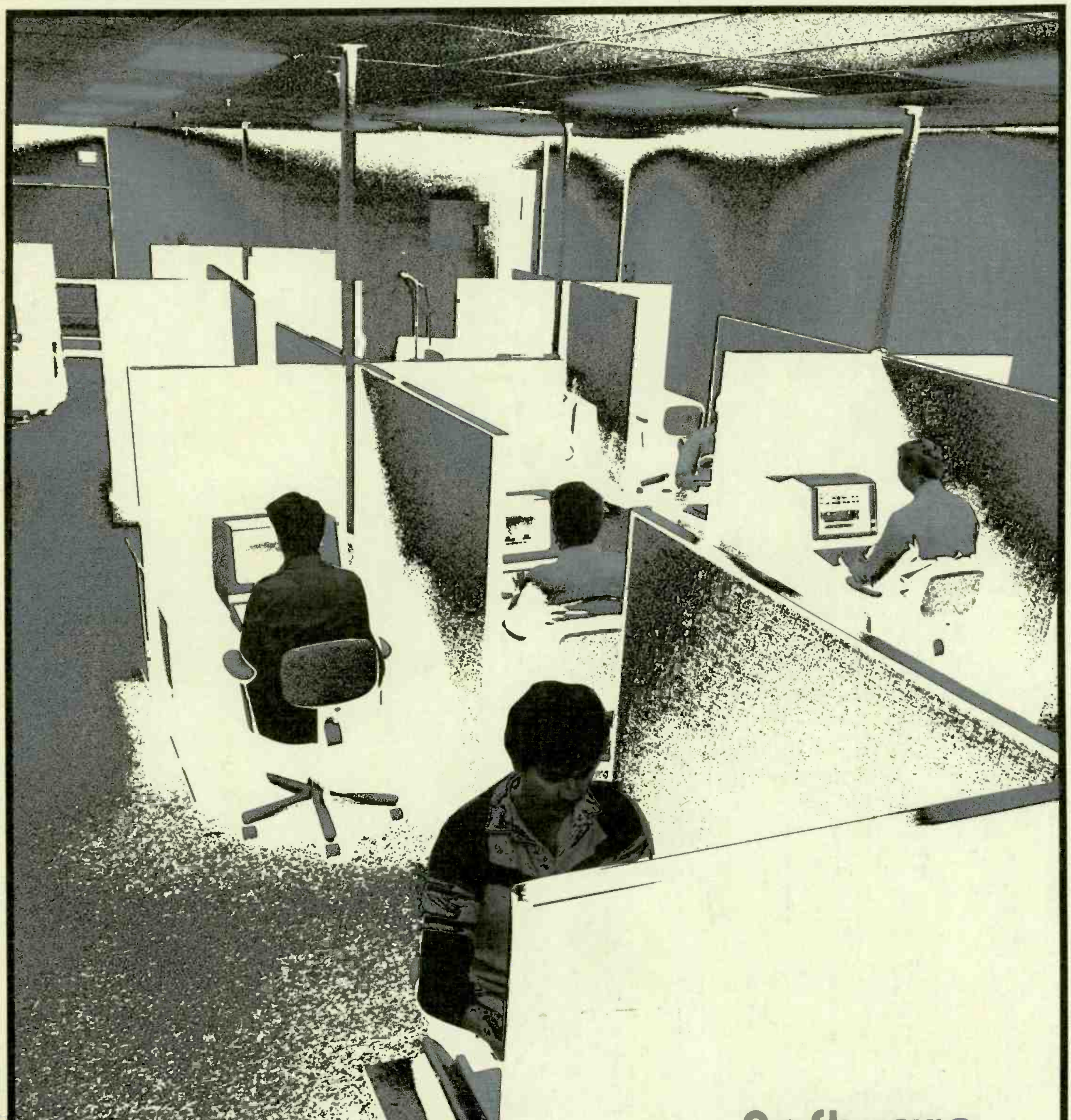
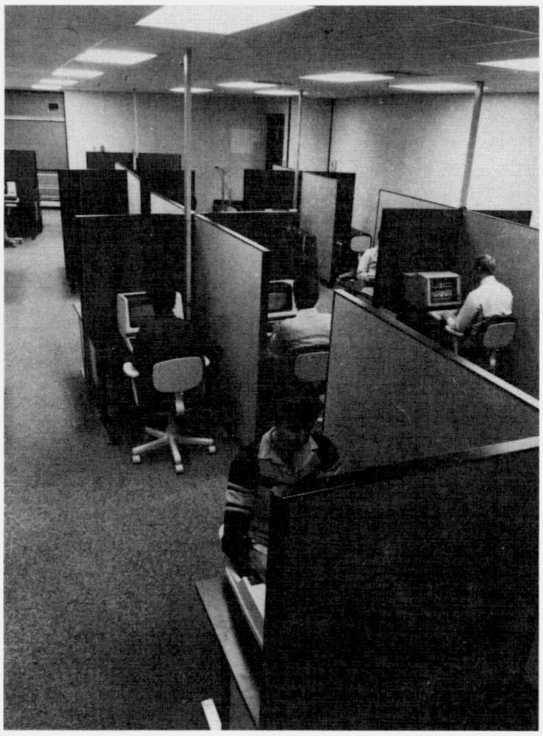
RCA Engineer

Vol. 29 No. 1 Jan./Feb. 1984





Architecture cast in a new light.
The cover for this software issue shows our printer's futuristic rendering of a terminal room at RCA Missile and Surface Radar's newly built computer center in Moorestown, N.J. (see photographs by Ken Kleindienst and article by Charles Liggett on page 37). In this room, software engineers have access to a Program Generation Center for compilation, test and debug of computer programs. Here they use graphics terminals for simulation and design work. Certainly, at RCA, more is being built than meets the eye.

The engineer's creations cast the heretofore tangible business of engineering in an entirely new light. The sheer magnitude of recent software development represents the bringing together of cell upon cell of human activity to build module upon module of computer code. This effort literally leads to a new, software-based architecture, created by a society of talented engineering experts working in concert.

—MRS

RCA Engineer

A technical journal published by
RCA Technical Excellence Center
13 Roszel Road
P.O. Box 432
Princeton, NJ 08540
TACNET: 226-3090 (609-734-3090)

• 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 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.

The maturing of software technology

Computer science, having come of age as a discipline in its own right, has begun to provide advanced software technology. During the last decade, software applications have grown at an unprecedented rate. Both this issue and the last issue of the *RCA Engineer* feature articles that describe some of the advanced developments in RCA software and their diverse applications.

Transfer of technology from scientist to engineer is the concept of early successful structured programming, which is the subject of an article by Braude and Mebus. Many other important ideas in computer science are finding their way into practice, including the sophisticated ideas of packages and parallelism found in Ada (see Blasewitz's article). In fact, software has become so important a part of RCA's business that special consideration must be given to the protection of RCA's software assets (Tripoli).

Among the newest software applications are those which perform functions normally associated with humans. The article by Zapriala *et al.* in the November/December issue describes an example of RCA's concerted thrust in Artificial Intelligence software. An example of the diversity of RCA software is the control of VideoDisc players (Auerbach *et al.*).

RCA not only uses software, but actually depends on it. We are now approaching a point in VLSI design, for example, at which the quality of our tools can hardly be separated from the quality of our products. Even the development of software itself requires sophisticated software, as pointed out by Suhy in the last issue of the *RCA Engineer.*

What of the future? RCA will be intimately involved with all the benefits that the Information Age has to offer. The software of this Age will assist us to give legal, medical, investment and defense advice and to solve problems in ways we can only now begin to imagine.

RCA is a founding member not only of the electronic industry's cooperative Microelectronics and Computer Technology Corporation (MCC) but also of the MCC Software Group. Through our software technology thrusts and those of the MCC, we will be ready for the challenges facing us in the next generation of computing technology.

Ronald A. Andrews
Director, Advanced Technology Laboratories
Government Systems Division

RCA Engineer
Vol. 29|No. 1 Jan.|Feb. 1984

in this issue ...
software

■ **Auerbach, *et al.*:** *"A design constraint was that 'no electrical modifications to the basic circuit boards were allowed.' This meant that the only modifications to the basic player chassis allowed would be in the software of the existing microprocessors."*

■ **Mebus/Braude:** *"Since hundreds of computer languages exist, some have perceived the situation as a modern 'Tower of Babel.'"*

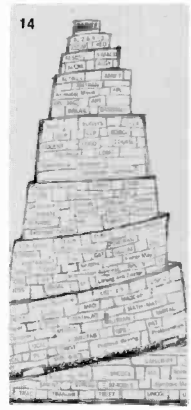
■ **Blasewitz:** *"Ada's adoption as a formal military standard guarantees its use within the defense community, but the commercial applications of Ada may surpass its military use."*

■ **Ed Reps:** *A special pull-out section shows the people in your division who will get your papers published.*

■ **MSR Facility:** *"The realization of this powerful center unifies MSR's computer programming functions under one roof . . . ."*

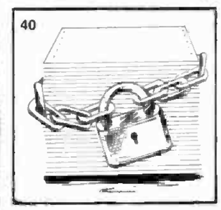
■ **Tripoli:** *"This article will explore the present-day legal alternatives for protecting computer software."*


4


14

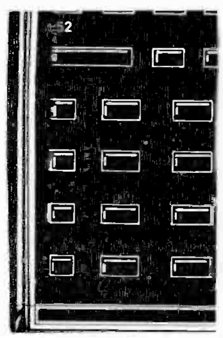

37


23


40

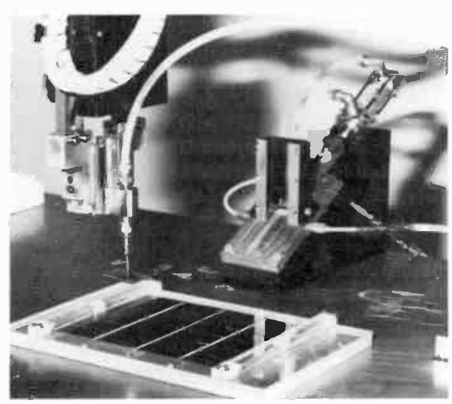
■ **Knapp:** *"One of the benefits of building, operating, and maintaining this complex system is that WNBC-TV has been a leader in local news for several rating periods."*

■ **Stein/Winarsky:** *"Computation errors are especially pernicious when the computer being used is part of a larger system, such as might be used in automated manufacturing."*


52


47

in future issues...
manufacturing,
automating the engineer's workplace,
technical excellence,
materials engineering



V. Auerbach | T.Y. Chen | A.L. Greenberg | N.J. Kiser
T.F. Lenihan | K. Kelleher | J.J. Power | W.G. Gibson

Multiprocessor control system for the SJT-400 random-access player

Direct random access of VideoDisc material allows RCA to explore new opportunities in nonlinear interactive video applications.

Since the introduction of the CED VideoDisc System in 1981, RCA has designed and built machines that play the disc in a linear manner from beginning to end. It had always been realized that the system could support nonlinear interactive applications, and a new VideoDisc player, the SJT-400, has been introduced to support these applications.

The SJT-400 allows the user to directly search to a time, band, or "page" of the disc. It allows custom programming of the video material by time or band. It contains an automatic stop feature to allow users to select alternate branch directions with specially prepared RCA interactive discs. A user-friendly operating system and on-screen display allow easy operation of the player. The design of this player will be reviewed in this paper.

At the inception of the SJT-400 design effort, it was realized that the only cost-effective way to produce this type of player was to modify the high-volume, linear-play SJT-200 player to meet our requirements. A design constraint was that "no electrical modifications to the basic circuit boards were allowed." This meant that the only modifications to the basic player chassis allowed would be in the software of the existing microcomputers. The design goals for the SJT-400 player were met by modifying the software of the existing player microcomputer to include an interface to a second add-on circuit board with a supervisory microcomputer and support ICs. To understand the operation of the SJT-400 player (Fig. 1), one must first understand how the basic SJT-200 chassis operates, and then note the changes needed to upgrade its capabilities.

Abstract: *The microcomputer control system for the 1983 series of VideoDisc players is reviewed. The supervisory control system of the SJT-400 Random-Access VideoDisc player, composed of four major parts, is described. System architecture, on-screen display, remote control, and special interactive modes are explained in detail.*

Final manuscript received November 21, 1983.
Reprint RE-29-1-1

Fig. 1. *Interactive VideoDisc system composed of SJT-400 player and Apple® computer.*

Basic SJT-200 control blocks

The control system of the RCA SJT-200 VideoDisc player consists of three primary integrated circuits, associated peripheral circuitry, and three motors. The three primary integrated circuits are as follows:

1. The mechanism microcomputer
2. The player control microcomputer
3. The digital information buffer (DAXI buffer, see box on page 7)

The three motors are as follows:

1. The function motor—A dc motor that operates the player mechanics to automatically insert and remove the disc.
2. The arm motor—A stepper motor that positions the stylus over the play area of the disc.
3. The turntable motor—A brushless dc direct-drive motor that spins the disc at 450 rpm.

The operation of the basic player will be presented by describing the interaction of the above components.

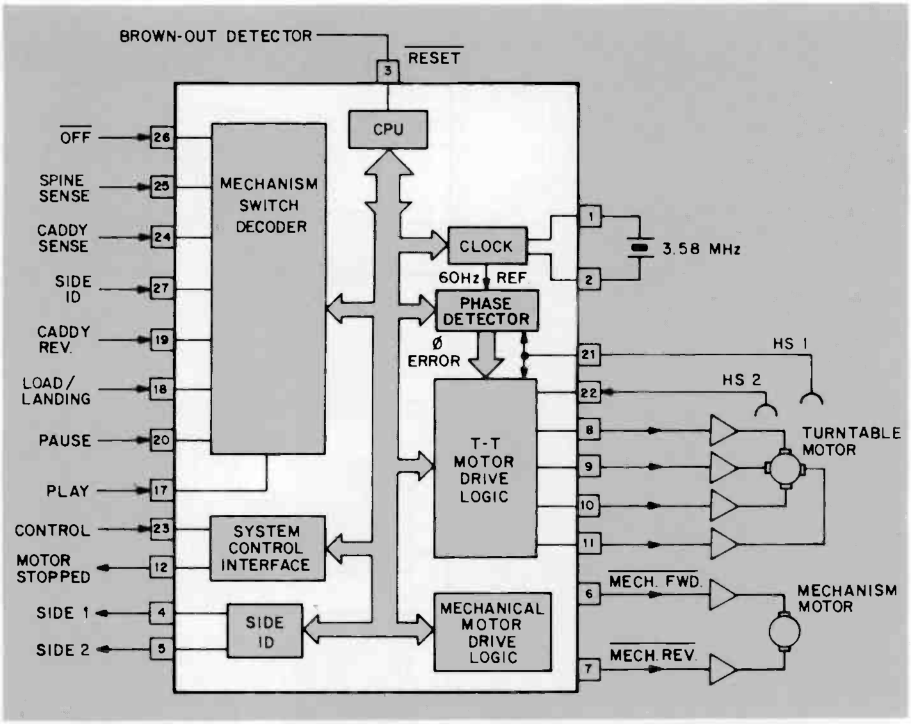


Fig. 2. *The mechanism microcomputer is packaged in a 28-pin carrier. It has 1024 bytes of instructional read-only memory (ROM), 128 nibbles of read/write memory (RAM), and 23 input/output ports.*

Mechanism microcomputer

The mechanism microcomputer, as the name implies, is predominantly responsible for controlling the various electromechanical devices in the player (Fig. 2). This microcomputer, like the others in the player, is a 4-bit commercially available device. Its responsibility starts during a caddy-load operation, when it controls the operation of pulling the disc caddy into the player, removing the disc from the caddy, rejecting the empty caddy, and lowering the disc onto the turntable platter.

With the disc in the normal play position, the mechanism microcomputer shuts off the function drive and activates the turntable drive to accelerate the disc to the operating speed of 450 rpm. The turntable drive is provided by a modern dc brushless motor. The mechanism microcomputer measures the velocity of the turntable by looking at two Hall-effect sensors (magnetically controlled electronic switches) mounted under the turntable motor magnet. Speed can be calculated from the timing of the Hall sensor outputs, while direction can be determined by the phasing of the signals. Electrical energy is transferred to the turntable platter through orthogonally mounted coils that interact with the permanent magnet rotor mounted to the turntable itself. The microcomputer controls the speed by selectively driving these motor coils with pulses of a calculated width, and controls the direction by phasing the timing of these drive pulses to the motor coils. When the speed is stable, the mechanism microcomputer signals the remaining microcomputers to begin the play cycle of the disc.

When the player microcomputer signals that the disc is to be stopped and removed from the player, the mechanism microcomputer reverses the phasing of the drive signals to stop the motor in the minimum time. Using the Hall-effect switches' information, the microcomputer stops the turntable in one of four precise locations where the disc can be raised for removal. After stopping the disc, the mechanism microcomputer commands the function motor to lift the disc off the turntable for reinsertion into the caddy.

In summary, the mechanism microcomputer provides the intelligence to bring the disc into the player, remove it from the caddy, and spin it up to 450 rpm for playing; it finally provides for stopping and removing the disc.

