data being loaded into the refresh memory(s) from the host computer.

Each image can be represented as a two-dimensional array of intensities (gray levels). Obviously, attempts to evaluate the results of an experiment by paging through reams of printed output, where numbers begin to blur together after awhile, can be both a painful and mystifying experience. The use of an impact printer to create output imagery, where one tries to correlate character density with gray level, can be an extremely time-consuming and inexact process.

In the past, we were somewhat better off than most because we had the option of creating specially-formatted computer tapes that were used to "write" out imagery onto film with an Optronics photowriter. Then, after the film was processed, we had to spend hours examining the film with a magnifying glass to determine how well our algorithm had performed.

Obviously, it is better to watch one's simulator in action, where each image, along with color graphic overlays of pertinent information, could be viewed on a color video monitor immediately following computation. Our IPF now permits an engineer to correlate input parameters with performance, and to possibly get a "feel" for frame-to-frame events. Simulations in this system are extremely well facilitated.

Each of the simulators being developed was originally implemented without the use of the image computer for display purposes. This was done to verify correct operation before undertaking the task of interfacing the computational side of the simulator to the display terminal.

With the use of subtasking, one can create a simulation program that will communicate with a "display" function that has been integrated into the S500 function library.

Due to the inherent structure of S500, user-written additions are easily and quickly accomplished. The addition of a new function into the function library requires only two steps: (1) load the program on the host computer; and (2) insert the new function's name into System 500's function-name directory so that the command interpreter will know of its existence.

Every function in the function library makes use of:

- The centralized parameter subsystem when prompting the user for program parameters.
- The utility tasks, which afford the programmer easy access to the functional capabilities of the Model 70 (including a common set of I/O routines that are used to perform all image I/O operations).

The simulation-display function is fairly easily assembled with the use of modules that control and converse with the functional subunits present in the Model 70. Its primary purpose is to act as the liaison between the simulator and the video display that is exhibiting its performance.

The Model 70 can be equipped with up

to twelve 512×512 pixel refresh memories. Presently three are available. Each of these refresh memories is 8-bits "deep," permitting each pixel to take on any one of 256 (0-255) gray, or intensity, levels. All nongraphic information is written into these refresh memories.

A single array of up to eight (three are presently available) independent 1-bit memories is available in which to write graphic overlays of binary data. Each graphics plane may be assigned a unique color, with a different color assigned to each of the possible combinations of graphics channels. The annotation of images can be easily accomplished by placing alphanumeric character images in the graphics planes.

Interactive input to any of the simulators can be accomplished with a terminal keyboard or via a trackball unit to selectively control the (programmable) cursor, operating as a pseudo-graphics channel, in the x and/or y direction (independently or simultaneously). Four function buttons are provided with the trackball. The status of these buttons can be checked by the host computer, thereby allowing rapid user indication of desired menu choices.

Video output and hard copy

Just as the Model 70 is compatible with NTSC and RS-170 formats on input, it is also capable of producing RS-170 and NTSC video on the output side using the NTSC encoder that is included in the facility. The encoder permits RGB output (being sent to the display monitor) to be converted and transmitted to devices such as VTRs and video disk units.

Although our simulations do not, obviously, take place in real time, their results can be recorded on a video disc one frame at a time (an option built into every simulation package) and later replayed in real time (Fig. 13).

The IPF is also equipped with a Matrix Instruments Multi-Format Hard Copy Camera (Model 4007) allowing the RGB information to be "filmed" on Polaroid film sheets (either 4×5 or 8×10) in several formats as well as in 35-mm or (Polaroid) SX-70 format. The imagery in this article was recorded using the Matrix camera.

Current applications

There are two principal application areas that one can speak of when the subject of image processing arises: the improvement of pictorial information that is to be interpreted by humans, and the processing of image data for machine perception.

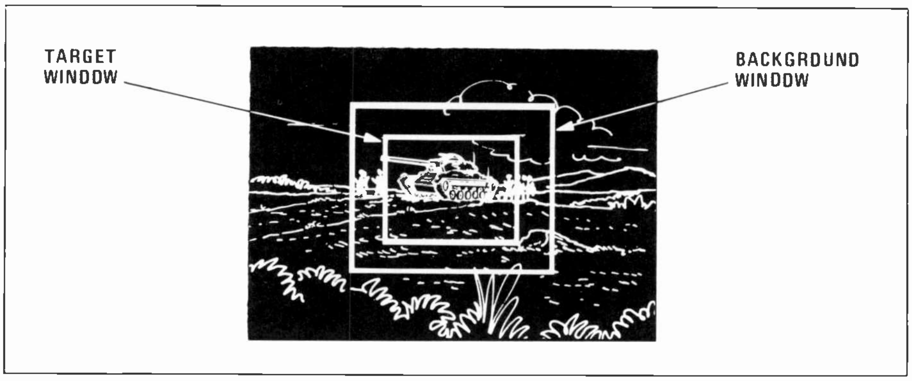


Fig. 10. *Tracker window concept.*

Although machine perception algorithms and methods can be used to perform character recognition and fingerprint processing, weather-prediction and crop-assessment using both aerial and/or satellite imagery, and product assembly and inspection using industrial robots,[3] our primary interest area is that of military applications such as the development of target detection, tracking, and identification techniques.

The military has been actively pursuing the development of "smart sensors" that would augment or even replace the human observer that is presently the keystone of the target acquisition and munitions guidance process. This process includes:

1. The detection of objects that could be targets.
2. The tracking of these objects until they can be recognized with some level of confidence.
3. The identification of the object as a sought-after target to be engaged.
4. Munitions launch.
5. Reacquiring the target, because it may have moved during the transient introduced by Step 4.
6. Tracking the target until final munitions impact.

Automatic tracking systems possessing some "intelligence" would be of great aid in systems where a human cannot always be a part of the classic control loop, and a must for "fire and forget" (fully autonomous) weaponry.[4]

Multifeature Bayesian Intelligent Tracker

We have developed a Multifeature Bayesian Intelligent Tracker (MFBIT) that provides superior capability for separating targets from background under a multitude of conditions, which will enhance the performance of many military systems that use electro-optic sensors.[5,6] The breadboard testbed is integrated into the IPF.

In a nutshell, the MFBIT derives image statistics such as intensities and edge magnitudes from two disjoint regions (Fig. 10): (1) a target window which encompasses a designated area, and (2) a background window—the immediate surround (window frame).

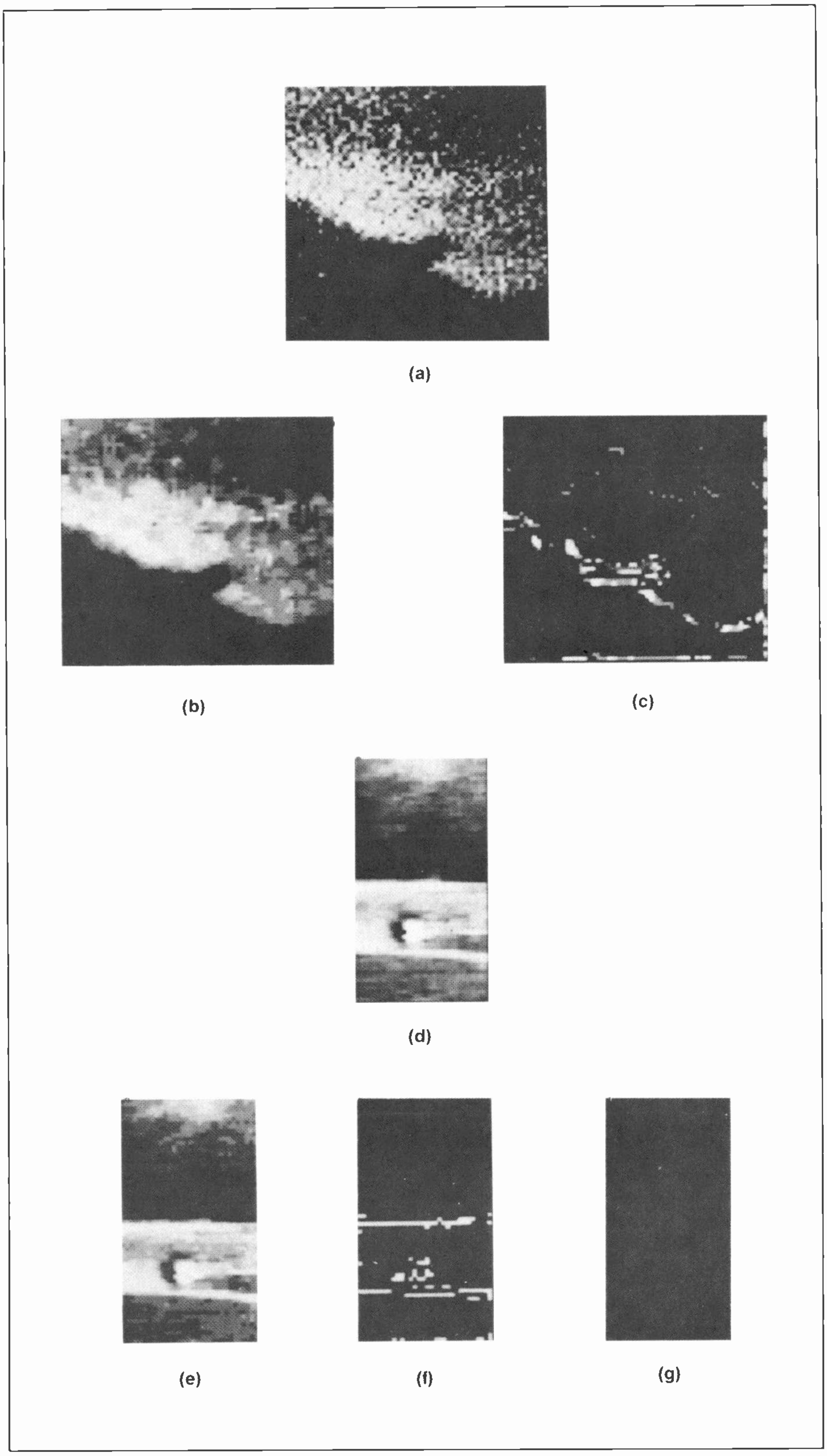
These windows form joint-feature distributions from these statistics. These statistics are combined into a single vector used to segment the target from the background (Fig. 11).

In the past, simple thresholding has often been used to separate targets from their background. The computation of the threshold can be quite sophisticated and is usually accomplished by investigating the statistics of the incoming video signal. The problem lies with the fact that it has already been shown that simple thresholding is severely limited in its ability to perform satisfactorily.

At the present time, a typical multimode tracker executes a contrast, edge, and correlation tracker in parallel. A supervisory or executive process is then used to determine which of the modes is providing the best estimate of target position. The MFBIT overcomes the necessity of making this type of judgment by combining the available information into multidimensional statistics that represent target and background.

As an example, let us assume that we are using intensity and edge magnitude as our two features. What we essentially have is a case where the MFBIT is encompassing all three modes while operating in a single mode:

1. When intensity is the best separator of target versus background, the tracker operates as a contrast tracker.
2. When edge magnitude is predominant, then what we have is an edge centroid tracker.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 11. *Two examples of resultant feature images:* **(a)** *aircraft image from visible TV data,* **(b)** *4-bit median filtered intensity feature,* **(c)** *3-bit edge magnitude feature,* **(d)** *32×64 IR/FPA (3-5 micron) image of truck on road,* **(e)** *4-bit intensity feature,* **(f)** *2-bit edge magnitude image, and* **(g)** *1-bit spatial frequency feature. Initially, a median filter is applied to the input imagery. The filter is a nonlinear technique that is useful for suppressing noise in imagery. With this method, the central pixel of a one-or-two dimensional window is replaced by the median of the pixels inside the window.*

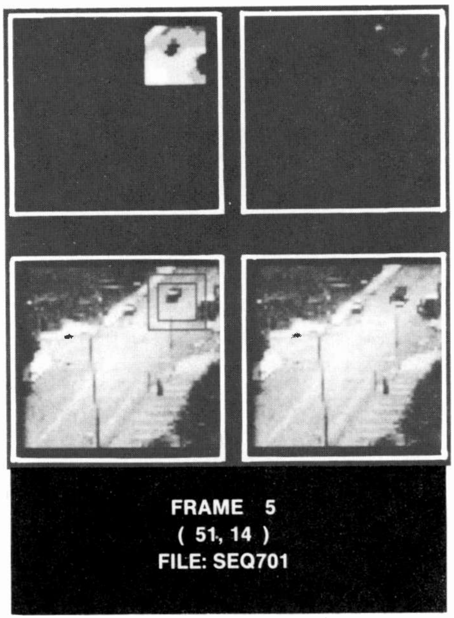


Fig. 12. *Typical tracker-simulator output.*

3. We are always operating in a mode that resembles a correlation tracker because the technique is forever searching for pixels in the current image that are statistically as close as possible to what were believed to be the target pixels in earlier frames.

Why does this combination of features work? Simply stated, there are target/background conditions which are inseparable when the two features are considered independently, but readily separable using the exact same features jointly.

How does the tracker operate in its statistical tracking mode? The preprocessed video is used to generate multiple feature images that will be subsequently used in the decision process. These features are combined into a probability density function for the target tracking window and the background window regions. These distributions are the basis of a statistical decision process that classifies the image pixels found inside the tracking window as belonging to either the target or the background.

The tracker uses a Bayesian decision rule that attempts to minimize the cost or risk associated with a particular choice, as follows:

> *A pixel is called* target *if, and only if, the penalty associated with calling it* background *is greater than the penalty of associating it with the target.*

Because we are using some number, *M*, of features, the tracker attempts to approximate an ideal bit allocation by trying to find the allocation of *N* bits among the *M* features that will minimize the sum of Bayes risk over the *M* features.

The information generated for each frame is then used for subsequent frame processing. The information controls the size and position of the window region, which acts as the source of the pixels used in the process.

One of the MFBIT's (Fig. 12) more important areas of potential growth is associated with the critical step of aimpoint selection. The statistical tracker is unable to isolate a specific aimpoint on the target that it has been attempting to track. To overcome this problem, we have been investigating the use of pattern recognition techniques—shape detection being the primary point of focus—inside the statistical tracking window.