Mechanism microcomputer functions

- ☐ Operate and control the motor to pull the disc caddy into the player.
- ☐ Control disc lifting and lowering to turntable.
- ☐ Control dc brushless motor to spin disc at 450 rpm.
- ☐ Stop disc in one of four precise locations.
- ☐ Provide turntable reference signal to on-screen display to prevent picture rolls when switching stylus modes.

Revised player control microcomputer and DAXI buffer

The interaction and interdependence of the player control microcomputer and the DAXI buffer are so close that we will discuss them together. The player microcomputer is a 4-bit device with 2048 bytes of ROM and 96 nibbles of RAM (Fig. 3). The DAXI buffer is a custom IC that captures the digital control data in line 17 of every VideoDisc field.

Modifications to the SJT-200 control microcomputer were made to make this part usable for the SJT-400. Some functions that were better suited for the feature microcomputer were removed from the control microcomputer in the SJT-400. The additional ROM that was made available was used to communicate with the feature microcomputer, to provide single-groove *Freeze*, and to add an automatic stop operation for interactive disc applications. Wherever possible, the characteristics of the control functions of the SJT-400 player are identical to those of the nonfeatured players.

The control microcomputer has five basic modes of operation: *Load, Play, Pause, Scan,* and *Freeze.* Except for *Freeze,* each mode has other automatic and transitory modes associated with entering or exiting that mode. For example, the *Load* is always followed by a *Spinup* mode whereby the

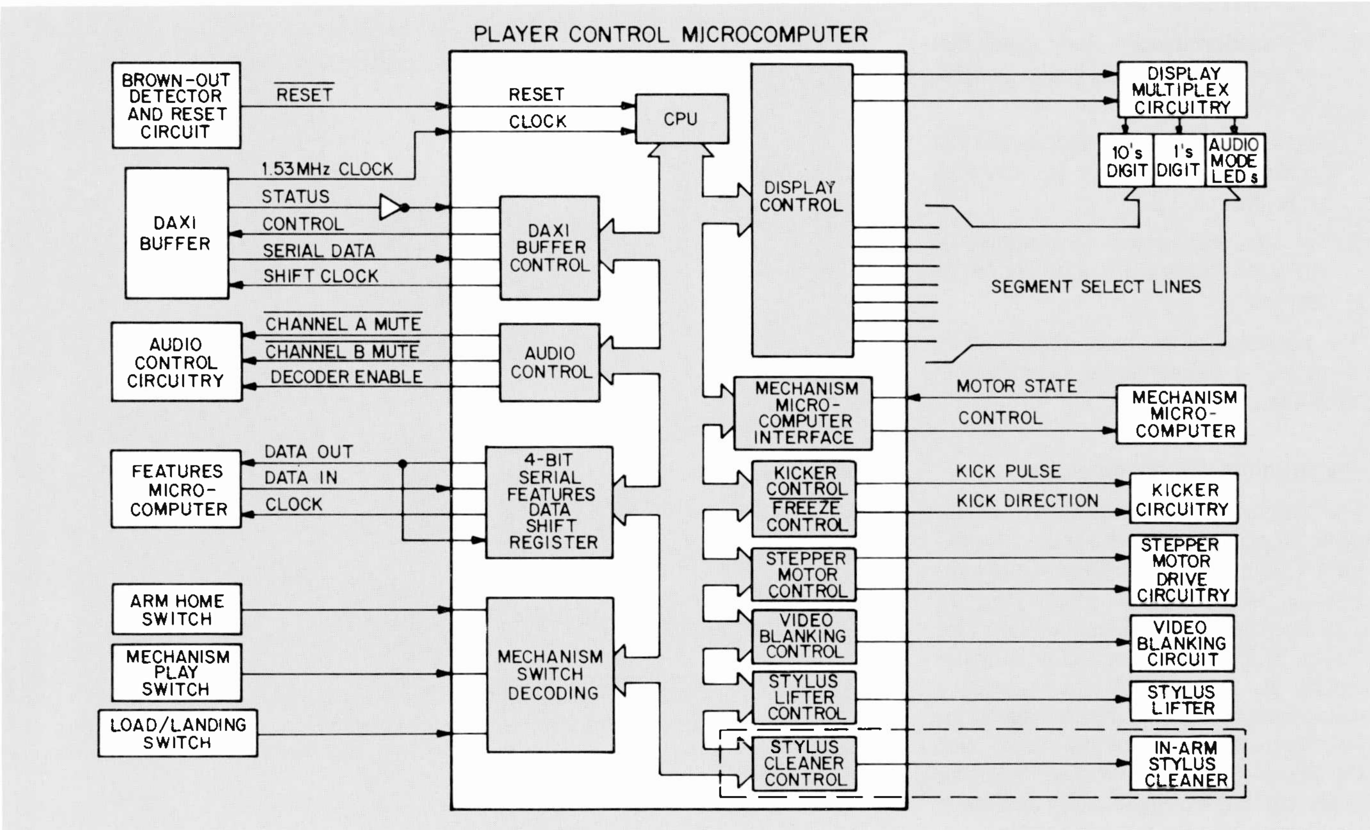


Fig. 3. *The player control microcomputer is packaged in a 42-pin carrier. It contains 2048 bytes of ROM, 128 nibbles of RAM, and 23 I/O pins.*

disc is lowered onto the turntable, the turntable comes up to speed, and the arm is positioned over the disc. In the *Load* mode, the player is waiting for a disc to be inserted. Once the disc is inserted, the control microcomputer automatically proceeds to the *Play* mode at the outer portion of the disc. The commands from the feature microcomputer are inhibited until the arm is over the playable area of the disc. When the feature commands are enabled, commands to enter any of the other modes may be executed. A return to the *Load* mode allows the current disc to be removed from the player.

During the normal play of the disc, the player microcomputer is constantly calculating the desired arm position to keep the stylus centered about the groove being played. It knows the stylus position by reading a digital code stored in the vertical interval of every field. This information is retrieved from the video signal by the DAXI buffer IC. When "enabled" by the player microcomputer, the DAXI buffer searches the video signal for a unique pattern of ones and zeros that is defined as the start code. It then captures the field, band, and flag information, checks it for errors, and signals the player microcomputer that it is available for use. The player microcomputer then transfers the binary data from the buffer, at its convenience, into its own memory via a serial link between the two ICs.

For the *Play, Scan,* and *Freeze* modes to operate properly, the control program must be synchronized with the DAXI code

Control microcomputer functions

- ☐ Communicates with the mechanism microcomputer, to load and unload the disc.
- ☐ Communicates with the feature microcomputer, to send control status and DAXI information and to receive commands and display data.
- ☐ Displays information received from the feature microcomputer on two LED 7-segment digits.
- ☐ Stores DAXI information (field number, band number, flags).
- ☐ Maintains synchronization with DAXI code and checks validity.
- ☐ Mutes appropriate audio channel(s) either automatically or when commanded by the features microcomputer.
- ☐ Blanks video when required.
- ☐ Controls the stepper motor to move the stylus carriage to the proper position.
- ☐ Generates appropriate kick pulses to the kicker coils, so that the stylus can scan across grooves or freeze in a single groove.
- ☐ Generates a signal to lift the stylus when required.
- ☐ Provides corrective measures for many improper player conditions.
- ☐ Handles *Stopbit* processing.

DAXI flags

All CED VideoDiscs have a 77-bit **D**igital **AuX**iliary **I**nformation (DAXI) code on horizontal line 17 of every field. This information is essential for maintaining the normal sequence of video in the presence of disc defects and for identifying the different disc formats and program content that may be present on a CED VideoDisc. Since every field is identified by a unique number in the DAXI code, nonsequential play of a VideoDisc in an interactive environment can be done quickly and accurately (see even- and odd-field DAXI information below).

The first thirteen bits of the DAXI code in each field identify the start of the data. The second thirteen bits are check bits. The field identification number is located in bits 54 through 71. Band number or system status (flags) are in bits 72 through 77 on alternate fields. Bits 27 through 53 are currently unassigned. Preprogram material is identified by bits 72 through 77, all zeros. Post-program material is identified by bits 72 through 77, all ones. During program-play material, bits 72 through 77 are defined to be the band number during even (DAXI) fields and to be system flags in the odd fields. In the current mastering specification, band numbers must be a minimum of ten seconds or 300 frames and no flag can change state more frequently than once in 24 frames.

The system flags define the status of the audio material and identify the occurrence of automatic stop zones on the disc. The audio may be mono, bilingual, or stereo.

Mono (bit 76 = 0, bit 75 = 0) is interpreted as audio on channel A only. Bilingual (bit 76 = 1, bit 75 = 0) means the disc has two independent audio channels. Stereo (bit 76 = 0, bit 75 = 1) has L + R on channel A and L − R on channel B. Audio noise suppression encoded, that is, CX encoded (bit 74 = 1) is identified by this flag. The *stop* zones are identified by the presence of a *stop* flag (bit 73 = 1) and are separated by the absence of the *stop* flag (bit 73 = 0). Playing a disc into a *stop* zone causes the SJT-400 player to repeatedly play the first full groove in the *stop* zone. The guard bit (bit 77) is used to guarantee that the system flags are not interpreted as pre-or post-program material. If any one of the other flag bits (72 through 76) is a logical one, then the guard bit is a logical zero. The guard bit is a logical one when all the other flag bits are zero. Currently, bit 72 is unassigned and will be a logical zero until it is assigned.

Even-field DAXI information:

Bit number	1 — 13	14 — 26	27 — 53	54 — 71	72 — 77
DAXI Code	Start Code	Check Bits	Unassigned Bits	Field Number	Band Number

Odd-field DAXI information:

Bit number	1 — 13	14 — 26	27 — 53	54 — 71	72 — 77
DAXI Code	Start Code	Check Bits	Unassigned Bits	Field Number	System Flags

on the disc. This synchronization is done in one of the transitory modes whenever the stylus is lowered onto the disc and

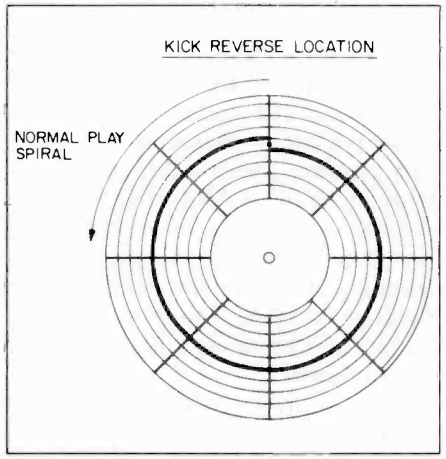


Fig. 4. *The CED VideoDisc stylus frozen in one groove.*

whenever there are sixteen consecutive fields of bad or missing DAXI code. Several safeguard measures are included in the *Play, Scan,* and *Freeze* modes to avoid undesirable player conditions such as scanning past band zero at the outer edge of the disc or the end band (band 63), staying in a locked groove, and continuing to play with debris on the stylus. The feature microcomputer may request the *Play* mode with or without the *Stopbit* processing enabled. *Stopbits* are flags in the DAXI code that can automatically cause the player to *Freeze* in a groove, presumably where the video is a still picture (Fig. 4). A *Stopbit* zone consists of consecutive frames with the *Stopbit* set. The zone is currently specified to be a *minimum* of 24 frames (or 48 fields, or 6 grooves). The recommended *Stop* zone is 48 frames of still picture video with the *Stopbit* set in frames 24 through 48 inclusive. Therefore, 2250 *Stop* zones can be stored on each side of a VideoDisc. When *Stopbit* processing is enabled, four consecutive frames with the *Stopbit* set cause the control microcomputer to go into the *Freeze* mode.

In the *Play* mode, the video progresses in the normal linear manner, in which the video was mastered on the disc.

The *Scan* modes provide nonlinear play of the video by moving the stylus in a radial direction at the time appropriate for the desired scan rate. The stylus is moved by electromagnetically kicking the stylus from groove to groove and moving the stylus arm carriage. The control microcomputer always kicks the stylus at the bottom of the video picture, so as to disturb the video as little as possible. Of the six different types of *Scan* modes, four represent two different speeds (16× and 128× normal) both forward and reverse. The other two types of *Scan* are used to go to the

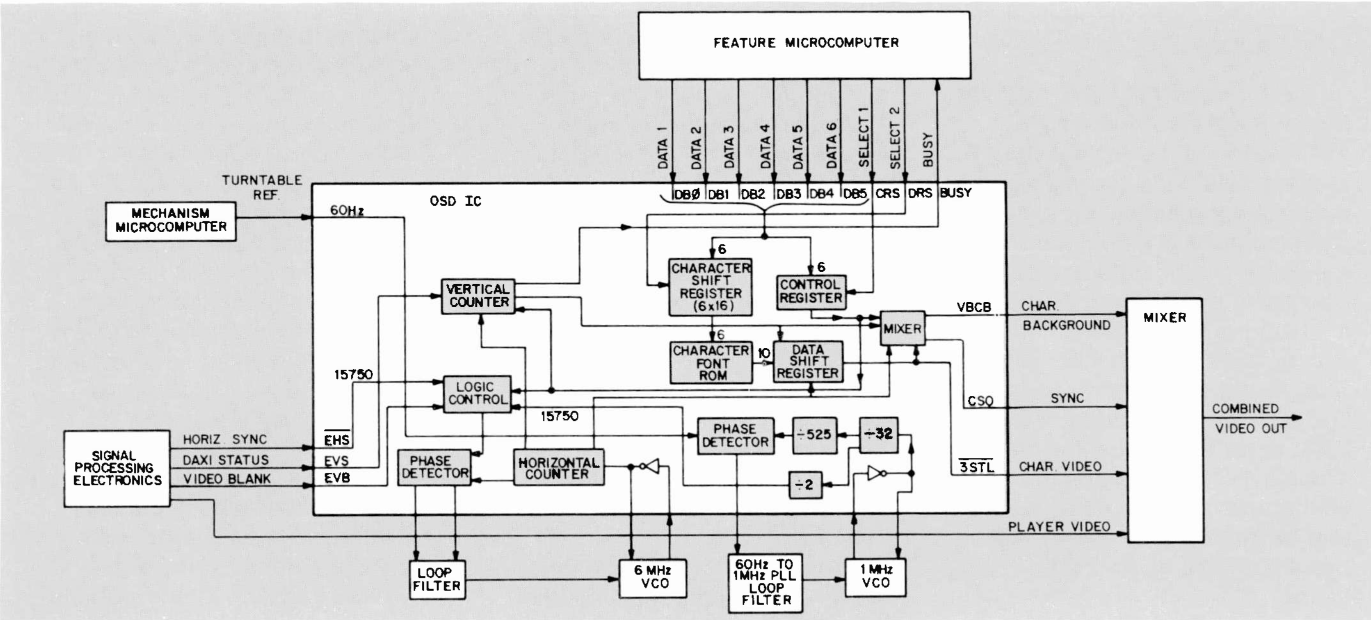


Fig. 5. *The on-screen display is an RCA-developed part that allows superimposing text on video. The part was designed to be used in both TV and VideoDisc applications.*

next or the previous *Stopbit* zone. To locate the next *Stopbit* zone, the video proceeds forward at the *Play* rate with the audio muted and the *Stopbit* processing enabled. The previous *Stopbit* zone is located by moving one groove in reverse during the even (band) DAXI fields with the audio muted until the beginning of the previous *Stopbit* zone is passed, and then proceeding into the *Stopbit* zone at the *Play* rate. Once the *Stopbit* processing detects a zone, the control microcomputer goes into the *Freeze* mode. In all six of the scans, the stylus remains on the disc, and video blanking is controlled by the feature microcomputer.

The *Pause* mode causes the stylus to be lifted from the disc surface. In this mode the arm carriage is not moved. For the most part, the control microcomputer is idling, waiting for the next feature command.

During the *Freeze* mode, the control microcomputer adapts the kick-pulse signal to the kicker coils to maintain a single-groove reverse kick, once per groove. The kick-pulse energy is also minimized whenever the location on the disc of the stylus kick is changed. The feature microcomputer can command an offset to the kick location at any time during this mode. This offset command is used when doing "plus" or "minus" a *Page*, or during *Page* seeks. A groove-reference flag is sent to the feature microcomputer to indicate when the last field of the groove or *Page* occurred.

Feature board control devices

As previously stated, no major changes to the SJT-200 player chassis were allowed in the design of the SJT-400 player. To get additional player features, an electronics board was added. This board provides the player with the capability of infrared (IR) remote control, computer interface, on-screen display of data and prompts, and direct random access to any time or band address on the VideoDisc. The three ICs added were the on-screen display, the feature microcomputer, and the remote keyboard microcomputer (RKM). Again, these parts will be explained separately.

On-screen display IC

The on-screen display (OSD) system was designed in a single integrated circuit (Fig. 5). It can display up to six lines of text with sixteen characters of text available on each line. A font of 64 selectable characters is stored in an internal ROM. Each character is shown in a 5 × 7 matrix format, 14 scanning lines high. For better visibility, each displayed character is set in a rectangular black background. The OSD receives coded alphanumeric data and control information from the feature microcomputer in the SJT-400 player. The OSD converts the coded data to video characters and uses the control information to set the appropriate display format. It receives horizontal and vertical timing from the video on the disc being played, and uses the timing to lock its output characters to the video signal recovered from the disc. This allows messages from the feature microcomputer to be superimposed on the video without any jitter. The OSD characters and video from the disc are combined in a mixing circuit external to the OSD IC.

In VideoDisc players without an OSD, sync and video are not available when the stylus is lifted off the disc. When the stylus of those players is lowered onto the disc, the vertical sync is often out of phase with its associated TV receiver. The picture on the TV receiver then exhibits a vertical roll. To display characters while the stylus is not on the disc and to prevent distracting vertical rolls, the OSD generates composite sync along with the characters and maintains vertical phasing information by using a vertical-rate reference signal from the mechanism microcomputer that drives the turntable.

As shown in the block diagram of Fig. 5, the 6-MHz dot clock is phase-locked to either 15.75-kHz horizontal sync (2100 pulses per disc revolution) or to a 15.75-kHz oscillator phased-locked to the turntable motor drive, depending upon the position of the stylus. In either case, the dot clock receives the same 15.75-kHz frequency and the vertical counter receives the same 31.5 kHz. This is how the vertical timing is maintained independently of stylus position.

Also, as shown in the block diagram, the DAXI status pulse, which occurs on line 17, is used to preset the vertical counter. The outputs of the horizontal and vertical counters are used to generate sync and character background, control the shifting of microcomputer data to the character font in order to select the appropriate 5 × 7 alphanumeric matrix, and control the shifting of the 5 × 7 matrix elements through the data shift register. The vertical counter sends a *Busy* signal to the feature microcomputer, which tells the microcomputer when new data can be strobed into the character shift register.

Although specifically designed for VideoDisc applications, the OSD IC is flexible enough to be used in other Consumer Electronics products, such as TV receivers and other video systems. This flexibility is obtained by building-in different operating modes and allowing selection in each application. The control register is used for this purpose. Under microcomputer control, it (1) controls the vertical position of the displayed messages, (2) selects a black outline around characters or a rectangular black background behind them, and (3) selects other operating modes.

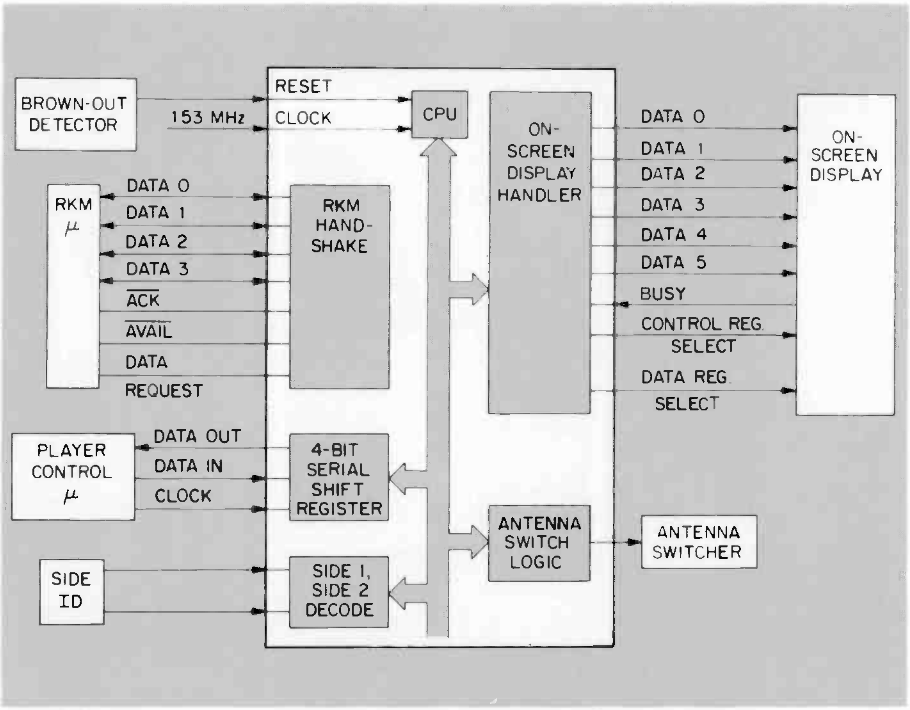


Fig. 6. *The feature microcomputer is packaged in a 42-pin carrier. It contains 4090 bytes of ROM, 192 nibbles of RAM, and 36 I/O ports.*

Feature microcomputer

The feature microcomputer is the master controlling element in the player (Fig. 6). It coordinates the work done by the other microcomputers and, if necessary, overrides their control actions. This computer chip is the largest and most powerful of this 4-bit series of ICs in the player. The feature microcomputer contains the logic that allows the player to search to a time (the user enters 4 digits of disc time in minutes and seconds), a band (the user enters 2 digits to select one of 62 bands on the disc), or a page (the user enters 6 digits of disc time in minutes, seconds, and field). The user may also program his player to customize the video to his own taste. With on-screen display prompts, the feature microcomputer guides the user in entering the programming to allow setting of a start time and a stop time of disc material to be displayed or in setting up a list of up to 5 bands for display. The logic to make this player "user friendly" and easy to operate required many iterations and most of the processing power of the microcomputer. At the start of this project, it was felt that there was adequate margin to complete all the forseeable tasks with this part. As happens in all computer projects, new requirements grow to exhaust any available excess capability, and the largest part of the programming effort was spent on code compression to accommodate these new requirements.

Only by examining the flowcharts of the program logic can one fully understand the details of the feature microcomputer's function. To simplify the explanation of the logic, with the exception of the initialization, microcomputer operation will be described by summarizing its processing during one field (1/60 second).

At power-up, the feature microcomputer clears its RAM, sets its flags, resets the on-screen display and loads the appropriate message, updates the LED digit displays and discrete annunciators on the player front panel, and initializes the communications exchanges between the RKM and player control microcomputers.

During each field, the microcomputer does a sequence of tasks: it first waits for an interrupt (a suspension of normal processing) that occurs every 1/60 second from the control microcomputer to indicate that new disc DAXI information is available. The feature microcomputer does a serial exchange, with the player control microcomputer using shift registers that are built into both parts. The feature microcomputer takes this new DAXI information from the player microcomputer and converts the field number from binary to base 60 for display. It then determines if band (even field) or DAXI flags (odd field) are available for processing (see DAXI flags box, page 7). Random-access seeks can take many fields. At the start of the seek, the seek type and termination data is set in

Feature microcomputer functions

- ☐ Supervise all other player microcomputers in operating the player.
- ☐ Communicate with the player control microcomputer, to receive control and DAXI information and send comands.
- ☐ Communicate with the RKM microcomputer, to receive local key, IR, or computer inputs and to send control and DAXI information.
- ☐ Update the on-screen display with player time, band, and display prompts.
- ☐ Update the LED indicator information.
- ☐ Convert binary time and band data to base 60 decimal.
- ☐ Check user inputs for validity and issue friendly prompts for inproper responses.
- ☐ Execute random-access commands to disc locations.
- ☐ Execute programmed operations.

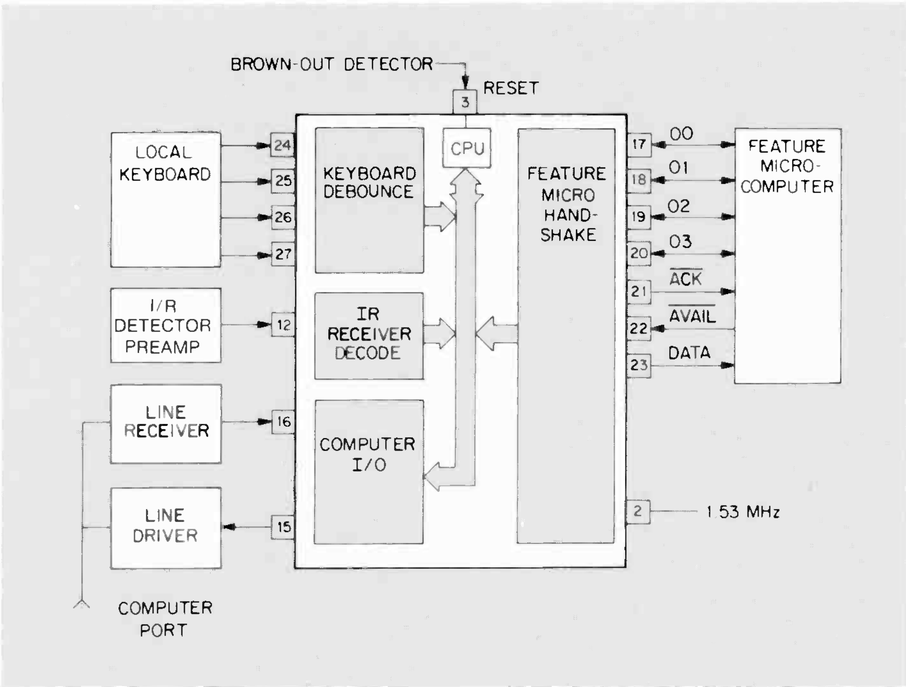


Fig. 7. *The RKM microcomputer is packaged in a 28-pin carrier. It has 2048 bytes of ROM, 128 nibbles of RAM, and 23 I/O ports.*

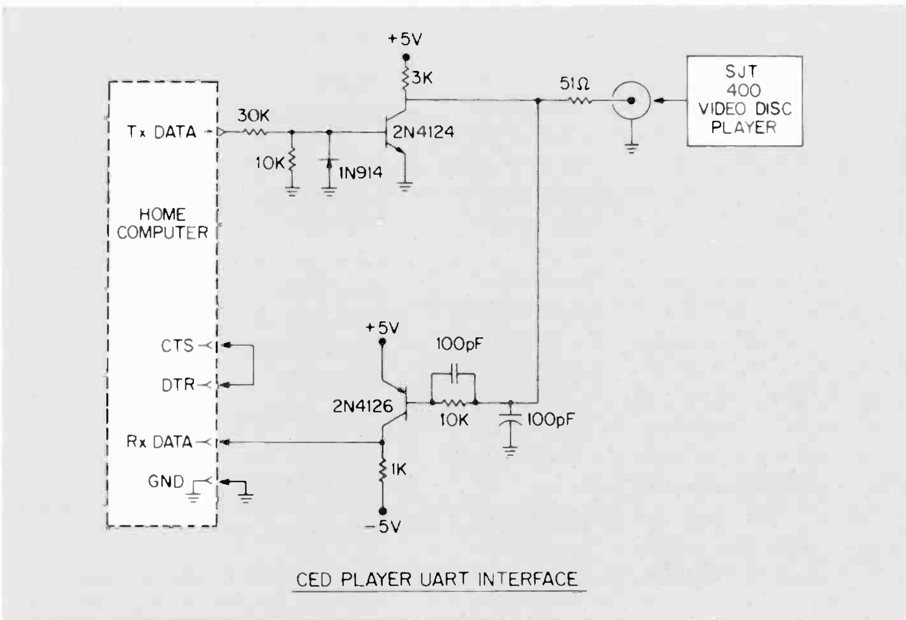


Fig. 8. *SJT-400 to RS-232 level shifting and demultiplexing schematic. This circuitry must be added between the player and the computer to convert from the bidirectional single-wire interface to the more conventional RS-232 interface found in most home computers.*

RAM and a timer is started. At each field following the recognition of a new DAXI code, the seek-termination data are examined. Depending on the distance to the target groove or band, the search speed and direction are commanded. When the new DAXI information indicates the target has been reached, the feature microcomputer tells the control microcomputer to stop the search and *play* or *freeze* depending on the type of random access. If the feature microcomputer has not found the target, then it checks the timer to determine if the search should be aborted. The timer is included, in the rare event that some disc defect prevents the completion of a search to target the disc location.

Having completed the disc search operation, the feature microcomputer next determines if the on-screen display is on and if it needs updating with new DAXI band and time information or a programming prompt.

The feature microcomputer now does its exchange with the Remote/Keyboard microcomputer (RKM). In this parallel exchange, the feature microcomputer sends time, band, and player flags and receives back external commands from the local keyboard, the IR remote, or the external computer.

With the new inputs from the RKM, the feature microcomputer is now able to enter its mode-dependent processing of managing the programming inputs, setting up RAM for the next micro-to-micro exchange, and determining the validity of both the data and command inputs. With these operations completed, the feature microcomputer waits for a new interrupt from the control microcomputer that signals arrival of new data and the passing of one field of time.

The IR remote, keyboard, microcomputer

Like all of the other microcomputers in this player, the RKM handles data in 4-bit nibbles. Although this microcomputer contains 2048 bytes of ROM, like the player microcomputer, it uses a 28-pin package since it has fewer I/O requirements. The RKM microcomputer is the I/O port for the SJT-400 player (Fig. 7). It receives player status from the feature microcomputer every 16.6 milliseconds and passes along commands that it receives from the outside world via the IR link or the hardware port. The RKM microcomputer is partitioned into two distinct and mutually exclusive subsections, the infrared remote (IR) handler, and the computer interface. In addition to these tasks, the RKM also monitors the keys mounted on the player's front panel (referred to as the local keys).

Many computer-controlled applications, such as advanced game applications, require a fast and comprehensive transfer of data from the player to an external computer. This is the reason why the high-speed computer interface was developed. The computer uses, as its interface, an RCA phono connector mounted on the player's back panel.

The RKM microcomputer, on powering up, monitors the computer port. If it fails to see activity on the bus, it reverts to IR mode. While in the IR mode, the microcomputer tests the hardware port periodically to see if there is an external computer pulling the bus low. Upon detecting a low level, the RKM begins counting to see how long the bus remains low. If it is low for a period greater than one half second, the microcomputer assumes that the bus is shorted and reverts to IR mode. It is assumed that most players will not have a computer attached, therefore, the failure states always cause the player to drop back into IR mode. A low that lasts longer than 160 milliseconds but less than one-half second forces the RKM into the computer-interface portion of its program.

The high-speed computer interface was designed to communicate with most low-cost personal computers. Routines have been written and interfaces built to allow communication with an Apple II, a Commodore Vic-20, an Atari 800, an IBM PC, and even a Timex-Siclair 1000 (Apple II, Commodore Vic-20, Atari 800, IBM PC, and Timex-Sinclair 1000 are registered trademarks).

To talk to an Apple II computer, the player requires a very simple hardware interface (Fig. 8). This interface handles the signal conversion from standard RS-232C to TTL levels and from a two-wire system to one that uses a single bidirectional coaxial cable.

Since the bus is bidirectional, some method must be used to prioritize the transmissions or bus contention will result. Because the RKM is locked with the feature microcomputer in a 16.6-millisecond cycle, the natural choice is to let the player dump its status and then listen for a command in reply from the controlling computer. Communication is bit serial, 9600 baud. The player transmits seven characters. Each is composed of a start bit, two 4-bit nibbles of data, and a stop bit (Fig. 9). It then opens a 4-millisecond "window" for the external computer to reply. Again, if after sixteen consecutive tries, there is no response, then the player reverts to the IR mode.

The data sent from the player once every 16.6 milliseconds include stylus-position information (time and/or band), mode (play, pause, scan), features information (seek-in-progress, repeat, etc.), and disc information (stereo, side, stop bit encountered).

As mentioned earlier, the RKM and feature microcomputers are locked together on a cycle that takes approximately 16.6 milliseconds, that is, video field rate (Fig. 10). From the viewpoint of the RKM microcomputer, time 0 is the point when the feature microcomputer asserts the /DATAVL line (Fig. 10). At this point, the transfer of thirteen data nibbles to the RKM begins (preamble is not one of those transferred). Upon completion, the RKM presents a command to the feature microcomputer. If no command exists to be executed, the RKM supplies a "no-key" code. There is, of course, appropriate "handshaking" during this time to ensure that the data are received correctly. Since this is a parallel data exchange, it occurs relatively quickly.

This entire process takes about one millisecond. At the end of this time, the serial exchange with the external computer begins. At 9600 baud, each bit is aproximately 104 microseconds long. Seven characters are sent, each ten bits in length, and one is received. Therefore, the total communication time requires about 8.3 milliseconds. This leaves about 7.3 milliseconds for the RKM to process the new command received so that the next exchange with the feature microcomputer will not be missed.

At irregular intervals along the way, calls are made to the local keyboard debounce routine to check for local key presses.

RKM microcomputer functions

- ☐ Scan and debounce local keyboard.
- ☐ Communicate with a computer.
- ☐ Scan the IR receiver for remote commands.
- ☐ Check all commands for validity and application, and then forward them to the feature microcomputer for action.

Infrared receiver (IRR) decode algorithm

The IRR infrared receiver portion of the RKM microcomputer is charged with the task of accurately decoding commands from the Digital Command Center (remote control hand unit), while rejecting IR background noise, yet maintaining continuity by "forgiving" minor interruptions of the IR transmission. This task had to be accomplished by careful balancing of system parameters, since the restrictions placed upon the goal seem to be in conflict.