The statistical tracker can provide an estimate of target size, thereby overcoming the primary hurdle associated with shape detection, namely size variations. Because this information is available from the MFBIT, at virtually no "cost," we are considering the use of the generalized Hough transform to perform the detection of predefined shapes within the tracking window.

The Hough transform exploits the inherent duality between the points that comprise an arbitrary curve and that same curve's parameters. The transform is an interesting and computationally efficient method that can be used in the detection of lines in images, a conceptually simple step in scene analysis.

Conclusion

Additional applications for the IPF may come from future RCA endeavors in areas such as millimeter-wave and shortwave IR sensors, robotics, bandwidth-compressed secure TV-conferencing, and the autonomous processing of weather-satellite imagery (ground or on-board).

The merit of the IPF's usefulness in such diverse applications was recently demonstrated when we used the IPF as the data source/output display unit and analysis/device controller for a Facsimile Data/Text Compression (with Error Detection and Correction) Demonstration System (FAX). In this application, the IPF was used in every phase of the system's development cycle (Fig. 14).

ATL's Image Processing Facility has consistently shown itself to be an extremely resourceful and valuable tool, and will continue to be in the future as more and more individuals learn of its capabilities.

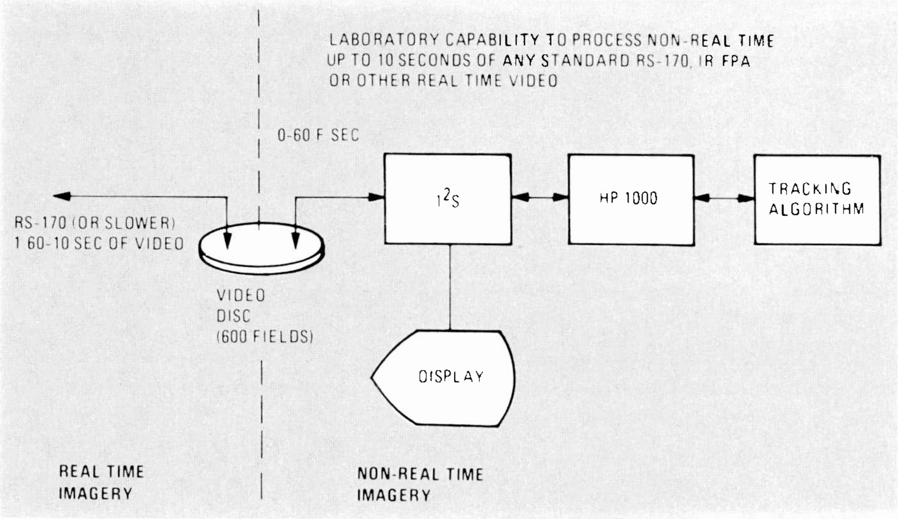


Fig. 13. *Facility for demonstrating real-time tracker performance.*

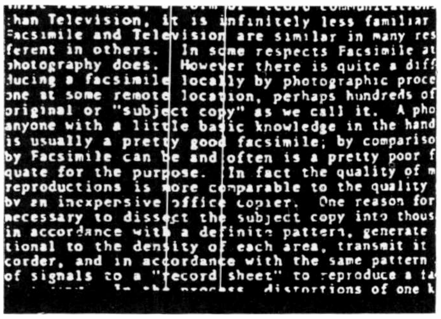
Authors Zapriala (seated) and Schaming (standing).

Bernie Schaming is Unit Manager, Image Technology Laboratory, part of the Advanced Technology Laboratories in Camden, N.J. Mr. Schaming joined the Astro-Electronics Division of RCA in 1962. In 1964 he transferred to the Advanced Technology Laboratories where he has been engaged in projects requiring computer solution and simulation. Since 1973 he has been working chiefly on image processing and he is currently responsible for RCA's development of the multifeature Bayesian statistical tracker.

Contact him at:
Advanced Technology Laboratories
Camden, N.J.
TACNET: 222-3942

John Zapriala is a Member of Engineering Staff, in the Software Technology Laboratory of the Advanced Technology Laboratories, Camden, N.J. He is currently investigating the development of an operational expert intelligence system, a computerized decision aid which will attempt to augment and reproduce the expertise of an experienced tactical intelligence analyst in performing intelligence fusion. Mr. Zapriala joined RCA in 1980 and worked primarily in the areas of pattern recognition and image processing on a shape detection technique for low-resolution IRCCD array imagery which utilized the generalized Hough Transform.

Contact him at:
Advanced Technology Laboratories
Camden, N.J.
TACNET: 222-6244



:han Television, it is infinitely less familiar
Facsimile and Television are similar in many res
ferent in others. In some respects Facsimile at
photography does. However there is quite a diff
ducing a facsimile locally by photographic proce
one at some remote location, perhaps hundreds of
original or "subject copy" as we call it. A pho
anyone with a little basic knowledge in the hand
is usually a pretty good facsimile; by compariso
by Facsimile can be and often is a pretty poor f
quate for the purpose. In fact the quality of m
reproductions is more comparable to the quality
by an inexpensive office copier. One reason for
necessary to dissect the subject copy into thous
in accordance with a definite pattern, generate
tional to the density of each area, transmit it
corder, and in accordance with the same pattern
of signals to a "record sheet" to reproduce a fa
In the process distortions of one k

Fig. 14. *Typical facsimile imagery.*

Acknowledgments

The authors would like to express their appreciation to J.J. Rudnick and O.E. Bessette for their helpful comments during the preparation of this manuscript, and to A.F. Collabelli and B. Ptashinsky for preparing the imagery used herein.

References

1. K.R. Castleman, *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, N.J., p. 3 (1979).
2. J.A. Tegnelia, "DARPA Unveils Staring Focal Plane Array," *Defense Electronics*, pp. 110, 112 (October 1980).
3. R.C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, Pa., pp. 1, 5 (1977).
4. B. Reischer, "Target Tracking Methodologies Present and Future," *Proceedings of the Open Sessions of the Workshop on Imaging Trackers and Autonomous Acquisition Applications for Missile Guidance (GACIAC-PR-80-01)*, p. 156 (November 1979).
5. W.B. Schaming, "An Adaptive Gate Multifeature Bayesian Statistical Tracker," SPIE 26th Annual International Technical Symposium and Instrument Display, San Diego, Calif. (August 1982).
6. D.J. Shazeer, "Performance Measures for Statistical Segmentation," SPIE 26th Annual International Technical Symposium and Instrument Display, San Diego, Calif. (August 1982).
7. R.O. Duda and P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, Vol. 15, No. 1, pp. 11-15 (January 1972).

Technical Excellence Committees support engineers' viability

To remain viable throughout his career, the engineer must commit to continual updating. RCA has developed the Technical Excellence Committees (TECs) as an aid in these efforts.

The first TEC started over 15 years ago and today 20 such activities have become an integral part of most of RCA's engineering groups.

The activities, tailored to the needs of the locations, cover the four charter areas shown on this page.

Your local TEC is attempting to support you, to help you develop and maintain a successful and satisfying engineering career. You can help the TEC help you by participating actively and contributing your ideas regarding its programs.

TECHNICAL EDUCATION. To provide formal and informal sources of learning of engineering subjects.

Continuing Engineering Education (CEE) planning of course offerings

Recommendations for future courses

Develop course resource list (internal and external)

Study need for courses and reasons for taking/not taking

Mini courses—by local instructors

Hands-on computer lab

Technology lectures—regular/irregular

Noontime movies/video tapes

Replay of Princeton colloquia

Workshops/brainstorming

Technical symposia

TECHNICAL INFORMATION. To provide knowledge of and access to the sources of pertinent available information.

Library committee

Library orientation

Computerized reference search

Plan for better accessibility of technical literature

Scientific bookmobile visit

Engineering forum: a social event with engineering lecture

Technical reference file

Engineering skill register

TEC newsletter

Business highlights—circulate to keep updated

Crosstalks—interdepartmental communications and reviews

RECOGNITION. To support and provide recognition of demonstrated technical excellence.

Technical excellence awards (quarterly, annually, etc.)

Author recognition (papers, presentations, patents)

David Sarnoff awards

Nominations for professional society awards

Nominations for IEEE Fellow

Recognition in TEC newsletter

Recognition in family news

Recognition in local papers

Recognition for teaching and holding office in professional societies

Recognition for educational accomplishments (CEE certificates, degrees, PE license, etc.)

PROFESSIONAL ACTIVITIES. To encourage activities supportive of engineering professionalism.

Membership in professional societies

Participation in activities of professional societies

Holding office in professional societies

Course for PE license

Obtain PE license

Encourage publication

Encourage patent disclosures

Encourage technical presentations

Support of high school science programs

Computer clubs, users groups

J. Nizet | D. Shand

The CDP 1800 family in intelligent telephone applications

Intelligent telephones that provide improved, more reliable service at home, in the office, and in the telephone booth, owe their "smarts" to the CDP1800-series CMOS microprocessor.

Abstract: *An overview of the state of development, features, programming, and future prospects for the intelligent microprocessor-driven telephone is presented. Special attention is given to the advantages of the CMOS microprocessor in this application.*

The expanding telecommunications market is demanding increased sophistication and innovation in its products. These demands are assuring that, in the future, home telephones will store previously dialed numbers and provide a catalog of prestored numbers, among other features. Both home and public telephones will dial an emergency number, for example, with the depression of any single key. The user can also expect magnetically or optically operated credit-card "coinboxes" that will deduct the cost of a call from his account.

In the office, the user will find (if he has not already) increasingly complex telephones and exchanges that offer more features at lower costs. If his curiosity is piqued, and if he investigates beyond the instrument itself, he will find the cause of this revolution in sophistication and innovation—the microprocessor. This device, being incorporated in a multitude of telecommunications equipment, is directly affecting the user through improved equipment performance and, as a result, improved service.

The microprocessor

The complementary metal oxide semiconductor (CMOS) microprocessors that are used in the intelligent telephones—the CDP1802, 1804, 1805, and 1806—are 8-bit devices with sixteen 16-bit registers, any of which can be designated as a program counter or stack pointer. Additionally, these microprocessors can address a full 64 kbyte of random-access memory (RAM) or read-only memory (ROM) with no restrictions, via a multiplexed memory-address bus and separate data bus. Figure 1 shows the telephone circuit board.

The availability of many registers is a great advantage in program design, and leads to reductions in the amount of RAM needed. Along with the designation feature, this availability also permits reductions in program length through the use of techniques not normally possible with other microprocessor architectures.

Microprocessor input/output (I/O) is on chip, and consists of one output and four inputs together with three encoded select lines for byte I/O. One interrupt input and, for fast I/O transfer, direct memory access (DMA) input and output are also available.

CMOS microprocessor superiority

The CMOS technology offers many advantages over competitive technologies, especially in the telecommunications market, and in particular in devices such as telephones, which have to be powered from the line.

Paralleling the growth of the microprocessor has been the increasing price of energy, and with it the corresponding pressure by telecommunications authorities and companies to reduce the power consumption of telecommunications systems, including elements such as telephones, coinboxes, and exchange equipment. The properties of the CMOS microprocessor, specifically its low power consumption, make it nearly the perfect match for the required energy limitations. A typical CMOS microprocessor system consumes one-tenth the power of systems designed from alternative technologies, such as NMOS. Not only does

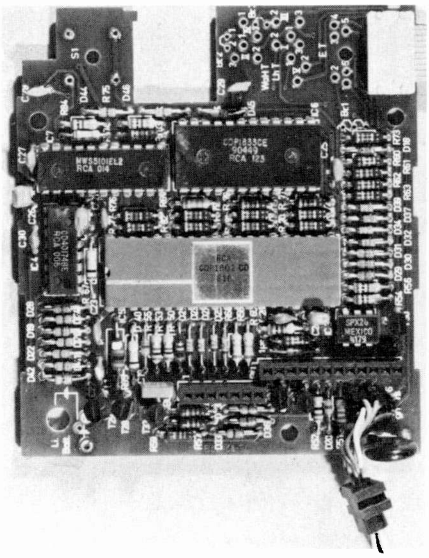


Fig. 1. *An intelligent-telephone circuit board. This board contains all the components shown in the block diagram of Fig. 2. The central unit on the board, in terms of providing the system intelligence, is the CDP1800-series microprocessor.*

Final manuscript received October 13, 1982.
Reprint RE-28-1-11

this mean that a telephone system designed around a CMOS microprocessor operates with very low current, but also that the system, in consuming less power, is cheaper to run and, very importantly, dissipates little heat, therefore requiring no complex or expensive cooling equipment. Without the mechancal cooling fans, the CMOS-based system can operate more reliably than alternative systems.

The reduced power consumption of CMOS devices results in the microprocessor's wider operating temperature range (most CMOS LSI devices are available in versions that operate over a temperature range of −40°C to +85°C), and extends its use to outdoor applications. Yet another important advantage of the CMOS microprocessor in telecommunications applications is its ability to operate from a varying voltage, and to make the system less susceptible to external electrical noise. For this reason, a microprocessor-based telephone system can operate normally nearby an electric motor.

Intelligent telephone features

Figure 2 shows a block diagram of an intelligent telephone system. The features of this system, made possible through the use of the CDP1802 microprocessor, are described below. Several systems in production now have the following features:

- Normal telephone functions
- Pulse dialing at the correct rate
- Redial of last-dialed number in memory
- Repertory dialing functions (12 or 20 numbers)
- Optional display with time-of-day clock
- Display of the cost of a phone call
- Display of total cost of all phone calls from that telephone during a billing period
- Display indicating busy phones, when the instrument is part of an internal subset
- Communications with a central office
- Security number; a number dialed out when any single key is depressed.

The special features of the pay or coinbox phones are:

- Coin discriminator
- Tariff and teletaxes computation
- Warning-tone generation
- Free-call control
- Dialing with operator control
- Communication with a central office.

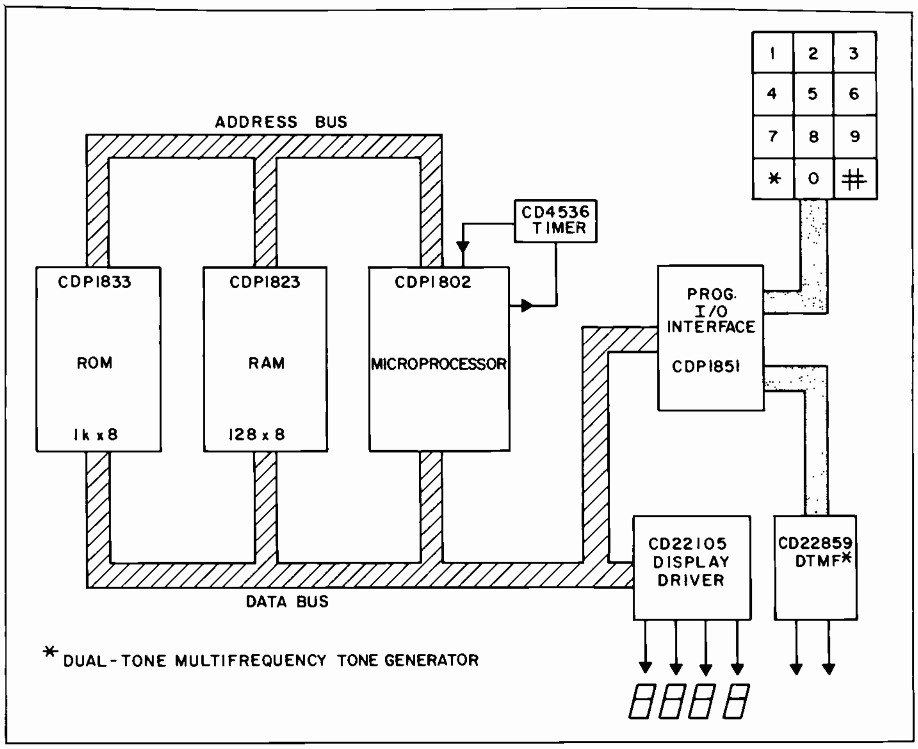


Fig. 2. *The typical intelligent telephone system. Note the input keyboard, microprocessor, memory, I/O, and display elements, and data transmission device.*

The special features

Coin discriminator. Until recently, coin discrimination was handled by a mechanical coin filter, whose main problem was blockage of the channel along which the coins move—a blocked channel meant a phone out of order. In modern systems, the coins move past optical and electromagnetic sensors that detect the size and magnetic characteristics of each coin. On the basis of inputs from the sensors, a microprocessor determines whether the coin is valid or not. The electromagnetic portion of this sensor/microprocessor system involves analog-to-digital (A/D) conversion; however the system as a whole is economical and is essentially immune to coin blockage.

A major advantage to the pay-phone manufacturer of the coins discriminator is its adaptability to varying coin characteristics. Since coin parameters are checked by the microprocessor against those on a small erasable programmable read-only memory (EPROM) in the instrument, the manufacturer can set up the system to handle coins of any country, or new coins, simply by appropriately programming, or replacing, this EPROM.

Teletaxes. The telephone company determines the price of a call by counting the number of time intervals called "periods" that pass during the call. The tariff per period is fixed, but period length varies. For example, a given amount of money might buy three minutes of a regional call, but only a few seconds of an intercontinental call. The length of a given period is defined by the generation of a 12- or 16-kHz signal, called a "teletax pulse," on the line. To determine the cost of a call, the telephone electronics has only to count the number of pulses issued during the call, and to multiply this number by the tariff. Each time a teletax pulse is recognized, the central processing unit (CPU) decreases the credit (originally determined by the coin discriminator) by the amount of tariff. The remaining credit is displayed on an LCD (liquid-crystal display). When the credit becomes zero, the phone sounds a warning tone; if no more coins are put in, the phone switches off the line after 10 seconds.

At present the telephone company writes off the 10 seconds per customer. However, in the new generation of microprocessor-controlled coinbox or pay phones, the CPU calculates the rate at which the credit is decreasing, anticipates by 10 seconds the time when it will be zero, and sends the warning tone 10 seconds before that time.

Free dialing. The intelligent pay phone retains a list of numbers in memory that

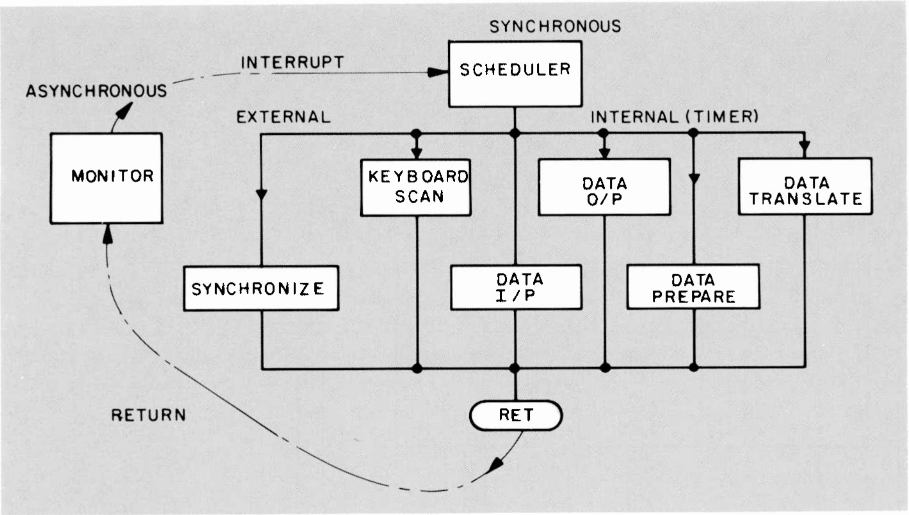


Fig. 3. *A typical intelligent-telephone software system. The system software determines when the phone is being operated, and carries out the instructions input by the user as he dials a number.*

may be called at no charge (police, fire, and so on). The CPU checks a dialed number to see if it is on the no-charge list, and if so, makes the connection without cost to the customer.

Operator. When a customer inserts coins while talking to an operator, the CPU will generate a frequency on the line that will identify to the operator the amount of money put in.

Security. Callers attempting to use a phone without paying will be thwarted by a system watchdog, a monostable flip-flop that must be reset at a fixed rate. If not reset in the required time interval, because something interferes with the normal program flow, the watchdog flip-flop will generate a reset of the total system on the CPU and so terminate the call in process.

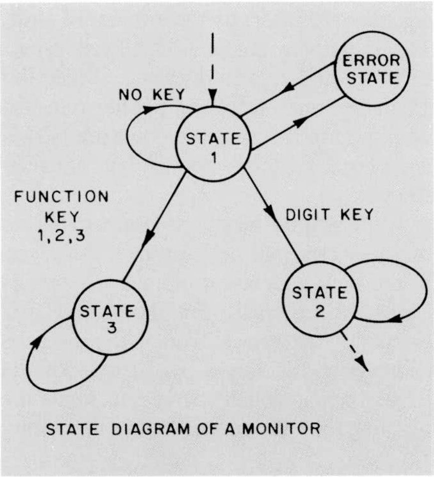


Fig. 4. *The asynchronous or monitor routine uses the state-diagram technique.*

Communications with a central office. Some systems have one communication line dedicated to the central office, most do not. In the latter case, when communications are necessary, they are handled by means of dummy teletaxes from the central office to the pay phone and with dummy dialing pulses from the telephone to the central office. In one system now in service, this central office is driven by another CDP1802 microprocessor.

Telephone programming

The program used with the intelligent telephones is made up of two major routines, the synchronous (interrupt) routine and the asynchronous (monitor) routine, Fig. 3.

The *synchronous* routine is interrupt driven. Interrupts are received at a fixed rate (for example, every 16 milliseconds). The CPU enters interrupt and scans the keyboard, refreshes the display, tests the communication line, and looks for a teletax or for a coin insertion, Fig. 4. In other words, the synchronous routine schedules time-dependent tasks.

All of these different tasks can be done at each interrupt or divided among interrupts. For example, the CPU can do a display refreshment plus a keyboard scan at one interrupt, and at the next do a teletax test and a second keyboard scan. When an event is detected and debounced, the CPU sets a flag and stores the event code at a RAM address or in a register.

To understand the *asynchronous* or monitor routine, consider a telephone in the 0 state: receiver down, with no key pressed. If the receiver is lifted, the system jumps from the 0 state to the 1 state (hook lifted, no key pressed). If a key is pressed, the system goes from state 1 to state 2, but if no key is pressed, it will perform the task of state 1, then will stay in that state awaiting an event to cause it to jump to another state. In short, the monitor waits for an event, determines which state it must jump to, jumps into the new state, does the task of that new state, and waits for another event. It is the synchronous (interrupt) routine that detects this event and signals the monitor.

Systems in service

The versatility of the CDP1800 family has sold a number of telecommunication companies in Europe on the CMOS microprocessor-controlled telephone design. The several systems now in production or design are given in Table I.

Table I. *Systems in Europe have CMOS microprocessor-controlled designs.*

Pay phone

1 company in Switzerland
1 company in Belgium
1 company in Spain
2 companies in U.K.
1 company in Sweden
1 company in Germany
1 company in Austria

Card phone

1 company in Switzerland
1 company in France
1 company in Belgium

Intelligent telephone

4 companies in Germany

Line concentrator

1 company in France
1 company in Switzerland
1 company in Austria

Military telephone

1 company in U.K.

Small PABX (exchange)

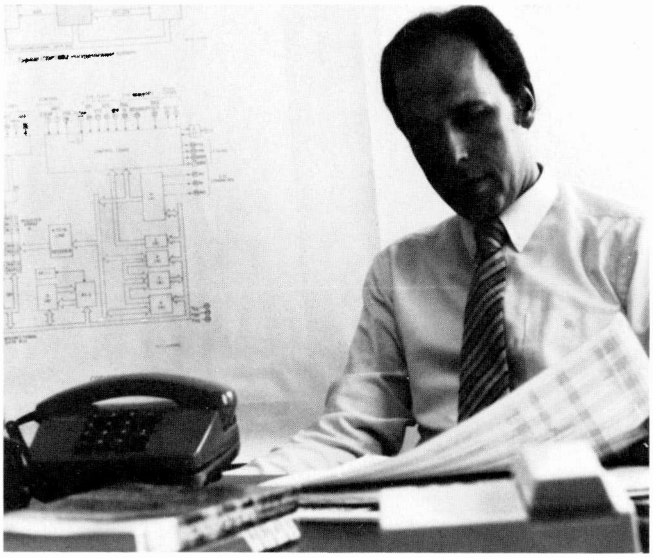
1 company in France
1 company in Finland
1 company in Germany

Intelligent modem

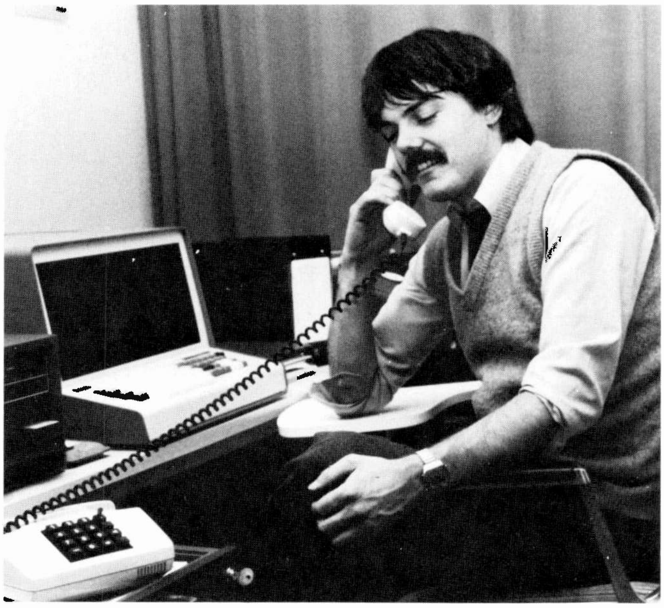
1 company in Germany

Transaction terminal (through telephone)

1 company in U.K.



Jean Nizet is responsible for semi-custom IC design (cell library system) at the European Design Automation Center in Brussels. He joined RCA in 1973 and worked until 1978 in the design and maintenance of computer-based systems for testing power transistors at RCA's Liege, Belgium facility. From 1978 until 1981 he was responsible for microprocessor design in European Applications Engineering in Brussels.
Contact him at:
RCA Solid State
European Design Automation Center
Brussels, Belgium



David Shand joined RCA Quickborn, Germany, in 1979, and has since then been a Microprocessor Field Application Engineer in the German Region. His experience with RCA includes the design of intelligent telephones for public and PABX systems for OEM customers. He graduated B.Sc. (Hons) in Physics from Aberdeen University and completed the M.Sc. course in Digital Techniques at Heriot-Watt University in Edinburgh, Scotland.
Contact him at:
RCA Solid State
Quickborn, Germany

Conclusions

All of the phones described in this paper have one common feature—they are all designed around a general type of microprocessor, that is, a device designed for use in a broad range of applications. The current trend in the continuing effort to bring more features at lower cost to the telephone user, as well as design economy, flexibility, and expansibility to the manufacturer, is toward the development of a telephone-oriented version of a standard microcomputer. This device, already on the drawing boards at RCA and a number of other companies, will optimize telephone design by including on-chip such features as a keyboard interface with interrupt possibility in a low-power mode, a serial bus to interface to components such as serial random access memory, liquid crystal display driver, tone dialers, and real-time clocks, and a 16-bit multifunction timer to simultaneously generate a time base and measure pulses like the teletax pulse. The primary feature of the microcomputer will be, however, the CMOS technology, which offers low power consumption and low operating voltage—immense benefits in telephone design.

Technological advancements in the CMOS area will also help to minimize manufacturing and operational costs of coinbox telephones. New design automation techniques are now readily available for fast generation of low-cost custom ICs (so-called "semicustom ICs"). ICs built by these techniques will be used to "swallow up" the discrete circuitry now external to the central microprocessor, and will, along with advancements in high-density VLSI CMOS ROMs, help to achieve the minimum-cost goal. The advent of low-cost per-bit CMOS ROMs is already increasing the use of higher-level languages in coinbox telephones and further reducing development cycle time, costs, and maintenance.