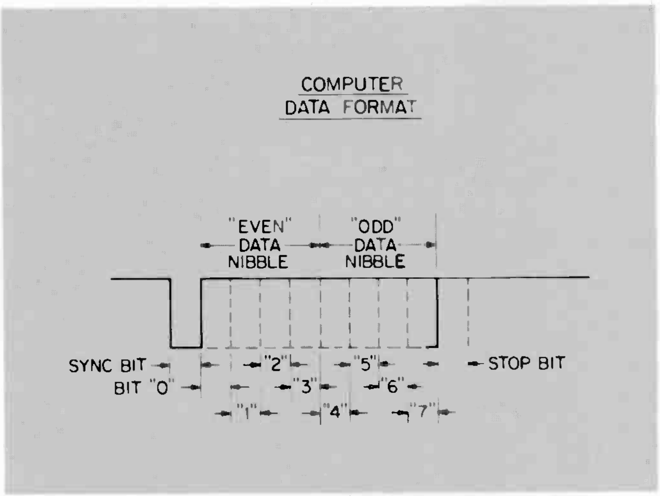


Fig. 9. *Computer data sent and received by the player use the above industry standard format. This format is easily understood by USART chips found in most home computers.*

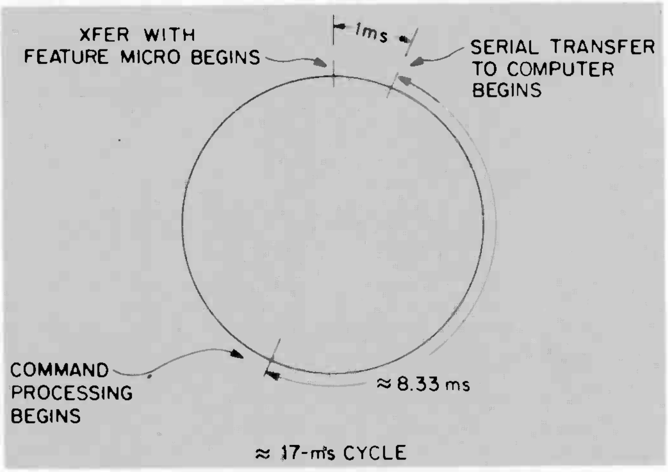


Fig. 10. *The RKM and feature microcomputer exchange information in every field. This adds delays between when external commands are sent and when they are actually executed.*

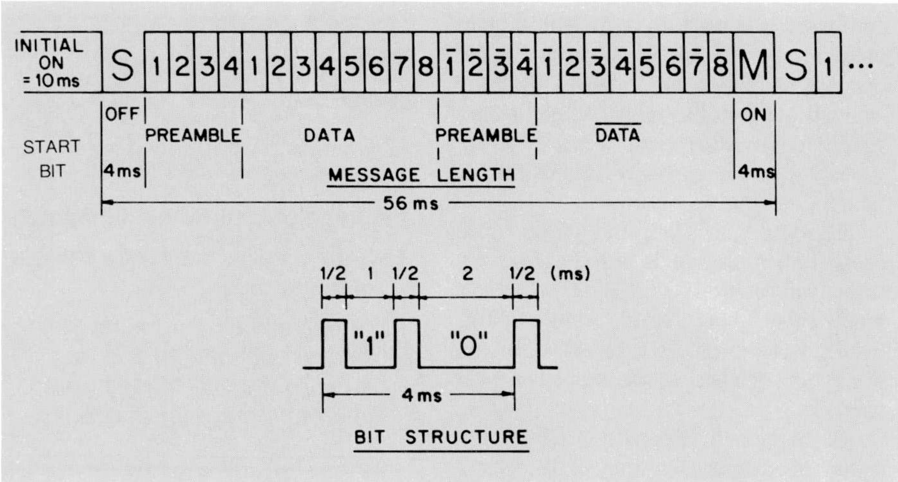


Fig. 11. *The remote transmission is composed of a mark bit, a space bit, twelve "true" data bits, and twelve "complemented" data bits. The time needed to transmit the data is 56 milliseconds. This code structure is used in TV, videotape, and VideoDisc products and makes possible the unified remote product that can control multiple devices.*

The RKM microcomputer polls the IRR input port every 0.5 milliseconds to see if an IR signal is present. During this time, it is also checking to see if a computer is trying to establish contact, and it is also monitoring the local keys on the player's front panel. If the software detects the presence of what might be an IRR command stream, it begins measuring the start bit time. As Fig. 11 shows, the start bit may be 4- to 10-milliseconds long. If it is too short, the timing phase is aborted. If it is too long, an immediate "keyup" instruction is passed to the feature microcomputer (this prevents sunlight from "latching" a scan command after the button is released).

Having accepted the start bit, the program abandons computer mode until after the IRR transmission is received and processed. The RKM then polls the IRR port on a 0.25-millisecond interval, devoting itself full-time to the task.

The space also has to be within certain limits. Too short a space causes return to "mark" detection. Too long a space causes a return to standby. If both mark and space are correct, we proceed to the IRR bit decoding.

IR bit decoding

Since a "zero-bit" is longer than a "one-bit," a method was chosen to increment a count while monitoring the IRR input port. In addition, the code was written to compensate for differences in pulse widths from various remote transmitters by performing a timing phase correction continually. After receiving the 24-bit message, the first four bits of the "true" and "complemented" portions are checked to see if they are equal to "2," the preamble assigned to the VideoDisc player. If the preamble is not correct, then the remainder of the message is ignored. Having found a proper preamble, the program then compares the "true" and "complemented" data to see if they are equivalent. If they are not correct, the program will assume that the player is performing the desired function and will pass "continue function" code to the feature microcomputer. The "bad read" flag is incremented when this occurs and only seven consecutive "bad reads" will be tolerated. On reception of the eighth one, a key-up code is generated and passed to the feature microcomputer. This "forgiveness" code is necessary to correct for quick interruptions of the beam (for example, a fly darting in front of the transmitter).

Once the command is received and validated, it is then checked to see that it is a legal code for the VideoDisc player and it is sent to the feature microcomputer. Input polling time is now reset to 0.5 milliseconds (normal check for IRR signal time) and the program starts looking for the next IR transmission.

Conclusion

A control architecture consisting of multiple microcomputers can be developed to allow construction of a higher-cost "feature player," without burdening the high-volume low-end player design with additional parts and their associated cost. This type of architecture provides for a cost-effective family of players whose performance can be enhanced by simply adding more computer power.

The SJT-400 adds a new dimension in player capability and performance not previously found in the CED system. The SJT-400 allows the viewer the option to interact with disc programming. Specially prepared software, such as a mystery disc with multiple endings, can provide a new form of entertainment not previously available in the CED format. The computer interface allows the player to be used in industrial application—one example is an arcade football game.

Authors (left to right) Power, Lenihan, Chen, Gibson, and Auerbach.

Authors Kelleher (left) and Kiser.

Walter Gibson is a Senior Member of the Technical Staff, in the VideoDisc Player Control Group at the Laboratories. He joined RCA in 1946. He was involved in the early stages of color TV development. Recently he has been engaged in VideoDisc Systems research. He has received three RCA Laboratories Outstanding Achievement Awards and the David Sarnoff Award for Outstanding Technical Achievement. He holds 21 U.S. patents and has published 5 technical papers.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-3038

Kevin Kelleher is a Senior Member of the VideoDisc player engineering staff. He received a BSEE degree from Iowa State University in 1973 and worked at Motorola in CCTV camera design. He joined RCA in 1979 after receiving his MSEE from Illinois Institute of Technology. His major responsibility is design of the player interface between electronic and mechanical systems.
Contact him at:
"SelectaVision" VideoDisc Operations
Indianapolis, Ind.
TACNET: 426-3289

Ned Kiser is currently Manager, Digital Control Systems, VideoDisc Player Engineering. He received a BSEE degree from Iowa State University in 1976 and joined RCA in 1979. He has been involved in both the signal processing and control system aspects of VideoDisc player design. In his present position, he is responsible for all VideoDisc player applications of microcomputers and microprocessors.
Contact him at:
"SelectaVision" VideoDisc Operations
Indianapolis, Ind.
TACNET: 426-3291

James Power is Group Head of the VideoDisc Player Control Group at RCA Laboratories. He received his BS and MS in Electrical Engineering from UCLA in 1971 and 1973 respectively.

He joined RCA Astro in 1973 in the Advanced Development Group and worked on the application of microprocessors in spacecraft systems. He joined RCA Laboratories in 1978 in the Microprocessor Application Group. Since then he has worked on microprocessor designs of TV remote controls and VideoDisc control systems. He received the RCA Laboratories Achievement Award, as a team member, for the development of software for the VideoDisc player.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2726

Thomas Chen graduated from Purdue University in December 1973 with a BS degree in Electrical Engineering. He received his MSEE degree in 1976 from Rutgers University. He joined RCA Laboratories in 1973 in the systems research group. From 1974 to 1979 he worked in the LSI systems design group of the RCA Solid State Technology Center, in the area of microprocessor system development.

Since 1979, he has been working in the VideoDisc Player Control Research Group of RCA Laboratories. He has been involved in the development of an integrated circuit for either video character generation or character overlay on video. More recently, he has been developing control software for advanced multi-microprocessor-based VideoDisc players. In 1980, as a member of this group, he received the RCA Laboratories Achievement Award.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2175

Victor Auerbach received a BEE degree in 1956 from the City College of New York and a MSEE degree from University of Southern California in 1958. He joined RCA Astro-Electronics in 1958 and was involved with numerous digital designs of spacecraft systems.

He joined RCA Laboratories in 1979 as a Member of the Technical Staff in the VideoDisc Player Research Group. Since then he has developed software for IR control of the player and other new player control systems.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2981

Thomas Lenihan joined RCA Laboratories as a Research Technician in August 1975. He has worked as a member of the Microprocessor Research Group progressing from RT to STA. In May 1981, he graduated from La Salle College in Philadelphia with a Bachelor of Science degree in Electronic Physics. He transferred to the VideoDisc Player Control Research Group in October 1981 as an AMTS.

He has been responsible for implementing the computer portions of the RKM microprocessor's code, and for interfacing several home computers to the VideoDisc player. As a member of the Microprocessor Research Group, he received the RCA Laboratories Achievement Award and a U.S. patent.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-3218

Arthur Greenberg graduated from Drexel University in 1977. He joined RCA Laboratories in September of 1978. He received the RCA Laboratories Achievement Award, as a team member, for the development of software for the VideoDisc player. Mr. Greenberg is no longer with RCA.

G.W. Mebus | E.J. Braude

Computer languages—A view from the top

After just thirty years, hundreds of computer languages populate a modern "Tower of Babel." The important ones are giving us better insight into computation and better ways to accomplish it.

Abstract: *This overview of computer programming languages discusses "language" in the relatively new context of "computing." The article gives the origins of a variety of computer languages, their development, and their special characteristics. It shows how differing language requirements have led to markedly different languages. It also points to new directions languages are taking, and the capabilities they will provide.*

Final manuscript received November 16, 1983.
Reprint RE-29-1-2

Languages allow humans to communicate. Computer languages allow humans to get computers to do their bidding. We can get a computer to do a tremendous variety of tasks such as tracking inventories, computing payrolls and printing the checks, communicating with and controlling spacecraft near distant planets, providing video games, and simulating human intelligence.

A simple example of a man/machine interface is a radio. The knobs on a radio set constitute a "language" (a means of communication) that gets a radio to do our bidding; but we do not write articles about that "language" because the variety of things a radio can perform is many orders of magnitude smaller than the variety of tasks a computer can perform.

By comparison, a calculator's "language" (pushbuttons) is significantly more complex than the radio's because there are more instructions the calculator responds to, and sequences of instructions now become important. People do write articles and even books on how to use calculator "languages," simple as they are.

The potential of modern computers—with hundreds of instruction types, the ability to make decisions, and high operating speeds—is staggering. There must be means for communication with computers that are commensurate with the sophistication of both the computer and its intended applications. This is the genesis of computer languages.

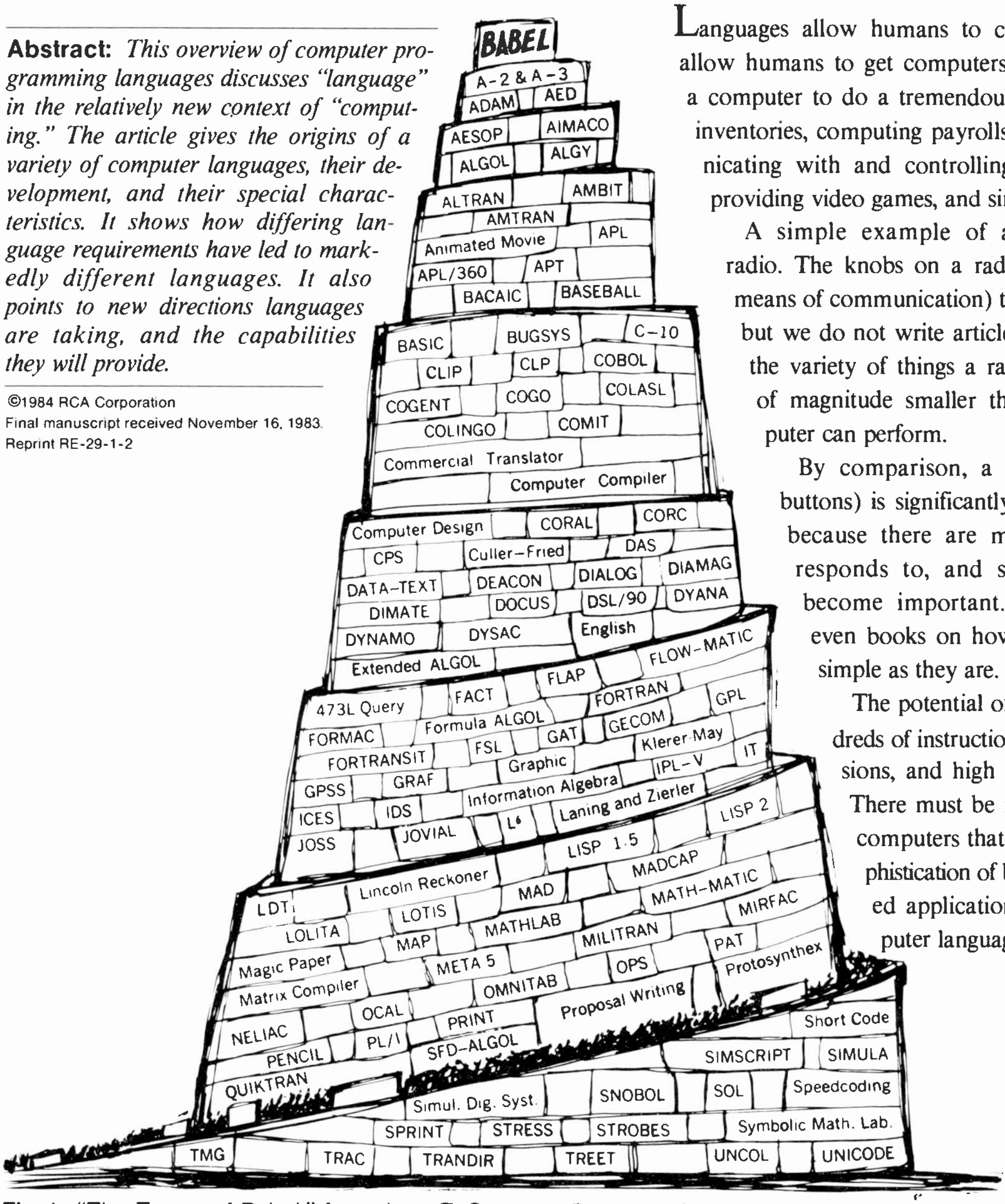


Fig. 1. "The Tower of Babel," from Jean E. Sammet, *Programming Languages: History and Fundamentals,*[1] © 1969, front endpaper. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey. The tower has become much higher since 1969.

Assembled Instructions	S	AA	I	F	C	Line No.	Loc	OP	ADDRESS	Remarks
						01		ORIG	1000	Load at location 1000
1000:	+	1013	0	2	32	02	GCD	STJ	EXIT	Subroutine linkage
1001:	+	1	0	5	15	03		LDX	1	Load x value into rX
1002:	+	2	0	5	63	04	LOOP	CMPX	2	Compare x to y
1003:	+	1013	0	5	39	05		JE	EXIT	Exit if equal
1004:	+	1009	0	4	39	06		JL	SWAP	Else, if x<y swap them
1005:	+	1	0	1	55	07		DECX	1	Otherwise decrement x
1006:	+	0	0	2	48	08		ENTA	0	Clear rA (upper digits)
1007:	+	2	0	5	04	09		DIV	2	Get remainder in rX
1008:	+	1	0	0	55	10		INCX	1	Increment remainder
1009:	+	2	0	5	08	11	SWAP	LDA	2	Load y value into rA
1010:	+	1	0	5	24	12		STA	1	Store into x location
1011:	+	2	0	5	31	13		STX	2	Store new y value
1012:	+	1002	0	0	39	14		JMP	LOOP	Repeat the process
1013:	+	1013	0	0	39	15	EXIT	JMP	*	Return to where called
						16		END	GCD	End of program

Fig. 2. Greatest common divisor (GCD) algorithm in MIX assembly language. MIX is a fictitious computer developed by Donald Knuth,[2] for teaching programming and algorithms. While the assembly code (right side) is detailed and difficult, the machine code (left side) is much less readable. It would be even worse if displayed in internal form of 41 bits per instruction word.

Tower of Babel

Since hundreds of computer languages exist, some have perceived the situation as a modern "Tower of Babel" (Fig. 1). This is due, primarily, to the many separate application domains for these languages. For each application, some languages are highly expressive, while others are abysmally inadequate. This diversity also arises from technological advances in both computer hardware and language technology.