The general public, if they are using advances in telephone design as a barometer of technological advancement, may not be surprised to find that the coinbox itself may soon give way to progress. Field trials are underway to test the feasibility of cardphones. Cardphones are public phones that are operated by either magnetic or holographic credit cards. If technological progress is measured and somehow justified by its benefit to the public, then the intelligent telephone must be one of the best examples of that progress.

R. Williams | R.W. Nosker

Adherent dust particles

Dust sticks to everything and everyone, and critically affects stringent requirements for manufacturing today's sensitive electronic products.

Abstract: *Dust particles are an important factor in the manufacture and use of electronic products. Strongly adherent particles are especially troublesome. We have investigated the mechanism of dust adhesion and found that it often involves water-soluble materials. Excursions of the relative humidity above and below a critical value can trigger a cementing effect that we believe leads to most of the damage caused by dust particles.*

Dust is such an ordinary part of daily life that we rarely notice it. Occasionally, a sunbeam at just the right angle or the sight of a laser beam in a darkened room reminds us of the multitude of particles that are always there in the air around us. Many of these particles are so small that they remain suspended in the air for days, but a single one of them, landing on an electronic device during manufacture, can cause its malfunction. Modern electronics—exemplified by marvelously ingenious integrated circuits, kinescopes, and the VideoDisc—has come to depend more and more on large arrays of very small elements. Malfunction of one or a small number of these elements can render a whole system useless. Large systems can also be damaged by airborne particles. Outdoor solar-cell arrays many square meters in area become coated with dust that obscures incident light with a serious loss of efficiency.[1]

Though the problem of dealing effectively with particles in manufacturing is formidable, there are some general principles that apply. These help us to understand what is happening and may help us to control the important conditions. Not all particles are the same. Some stick to surfaces more strongly than others, and these cause most of the trouble. We have investigated the nature of dust particles and the physical processes that lead to strong particle adhesion on surfaces. In what follows we will describe some of the things we have learned.

The role of electrostatic charge

Many dust particles are charged. They are attracted to neutral surfaces by electrostatic image forces. Conversely, the surface of an insulator can acquire an electrostatic charge and attract neutral dust particles to it by the same mechanism. Either way, more particles land on a surface when electrostatic charges are acting. In this sense electrostatic effects aggravate the dust problem. However, charged particles that reach a surface are not very much more firmly bound than neutral particles. By measurements that we will describe below, we have found an effect, due to water soluble salt crystals, that causes some particles to be

Final manuscript received October 8 1982
Reprint RE-28-1-12

very much more firmly bound than others. To get an overall perspective, we first describe the particles normally found in the atmosphere.

The properties of atmospheric dust particles

> *Time which antiquates antiquities, hath an art to make dust of all things.*[2]
>
> —Sir Thomas Browne

Atmospheric particles range from 0.002 to 100 μm in diameter. Particles at the lower limit are generally unstable, since small solid particles coagulate to form larger particles, and small liquid droplets tend to evaporate rapidly. Particles at the larger limit sediment quickly under the influence of gravity. Solid particles are usually irregular in shape and often consist of aggregates. In general, small particles greatly outnumber large ones. Figure 1 shows typical particle-size distributions that have been measured in ordinary rooms and in various clean-room environments.[3] Within RCA we carry out many processes in Class 100 installations. These provide an environment in which the number of particles per cubic foot that have diameters greater than 0.5 μm is 100 or less.[4]

The rate at which a particle falls in quiet air is a strong function of its diameter. Table I shows the time required for particles of various sizes to fall one foot. A 3.0-μm particle, for example, falls 0.006 ft/s. A moving stream of air can carry such particles along for great distances.

There are many sources of airborne particles. Each source contributes its own type of material with a characteristic particle-size distribution. Table II shows examples. It has been estimated that a total of about 4.3 × 10^7 tons of particles fall on the U.S. each year. About 72 percent of the material derives from natural

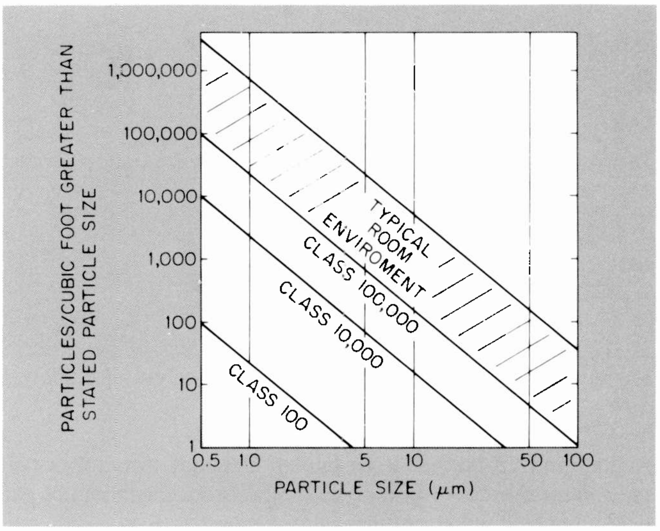
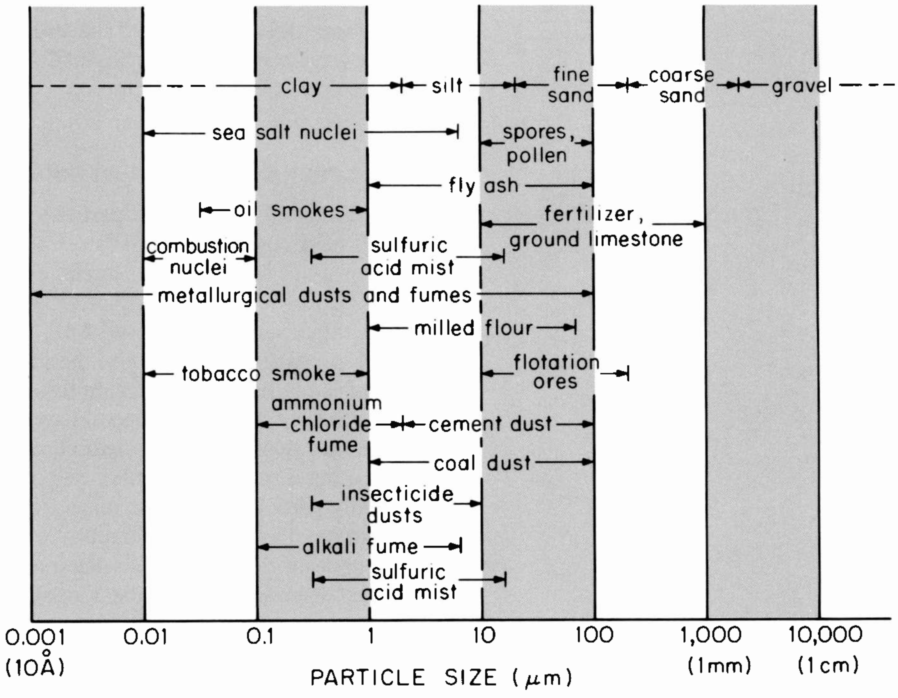


Fig. 1. Approximate distribution of particle sizes in a typical room atmosphere, and in three classes of filtered, clean environments as defined by Federal Standard Number 209a.[3]

Table I. Time needed for a particle to fall one foot in still air.

Size of particle in μm	*Time needed to fall 1 ft*
0.25	10 hr
0.5	3 hr
1	1 hr
2	15 min
3	3 min

Table II. Approximate size distribution of a variety of airborne particulate contaminants. Data from *Chemical Engineer's Handbook*, 5th Edition, Ed. R.H. Perry and C.H. Chilton, McGraw-Hill Book Company, New York, pp. 20-79.



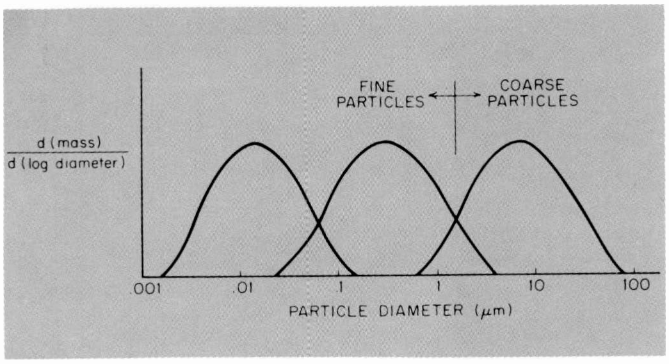


Fig. 2. Typical mass distribution of particles as a function of size.

sources and 28 percent from human activities. Around heavily industrialized areas or primary natural sources, the fallout of particles may be hundreds of times the average.

The largest natural sources are blowing soil, ocean spray, forest fires, and volcanic activity. Blowing soil is most important in arid regions, such as the U.S. Southwest, that have little natural vegetation, although it is a very important source everywhere. Ocean salt spray dries in the air, leaving behind salt crystals that make up a large source of dust particles, worldwide. Concentrations are high near the seashore but, in some areas such as California, onshore winds carry particles inland for hundreds of miles. Volcanic activity is usually a minor source, but an occasional major eruption can produce a dust cloud that encircles the earth and produces vast amounts of sedimentation for a year or more. Forest fires account for about 7 percent of the total. Plant pollen and spores contribute about 3 percent, and, at certain times of the year, may be the dominant particle source. Table III shows the sources of particulate material due to human activity.

Table III. Breakdown of airborne particulate contamination due to human activity (from Reference 5).

Activity	*Percent of total mass*
Transportation	5.6
Fuel combustion in stationary sources	54.5
coal (incl. coal refuse burning)	(39.8)
fuel oil	(1.3)
natural gas	(0.9)
wood & plant matter (incl. structural fires)	(12.5)
Industrial processes	39.9
iron & steel	(9.3)
other metals	(0.8)
cement	(4.1)
stone, sand, rock, etc.	(4.1)
coal cleaning	(0.9)
phosphate rock	(0.9)
lime	(2.1)
asphalt batching	(2.5)
other mineral products	(0.8)
oil refineries	(0.5)
other chemical industries	(0.4)
grain handling & storage	(3.7)
pulp & paper	(3.3)
flour & feed milling	(1.4)
solid-waste disposal	(5.1)
Total	*100.0*

Table III does not include occasional human sources, such as salt crystals of sodium chloride and calcium chloride. These are put on highways to melt snow and ice. The resulting solutions are later sprayed into the air by moving traffic and end up as atmospheric particles.

Recent studies[6] show that there can be three distinct peaks in the size distribution of atmospheric particles, as shown in Fig. 2. Coarse particles have a mean diameter of about 9 μm and they derive from fly ash, mechanical and industrial processes, larger sea-spray particles, wind-blown dust, volcanoes, and plant particles. They leave the atmosphere primarily by sedimentation. There is also, typically, a peak in the distribution for diameters around 0.02 μm. These particles are the primary nuclei from combustion and have enormous concentrations near combustion sources. With time they grow by condensation and coagulation to a mean diameter of about 0.3 μm. Smaller sea-spray particles and wind-blown dust also contribute to the 0.3-μm peak. Many of these fine particles are at least partly liquid. They are removed from the atmosphere primarily by rain. On a mass basis, there is relatively little coagulation of fine particles with coarse particles.

The above description applies to the particles found in an outdoor environment. Indoors, there are additional things to consider. First of all, how do particles make their way from the outside to the inside? Particles will enter via an air exchange if the air speed at the points of entry is high enough (Table I) to keep the particles in suspension. This is especially important during the heating season when there is a high rate of air exchange. Many particles are carried in by people. We inevitably accumulate and redistribute particulate material. In addition to this transfer process, particles are also generated indoors in great abundance. Home cooking exhausts, mists, smoke, and industrial fumes, along with the inevitable human emissions, all contribute.

It is hard to overemphasize the role that people play as purveyors of dust. A person moving about transfers prodigious quantities of dust onto clothing by sitting, leaning and touching, then flings the particles hither and yon with sudden movements. In the home, particle counts increase by factors of ten to twenty when people are moving about. In the lower background of the clean room these effects stand out even more. The use of clean room overgarments is only marginally effective. Too little attention is paid to this human factor.

Experimental study of dust adhesion

We have found that the dust particles that settle on a surface are not all held equally strongly. Most are bound only by electrostatic or weak intermolecular forces, and can be easily moved by a jet of air or a probe. A few of the particles stick very tightly.[7] Often, they cannot be removed from the surface without breaking off the particle or a piece of the substrate. Clearly a different mechanism of binding is acting in this case.

To measure adhesion properties we applied known quantities of standard dust samples by sedimenting the dust onto test surfaces, using a suitably dimensioned box with a timed shutter, so that only particles in the size range from 1 to 30 μm landed on the surface. Particle concentrations were determined by counts done under a microscope. To determine the degree of adhesion we used a modification of the VideoDisc system as a tool. The stylus can be used to scan a large area systematically so that it encounters nearly every particle on the surface. Immediately after dusting the surface, and for any time thereafter, as long as the relative humidity in the room had not risen above 80 percent, the stylus traversed the grooves of the disc and pushed nearly all particles out of its way without incident. Almost none of the particles adhered strongly to the surface.

However, if the dusted surface was exposed to a humidity cycle in which the relative humidity rose above 80 percent for an hour or more and then returned to a lower value, the behavior of the particles was quite different. Some particles then adhered strongly and were literally cemented to the surface. When the stylus hit one of these particles, it was kicked out of its groove. By using a suitable circuit, these events were recorded and counted. In this way, we found that, after the humidity cycle, about 1 percent of the particles had become strongly adherent. Microscopic examination of the dusted surface at this point showed that some particles were surrounded by a stain. These were usually adherent particles. We had directly measured the force needed to move these particles. It was orders of magnitude higher than the force needed to move nonadherent particles and was often roughly equal to the mechanical yield strength of the substrate material. The particle had been glued to the surface by the action of the humidity cycle.

These results suggested to us that the adhesive effect triggered by the humidity cycle is due to the presence of water-soluble particles in the dust. These may be present alone or together with another particle as part of a composite particle. Since water-soluble particles make up only a small fraction of all dust particles, this would explain why only a few particles become strongly adherent after the humidity cycle. In a humid atmosphere, soluble particles adsorb water on the surface until part of the crystal dissolves. This process is called deliquescence and we will describe it more fully below.