Avoiding 0's and 1's

In theory, any computer program can be implemented by stocking the computer's memory with 0's and 1's. This approach is impractical, however, because it is an unnatural language for humans; it's extremely tedious and error-prone. A language with some words rather than "bits" (binary digits, that is, 0's and 1's) is clearly needed.

The simplest way to introduce words is to write the name of the instruction rather than the code that the computer wants to see. Names can also be applied to the locations of data in memory and the positions of instructions in the program, and decimal numbers can be written instead of some other bit strings. With this simple language, the programmer has a much easier time writing, reading, and understanding the program. In fact, this same language could be used to write programs for different kinds of machines, computers that have a similar complement of instructions in their repertoire, but that have different bit codes assigned to them (that is, an ADD instruction may be defined as 000000 on one machine and as 11000110 on another). The only problem is that this form of the program must still be translated into the 0's and 1's that the computer understands.

Assembly languages

Determining the bit values for the names used above is necessary but still tedious and error-prone. The very mechanical nature of the process, however, makes it an ideal application for a computer! This computerized translation process is called "assembly," the translating program is called an assembler, and the language we have developed is called an assembly language. An example of an assembly program and the corresponding machine codes is shown in Fig. 2.

```
GCD(X,Y)
  DO UNTIL X=Y
    IF X>Y
      THEN replace X by X-Y
      ELSE exchange X and Y
    ENDIF
  ENDDO
  answer is X
```

Fig. 3. A PDL description of the GCD algorithm. This is not a program, but a "structured English" description. The DO-ENDDO and IF-THEN-ELSE-ENDIF constructions follow the discipline called "structured programming."

An example for comparison

We offer a flavor of various languages by writing a "greatest common denominator" (GCD) program in each. The greatest common denominator of two given positive integers is the largest integer that divides both. For example, the GCD of 8 and 20 is 4, and the GCD of 8 and 21 is 1. The classical method (algorithm) for finding the GCD of two integers is to replace the larger by the difference between the two numbers. If the two numbers are now equal, then their mutual value is the GCD, otherwise, repeat the process.

Program Design Language

In a Program Design Language (PDL), used to describe algorithms, the GCD process could be expressed as shown in Fig. 3. A PDL is not a programming language. It is a language that communicates program designs between people. We will see how programming languages convey this information to computers.

An improved example

As a point of professional responsibility, let us briefly consider the efficiency of the GCD algorithm. If the sizes of the two numbers differ greatly, time could be wasted with numerous subtractions. The intent of the process is to find the remainder. Fig. 4 thus uses $((X - 1) \bmod Y) + 1$, instead of subtraction, to speed execution. This is essentially the modulo (remainder, residue) function modified to produce Y instead of 0 whenever Y exactly divides X. In that way we will still consider only positive integers.

Really useful programs should also determine whether the input data is proper and give appropriate informative error messages if it is not. These examples are only given for illustration; they would have to undergo rigorous analysis and testing for correctness before being considered reliable software products.

Widely used first-generation languages

FORTRAN

Assembly language programming requires the programmer to know the exact makeup (the "architecture") of the machine. The first attempt to give programmers the ability to write programs without requiring detailed machine knowledge was FORTRAN (FORmula TRANslator)[3]. Because we now take FORTRAN for granted, few realize how big a step it was. The first few years of frustrations in its development convinced many that such automatic translation could not be accomplished. Obviously, it finally was successful. The translation process is called "compilation," and the program that does the translation is called a compiler.

```
GCD(X,Y)
  DO UNTIL X=Y
    IF X>Y
      THEN replace X by ((X-1)mod Y) + 1
    ENDIF
    exchange X and Y
  ENDDO
  answer is X
```

Fig. 4. A second PDL description. The use of ((*X*−*1*) *mod Y*)+*1* instead of *X*−*Y* greatly reduces the number of "DO loops" performed, thus, speeding the algorithm, often substantially.

```
      INTEGER FUNCTION GCD(IX,IY)
123   IF (IX.EQ.IY) GOTO 987
      IF (IX.GT.IY) IX = IX-IY*(IX-1)/IY
      ITEMP = IY
      IY = IX
      IX = ITEMP
      GOTO 123
987   GCD = IX
      RETURN
      END
```

Fig. 5. A FORTRAN IV version of the GCD algorithm. Note that in FORTRAN **(a)** names beginning with a letter from I to N indicate integers; **(b)** ".EQ." is a test for equality; **(c)** "=" is used to assign values; **(d)** line labels look like numbers (but aren't: GOTO 3*41 cannot be used instead of GOTO 123).

```
100  DEF  FNG(X,Y)
110     IF X = Y THEN 180
120     IF X < Y THEN 140
130     LET X = X-Y*INT((X-1)/Y)
140     LET Z = Y
150     LET Y = X
160     LET X = Z
170     GO TO 110
180     LET FNG = X
190  FNEND
```

Fig. 6. A BASIC version of the GCD algorithm. In BASIC, every line has a label number and "=" is used for both equality tests and assignment. Only 26 functions can be defined (named FNA through FNZ).

The FORTRAN language allows programmers to write algebraic expressions in nearly natural form, rather than in sequences of one-word commands. Due to the limitations of card punches and printers the symbols *, / and ** were adopted for multiply, divide and power operations—conventions held by most subsequent languages. The IF and DO statements are closer to natural descriptions of decisions than assembly language can provide. Arrays and tables of numbers are established by declaring them, and their elements can be changed and accessed by specifying their indices. A variety of types of numbers (integer, real, double precision, complex) can be declared; appropriate arithmetic rules for their computation are automatically enforced. Language features for specifying the formats of listings were also incorporated.

The compiler program produces machine instructions for transferring control and data to and from "subroutines," separate pieces of software that can be used several times at various places in a program. A statement in FORTRAN (and any other compiled language) typically generates several machine instructions, unlike assembly languages that typically are one-for-one. But best of all, the programmer does not need to know the internal structure of the computer to write a computer program—the compiler handles that automatically. Fig. 5 shows the GCD algorithm in FORTRAN IV. Subsequent updatings of FORTRAN,[4] especially FORTRAN 77, have added contemporary features that we will see in other languages.

BASIC

A direct descendent of FORTRAN is BASIC (Beginner's All-purpose Symbolic Instruction Code)[5] originally intended as a simple introduction to FORTRAN. The number of constructs and capabilities are reduced, but the important features remain. Its adoption as a standard language for home computers illustrates the success of these simplifications. A BASIC version of our GCD program is shown in Fig. 6.

COBOL

COBOL (COmmon Business Oriented Language)[6] is another language that grew up with FORTRAN. As an historical note, the two companies that first implemented COBOL were Remington Rand and RCA. Its designers wanted to provide a self-documenting language that managers would be able to read easily. COBOL's acceptance by the business community has been enormous. Many feel that COBOL's wide acceptance is due to early Government regulations mandating COBOL. Its readability stems from its nearly exclusive use of English rather than algebraic symbols. For example, in COBOL we may say

ADD APPLES TO BANANAS GIVING PRODUCE

rather than

PRODUCE = APPLES + BANANAS

as in FORTRAN. Many special facilities for generating reports, handling large amounts of data in record formats appropriate for business, and data sorting and searching are embedded in COBOL. Figure 7 shows COBOL code. Evidently, COBOL is inappropriate for our GCD example.

Languages designed for special applications

ATLAS

Some programming languages seem to serve the purpose that jargon does in natural languages. In fact, specialized applications often have their own programming languages with embedded words and phrases peculiar to the discipline. There are languages for electrical circuit design, for civil engineering, and many other restricted disciplines. ATLAS[7] is a standard language for directing automated test equipment (ATE). "ATLAS" originally meant Abbreviated Test Language for Avionics Systems. The "Avionics" later became "All" in recognition of wider applications. Like COBOL, ATLAS leans heavily on English to provide good documentation; the programs can instruct manual testing as well as computer-driven testing. And, like COBOL, ATLAS is remarkably inappropriate for the GCD algorithm. Figure 8 shows an ATLAS test procedure named GAIN CHECK.

```
IDENTIFICATION SECTION.
PROGRAM-ID.  EXAMPLE.
REMARKS.   THIS IS WISPS OF COBOL, FOR FLAVOR ONLY.

ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
    SOURCE-COMPUTER.  VAX-11/780.
    OBJECT-COMPUTER.  VAX-11/780.
   INPUT-OUTPUT SECTION.
    FILE CONTROL.  SELECT FIRST-FILE ASSIGN TO '2020' UTILITY.
       .
       .
       .
DATA DIVISION.
  FILE SECTION.
    FD FIRST-FILE BLOCK CONTAINS 4 RECORDS.
    RECORDING MODE IS ...
       .
       .
    01  INPUT-RECORD.
       02  FIELD-1   PICTURE 9 (20).
       .
       .
       .
PROCEDURE DIVISION.
  OPEN INPUT FIRST-FILE ...
       .
       .
       .
  CLOSE FIRST-FILE.  STOP RUN.

  RECORD-SELECTION SECTION.
    PARAGRAPH-1.  READ FIRST-FILE AT END GO TO PARAGRAPH-2.
       IF FIELD-1 = FIELD-2 GO TO PARAGRAPH-1 ELSE ...
       .
       .
       .
  EXIT.
```

Fig. 7. An example of COBOL form and style. A complete COBOL program, even a trivial one, takes lots of space. This sketch indicates the concerns of a COBOL program (for example, the Environment Division identifies all the hardware to be used) as well as its readability and verbosity.

```
220000 DEFINE,
       'GAIN CHECK',PROCEDURE,('DC-IN','OUT-HI',
       'OUT-LO',UP LIM','LOW LIM','FAIL EXIT')
       RESULT('GAIN')$
    01 DECLARE,DECIMAL,STORE,,'DC-IN','UP LIM',
       'LOW LIM','LOCAL','GAIN'$
    02 DECLARE,STP,STORE,'FAIL EXIT'$
    03 DECLARE,CONN,STORE,'OUT-HI','OUT-LO',6 CHAR $
    10 APPLY,DC SIGNAL,VOLTAGE'DC-IN' V RANGE 1V
       TO 10V BY 0.1V,CNX HI J1-8 LO J2-9 $
    12 MEASURE,(VOLTAGE ERRLIM +-.001V INTO'LOCAL'),
       DC SIGNAL,VOLTAGE MAX 10V,CNX HI J1-8 LO J2-9 $
    14 MEASURE,(VOLTAGE ERRLIM +-.001V),DC SIGNAL,
       VOLTAGE MAX 100V,CNX HI 'OUT-HI' LO 'OUT-LO'$
    15 CALCULATE,'GAIN'='MEASUREMENT'/'LOCAL'$
    16 COMPARE,'GAIN',UL 'UP LIM' LL 'LOW LIM'$
    18 GO TO,STEP 'FAIL EXIT' IF NOGO $
    20 END,'GAIN CHECK'$
```

Fig. 8. An example of ATLAS coding. The readability of ATLAS is apparent, as is its limited domain of application. From *ANSI/IEEE Std. 416-1978:7,* © 1978, pp. 6-22, 23. Reprinted by permission of The Institute of Electrical and Electronics Engineers, Inc. New York, New York.

Languages that reduce their worlds to mathematical forms

APL

In 1962 Kenneth Iverson published a book named "A Programming Language"[8] that proposed a new mathematical notation, a kind of algebra, for describing computations. Although it looked strangely hieroglyphic, this notation combined similar concepts from diverse areas of mathematics under a small set of new symbols. It was given an unusually simple and consistent set of "grammar" rules. A major contribution of this language was to allow treatment of multi-dimensional arrays of data as single data items. There were symbols for rotating, reversing, and transposing arrays; selecting subarrays; and evaluating polynomials from an array of coefficients. Even change of control (branching) became a data manipulation, taking an integer to be the next program line number to execute. The use of arrays removed much need for branching and looping in computations. "Iverson's notation" was used for a variety of processing description applications. For example, it was used to describe the operation of significant parts of the IBM/360 computers.

In 1968 IBM released a commercial software product named APL\360. Although based on the notation of Iverson's book, the language was even simpler, more consistent and more powerful. APL is interpreted, not compiled. It is executed, symbol-by-symbol, from right to left.

APL implementations are interactive; one uses the system as a powerful algebraic calculator. When an expression is typed, the system responds with the value of the expression. Data is created and recreated dynamically (that is, when the program is executing). APL's concise notation and interactive support system allow programs to be designed, coded and debugged in one-fifth to one-half the time other languages typically require. It is excellent for algorithm development and experimentation, for one-time computations and modelling.[9,10]

Figure 9a shows an APL implementation of the GCD algorithm, and Fig. 9b shows a modified version able to handle arbitrary arrays of *X* and *Y* data, producing a similar array as a result. But a different solution (more in the spirit of APL) has been proposed[11]: embed the GCD function as a primitive of the language. The OR function (symbolized by the inverted caret) currently takes 0's and 1's as arguments and returns 0's and 1's as results. The domains could be compatibly expanded to be the integers without affecting the operation for bit values. Thus $8 \vee 20$ would produce a 4 and $8 \vee 21$ would produce a 1. Similarly, the AND function (caret) could become the Least Common Multiple function.

A revised example

You may notice that in the APL examples the defined functions named GCD invoked *themselves* on their last lines. This practice is called "recursion" and was not available in languages discussed previously. Modern programming languages generally allow recursion as another means of simplifying the program statements, so let us revise our PDL description of the GCD algorithm to reflect this capability. Figure 10 shows a recursive form of the GCD.

Languages that embody modern software engineering principles

The modern notion of software engineering is based largely on the ideas of Edsgar Dijkstra[12] and others. Dijkstra developed a method for writing reliable programs ("reliable" means that the program always performs according to its prespecified behavior). The most important software engineering principle is "top-down" programming. Briefly stated, the top-down approach is to break the problem into a manageable number of sub-problems, and so on, until the sub-sub-problems can be conveniently written *in toto.* The resulting software is composed of relatively small "modules" which tend to be self-contained and independent. Each module is easier to understand, write and change than one huge program containing everything. Several teams of programmers can work simultaneously on different parts of the software system.

```
        ∇Z←X GCD Y
[1]      Z←X
[2]      →(X=Y)/0
[3]      →(X>Y)↓L1
[4]      X←1+Y|X-1
[5]     L1:Z←Y GCD X
        ∇
```

(a)

```
        ∇Z←X GCD Y
[1]      Z←X
[2]      →(0∊X=Y)↓0
[3]      Z←X⌊Y
[4]      Z←Z GCD 1+Z|(X⌈Y)-1
        ∇
```

(b)

Fig. 9. Two APL versions of the GCD algorithm. Each strange symbol stands for a built-in operation of the language. **(a)** handles single values for *X* and *Y*, as do the other languages. **(b)** could accept, say, 17-dimensional arrays of integers for *X* and *Y* values. The APL approach is as different as its appearance.

```
GCD(X,Y)
  IF X=Y
    THEN answer is X
    ELSE
      IF X>Y
        THEN answer is GCD(Y,((X-1)mod Y)+1)
        ELSE answer is GCD(Y,X)
      ENDIF
  ENDIF
```

Fig. 10. A PDL description of the recursive GCD algorithm. An algorithm defined in terms of itself can often avoid looping (note that DO-ENDDO is gone). Some algorithms are naturally recursive. For example, *"0! = 1,* and *n! = n*(n−1)!* for *n>0"* fully describes the factorial function.

ALGOL

This top-down principle has dictated the basic structure of ALGOL (the ALGOrithmic Language)[13] by requiring a much richer control structure capability. Unlike prior programming languages we have seen above, ALGOL did not rely on GO TO instructions to change the order of executing the program statements, but used two revolutionary constructions:

if <condition> then <action> else <action>

and

for <variable> = <expression> while <condition>
do <action>

where the items in the <> brackets stand for any appropriate names, expressions or statements in the language.

Dijkstra had pointed out[14] that the GO TO statement was a dangerous language feature because it could easily be (and was) used to develop incomprehensible programs, ones with rat's nests of branching. One could not be sure in such programs just how the program would really act under all possible sets of input data. Repair of these programs often introduced even more baffling branching. The new ALGOL constructs provided an enforced organization of control and of thinking about control. They were eventually proven to be sufficient for all program control; that is, the GO TO was no longer even needed.

In addition to the control structures, the nesting of those structures naturally accompanied them. ALGOL also introduced the idea of the "scope" of data within the blocks and of dynamic storage allocation. We won't go into those items here, but they all had significant impact on subsequent languages. In the U.S., ALGOL never became a popular programming language in its own right, largely because the designers did not specify any input-output capabilities in the language.

It did become a popular language in universities as a pedagogical tool, however. Besides having the excellent control structures ALGOL was the first really rigorously defined language. The definition was done in a language used to describe languages (a "meta-language") called BNF[15]. The abbreviation meant either Backus Normal Form or Backus Naur Form. John Backus invented the language and Peter Naur used it to describe the syntax ("grammar") of ALGOL. Because of ALGOL's formal BNF definition it was an excellent language for teaching techniques for designing compilers. Figure 11 shows how ALGOL can do the GCD algorithm.

PL/I

After ALGOL a language named PL/I[16] was developed to be a kind of language for all applications with something for ev-

```
integer procedure gcd(x,y);value x, y;
    integer x, y;
    begin if x=y then gcd := x
         else if x>y then
                 gcd := gcd(y,((x-1)mod y)+1)
               else gcd := gcd(y,x)
    end
```

Fig. 11. An ALGOL version of the GCD algorithm. Control structures ("if-then-else" and "begin-end") first appeared in ALGOL as did the use of ":=" to indicate assignment and semi-colons as statement terminators. The procedure section is actually a single statement.

eryone. It combined features of FORTRAN, ALGOL, COBOL, and even APL. PL/I has hundreds of reserved words. It is a very large language requiring a very large compiler program. Despite strong support from IBM, PL/I never became truly popular. Figure 12 shows a PL/I treatment of the GCD.

Pascal

In reaction to excesses such as those in PL/I, Nicklaus Wirth developed a small but capable language requiring only a small compiler. This language, Pascal[17] (named in honor of mathematician Blaise Pascal), was based on the ideas of ALGOL. Its aims were to be a language "suitable to teach programming as a systematic discipline based on certain fundamental concepts clearly and naturally reflected by the language" and to "develop implementations of this language which are both reliable and efficient on presently available computers."

Pascal follows the flavor of ALGOL. Notable additions are input-output features, new data structures and user-definable types (as compared to built-in types such as integer and real). Although designed for teaching, Pascal has enjoyed great popularity lately, being implemented on both large computers and microcomputers. It seems that "small is beautiful." Figure 13 shows the Pascal version of GCD.

*Ada**

In the 1970s the U.S. Department of Defense (DoD) realized that a software crisis was looming. In addition to the many kinds of military computers then in use, there was an unmanageable number of programming languages (both assembly and higher level) in military systems. Several of the "standard" languages, notably CMS-2 and JOVIAL, had a number of incompatible dialects besides. DoD decided that a standard language for all military embedded computer systems should become a requirement.

To define the language,[18] DoD requested suggestions from military (U. S. and foreign), academic and industrial institutions several times during the development of five increasingly detailed requirements documents (named Strawman, Woodman, Tinman, Ironman and Steelman).[19] Four candidate languages (named Red, Green, Blue and Yellow) were developed and Cii Honeywell Bull's Green was finally selected. Soon afterward it was named Ada after Lady Augusta Ada Lovelace (who was the daughter of Lord Byron and probably the first real programmer). All four candidates were based on Pascal, a tribute to Pascal's careful design.

* Ada is a Registered Trademark of the U.S. Department of Defense (Ada Joint Program Office).

```
GCD:PROCEDURE(X,Y) RETURNS(INTEGER) RECURSIVE;
    DECLARE(X,Y) INTEGER;
    IF X=Y THEN RETURN(X);
        ELSE IF X>Y
            THEN RETURN(GCD(Y,MOD((X-1),Y)+1));
            ELSE RETURN(GCD(Y,X));
    END;
```

Fig. 12. A PL/I version of the GCD algorithm. In PL/I, you must explicitly declare the procedure to be recursive for recursion to be implemented by the compiler. The "ELSE" statement is separate from the "IF-THEN" statement rather than the single statement in ALGOL.

Ada, however, goes far beyond the simple aims of Pascal due to the extra requirements imposed on it. First, Ada supports the orderly development of truly large software systems. To do this, Ada has separately compileable "program units" so that many pieces of a large software system can be written at the same time. Another structural feature is the "packages" of data and program units with two distinct parts. One part, called the "specification," holds declarations of the interface, or the user's view of the items in the package. The other part, called the "body," holds the implementation of the package items. Various packages can be independently specified by many people and the compiler can verify that their interfaces are compatible before the implementations are written. Bodies can later be written or completely changed without having to recompile the entire program system.

Ada produces reliable programs. The package concept helps here, as does the strict enforcement of "typing" (like Pascal) to ensure that all data are used properly.

Ada supports concurrent processing (for example, for systems of computers). A type of program unit called a "task" identifies pieces of software that can be run at the same time as other pieces. There are also facilities for communication between tasks and a means for handling errors automatically.

Ada supports the sharing of software among programmers and the transportability of software among computers, both to reduce the amount of software developed. "Generics" are templates or skeletons of programs or data with parts unspecified. When a specific version of the generic software is to be used in a specific application, then only the missing parts are specified to allow the compiler to create that version. Transportability is supported by requiring that all Ada-related tools, including the Ada compiler, be written in Ada. Importantly, a standard programming environment[20] (tool set) is also being prepared and will be required for all Ada development.

The Ada language and its compilers are much larger than Pascal and more capable.[21] It remains to be seen how easy Ada is to use. As a strong incentive to establish Ada, DoD has required that, beginning in 1984, all new DoD software developments must be done in Ada. Additionally, Ada must be used as a PDL for development of software requirements and specifications. Figure 14 shows an Ada version of GCD.

Languages for artificial intelligence

The LISP (LISt Processing)[22,23] language was developed in 1960 by John McCarthy and his team in M.I.T.'s Artificial Intelli-

```
function GCD(X, Y : integer) : integer;

   begin (* function GCD *)
      if (X = Y) then
         GCD := X
      else
       begin (*else*)
         if X > Y then
           GCD := GCD(Y,((X-1) mod Y) + 1
         else
           GCD := GCD(Y,X)
       end; (* else*)
    end; (* function GCD *)
```

Fig. 13. A Pascal version of the GCD algorithm. Pascal follows the structure conventions of ALGOL, except that greater use of "begin-end" bracketing is required to avoid ambiguities about which "if" an "else" refers to.

```
function GREATEST_COMMON_DIVISOR(X,Y : integer) return integer is

begin
   if X = Y then
       return X;
   end if;
   if X > Y then
     return GREATEST_COMMON_DIVISOR(Y,((X-1)mod Y)+1);
   else
    return GREATEST_COMMON_DIVISOR(Y,X);
   end if;

end GREATEST_COMMON_DIVISOR;
```

Fig. 14. An Ada version of the GCD algorithm. Although this function could also have been called "GCD," this shows Ada's support of long names (like COBOL). Ada uses an "end if" phrase to mark the end of an "if" statement. This small program does not illustrate Ada's several features for developing large programs.

```
  (def gcd (lambda (X Y)
          (cond
           ( (equal X Y)      X)
           ( (greaterp X Y) (gcd Y (plus (mod (subl X) Y) 1) ))
           ( t               (gcd Y X) )
          )))
```

Fig. 15. A LISP version of the GCD algorithm. As all LISP functions, this is a single expression in the form of a function name followed by arguments, and each argument is in the same form. Lots of parentheses result from this inherently recursive form.

gence Group. Its roots are in Artificial Intelligence (AI), and LISP remains the language of choice in the AI community. McCarthy's work on a system to make deductions from imperative and declarative sentences required a programming system to manipulate formal expressions. He developed a scheme for representing an appropriate class of computations (partial recursive functions on symbolic expressions) as the basis for LISP.[24]

The resulting programming language is radically different from others in several ways. The data is in the form of lists containing symbols and other lists. The functions of LISP are based on recursion (McCarthy also influenced the inclusion of recursion in ALGOL at about the same time). Programs are lists of symbols—identical to data structures. Thus, programs can easily manipulate other programs or even themselves.

The appearance of LISP programs is radically different too. Parentheses abound, functions all use "Polish" notation with the function name followed by the arguments, and (in classic LISP) there is no assignment or use of variables. Programs are not sequences of operations (do this, then do that) but are functions of functions of functions . . . . Although LISP is only well suited for list processing and general symbolic manipulation, it has basic arithmetic operations. Figure 15 shows the GCD in LISP.

LISP has few primitive operations; all interesting functions are built from them. In this way, one easily extends the language by building layers of capability. Functions that produce other functions are much easier to develop and use in LISP because programs are treated as data. Automatic production and modification of new functions are instrumental to automated learning, and very useful in AI applications.

Layers of definition and recursion also make processing slow. Several LISP machines are now in production, specifically designed to efficiently execute LISP programs.

Languages for saying what to do, not how to do it

PROLOG

Expert Systems are programs that mimic the diagnostic talents and counselling of human experts, usually in specialized areas of knowledge. Generally, the expert knowledge is expressed as a set

```
GCD(J,J,J).

GCD(X,Y,Z) :- X>Y,
              Dl is X-1,
              D2 is Dl mod Y,
              D  is D2+1,
              GCD(Y,D,Z).

GCD(X,Y,Z) :- GCD(Y,X,Z).
```

Fig. 16. A PROLOG version of the GCD algorithm. These three statements are the simplest expression of the GCD yet (except for our modified modulo algorithm).

of "rules," statements associating related pieces of information with no "order of execution." PROLOG (PROgramming in LOGic)[25] is often used to implement Expert Systems. PROLOG uses sets of rules and statements of "theorems." A theorem acts as a high-level goal to be proven or satisfied. The PROLOG system automatically determines subgoals through which the goal could be achieved, then repeats the process until rules are found that satisfy the chain of subgoals. A practical feature of this process is that programmers specify neither the order in which the rules are used or the subgoals to be satisfied.

PROLOG shares some features with LISP. Figure 16 shows a PROLOG version of the GCD. Although our modified modulo function is not neatly handled, note that PROLOG needs to make only the original statement of the problem. Thus, it is easily seen to be perfectly correct.

PROLOG was developed in France and has recently become popular in Japan where that country's Fifth-Generation effort aims to make widespread, practical use of AI in a short time.

Very high level languages

SETL

The less detail one must specify for a program, the easier it is to write the program. This is the basic idea behind high-level languages. They save program development time by letting the computer do as much of the work as possible. The level that a language achieves is related to how easily concepts can be expressed in that language. SETL (SET Language)[26] is a so-called Very High Level Language. It easily supports sets, maps, graphs and trees as data types. It does arithmetic, set, and string operations. It uses existential and universal quantifiers to conveniently identify subsets satisfying given properties.

A major triumph of SETL was the quick implementation of the first Ada interpreter. Although the interpreter is very slow, it provides use of the entire Ada language. The Ada interpreter was also the first Ada translator (including all of the Ada compilers being developed) to pass the stringent Ada certification tests.

FP

John Backus, in his Turing Award lecture,[27] presented a complaint about current programming languages. Backus, the chief designer of FORTRAN, contended that our programming languages are too closely tied to the machines on which they run, that they suffer from a malady he termed "the Von Neumann Bottleneck." This bottleneck is the passageway between the conventional computer processing units and the data memory. Not only do the data and the instructions to be executed pass through the bottleneck, item-by-item, but so do the addresses of the data and the instructions. Too great a portion of the processing has no direct relation to the intended computations.

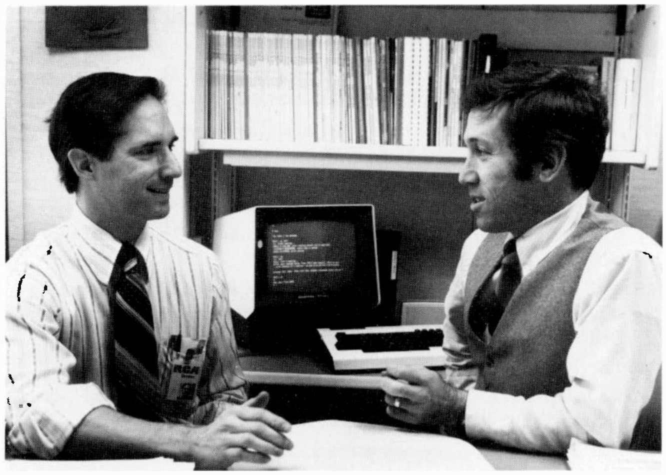
To break away from this bottleneck, Backus proposes a functional language based on the advances of LISP and APL. Its basic operations operate on arrays and sets and on functions (programs). Programs should satisfy algebraic rules with proveable characteristics. Not only will we have functions of functions, but also higher levels of functional forms capable of spawning whole classes of new functions. Programs would be more abstract, smaller, and manageable. This development should also affect the design of machines on which such programs would run. FP is still in development, but it holds great promise.

Summary

Programming languages have changed enormously during their three decades of life. They permit us to specify complex computations and processes in simple, convenient ways. Each purpose for a computer language has led to new languages in the past, and the trend continues. But, whereas the machine once dictated the form and style of programs, the programming languages are now influencing the structure of the machines that they direct. The languages are also giving humans better insight into computation and ever better ways to accomplish it. They are even teaching us more about language itself.

Acknowledgment

The authors wish to thank Helen Wu for supplying Ada and Pascal examples, and Jim Whitehead for the LISP example. Thanks, too, to the students in Eric Braude's "Programming Languages" class (taught at the New Jersey Institute of Technology), and especially to Janice Ward, for their review of an early draft of this paper.



Authors Mebus (left) and Braude.

Eric J. Braude is Manager of the Software Technology Laboratory at RCA's Advanced Technology Laboratories, where he is responsible for several projects in artificial intelligence, program verification and compiling.

Dr. Braude was educated at the University of Natal (South Africa), Miami and Illinois, and received a Ph.D. in mathematics from Columbia University.

He was formerly a Professor at the City University of New York and Penn State University, and was a Computer Scientist at the Naval Surface Weapons Center. He has published work on Descriptive Set Theory, Lattice Theory and Compiling Technology.

Contact him at:
Advanced Technology Laboratories
Camden, N.J.
TACNET: 222-4067

George Mebus is a Unit Manager, Software Technology Laboratory at the Advanced Technology Laboratories, Camden, New Jersey. He joined RCA in March, 1983. His current responsibilities are in programming languages and their processors in the areas of microcode development, software engineering environments, and formal verification of computer systems. Previously, Mr. Mebus worked at the Naval Air Development Center, Warminster, Pennsylvania, where he led the development of the Facility for Automated Software Production (FASP) for software support of the Navy LAMPS project. He is a member of IEEE CS, ACM, and Tau Beta Pi and holds a U.S. patent. His favorite programming language is APL.

Contact him at:
Advanced Technology Laboratories
Camden, N.J.
TACNET: 222-3597

References

1. J.E. Sammet, *Programming Languages: History and Fundamentals,* Prentice-Hall, Englewood Cliffs, New Jersey (1969).
2. D.E. Knuth, *The Art of Computer Programming,* Vol. 1, 2nd Ed., Addison-Wesley, Reading, Massachusetts (1973).
3. J.W. Backus, *et al.,* "The FORTRAN Automatic Coding System," *Proceedings of the Western Joint Computer Conference,* Los Angeles, California, pp. 188-198 (1957).
4. L.R. Meissner and E.I. Organick, *FORTRAN 77: Featuring Structured Programming,* Addison-Wesley, Reading, Massachusetts (1980).
5. H.D. Peckham, *BASIC: A Hands-On Method,* McGraw-Hill, New York, New York (1981).
6. *COBOL—1961: Revised Specifications for a COmmon Business Oriented Language,* U.S. Government Printing Office : 1961 0-598941
7. *An American National Standard IEEE Standard ATLAS Test Language,* Institute of Electrical and Electronics Engineers, New York, New York, ANSI/IEEE Std 416-1978, (1978).
8. K.E. Iverson, *A Programming Language,* Wiley, New York, New York (1962).
9. L.Gilman and A.J. Rose, *APL—An Interactive Approach,* Wiley, New York, New York (1976).

10. J.G. Solomon, "APL and Engineering Productivity," *RCA Engineer,* Vol. 28, No. 1, pp. 18-24 (January/February 1983).

11. E.E. McDonnell, "A Notation for the GCD and LCM Functions," from *APL75,* Association for Computing Machinery, New York, New York (1975).

12. O.J. Dahl, E.W. Dijkstra, and C.A.R. Hoare, *Structured Programming,* Academic Press, London, New York (1972).

13. P. Naur, Editor, "Revised Report on the Algorithmic Language Algol 60," *Communications of the ACM,* Vol. 6, pp. 1-17 (1963).

14. E.W. Dijkstra, "Go To Statement Considered Harmful,"*Communications of the ACM,* Vol. 11, No. 3, pp. 147-148 (March 1968).

15. J.W. Backus, "The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference, *Proceedings of ICTP,* UNESCO, Paris, pp. 125-132 (1959).

16. J.K. Hughes, *PL/I Programming,* Wiley, New York, New York (1973).

17. K. Jensen and N. Wirth, *PASCAL User Manual and Report,* 2nd Edition, Springer-Varlag, New York, Heidelberg, Berlin (1974).

18. *Ada Programming Language,* Department of Defense, Washington, D.C., ANSI/MIL-STD-1815A-1983 (January 1983).

19. *Requirements for High-Order Programming Languages, STEELMAN,* Department of Defense, Washington, D.C. (June 1978).

20. *Requirements for the Programming Environment for the Common High-Order Language, STONEMAN,* Department of Defense, Washington, D.C. (February 1980).

21. N.M. Habermann and D.E. Perry, *Ada for Experienced Programmers,* Addison-Wesley, Reading, Massachusetts (1983).

22. J. McCarthy, *et al., LISP 1.5 Programmer's Manual,* M.I.T. Press, Cambridge, Massachusetts (1962).

23. P.H. Winston and B.K.P. Horn, *LISP,* Addison-Wesley, Reading, Massachusetts (1981).

24. J. McCarthy, "A Basis for a Mathematical Theory of Computation," *Computer Programming and Formal Systems,* P. Braffort and D. Hirschberg, Editors, North-Holland, Amsterdam (1967).