We confirmed the idea that soluble salt particles are responsible for most of the strong dust adhesion by washing the dust in water before using it for the adhesion test. The dust was washed in pure water, then dried and used as in other tests. When this was done, we found almost no strongly adherent particles. This result confirms that water-soluble material present in natural dust is responsible for the strong dust adhesion.

Deliquescence

At high relative humidities the atmosphere has a high moisture content. When the relative humidity exceeds a certain critical value, the deliquescent salt takes up moisture to form a solution. The solution continues to absorb moisture until its vapor pressure is equal to that of the water vapor in the atmosphere. If the relative humidity now falls below the critical value, the solution will lose water until the salt crystallizes again. The critical value of relative humidity varies from one material to another. The

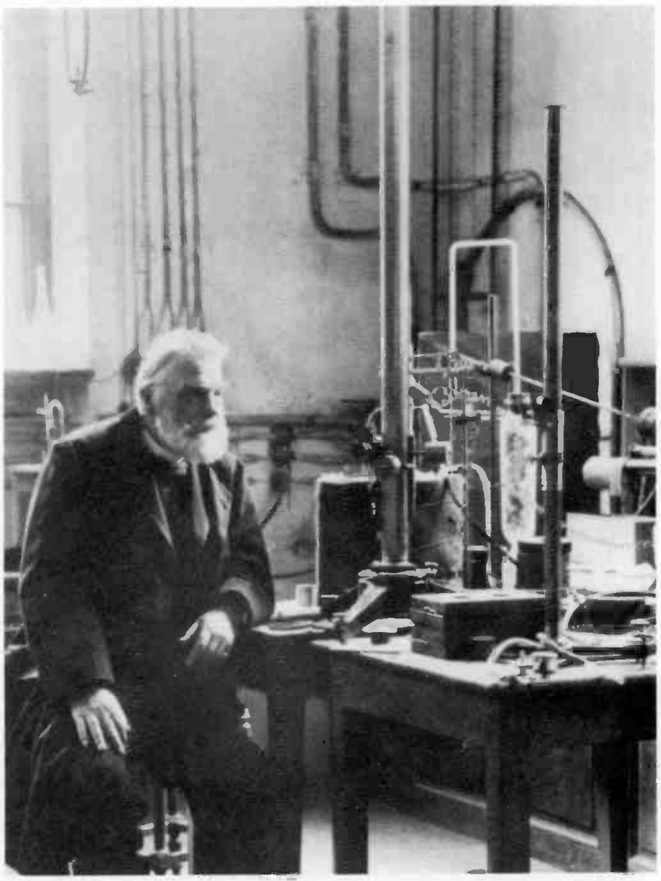


Fig. 3. François Marie Raoult (1830-1901) in his laboratory at the University of Grenoble, France. His theory of solutions, known as Raoult's Law, describes how the vapor pressure and other properties of a solvent change with the amount of dissolved material. It explains a wide variety of everyday phenomena, ranging from deliquescence to the behavior of antifreeze solutions. (Photograph provided by the Edgar Fahs Smith Collection, University of Pennsylvania.)

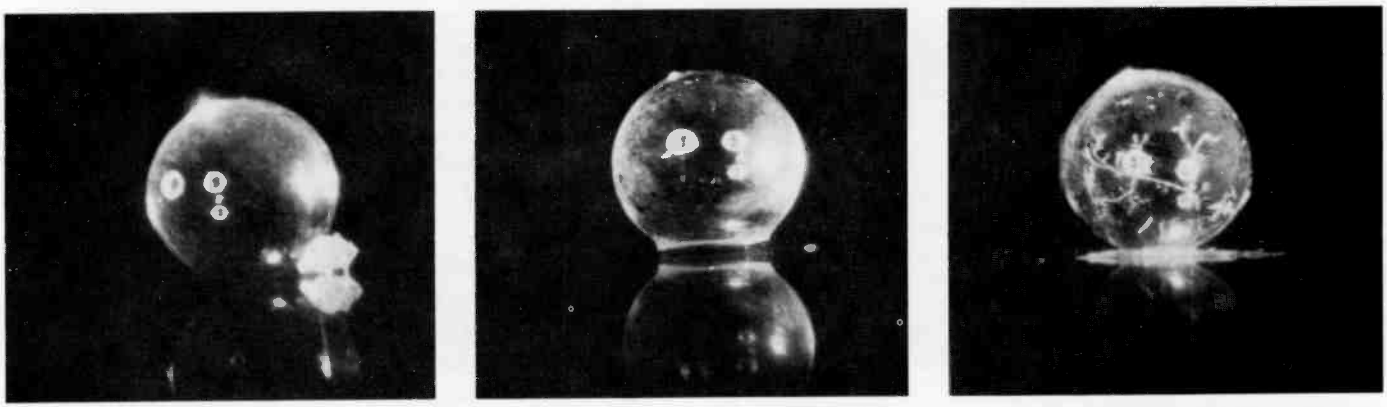


Fig. 4. A glass ball, 2.5 mm in diameter, rests on a glass substrate. A sodium carbonate particle, mass 300 times less than the ball, lies next to the ball. In **(a)**, the humidity is below 78 percent and the particle is dry. In **(b)**, the humidity has been raised to 90 percent and the sodium carbonate is in water solution around the ball. In **(c)**, the humidity has dropped below 78 percent and the sodium carbonate has redeposited around the ball, gluing it to the substrate. The force required to move the ball in **(c)** is 10^5 times greater than in **(a)**.

Table IV. Critical values of relative humidity for some water-soluble salts that are found in the atmosphere. The critical humidity is defined in the text.

Soluble salt	*Critical relative humidity* (%)
LiCl	15
$CaCl_2$	32
$MgCl_2$	33
K_2CO_3	44
NaCl	76
Na_2CO_3	78
NH_4Cl	79
Na_2SO_4	83
KCl	86
K_2SO_4	97

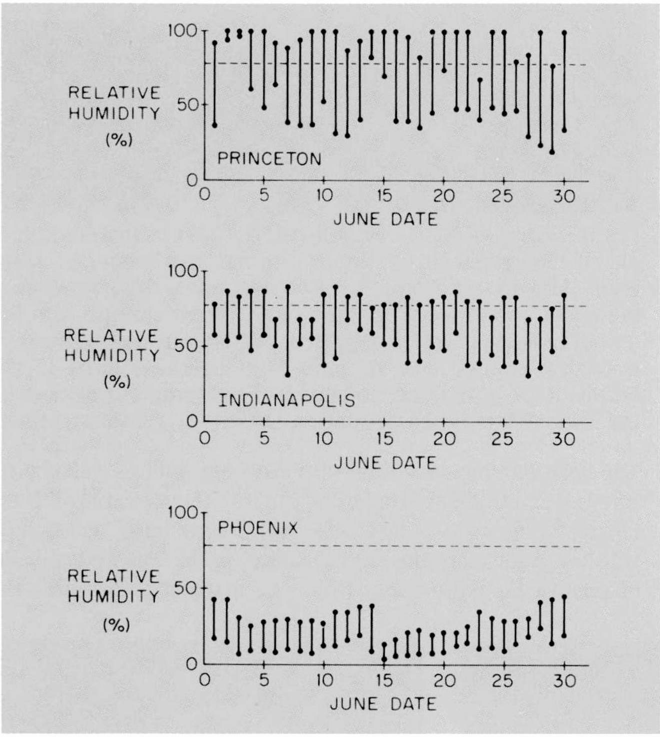


Fig. 5. High and low values of relative humidity for each day of June, 1981, for three locations: Princeton, New Jersey; Indianapolis, Indiana; and Phoenix, Arizona. The horizontal line at 76 percent is the critical relative humidity for deliquescence of sodium chloride. Princeton data were supplied by D.A. Kramer. The other data were supplied by the U.S. Department of Commerce. Such data can be obtained for any station for which data are recorded by calling Asheville, North Carolina, (704) 258-2850, extension 683, and requesting a Local Climatological Data Form.

Fig. 6. Suppliers of table salt have found that the clogging of salt shakers, due to changes in relative humidity as described in the text, can be ameliorated by adding 0.5 percent of sodium aluminum silicate. This inert material coats the particles and reduces adherence. (Morton Salt trademark used with permission of Morton Salt Division of MortonNorwich.)

more soluble the salt the lower the critical value (Fig. 3). In Table IV we list the critical values of relative humidity for several soluble salts that are found as particles in the atmosphere.

In Fig. 4 we show how deliquescence can cement a particle to a surface. In this case, we have used a small glass sphere on a horizontal surface. The soluble salt is a particle of sodium carbonate, shown in Fig. 4a, touching both the sphere and the surface. Sodium carbonate particles are often found in the atmosphere and are emitted by the cooling stacks of power stations and industrial plants. Figure 4b and 4c show how, when the humidity cycles above and below the critical value of 78 percent, the salt dissolves, wets the sphere and the support surface, makes intimate contact, and then recrystallizes to form solid-solid bonds. This cements the sphere to the surface. At this stage, the force needed to dislodge the sphere from the surface is 27 grams, or 10^5 times the force needed to move it at the original stage shown in Fig. 4a. This sequence of flow and wetting, followed by solidification, is the normal way that an adhesive works. When the environmental conditions are right, any water-soluble salt can act as an adhesive.

The natural cycling of the relative humidity

Figure 5 shows high and low values of the outdoor relative humidity for each day in June of 1981 for three different locations: Princeton, New Jersey; Indianapolis, Indiana; and Phoenix, Arizona. As a reference point, the critical relative humidity value for the deliquescence of sodium chloride, 76 percent, is drawn in each graph as a horizontal line. The humidity cycle that we have described above takes place on those days that have one point above the critical line and one below it. In Princeton, the cycle took place on 25 days out of 30 in this month, and in Indianapolis it took place on 19 days out of 30. In Phoenix it did not take place at all during this month.

Indoors, the humidity does not normally swing through as wide a cycle as it does outdoors. In one of the above locations, a well air-conditioned room might have a mean relative humidity of 50 percent with variations up and down of 10 to 15 percent. However, concern about energy usage has led to less air-conditioning. In June of 1981, for example, the indoor relative humidity in RCA Laboratories averaged about 70 percent. There were many days when the relative humidity went through the excursion that gives rise to sodium chloride particle adhesion. In winter the average relative humidity indoors will run much lower but there may still be excursions around the critical value for one of the materials in Table IV. When this happens, there can be adhesion due to soluble particles.

In addition to its effect on dust particle adhesion, the deliquescence effect also gives rise to a common irritant in daily life—the clogging of salt shakers. Excursions of relative humidity lead to solution and recrystallization of minute quantities of salt, causing the grains to stick to one another. Figure 6 and its caption show how one company has solved this problem and given us one of our most enduring advertising slogans—"When it rains, it pours."

Acknowledgment

We are indebted to G.R. Auth and L.A. DiMarco for experimental help and to D.A. Kramer for supplying the relative humidity data for Princeton.

References

1. A.R. Hoffman and C.R. Macy, "Photovoltaic Soiling Studies," *DOE/JPL-1012-49* G.P.O. Washington, DC, p. 15 (1980).
2. Sir Thomas Browne, in *Bartlett's Familiar Quotations.*
3. Federal Standard 209a, available from General Services Administration, Specifications Activity, Printed Materials Supply Division, Bldg. 197, Naval Weapons Plant, Washington, DC 20407.
4. E.C. Douglas, "Advanced Process Technology at the Solid State Technology Center," *RCA Engineer,* Vol. 26, No. 2, pp. 8-17 (September/October 1980).
5. J.C. Giddings, *Chemistry, Man, and Environmental Change,* Harper & Row, New York, N.Y., p. 243 (1973).
6. K.T. Whitby and B. Cantrell, in *Int'l. Conf. on Environmental Sensing and Assessment,* IEEE, Inc., New York, N.Y., p. 1 (1976).
7. R.W. Nosker, L.A. DiMarco, and R. Williams, *Applied Physics Letters,* Vol. 38, p. 1023 (1981).

Authors Williams (left) and Nosker.

Richard Nosker joined RCA in Camden as a freshman co-op in 1961. He moved to RCA Laboratories in 1964, and received the BSEE from the University of Cincinnati in 1965. He received the PhD from Princeton University in EE (solid state) in 1970, with the support of a David Sarnoff Fellowship. Since 1966, he has worked on VideoDisc. For nearly a decade, he has concentrated on the mechanics and chemistry of the stylus-disc interface. He has received three RCA Laboratories Outstanding Achievement Awards, and is a member of the American Physical Society.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2807

Richard Williams received the AB in Chemistry from Miami University and his PhD in Physical Chemistry from Harvard University. He joined RCA Laboratories in Princeton, New Jersey, in 1958, and is now a Fellow of the Laboratories. Dr. Williams has worked in the fields of liquid crystals, semiconductor-electrolyte interfaces, internal photoemission, solar cells, properties of electrons on the surface of liquid helium, and the surface properties of the VideoDisc.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2762

Somewhere in the world another engineer has done something you ought to know about . . . Your RCA Technical Librarian will assist you in searching any of these online databases, including RCA's own proprietary database.