25. W.F. Clocksin and C.S. Mellish, *Programming in PROLOG,* Springer-Varlag, Berlin and New York (1981).

26. J.T. Schwartz, *On Programming: An Interim Report on the SETL Project,* 2nd Edition, Courant Institute of Mathematical Sciences, New York, New York (1975).

27. J. Backus, 1977 ACM Turing Award Lecture: "Can Programming Be Liberated from the Von Reumann Style? A Functional Style and Its Algebra of Programs," *Communications of the ACM,* Vol. 21, No. 8, pp. 613-641 (August 1978).

Upcoming *RCA Review* issue announced

RCA Review is a technical journal published quarterly by RCA Laboratories in conjunction with the subsidiaries and divisions of RCA Corporation. Copies of the issue are available (at a cost of $5.00) by contacting the *RCA Review* office at Princeton, TACNET: 226-3222.

Contents, December 1983

R.M. Blasewitz

Ada®—Not just another programming language

Ada promises to be the fourth generation *higher order language to bridge the gap in software technology, and begins—in earnest—the assault on the software crisis.*

"The Analytical Engine has no pretentions whatever to originate anything. It can do whatever we know how to order it to perform. It can follow analyses; but it has no power of anticipating any analytical relations or truths."

—*Augusta Ada Lovelace*

The long-term and growing trend toward software domination of systems, both commercial and military, has assumed the proportions of a landslide in recent years. Essentially every system under development or in the planning stages makes extensive use of computer technology and software. Digital systems are now commonly applied not only to control of the central function of devices, but also to inter-system communications.

Why is software so dominant and why is it becoming even more so? A principal reason is that, when a system modification is required, software changes are easier to make than physical system changes. Indeed, a Department of Defense (DoD) study has shown that hardware changes cost 38 times as much as software changes and take three times as long to implement.

Current annual DoD expenditures for software are measured in billions of dollars, and estimates by the Electronics Industries Association place military software costs at $32 billion by 1990 (Fig. 1). These estimates indicate that software costs are more than substantial, they are dominant. When one considers how many more commercial systems are required than military systems, and the attendant costs, then the impact of software is seen to assume enormous proportions.

The growth of the software development processes has shown a chaotic pattern from the beginning. Even now, after 25 years, we find little uniformity in the specifics of software development. It is also clear that there are not enough trained software professionals to meet today's demands. And this situation is steadily growing worse.

These issues have become increasingly clear in recent years, virtually crying out for an intelligent, planned approach to the problems. The United States Department of Defense, largely because of the visibility of its needs in this area, took the lead in the mid-1970s by sponsoring development of a new high-order language suited to the full range of military applications and incorporating state-of-the-art software practices. This development program and the language itself were called Ada.® The background and evolution of Ada make a fascinating study; a highly abbreviated summary is contained in the accompanying box.

Ada and its applications

DoD initiated the Ada program to save taxpayer money through standardization. These savings will come from the portability and reuse of operational software, more effective use of support software (including Program Design Languages), improved programmer productivity, and reduced software maintenance. Standards will also be beneficial for business concerns. There is little question that the entire software industry is in need of a modern, efficient, and highly portable system-implementation language and tool set. In summary, technical arguments about which language is best miss the point, because only Ada will benefit from DoD's investment in standards enforcement for compilers and supporting tools.[1]

Unquestionably, then, Ada has come of age as an accepted high-order language for use in embedded computer applications. The proposed standard for the language is extremely comprehensive and is recognized as very powerful.[2] Further, the ability of Ada to solve problems is as great as that of any other programming language. Ada's adoption as a formal military standard guarantees its use within the defense community, but the commercial applications of Ada may surpass its military use. A number of factors contribute to this commercial interest:

1. American and European industries were involved in the development of Ada design requirements and in the review and revision of the language.
2. Ada is now widely used as a teaching tool in academia.
3. Ada has been adopted as the standard language for process control by the European Common Market.
4. Commercial applications often mirror military applications in function or character; for example, in process control and word processing.
5. Ada is designed to ease future maintenance problems.

An indication of this commercial interest is seen in the parallel and cooperative Ada development efforts being carried out by the British and German Ministries of Defense, the European Economic Commission, and the Japanese Ministry of International Trade and Industry. Another indicator of interest is the number of compilers available and under development. Already there are at least three validated (by DoD) Ada compilers available, with 20 more expected in 1984; a number of other compilers have been developed for Ada and are awaiting DoD validation. Millions of lines of Ada code have been written and executed, and large Ada applications are currently in use. One such commercial user reports using a "bare bones" compiler to begin implementing all in-house and contracted software in Ada.[3] The results of the first effort—a payroll package and

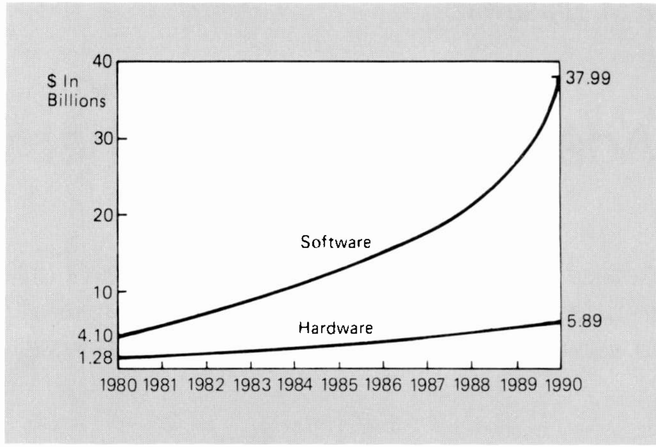


Fig. 1. *Software costs in DoD embedded computers will approach $38 billion by 1990, a margin of nearly $32 billion over hardware costs (courtesy EIA).*

Ada® is a registered trademark of the U.S. Government Ada Joint Program Office.

Abstract: *Ada® is more than just another high-order language; it is the basis for a modern perspective not only of programming but also of software engineering. As the programming system from which a new software culture will evolve and mature, Ada is the cornerstone of the Department of Defense initiative entitled "Software Technology for Adaptable, Reliable Systems" (STARS). This paper describes the rationale for the creation of Ada, and indicates progress to date in its broad-based application. The structure of Ada is presented, with primitive examples of its use in basic programming problems.*

Final manuscript received December 2, 1983.
Reprint RE-29-1-3

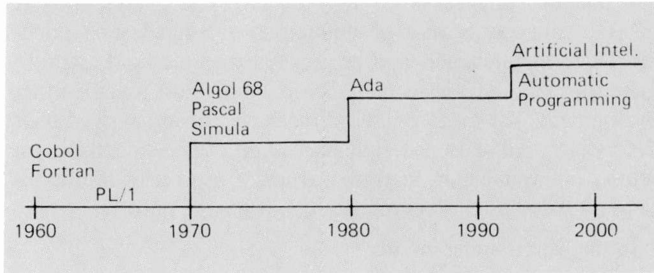
The evolution of Ada

In January, 1975, the Undersecretary of Defense for Research and Engineering initiated the development of Ada. Department of Defense directives and instructions were then issued requiring the use of high-order languages for military systems, designating existing languages (FORTRAN, JOVIAL, TACPOL, COBOL and CMS-2) as interim standards, and stating that the services should work together in developing a single language suited to military applications and incorporating state-of-the-art software practices. The DoD established a High Order Language Work Group (HOLWG) under the direction of Lieutenant Colonel William A. Whitaker to coordinate the effort.

The basic requirements for the language, which was intended to counter increasing software costs and poor-quality software, were defined in a sequential series of draft specifications developed between 1975 and 1977 by David A . Fisher at the Institute for Defense Analysis. These documents were expanded in specific detail and formalism as the HOLWG gained knowledge of the key tradeoffs and applications requirements of embedded computer software.

In 1977, DoD issued a competitive request for proposals from contractors to design a language that met a set of requirements called Ironman. Seventeen bidders responded, of which four were chosen to present independent preliminary designs. A final requirements document, entitled Steelman, integrated the knowledge obtained through the design contractors' attempts to meet the requirements. Thus, Steelman became the target requirements document for the competing contractors. The four finalists were Honeywell, SRI International, SofTech, and Intermetrics. The designs were presented in early 1978, with all the designs being distributed throughout the Ada-involved community for comment. These comments dictated that the Honeywell and Intermetrics design were preferred, and thus these two contractors were selected for the second, follow-on phase. The final designs were submitted in April, 1979. Using the worldwide Ada community for extensive evaluation, the DoD selected the Honeywell design that had been developed by an international team at Cii-Honeywell Bull, Paris.

The new language was named Ada, after Countess Augusta Ada Lovelace. It was she, the daughter of poet Lord Byron, who translated the work of Italian mathematician L. F. Menabrea, attaching her own commentaries on the variances between the "difference engine" and the "analytical engine."



Ada represents a major step in modern program language development, and should exist until either artificial intelligence or automatic programming replaces it.

> The Analytical Engine does not occupy common ground with mere calculating machines. It holds a position wholly its own; and the considerations it suggests are most interesting in their nature. In enabling mechanisms to combine together general symbols, in successions of unlimited variety and extent, a uniting link is established between the operations of matter and the abstract mental processes of the most abstract branch of mathematical science. A new, a vast, and a powerful language is developed for the future use of analysis, in which to wield its truths so that these may become of more speedy and accurate practical application for the purposes of mankind than the means hitherto in our possession have rendered possible. Thus not only the mental and the material, but the theoretical and the practical in the mathematical world, are brought into more intimate and effective connexion with each other.
>
> —Augusta Ada Lovelace, 1842.

Lovelace also delivered the notes and writings of Babbage (with her amendments) to the scientific community of the time, and is thought to be the first female programmer.

The Deputy Secretary of Defense announced the DoD intention to make Ada a military standard, opening another series of reviews and comments. Subsequent months were used to expose the language to the actual user community and to refine it according to the feedback received. The proposed Ada Standard Document, MIL-STD-1815 (the number was chosen to honor the Countess' birth year, 1815) was published in July, 1980. In the same year, DoD created the Ada Joint Program Office (AJPO) under the direction of Lieutenant Colonel Larry Druffel to manage the desired implementation and acceptance of Ada. Ada was submitted to the American National Standards Institute (ANSI) for consideration as an American standard under the ANSI canvass procedure in 1982, and approved in January, 1983. Ada is also a registered trademark of the U.S. Government (Ada Joint Program Office).

voucher entry system for a general ledger accounting package—elicited comments such as "superbly efficient" and "The ability to independently compile packages and insert them into a main program provides an inherent flexibility in design that is not readily available in any other commercial languages." Of special significance was a poll of the programmers on their language preference—Ada was the unanimous choice.[3]

What other factors are important in this language's climb to esteem? Many computer scientists believe that in addition to their use in coding, Ada-compatible methods and languages also have potential application in software life-cycle phases, such as definition and specification of requirements, design, testing, documentation, maintenance, and project management and control.[4] These uses of Ada are in their infancy and must await further development. Yet many organizations have already derived their own unique Ada-based design languages,[5] perceiving this as a method of capitalizing on three particular aspects of transitioning to an Ada-based programming environment, namely:

1. Immediate training in Ada
2. Experience in Ada's software potential as a design tool
3. Qualification to respond to Ada-related DoD Requests for Proposals.

Ada is also an important facet of a major DoD initiative undertaken to solve the software crisis that is now so well recognized.[6] It is widely believed that the U.S. has lost its lead in many of the mature technologies upon which our industrial and military base were founded. The same threat now challenges our lead in the computer, electronics, and software industries. DoD, in recognizing this factor, has begun a strong, aggressive action called "the software initiative," or the Software Technology for Adaptable, Reliable System (STARS) Program.[7] The purpose of this initiative is to describe and plan a management strategy for meeting the challenges of the future. Software is the essential element in the STARS directive, for it controls and defines the systems of today and tomorrow, embodies system "intelligence," and provides the flexibility to respond to changing threats, needs, and requirements.

The STARS effort, which will build on the achievements in the Ada Program, is designed to provide an integrated, automated environment that covers the entire software life cycle. Both STARS and the Japanese "Fifth-Generation" initiative are described briefly in the accompanying box.

What is Ada and how does it work?

As a modern programming language, Ada incorporates many of the advances made in programming languages and research conducted in the past decade. Ada is based on Pascal to a large extent, but this does not mean that Pascal is a subset of Ada; indeed, virtually all of the features of Pascal were altered for their introduction into Ada. The principal inheritance from Pascal is its philosophy—that both algorithms and data structures should be specifiable precisely and clearly, and that the logical consistency of a program should be ensured by the compiler wherever possible.[8] Thus Ada is concerned with software concepts such as readability, maintainability, efficiency and a concern for programming as a human activity. All of these characteristics stem from the basic need to alleviate the software crisis. Ada is also unique technically for the following reasons:

1. Ada supports the concept of a software components industry
2. Ada is a blend of modern software engineering principles
3. Ada is a language designed to support programming solutions throughout the life cycle of a software project
4. Ada will support production of very large and complex software systems
5. Ada has numerous provisions for portability.

When we examine the problem domain for which it was designed (embedded computer systems), we see that Ada is also a language of considerable expressive power. In particular, the DoD requirements mandated a language that would support structured constructs, strong typing, relative and absolute precision specification, information hiding, data abstraction, concurrent processing, exception handling, generic definition, machine-dependent facilities, and separate compilation.

Before we investigate how Ada accomplishes its major tasks, let us consider the languages preceding Ada and make some pointed comparisons. A programming language at any level must provide three primary functions:

1. A vehicle that allows human beings to communicate more effectively.
2. A tool that enables expression of thought.
3. A means of enabling humans to instruct machines (computers).

Looking at some of the major languages of the past we can make capability assessments (Table I) that are, of course, sensitive to the problem domain. In essence, Pascal was designed for teaching sound software principles and Ada was designed for major military software systems. Ada is compared with Pascal for a number of reasons, but primarily because of Pascal's wide acceptance as a well-designed high-level language. Ada improves substantially on the Pascal base as illustrated in Fig. 2. Ada not

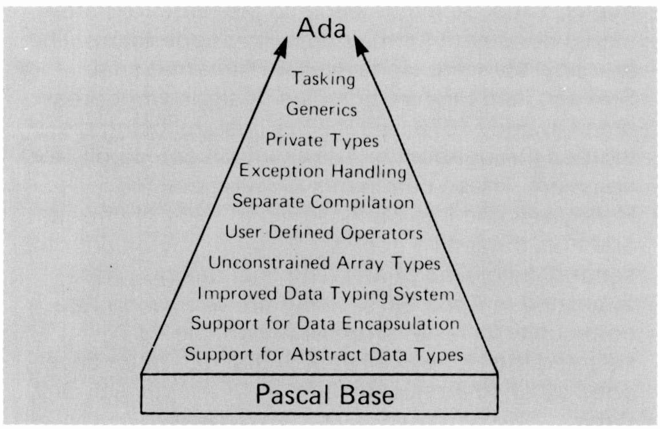


Fig. 2. *Use of Pascal as the building base for Ada permits incorporation of state-of-the-art advances in programming practices.*

Table I. *Language assessment for the three major language functions.*

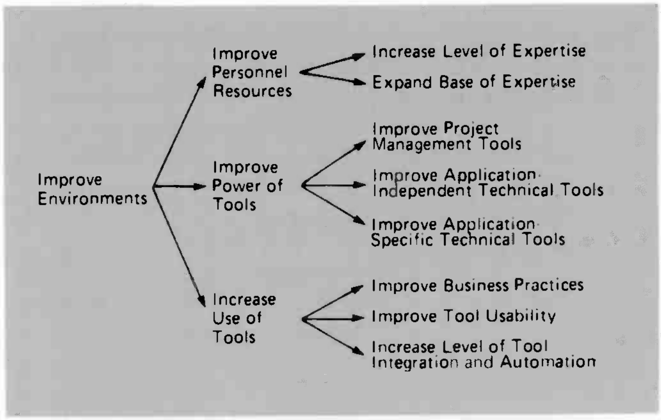
	Ada	*Pascal*	*Fortran*	*Assembly*
Function				
Communication	Excellent	Good	Average	Below Average
Tools of Thought	Very Good	Good	Average	Below Average
Instructing Computers	Good	Good	Average	Excellent

STARS and the Fifth Generation

A principal goal of the STARS Program is to improve productivity while achieving greater system reliability and adaptability.[17] The DoD planners knew that attainment of this goal would involve broad improvements in the environment in which software is first developed and then supported. The Ada program provides the initial focus for the development of a common, sharable (portable) software base. The STARS Program goes one step further and broadens the scope of attention to the entire environment in which software is conceived and evolved. The Ada Programming Support Environment (APSE) widened the focus beyond programming notations to the automated support for the software life cycle as a first thrust. The STARS Program expands this focus to encompass the unautomated (and in many cases unautomatable) aspects of the environment, the personnel working in the environment, and the efficient and smooth modernization of the environment as effective new technology becomes available.