ABI/INFORM (Business management and administration) □ ADTRACK (Index of advertisements) □ AGRICOLA (National Agricultural Library) □ AIM/ARM (The Center for Vocational Education) □ ALCOHOL USE/ABUSE (Hazelden Foundation) □ AMERICA HISTORY AND LIFE (U.S. and Canadian history) □ **AMERICAN MEN AND WOMEN OF SCIENCE (R.R. Bowker)** □ APTIC (U.S. Environmental Agency) □ AQUACULTURE (National Oceanic and Atmospheric Administration) □ AQUALINE (Water Research Centre, England) □ AQUATIC SCIENCE AND FISHERIES ABSTRACTS □ ARTBIBLIOGRAPHIES MODERN (Modern art and design literature) □ ASI (Congressional Information Service) □ **BHRA FLUID ENGINEERING (British Hydromechanics Research Association)** □ BI/DATA FORECASTS (Market forecasts for 35 countries) □ BI/DATA TIME SERIES (Economic indicators for 131 countries) □ BILINGUAL EDUCATION BIBLIOGRAPHIC ABSTRACTS □ BIOGRAPHY MASTER INDEX (Gale Research Company) □ BIOSIS PREVIEWS (Biosciences Information Service) □ BLS CONSUMER PRICE INDEX (Bureau of Labor Statistics) □ BLS EMPLOYMENT, HOURS, AND EARNINGS (Bureau of Labor Statistics) □ BLS LABOR FORCE (Bureau of Labor Statistics) □ BLS PRODUCER PRICE INDEX (Bureau of Labor Statistics) □ BOOK REVIEW INDEX (Gale Research Company) □ BOOKS IN PRINT (R.R. Bowker) □ BOOKSINFO (Brodart, Inc.) □ **CA SEARCH (Chemical Abstracts Service)** □ CAB ABSTRACTS (The Commonwealth Agricultural Bureaux, England) □ CALIFORNIA UNION LIST OF PERIODICALS □ CAREER PLACEMENT REGISTRY/EXPERIENCED PERSONNEL □ CAREER PLACEMENT REGISTRY/STUDENT □ CATFAX DIRECTORY OF MAIL ORDER CATALOGS □ **CHEMICAL INDUSTRY NOTES (American Chemical Society)** □ **CHEMICAL REGULATIONS AND GUIDELINES SYSTEM** □ **CHEMNAME (CA Chemical Name Dictionary)** □ **CHEMSEARCH (Chemical substances)** □ **CHEMSIS (Chemical substances)** □ CHILD ABUSE AND NEGLECT (National Center for Child Abuse and Neglect) □ CIS (Congressional Information Service) □ CLAIMS/CHEM (U.S. chemical patents) □ CLAIMS/CITATION (Cited patents) □ CLAIMS/CLASS (U.S. Patent Classification System) □ CLAIMS/U.S. PATENT ABSTRACTS □ CLAIMS/U.S. PATENT ABSTRACTS WEEKLY □ CLAIMS/U.S. PATENTS □ CLAIMS/UNITERM (Chemical patents) □ COFFEELINE (International Coffee Organization) □ **COMPENDEX (Engineering Index, Inc.)** □ COMPREHENSIVE DISSERTATION INDEX (Xerox University Microfilms) □ **CONFERENCE PAPERS INDEX (Data Courier, Inc.)** □ CONGRESSIONAL RECORD ABSTRACTS (Capitol Services International) □ CRIMINAL JUSTICE PERIODICAL INDEX (Justice and law enforcement periodicals) □ CRIS/USDA (USDA Cooperative State Research Service) □ **DISC (Personal computing)** □ DISCLOSURE II (U.S. Securities and Exchange Commission) □ DOE ENERGY (U.S. Department of Energy) □ **DOW JONES NEWS (Dow Jones and Company, Inc.)** □ DRUGINFO (University of Minnesota College of Pharmacy) □ DUNS MARKET IDENTIFIERS (Dun's Marketing Services) □ ECONOMICS ABSTRACTS INTERNATIONAL (Learned Information, Ltd., England) □ EIS DIGESTS OF ENVIRONMENTAL IMPACT STATEMENTS □ EIS INDUSTRIAL PLANTS (U.S. industrial economy) □ EIS NONMANUFACTURING ESTABLISHMENTS □ ELECTRIC POWER DATABASE (R&D projects) □ ELECTRONIC YELLOW PAGES-FINANCIAL SERVICES DIRECTORY □ ELECTRONIC YELLOW PAGES-PROFESSIONALS DIRECTORY □ ELECTRONIC YELLOW PAGES-RETAILERS DIRECTORY □ **ELECTRONIC YELLOW PAGES-WHOLESALERS DIRECTORY** □ ENCYCLOPEDIA OF ASSOCIATIONS □ ENERGY DATABASE (U.S. Department of Energy) □ **ENERGYLINE (Energy Information Abstracts)** □ ENERGYNET (Environment Information Center) □ ENVIROLINE (Environmental Information Center) □ ENVIRONMENTAL BIBLIOGRAPHY (Environmental Studies Institute) □ ERIC (Educational Resources Information Center) □ EXCEPTIONAL CHILD EDUCATION RESOURCES (Council for Exceptional Children) □ EXCERPTA MEDICA (Biomedical literature) □ FAMILY RESOURCES (National Council on Family Relations) □ FEDERAL INDEX (Capitol Services International) □ FEDERAL REGISTER ABSTRACTS (Capitol Services International) □ **FEDERAL ENERGY INDEX (U.S. Department of Energy)** □ FINANCIAL TIMES OF LONDON (Fintel, Ltd.) □ **FIND/SVP REPORTS AND STUDIES INDEX** □ FOOD SCIENCE AND TECHNOLOGY ABSTRACTS □ FOODS ADLIBRA (Food technology and packaging) □ FOREIGN TRADERS INDEX (U.S. Department of Commerce) □ FOUNDATION DIRECTORY (Nonprofit organizations) □ FOUNDATION GRANTS INDEX (The Foundation Center) □ **FROST AND SULLIVAN DM^2 (Defense Market Measures System)** □ **GEOARCHIVE (Geoscience Database)** □ **GEOREF (American Geological Institute)** □ GPO MONTHLY CATALOG (U.S. Government Printing Office) □ GPO PUBLICATIONS REFERENCE FILE (Government Printing Office) □ GRANTS (Grant Information System) □ HARFAX INDUSTRY DATA SOURCES (Ballinger Publishing Co.) □ **HARVARD BUSINESS REVIEW (Full text)** □ HEALTH PLANNING AND ADMINISTRATION (National Library of Medicine) □ HISTORICAL ABSTRACTS (World periodical literature) □ INPADOC (International Patent Documentation Center) □ **INSPEC (Physics Abstracts, Electrical and Electronics Abstracts, and Computer and Control Abstracts** □ INSURANCE ABSTRACTS (Life, property and liability insurance literature) □ INTERNATIONAL PHARMACEUTICAL ABSTRACTS (American Society of Hospital Pharmacists) □ **INTERNATIONAL SOFTWARE DIRECTORY (Computer programs)** □ IRIS (Instructional Resources Information System) (Water resources) □ **ISMEC (Information Service in Mechanical Engineering)** □ LANGUAGE AND LANGUAGE BEHAVIOR ABSTRACTS □ LEGAL RESOURCE INDEX (Information Access Corporation) □ LIFE SCIENCES COLLECTION (Cambridge Scientific Abstracts) □ LISA (Library and Information Science Abstracts) □ MAGAZINE INDEX (Information Access Corporation) □ MANAGEMENT CONTENTS (Business management and administration) □ MARC (U.S. Library of Congress) □ **MATHFILE (American Mathematical Society)** □ MEDLINE (U.S. National Library of Medicine) □ MEDOC (Eccles Health Science Library, University of Utah) □ MENTAL HEALTH ABSTRACTS (National Institute of Mental Health) □ **METADEX (Metals Abstracts/Alloys Index)** □ **METEOROLOGICAL AND GEOASTROPHYSICAL ABSTRACTS** □ **MICROCOMPUTER INDEX (Microcomputer Information Services)** □ MILLION DOLLAR DIRECTORY (Dun's Marketing Services) □ MLA BIBLIOGRAPHY (Modern Language Association) □ NARIC (National Rehabilitation Information Center) □ NATIONAL FOUNDATIONS (The Foundation Center) □ NATIONAL INFORMATION SOURCES ON THE HANDICAPPED □ **NATIONAL NEWSPAPER INDEX (The Christian Science Monitor, The New York Times, and The Wall Street Journal)** □ NCJRS (National Criminal Justice Reference Service) □ **NEW YORK TIMES INFO BANK-II** □ NEWSEARCH (Daily index of news stories) □ NICEM (National Information Center for Educational Media) □ NICSEM/NIMIS (National Instructional Materials Information System) □ NIMH (National Institute of Mental Health) □ NIMIS (National Center of Educational Media and Materials for the Handicapped) □ **NON-FERROUS METALS ABSTRACTS (Non-Ferrous Metals Technology Center, England)** □ **NTIS (National Technical Information Service)** □ OCEANIC ABSTRACTS (Marine related subjects) □ ONLINE CHRONICLE (News in the online industry) □ PAIS INTERNATIONAL (Public Affairs Information Service) □ PAPERCHEM (Institute of Paper Chemistry) □ **PATDATA (U.S. Patents)** □ PATLAW (Bureau of National Affairs) □ PHARMACEUTICAL NEWS INDEX (Data Courier, Inc.) □ PHILOSOPHER'S INDEX (Philosophy Documentation Center) □ PIRA (Research Association for the Paper and Board, Printing and Packaging Industries) □ POLLUTION ABSTRACTS (Cambridge Scientific Abstracts) □ POPULATION BIBLIOGRAPHY (University of North Carolina) □ PREDICASTS PROMPT/F&S INDEX □ PRE-MED (Current clinical medicine) □ PRE-PSYC (Current clinical psychology) □ PSYCINFO (American Psychological Association) □ PTS F&S INDEXES (Predicasts, Inc.) □ PTS INTERNATIONAL FORECASTS (Predicasts, Inc.) □ PTS INTERNATIONAL TIME SERIES (Predicasts, Inc.) □ PTS PREDALERT (Predicasts, Inc.) □ PTS PROMPT (Predicasts, Inc.) □ PTS U.S. FORECASTS (Predicasts, Inc.) □ PTS U.S. TIME SERIES (Predicasts, Inc.) □ RAPRA ABSTRACTS (Rubber and Plastics Research Association) □ **RCA TECHNICAL ABSTRACTS DATABASE (TAD)** □ RELIGION INDEX (American Theological Library Association) □ REMARC (U.S. Library of Congress) □ RESOURCES IN VOCATIONAL EDUCATION □ RILM ABSTRACTS (Repertoire International de Litterature Musicale) □ SCHOOL PRACTICES INFORMATION FILE (Education Service Group) □ SCISEARCH (Science Citation Index) □ SOCIAL SCIENCE CITATION INDEX □ SOCIOLOGICAL ABSTRACTS □ **SPIN (Searchable Physics Information Notices)** □ **SSIE CURRENT RESEARCH (Smithsonian Science Information Exchange)** □ STANDARD AND POORS NEWS □ **STANDARDS AND SPECIFICATIONS (National Standards Association)** □ STATE PUBLICATIONS INDEX □ **SUPERINDEX (Indexes to Reference Books)** □ **SURFACE COATINGS ABSTRACTS (Paint Research Association of Great Britain)** □ TRADE AND INDUSTRY INDEX (Information Access Corporation) □ TRADE OPPORTUNITIES (U.S. Department of Commerce) □ TRADE OPPORTUNITIES WEEKLY (U.S. Department of Commerce) □ TRIS (Transportation Research Information Service) □ TSCA INITIAL INVENTORY (Toxic Substances Control Act) □ U.S. EXPORTS (U.S. Department of Commerce) □ U.S. PUBLIC SCHOOL DIRECTORY □ ULRICH'S INTERNATIONAL PERIODICAL DIRECTORY □ UNITED STATES POLITICAL SCIENCE ABSTRACTS (University of Pittsburgh) □ USBE MOST-AVAILABLE TITLES (Universal Serials and Book Exchange, Inc.) □ WATER RESOURCES ABSTRACTS (U.S. Department of the Interior) □ **WELDASEARCH (The Welding Institute)** □ WORLD AFFAIRS REPORT (California Institute of International Affairs) □ WORLD ALUMINUM ABSTRACTS (American Society for Metals) □ WORLD TEXTILES (World Textile Abstracts)

Pen and Podium

Recent RCA technical papers and presentations

To obtain copies of papers, check your library or contact the author or his divisional Technical Publications Administrator (listed on back cover) for a reprint.

Advanced Technology Laboratories

W. Heagerty | M. Kernan | J. Pridmore
Simulation of Radiation Effects on CMOS/SOS RAMs—Presented at SOS/SOI Workshop, Provincetown, Mass. (10/5/82)

R. Kenville
Hardware Issues for Optical Mass Storage Systems—Presented at the Fifth IEEE Symposium on Mass Storage Systems, Boulder, Colo., published: *Proceedings* (10/27-29/82)

H. Li
VLSI Hierarchical Systems Architecture—Presented at a Seminar at Cornell School of Electrical Eng'g., Ithaca, N.Y. (10/5/82)

H. Li
(P. Sawkar | T. Forquer | E. Schernecks—MSR)
A Multiport Memory Organization for Use in Distributed Computing Systems—Presented at the Third National Conference on Distributed Computing Systems, Ft. Lauderdale, Fla., published in *Proceedings* (10/18-22/82)

M. Stebnisky | M. McGinnis | J. Werbickas
R. Putatunda | A. Feller
A Fully Automatic, Technology-Independent PLA Macrocell Generator—Presented at IEEE 1982 International Conference on Circuits and Computers, New York City, published in *Proceedings* (9/28-10/1/82)

Astro-Electronics

A. Aronson, *et al.*
Schottky-barrier IR CCD for Earth Sensing Applications—Government Microcircuit Applications Conference, Orlando, Fla. (11/2/82)

K. Budlong
Testing Large Spacecraft-Deployables—7th Aerospace Testing Seminar, Los Angeles, Calif. (10/13/82)

R. Buntschuh | R. Miller
RCA Direct Broadcast Satellite Configuration—National Telesystems Conf., Galveston, Tex. (11/7/82)

R. Buntschuh (A. Sadikeh—APL)
Nova Disturbance Control (DISCOS) Testing—Aerospace Testing Seminar, San Diego, Calif. (10/13/82)

F.H. Chu
Proposal for Organizing Symposium on Modal Test & Model Refine—ASME Gen. Comm. 103rd Winter Mtg., Phoenix, Ariz. (11/11/82)

L. Freedman
Space Shuttle Closed Circuit TV System—Nat'l. Telesystems Conference, Galveston, Tex. (11/10/82)

F. Gargione
Use of 8236 Plotter with 430 Offline Controller—Fall Tech. Mtg. of the Applicon Users Group, Philadelphia, Pa. (11/2/82)

S. Gaston
Unique Battery Reconditioning Cycle for RCA Geostationary Satellites—GSFC Battery Workshop (1982), Greenbelt, Md. (11/16/82)

J. Horan
TC Sensor Systems Highlights—*AIAA Astronautics and Aeronautics Journal,* (12/82)

L. Osborn
NOVA Flight Application of a Pulsed Plasma Microthruster—AIAA, New Orleans, La. (11/17/82)

S.F. Schiffer
RCA Astro Battery IR&D Program—Chemical Working Group, 1982 Independent Research and Development Meeting, Power Information Center, Philadelphia, Pa. (11/19/82)

S. Seehra | W. Slusark
Effect of Operating Conditions on the Rad Resistance of VDMOS Power FETs—*'82 IEEE Transactions on Nuclear Science* (12/82)

J. Staniszewski
Expectations Achieved NOVA 1—The Second Generation Navy Navigation Satellite—SAE Aerospace Control & Guidance System Committee, Williamsburg, Va. (11/2/82)

Automated Systems

R.C. Blanchard | R.E. Hanson
Simplified Test Equipment Applied to Turbine Engines—AUTOTESTCON '82, Dayton, Ohio (10/82)

L.B. Blundell | R.A. Lindley
AN/USM-410 Automatic Test Equipment Capability Augmentation—AUTOTESTCON '82, Dayton, Ohio (10/82)

S.E. Courcy | H.L. Resnick
Advanced Hardware and Software Techniques for Third Generation Field Portable ATE—AUTOTESTCON '82, Dayton, Ohio (10/82)

A. Eiranova | K.T. Ronan
Test Strategies for Microprocessor Based UUTs—AUTOTESTCON '82, Dayton, Ohio (10/82)

K.E. Ghostlaw | J.F. McCarthy
Test and Repair Subsystem Development for the AH-64 Attack Helicopter—AUTOTESTCON '82, Dayton,, Ohio (10/82)

W.C. Goodno | A.T. Hospodor
RCA Begins Production of ATE Test System—*Army Aviation* (10/82)

R.E. Hartwell
Simplifying the Development of ATE Software for Simplified Test Equipment—AUTOTESTCON '82, Dayton, Ohio (10/82)

S.A. Haslam | M.F. Krayewsky | J.F. Weiss
E3A Depot Automatic Test Equipment—AUTOTESTCON '82, Dayton, Ohio (10/82)

F.F. Martin
Schottky-Barrier Infrared Focal Plane Array Sensor Technology—Second Annual New England Symposium of the Assoc. of Unmanned Vehicle Systems, MITRE, Bedford, Mass. (11/82)

A.R. Pelletier
Shock/Vibration Applied to Military Design—Northeastern University Student ASME Chapter Presentation, Boston, Mass. (11/82)

P.E. Seeley
Electro-Optics: Broadening Our Vision and Capabilities—Northeastern University IEEE Student Chapter, Boston, Mass. (11/82)

L.B. Smith
Introduction to Engineering—Presented to Co-op Students, Wentworth Institute of Technology, Boston, Mass. (10/82)

L.B. Smith
Introduction to Engineering for Co-op Students—Presentation/Panel Discussion, Northeastern University, Boston, Mass. (11/82)

Government Communications Systems

D.W. Diehl | M. Lysobey | B.D. Pewitt
Frequency-Agile HF Antenna Coupler—Presented at Hi Freq. Comm. Symp., Griffiss AFB, Rome, N.Y., and published in *Proceedings.*

D. Hampel
Low-Power Dissipation Micro-Signal Processor for High Resolution Spectral Analysis—Presented at GOMAC Conference, Orlando, Fla., and published in *Classified Supplement to GOMAC Digest* (11/2/82)

E.J. Nossen
Robust HF Data Communications—Presented at Hi Freq. Comm. Symp., Griffiss AFB, Rome, N.Y., and published in *Proceedings* (11/16-18/82)

J. Springer
The Architecture of a Multi-Computer Signal Processor—Presented at Real Time System Symposium, IEEE, Los Angeles, Calif., and published in *Proceedings* (12/7-9/82)

K. Weir | J.J. Cohen
Operation and Maintenance Trainers for TRIDENT Integrated Radio Room—Presented at 4th Interservice Industry Training Conference, Orlando, Fla., and published in *Proceedings* (11/16-18/82)

Laboratories

D.F. Dixon
PASCAL vs. C: A Friendly Comparison—at Presented to Princeton Chapters of ACM and IEEE (11/9/82)

T.J. Faith | R.S. Irven | S.K. Plante | J.J. O'Neill
Contact Resistance: Al and Al-Si to n^+ and p^+ Silicon—Presented at American Vacuum Society National Symposium, Baltimore, Md. (11/16/82)

W. Kern | R.B. Comizzoli | G.L. Schnable
Fluorescence Techniques Allow Defect Study in Microelectronics—*Industrial Research & Development*, Vol. 24, No. 5, pp. 151-160 (5/82)

W. Kern
Fluorescent Tracers Simplify Detection of Microdefects—*Industrial Research & Development,* Vol. 24, No. 6, pp. 131-134 (6/82)

J.S. Maa | J.J. O'Neill
Reactive Ion Etching of Al and Al-Si Films with CCl_4, N_2, and BCl_3 Mixtures—Presented at American Vacuum Society National Symposium, Baltimore, Md. (11/16/82)

J.L. Vossen
Thin Film Deposition Methods—*SERI Amorphous Materials Review*, Arlington, Va. (11/22/82)

J. Zelez
Low-Stress Diamond-Like Carbon Films—Presented at American Vacuum Society National Symposium, Baltimore, Md. (11/16/82)

Missile and Surface Radar

O.G. Allen
The Control Chart As A Tool for Product Improvement—ASQC 26th Annual Symposium, Cherry Hill, N.J. (11/18/82)

R.M. Blasewitz
Ada As A Program Design Language—AFSC Standardization Conference, Dayton, Ohio, Conference *Proceedings* (11/30/82)

F.J. Buckley
Software Quality Assurance—Instructor, Software Quality Assurance Course, IEEE, Washington, D.C. (10/30-31 to 11/1/82)

F.J. Buckley
Software Quality Assurance—IEEE Seminar, San Francisco, Calif. (11/30/82)

F.J. Buckley
The Application of Distributed Computers to a Real-Time System—Stanford University, Stanford, Calif. (12/1/82)

F.J. Buckley
A Modest Proposal (With Apologies to J. Swift)—*Computer* magazine (11/82)

C.L. Christianson
Air Defense—Highly Mobile Tactics, Very Smart Weapons, New Survivable Architectures Drive Military Needs—Computers in Defense Conference, Data Processing Management Assoc., Washington, D.C. (12/2-3/82); and Boston, Mass. (12/13-14/82)

W. DiCecca | J. Golub | D. Kratzer
H. Pschunder
Application of Air Warfare Simulation in Determining Tactical Air Control System (TACS) Effectiveness—1982 Winter Simulations Conference, San Diego, Calif. (12/6-8/82)

M.L. Dorman, Jr.
Missiles From The Deep—*Sea Classics,* Vol. 16, No. 1, pp. 58-77, Part 2 of 3 parts (1/83)

J.W. Douglas
Building, Backing-up and Restoring the Disk—Applicon Users Group Fall Technical Meeting, Philadelphia, Pa. (11/1/82)

B.M. Fell, Co-author
The Impurity-Density Dependence of the Conductivity of a Metal—*Acta Physica Austriaca* (9/82)

T.J. Forquer
Potential Supercomputer Architectures—Computers in Defense Conference, Data Processing Management Assoc., Washing-
ton, D.C. (12/2-3/82); and in Boston, Mass. (12/13-14/82)

J. Haness
Effective Communications—IEEE PACE Meeting, Moore School, Univ. of Pennsylvania, Philadelphia, Pa. (12/6/82)

R.F. Kolc
VLSI Packaging at RCA Moorestown—IEEE VLSI Packaging Workshop, Gaithersburg, Md. (9/14/82)

H. Urkowitz
Synthesis of Multiple Random Processes Based Upon a Cholesky Decomposition—GLOBECOM '82 (1982 Global Comm. Conf.), Miami Beach, Fla., Conference *Record* (11/19/82)

F.E. Wuebker
The Impact of Nebula, MCF, & ADA on Real-Time Embedded Computer Systems—AFSC Standardization Conference, Dayton, Ohio, Conference *Proceedings* (11/30/82

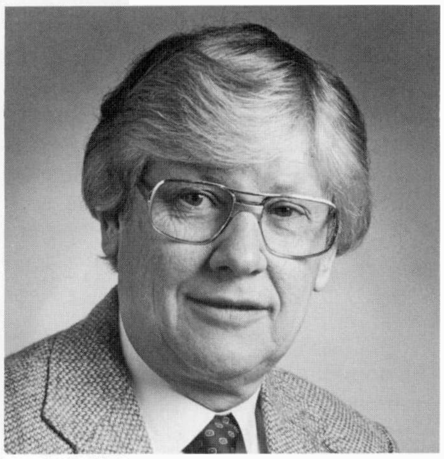
Solid State Division

K. Karstad
Microsystem modem makes dialing-up a breeze—*Electronics* magazine (12/15/82)

R.E. Stricker
Conductivity of Heavily and Lightly Ion Implanted Polysilicon Films for VLSI Circuits—1983 International Symposium on VLSI Technology, Systems and Applications, Taipei, Taiwan, R.O.C. (3/30 — 4/1/83)

Engineering News and Highlights

Fisher named engineering VP



John D. Rittenhouse, Division Vice-President and General Manager, RCA Picture Tube Division, announced the appointment of **Mahlon B. Fisher** as Division Vice-President, Engineering. Mr. Fisher will be responsible for all product and equipment engineering, design and development activities of the Division. Previously, Mr. Fisher was employed in a similar position by the N.A. Phillips Electronics Group, Inc. in Seneca Falls, N.Y. He received his B.S. degree in Electrical Engineering from the University of Washington and his M.S. degree from Stanford University.

Zang named to Corporate Engineering Education

Richard A. Zang joined the RCA Corporate Engineering Education group in January 1983 as Administrator, Engineering Education Programs. His most recent teaching experience includes a stint as Chairman of the Engineering Department at Stanly Technical College in Albemarle, North Carolina. In addition he has taught Math at the State University of New York (SUNY) at Binghamton, Engineering at the University of North Carolina at Charlotte (UNCC), and Microprocessors at IBM. He logged eight years in various positions at IBM, four of which were as an Engineer/Scientist. Rick holds a Bachelor's degree in Electro-Mechanical Technology and a Masters degree in Mathematical Sciences from SUNY at Binghamton. Contact him at: RCA Corporate Engineering Education, Cherry Hill, N.J., TACNET: 222-5020.

Staff announcements

Consumer Electronics

James J. Legault, Manager, Quality Control, announces the appointment of **David W. Keller** as Manager, In-Process Quality Control.

James E. Carnes, Division Vice-President, Engineering, announces the appointment of **Tom W. Branton** as Manager. Projection Project Engineering.

Laboratories

Richard E. Quinn, Staff Vice-President, Administration, announces the appointment of **William L. Harrington** as Head, Materials Characterization Research.

David Richman, Director, Materials and Processing Research Laboratory, announces his organization as follows: **Charles H. Anderson,** Head, Applied Mathematical and Physical Sciences; **Vladimir S. Ban,** Head, Video-Disc Materials and Diagnostics Research; **Chih Chun Wang,** Fellow, Technical Staff; **Glenn W. Cullen,** Head, Materials Synthesis Research; **Leonard P. Fox,** Head, Video-Disc Applied Process Research; **Richard E. Honig,** Staff Scientist; **Daniel L. Ross,** Head, Organic Materials and Devices Research; and **Robert J. Ryan,** Head, Polymer Processing Research.

Carmen A. Catanese, Director, Picture Tube Systems Research Laboratory, announces his organization as follows: **Curt R. Carlson,** Head, Image Quality and Human Perception Research; **Charles H. Anderson,** Fellow, Technical Staff; **Harry E. McCandless,** Manager, Advanced Development—Electron Guns (TTL); **David L. Staebler,** Head, Electron Optics and Applied Mathematics Research; **Roger L. Crane,** Fellow, Technical Staff; **Ralph W. Klopfenstein,** Fellow, Technical Staff; **Peter J. Wojtowicz,** Head, Magnetic Deflection Research; and **William H. Barkow,** Fellow, Technical Staff.

RCA Global Communications, Inc.

Joe T. Swaim, Vice-President, Switched Services Engineering and Operations, announces his organization as follows: **William A. Klatt,** Director, Network Operations; **Russell E. Blackwell,** Director, KCC Computer Operations; **John P. Shields,** Director, Network Engineering; **William D. Barnum,** Director, Field Support Services; **Richard L. Chory,** Manager, Software Engineering; **Rudolph K. Lang,** Program Manager, Telex Systems; **Kenneth H. Wendt,** Program Manager, Message Switching Systems; and **Jerome S. Bonow,** Administrator, KCC Administrative Services.

John P. Shields, Director, Network Engineering, announces his organization as follows: **Leo A. Tita,** Manager, Traffic Engineering; **Joseph G. Fitzgerald,** Leader, Network Planning; **James M. Patrick,** Leader, Facilities and TOCC; and **Dewey G. Snyder,** Leader, Computer Support.

James H. Muller, Director, Leased Facilities and Systems Engineering, announces his organization as follows: **Anthony J. Falco,** Manager, Central Office Engineering; **Alexander A. Avanessians,** Manager, Customer Engineering; **Solomon J. Nahum,** Manager, Construction and Installation; **Robert Ruben,** Manager, Equipment Engineering; **Roger C. Lewis,** Manager, Facilities Coordination and Telephone Services; and **James R. McDonald,** Administrator, Special Projects.

Obituary

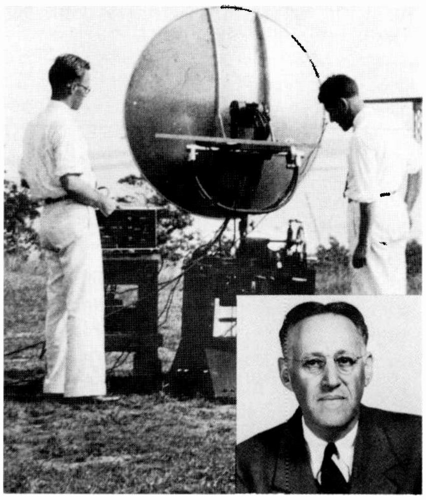
Irving Wolff, radar pioneer, dead at 88

Dr. Irving Wolff, a retired RCA Laboratories executive who pioneered in the development of radar, died at Princeton Hospital, in N.J. He was 88 years old. In addition to his work on radar, Dr. Wolff made important contributions in such diverse fields as acoustics, optics, radio, infrared detection, and radio frequency heating.

Dr. Wolff was born in New York on July 6, 1894. A Physics student, he was graduated from Dartmouth College in 1916, and received his Ph.D. from Cornell University in 1923. Following his graduation from Dartmouth, he served as a Physics instructor at Iowa State College and at Cornell where he became a Hechsher Research Fellow in 1924. Later that year, Dr. Wolff joined the RCA Technical and Test Laboratory at Van Cortlandt Park in New York. In the early 1930s, he was transferred to Camden, N.J., the base for his radar work. Shortly after World War II, he was appointed Director of the Radio Tube Research Laboratory of RCA Laboratories in Princeton, N.J. In 1951, he was appointed Director of Research, RCA Laboratories, and in 1954, he was elected Vice-President, Research. He retired in 1959.

In 1948, the Navy awarded him its highest civilian honor, the Distinguished Public Service Award, for ". . . one of the fundamental contributions to modern-day radar." The Franklin Institute, in 1959, presented him with its prestigious Elliott Cresson Medal "In consideration of his many important contributions to the science of electronics . . . ." Long active in professional and educational affairs, Dr. Wolff was a Fellow of the American Physical Society, the Acoustic Society of America, the American Association for the Advancement of Science, and the Institute of Radio Engineers. Dr. Wolff served as an advisor to several educational, governmental, and scientific organizations.

Some recollections of Irving Wolff



Dr. Irving Wolff (right) and Dr. Ernest Linder with microwave test equipment at Atlantic Highlands, N.J., in 1934.

In 1932, Dr. Irving Wolff initiated an RCA research program on extremely short radio waves, later called microwaves. He recruited me from Cornell University to assist him in that work. This was the beginning of an association of over 50 years that was not only fruitful scientifically, but resulted also in a warm personal friendship between us and between our families.

As a result of this research program, the future importance of microwaves first became evident in transmission tests at Atlantic Highlands, N.J., in 1934. Measurements showed no detectable absorption in the atmosphere, and thus cleared the way for important long-range applications such as radar, telephone relay systems, TV networks, navigation systems and, eventually space communications and survey systems. Tests were made also of the reflection properties of microwave beams. We found it possible to locate objects such as large trees, a water tank, and a ship, by reflection of the beam and its subsequent reception by our receiving equipment. Doc Wolff coined the term "radio vision" for this. Later, it was called "radar," but obviously Doc's term was much more prophetic of future technological possibilities.

During the 1930s, Doc and four RCA friends organized the Falcon Aero Club, bought a monoplane, and became enthusiastic amateur flyers. Doc's interest in aviation had a significant impact on the direction of RCA microwave development. Under his leadership, programs were undertaken on radar, altimeters, and various navigation and missile guidance systems, all of which were importantly successful. In recognition of these achievements, Dr. Wolff received numerous awards and his continued successes led, in 1954, to his election as Vice-President, Research, RCA Laboratories.

Doc enjoyed diversified interests. As an enthusiastic skier, he frequented the slopes of St. Moritz, the Poconos, and Aspen. He was a devoted family man. Ten years ago, Doc and I, accompanied by our wives, Connie and Jackie, toured Europe together for three months. This different environment brought out some of Doc's qualities I had not previously fully appreciated. His extensive knowlege of European history, his appreciation of art, his comprehension of music, his delight in gardens and flowers, all became more evident. He displayed a passable knowledge of the French, German, and Italian languages, and delighted in making use of it

But Doc Wolff was primarily a scientist and maintained continued direct contact with research in progress. Even when he was made Vice-President, he refused to move his office, preferring to remain where he was, surrounded by his research people, each of whom knew that Doc could be counted on for friendly interest and steadfast encouragement.

I found working with Doc to be a stimulating experience. He possessed keen intelligence, a fertile imagination, and an unusual ability to reduce experimental concepts to practical reality. Science in general, and RCA in particular, have benefited greatly from his talents.

—*Ernest G. Linder*

Professional activities

RCA employee participates in Eta Kappa Nu Jury of Award Meeting

1982 Jury of Award. *Seated (left to right): Ohlman, Renda, Musa, and Lueg. Standing, (left to right): D'Arcy, Rathjen, and Cruz.*

The 1982 Eta Kappa Nu (Honorary Electrical Engineering Society) Jury of Award convened at the headquarters of Indianapolis Power & Light Company on November 15, 1982, to select the Outstanding Young Electrical Engineer of the United States. The Jury Meeting was organized by **James A. D'Arcy** (RCA "SelectaVision" VideoDisc Operations), who is Chairman of the Eta Kappa Nu Awards Organization Committee.

The members of the Jury were:

Dr. Jose B. Cruz, Jr. (University of Illinois) Vice-President for Technical Activities, IEEE;

Dr. Russell E. Lueg, Professor of Electrical Engineering, University of Alabama (Past President, Eta Kappa Nu);

Dr. Samuel Musa, Staff Specialist for Electronic Warfare and target acquisition, Office of the Under Secretary of Defense for Research and Engineering;

Mr. Charles E. Ohlman, Senior Vice-President, Consumer Services, Indianapolis Power & Light Company;

Mr. Keith Rathjen, Vice-President for Advanced Technology and Engineering, Defense Electronics Operations, Rockwell International; and

Dr. R. Bruce Renda, Dean of the School of Engineering and Technology, IUPUI (Indianapolis).

The Eta Kappa Nu Recognition Award emphasizes among Electrical Engineers that their service to mankind is manifested not only by achievements in purely technical affairs, but in a variety of other ways. Since 1936, forty-six young men who were less than 35 years of age and who had received their Baccalaureate degree less than 10 years before, have received the Award and 100 men of similar characteristics have received honorable mention. The most recent RCA employee to receive this Award is **John G.N. Henderson** (RCA Laboratories) who was the 1977 winner. In addition, two other RCA employees, **Glenn A. Reitmeier** and **Dan Botez** (both of RCA Laboratories), were recognized as 1981 finalists in this award program. The 1982 winners will be honored at an award banquet on April 18, 1983, in New York City.

Technical excellence

Engineering recognition award announced

The Astro-Electronics Engineering Excellence Committee announced an Engineering Recognition Award for **Steven D. Pergament** for his outstanding analytical skills and professionalism in the performance of the thermal design tasks associated with the DE-A and DE-B satellites.

Steve's assignment was to provide thermal analyses in support of the design of the DE-A and DE-B spacecraft, to support the planning, setup, conduction and data analysis of thermal balance testing of these spacecraft, and finally to perform post-launch evaluation of the thermal subsystems. The assignment involved the performance of detailed thermal analyses of complete spacecraft configurations (both DE-A and DE-B). These analyses, and later the preparation for and conduction of thermal balance testing were performed in a high-pressure environment due to an extremely compacted program schedule to launch, and the untimely resignation of the former key thermal analyst. Through Steve's outstanding performance, not only was he able to supply all of the required thermal analyses, but he also assumed responsibility for various power subsystem analyses.

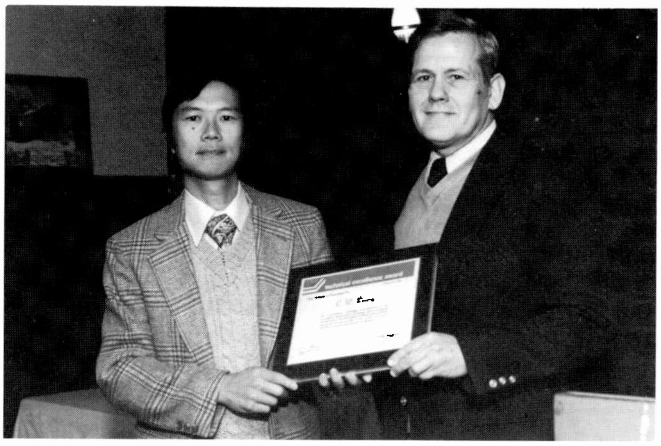
The results of these analyses were presented on an almost immediate basis to his supervisors, RCA program management, and to the customer (Goddard Space Flight Center). Steve's ability to interact with clarity and consideration with other DE team members was a motivating factor for the whole project. This professionalism and enthusiasm carried through to Steve's assignment as thermal subsystem engineer at the Spacecraft Operational Control Center at GSFC. Steve's efforts contributed directly to the on-time shipment of the DE-A and DE-B spacecraft. His analyses now serve as the basis for further missions using the DE type bus.

Solid State Division recognizes technical excellence

Solid State Division's second and.third quarter 1982 technical excellence awards were presented on November 19, 1982. Teams and individuals pictured on this page were honored at an awards luncheon.

Pictured from left to right are **Aaron Ashkinazy, Gary Gendel, Shirley Chen, Chris Davis** and **Larry Rosenberg,** Manager, Design Automation. Mr. Ashkinazy, Mr. Gendel, Ms. Chen, and Mr. Davis were recognized as team contributors in the design and development of the state-of-the-art MIMIC Logic Simulation Program.

Pictured from left to right are **Joseph Koskulitz,** Manager, Offshore Facility Development, **Roland Jacobus, Raymond Reusch, Walter Nething, Joseph Ingrassia, Ralph Thompson, Alvin Hom, Maurice Rosenfield,** Manager, Assembly Packaging Development, and **Leonard Gibbons,** Manager, Reliability Engineering. Mr. Jacobus, Mr. Reusch, Mr. Nething, Mr. Ingrassia, Mr. Thompson, Mr. Hom, and **Mr. Beelitz** (not shown) were recognized as a team in their efforts relative to the specification, design and installation of the RCA Assembly/Manufacturing Modernization Program within the Malaysia facility.



Bernie Leung (left) is being presented his award by **Andrew Dingwall** (right). As an individual contributor, Mr. Leung was instrumental in the design and development of processing techniques for the fabrication of Short-Channel VLSI Structures.

Richard Lydick (left) is being presented his award by **David Jacobsen,** Director Custom LSI (right). As an individual award recipient, Mr. Lydick was recognized for his development of LSI Emulating Controllers using the SOS APAR Automated Design System with sub-chip capability.

Editorial Representatives

Contact your Editorial Representative at the TACNET numbers listed here to schedule technical papers and announce your professional activities.

Cablevision Systems

		TACNET
*John Ovnick	Van Nuys, California	534-3011

Commercial Communications Systems Division (CCSD)

*Bill Sepich	Camden, New Jersey	222-2156
Clinton Glenn	Meadowlands, Pennsylvania	228-6231

Consumer Electronics (CE)

*Eugene Janson	Indianapolis, Indiana	422-5208
Francis Holt	Indianapolis, Indiana	422-5217
Chuck Limberg	Indianapolis, Indiana	422-5117
Don Willis	Indianapolis, Indiana	422-5883
Byron Taylor	Indianapolis, Indiana	426-3247

Government Systems Division (GSD)

Advanced Technology Laboratories		
*Merle Pietz	Camden, New Jersey	222-2161
Ed Master	Camden, New Jersey	222-2731
Astro-Electronics		
*Frank Yannotti	Princeton, New Jersey	229-2544
Carol Klarmann	Princeton, New Jersey	229-2919
Automated Systems		
*Paul Seeley	Burlington, Massachusetts	326-3095
Dale Sherman	Burlington, Massachusetts	326-2985
Government Communications Systems		
*Dan Tannenbaum	Camden, New Jersey	222-3081
Harry Ketcham	Camden, New Jersey	222-3913
GSD Staff		
*Ed Moore	Cherry Hill, New Jersey	222-5833
Missile and Surface Radar		
*Don Higgs	Moorestown, New Jersey	224-2836
Jack Friedman	Moorestown, New Jersey	224-2112
Graham Boose	Moorestown, New Jersey	224-3680

National Broadcasting Company (NBC)

*Bob Mausler	New York, New York	324-4869

Patent Operations

George Haas	Princeton, New Jersey	226-2491

Picture Tube Division (PTD)

*Ed Madenford	Lancaster, Pennsylvania	227-3657
Nick Meena	Circleville, Ohio	432-1228
Jack Nubani	Scranton, Pennsylvania	329-1499
J.R. Reece	Marion, Indiana	427-5566

RCA Communications

		TACNET
American Communications		
*Murray Rosenthal	Princeton, New Jersey	258-4192
Carolyn Powell	Princeton, New Jersey	258-4194
Global Communications		
*Dorothy Unger	New York, New York	323-7348

RCA Inc. (Canada)

Bob McIntyre	Ste Anne de Bellevue	514-457-9000

RCA Records

*Greg Bogantz	Indianapolis, Indiana	424-6141

RCA Service Company

*Joe Steoger	Cherry Hill, New Jersey	222-5547
Ray MacWilliams	Cherry Hill, New Jersey	222-5986
Dick Dombrosky	Cherry Hill, New Jersey	222-4414

Research and Engineering

Corporate Engineering		
*Hans Jenny	Cherry Hill, New Jersey	222-4251
Laboratories		
Eva Dukes	Princeton, New Jersey	226-2882

"SelectaVision" VideoDisc Operations

*Nelson Crooks	Indianapolis, Indiana	426-3164

Solid State Division (SSD)

*John Schoen	Somerville, New Jersey	325-6467
Power Devices		
Harold Ronan	Mountaintop, Pennsylvania	327-1633 or 327-1827
Integrated Circuits		
Dick Morey	Palm Beach Gardens, Florida	722-1262
Sy Silverstein	Somerville, New Jersey	325-6168
John Young	Findlay, Ohio	425-1307
Electro-Optics and Power Devices		
John Grosh	Lancaster, Pennsylvania	227-2077
Solid State Technology Center		
Judy Yeast	Somerville, New Jersey	325-6248

***Technical Publications Administrators, responsible for review and approval of papers and presentations, are indicated here with asterisks before their names.**

RCA Engineer

A technical journal published by Corporate Research and Engineering "by and for the RCA engineer"

Bldg 204-2, Cherry Hill, NJ 08358
Forwarding and return postage guaranteed

Printed in the U.S.A.
Form No. RE-28-1