It should be noted that although the STARS Program is motivated by a DoD need, the attendant issues are common to most of the software industry. The STARS Program is deliberately structured to encourage the rapid transition of new technology into practice. Initially the program does this by addressing the need for strong educational activities. Incentives will then be offered to develop and demonstrate improved techniques for possible incorporation into STARS products. The third component is the initiation of a Software Engineering Institute as a vehicle through which new technologies will be formed into products, validated, and brought into military practice. The Institute will be responsible for creating and operating a highly advanced software environment from which environments for military service standards will evolve. It will also ensure the availability of expertise as well as documentation and training support to reduce the risk involved in transitioning new software practices and tools on critical military systems.

The U.S. is not alone in realizing the importance of hardware/software technologies. Japan has proclaimed to the world that in ten years it intends to develop and market the fifth generation of computers—artificially intelligent machines that can reason, draw conclusions, make judgments, and even understand the written and spoken word.[18] In a crash program comparable to the U.S. space effort, Japan has gathered the best and the brightest under a charismatic leader, and backed the enterprise with substantial resources. A quote from E.A. Feigenbaum and P. McCorduck[19] illustrates Japan's commitment: "The wealth of nations, which depended upon land, labor, and capital during its agricultural and industrial phases—depended upon natural resources, the accumulation of money, and even upon weaponry—will come in the future to depend upon information, knowledge, and intelligence."



STARS program objectives illustrate the emphasis placed on improvement of present-day automated support environments.

Japan has also been pursuing the so-called software factory. This approach seeks much higher levels of software productivity with lower cost and schedule risks than those now achieved. The leading Japanese corporations have already implemented for their own use integrated systems of automated software tools that support the entire software life cycle. France and Great Britain have also made efforts to create a software technology research and development program, although neither to the degree or scope of the Japanese program. Quite simply, the Japanese are planning their future as the leaders of the knowledge industry, believing firmly that knowledge is power. This knowledge is based on both hardware and software quantum leaps over the technology of the past 30 years.

In summary, the U.S. is using Ada as a cornerstone for the advancement of software technology. The STARS Program strategy is to build upon existing DoD activities, and to meet the program objectives illustrated above. The objectives of the STARS Program are to improve the software state of practice by synergistically improving key aspects of the software development and support environment. STARS will rely on the planned evolution of the software environment, enhanced technically and by significantly improved acquisition strategies, management and business practices, and personnel upgrade programs. Central to the evolution of the environment and the transfer into the DoD community of the technology it embodies is the proposed national Software Engineering Institute. Ada is important in this program for it is more than a product: Ada is a process.

only adds those features specified in the figure, but also improves upon Pascal's syntax through the use of the Ada rules for using the semicolon.

Pascal patriots need not be alarmed, for Pascal is still said to have some advantages over Ada. These advantages include small size, simplicity/learnability, and wide availability of compilers. But (sorry, patriots), Pascal's problem-handling domain is also small, so that these advantages will dissipate rather rapidly. Ada is also technically unique because it is currently the only popular language matched to VLSI technology; it is the U.S. Army Military Computer Family's sole high-level language.

What is it that makes Ada a superior vehicle for software development? What are its primary technical features? How can the complexity of large-scale software development projects be reduced by the use of Ada? Let us examine these issues in some detail. In contrast to Ada, most existing programming languages emphasize syntax and features that are used to describe relatively small program units such as statements and procedures. Ada was designed with an overriding concern for a structured programming approach to software development: that is, the organization of a large program into components so that readability, reliability, and maintainability are maximized. Ada does provide excellent support for these characteristics, since the major features of the language explicitly reflect these qualities. Ada encourages, even demands, what has been called a "constructive" approach to programming.

The goals of modifiability, efficiency, reliability, understandability, and maintainability are reasonable choices for virtually any software system. The key to the production of software having these attributes is the sound application of software engineering principles that support these goals. Although not all-inclusive, an adequate set of principles might be:

- modularity
- uniformity
- information hiding
- completeness
- localization
- confirmability
- abstraction

Modularity is widely recognized as an important aspect of program structure. The motivations for designing modular software are based on Parnas' concept of information hiding. In recent years, the notion of information hiding has evolved into the concept of data abstraction. Again, a language that includes features inspired by these ideas is supportive of large-scale software development. To a lesser extent, a separate compilation facility also offers a means for modular grouping of related procedures and, possibly, some common data objects.

The package concept

The concept of the Ada package is thought to be the language's principal contribution to the programming science.[9] An Ada package consists of two components, the specification and the body. Packages permit a user to encapsulate a group of logically related entities. Therefore, packages directly support the software principles of data abstraction and information hiding. The following example illustrates the structure of a package:

```
        Package MY_EXAMPLE_PACKAGE is
        { -declarations                      } Visible and
Visible {      .                             } private parts
part    {      .                             } form the
        {      .                             } package
        {      .                             } specification.
        private
          -declarations                      } Private part
               .                             } is invisible to
               .                             } users.
               .
        end MY_EXAMPLE_PACKAGE

        Package body MY_EXAMPLE_
                   PACKAGE is                } Package
          -declarations                      } body
          -sequence of Ada statements        }
               .
               .
               .
        end MY_EXAMPLE_PACKAGE;
```

The package specification is simply a set of declarations concerning data objects. The sequence up to the occurrence of the word "private" is the visible part of the specification. The last part is called the private part and will be described further below.

The package body is the means by which the package provides the operations promised in the package specification. That is, the code for the operations to be performed (for example, compute the new time of day) is located within the body component. The body may also contain declarations needed to implement the operations, but these need not be usable outside the package body.

This package structure supports the principles of modularity, abstraction, localization, and information hiding. Programmers can apply these principles in other languages, but Ada packages enforce and encourage these principles. If a user violates the Ada rules with respect to package specification, the compiler will trap the errors at compilation time. Since the specification and body may be compiled separately, it becomes an easy matter to create the specification early in the software design process and then later to add the body as details about the operations become known. In other words, the package provides excellent support for top-down design. Possibly most important, Ada packages aid in the process of controlling the complexity of software solutions (as was done by Fortran subroutines) by providing a mechanism with which to physically partition related entities into a logical grouping.

Information hiding

Abstraction is the next major principle of concern in the production of engineered software implementation, which leads us to consider the concept of information hiding. Ada provides two mechanisms for hiding details of implementation while still making the mechanisms for using the implementation available to users. The purpose of hiding is to make inaccessible certain details that should not affect other parts of a system. Ada's package structure gives programmers the ability to decompose the entire package into a specification and body, where the body can hide the implementation details of the package. In fact, given the package specification, it is possible to compile programs that make use of the package before the package body even exists.

The second mechanism noted above is the private part of the package specification—a place for declarations that are necessary to define the physical interface of the package, but that need not be included in the logical interface. The physical interface provides the information that the compiler requires. The logical interface is simply the visible part of the package. All of this implies that the user of the package is unable to make use of the information contained in the private part in any way that will affect the correctness of his program.[10] This distinction between physical and logical interfaces permits enforcement of the following maintainability characteristics:

- Changes to the package body are guaranteed not to require modifications to the source of programs using the package, which in turn leads to no new recompilations;
- Changes to the private part are guaranteed not to require modifications in the source code of programs that reference the package, but may require recompilation of the source;
- Modifications to the visible part may require changes to the source code, followed by recompilation.

Although abstraction and modularity are perhaps the most important principles we may use to control the complexity of our software, they do not ensure that the final products will be consistent or correct. Hence the principle of uniformity directly supports the goal of understandability, and implies that programs use a consistent notation and are free from any unnecessary differences.

The principles of completeness and confirmability also support the goals stated earlier by aiding in the development of solutions that are correct. Whereas abstraction extracts the essential details of a given design, completeness ensures that all the necessary elements are present. Efficiency is also enhanced by strictly limiting the effect of lower-level implementation changes on higher-level programs. The principle of confirmability implies that we can decompose our software system so that it can be readily tested and validated. Ada supports these principles with such features as:

- Strong typing;

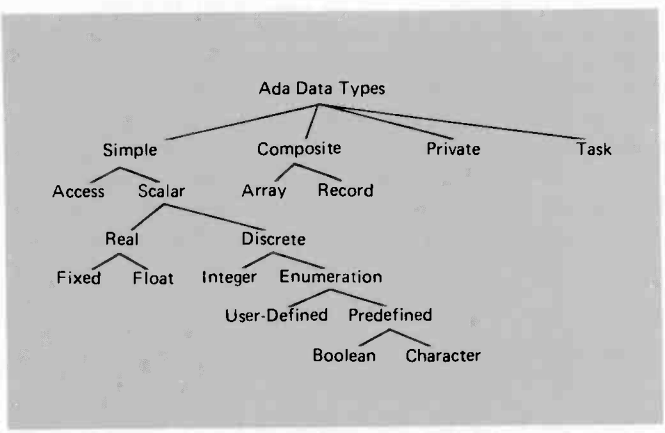


Fig. 3. *Ada's typing system abounds with power, robustness, and freedom of expression for potential users.*

- Configuration management over the naming space;
- Multiple methodological support (top-down and bottom-up);
- Full support of predefined data types; and
- Unique parameter-passing features and mechanisms.

Ada's strong typing allows a user to express a set of values and a set of operations applicable to those values in a unique and conclusive manner. Figure 3 illustrates the allowable data types within Ada. Notice that users may define their own class of types under the enumeration heading.

For example, the following is a valid Ada variable type:

```
TYPE COLOR is (RED, WHITE, ORANGE, BLUE)
```

Allowing the user this flexibility and freedom of expression in typing would seem to contradict good practice; however it is more than balanced by the greater security and efficiency gains produced.

Ada also facilitates the management of program name space by requiring each compilation unit to indicate the other compilation units with which the unit is to be compiled and upon which it depends.[11] For example, package ABC may begin with the clause "with A, B" which states that package ABC can refer to the names A and B and can use the names declared in A and B. This name qualification aids in keeping symbols distinct and more manageable, thereby aiding configuration management practices.

Ada supports bottom-up and top-down software development methodologies. Bottom-up methods can proceed by the development of packages at the body level first. Once the operations at the low level are known (such as, compute radar dwell-time), the algorithms can be implemented via the package body mechanism. The specification part of the package can be compiled separately. In Ada, library units are used to implement this methodology. Therefore, after the creation of packages that provide the primitive facilities we need, we can build up from the tools they give us. Top-down development is accomplished with the aid of program stubs and subunits. A stub indicates the place where a separately compiled subunit must eventually be provided. For example, while developing package PROM, a programmer may write:

```
Package body PROM is
    .
    .
    .
type MY_TYPE is array (1 . . . 100) of Integer
procedure NOT_DONE_YET
(X:MY_TYPE) is separate; stub
    .
    .
    .
begin
    .
    .
    .
end PROM;
```

The reserved word "separate" in this example informs the compiler that a subunit consisting of the procedure named

NOT_DONE_YET, with one parameter MY_TYPE, will be compiled later. The compiler saves in the data base all of the contextual information needed from PROM. For the subsequent compilation of NOT_DONE_YET, a programmer can work with the following subunit:

```
Separate (PROM)
procedure NOT_DONE_YET (X:MY_TYPE) is
        .
        .
        .
begin
        .
        .
        .
end;
```

In this manner, orderly top-down design and implementation may proceed.

Ada's technical features cannot be dismissed without also looking at the following key elements (regardless of how they may support engineering principles or goals):

1. Tasks
2. Exception handling
3. Generic program units
4. Representation specification
5. Input/output.

The Ada tasking model is based on the concept of communicating sequential processes.[12] In other words, tasks can be viewed as independent, concurrent operations that communicate with each other by passing messages. When two tasks pass a message to each other, they are said to rendezvous. The basic Ada mechanism for this communication is provided through the *entry* and *accept* statements. Like a package, a task is also divided into a specification part and a body. Likewise, the modularity and abstraction concepts discussed for packages generally apply to tasks. For real-time applications, Ada provides this strong facility for multi-tasking or for logically parallel threads of execution that can cooperate in a controlled manner.

In most operations, a system must be able to recover gracefully from error conditions—a beneficial attribute of *any* software program. In most programming languages, an input format error, divide check, and so on would cause the program to crash and pass control back to the operating system. Ada permits a block-structured exception-handling capability. Exceptions, which are typically error conditions, may be predefined (such as divide by zero) or user defined (such as buffer is full condition). In Ada, if an exception occurs, normal processing will be suspended and control will pass to the exception handler (a program written for handling exceptions), which normally will respond to the exception noted. In the case where total recovery may be impossible, such as in peripheral device failures, Ada will allow graceful (that is, gradual) degradation of computer performance through the exception-handler mechanism.

The designers of Ada realized that its large and encompassing applications domain would force the emergence of some special features. Primary in this search for special features was the need to reduce the maintenance costs of complex software systems. A feature generated for this purpose was the ability to compile separate units, which was lacking in its predecessor, Pascal. In addition, Ada gives generic program units the ability to factor out common algorithms or procedures used on different data types. For example, we may need to sort an array of numbers from highest to lowest, or we may need to sort through an alphabetized list of names. The algorithm to process such information would be the same; only the data types would be different. A programmer may take this common algorithm and provide parameters in the form of data items, thus creating a generic unit.

At times a programmer must be able to exploit hardware features. In most languages this is done by creating a separate assembly-language routine, and then linking it to the high-level language. Ada, however, includes features to specify implementation-dependent features and data representation. In particular, Ada permits the specification of address, enumeration-type representation, length, and record-type representation.

Embedded computer systems are noted for their unique input/output devices or demands. Very few within the military domain use such standard devices as printers or terminals; instead they interface to signal processors, radars, and so on. To handle such a wide variety of devices, Ada input/output (I/O) is achieved through several packages. In particular Ada has predefined packages for high-level I/O (including binary I/O and text I/O), and low-level I/O. Ada also allows programmers to create their own unique I/O packages as required. In fact, with the extensibility provided by Ada's packaging mechanism and generic facilities, one need not provide any special language features to accommodate I/O.

Obviously not all of Ada's features can be examined here. Only some of the more notable have been briefly summarized, with some emphasis on their use or function. The language draws on many years of research relating to algorithmic languages and methodologies to support modern programming principles. However, there is less than total support for Ada, with some critics claiming that Ada is simply too complex.[13] Possibly the only answer to these critics is that the features that make it so complex also make it a superior vehicle for large-scale software development.

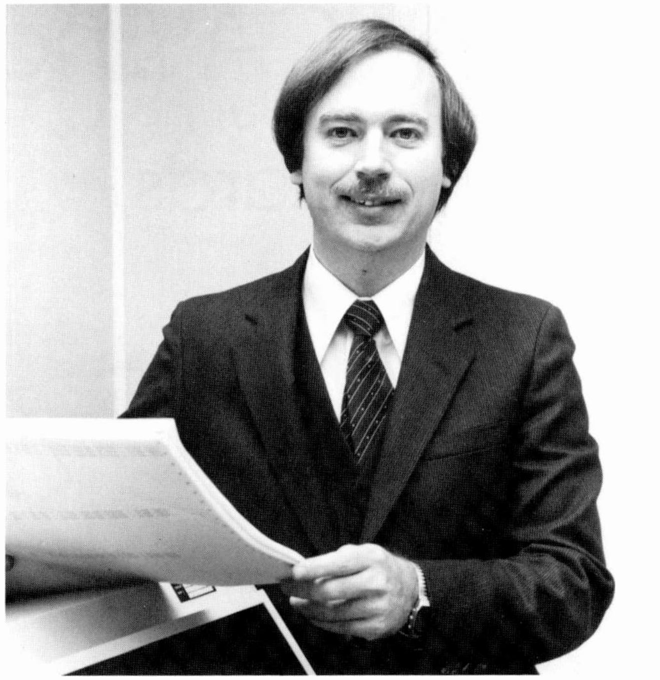
Training for a language of this robustness is not a simple matter.[14,15] For the first time, a language is more than its syntax, thus making it more of a method than a language. Its components are complex and are difficult to understand and to apply optimally. The design of real-time systems using Ada is a process that requires special skills and backgrounds, and dictates that choices for design be founded in sound engineering principles. Ada's approach to software development is truly unique, and to reap its benefits users will need to be trained and skilled in using its philosophy of software development—regardless of the methodology selected. As a relatively new language, and one with almost no track record with validated compilers, Ada presents the potential user with a very real risk factor. However, with DoD's firm support and a growing recognition of its inherent superiority, Ada shows every sign of becoming a spectacular success.

Conclusion

Clearly, Ada is a language whose time has come. It is real, not only because it was painstakingly developed to meet the long-term needs of advanced system development, but also because the demand for an Ada-like solution to the growing software development and maintenance problems is becoming increasingly intense. Project managers and system developers share an understanding with software managers that the demand for software is escalating rapidly, and that the costs for software usually (if not always) dominate the project cost. And although much of the impetus in this area has come from the Defense Department, the problems and their solutions have an even greater long-term impact in non-military applications.

The final solution to the basic problems of the software crisis (namely, human limitations) lies in applying modern software methodologies that are supported by a higher-order language such as Ada that encourages and enforces these principles.[16] The coupling of Ada with the tools and methodologies derived from the STARS program offers the software industry a significant inroad into the solution of the software crisis and its inherent problems.

The very viability of future systems rests with the successful infusion of Ada as the programming language. The DoD has focused its energy on solving the software crisis and has pleaded for industry support in this endeavor. Corporate awareness and response to this plea should be of paramount importance for future business considerations. Alert organizations will be prepared for the challenges presented by state-of-the-art technology in both the hardware and software domain, and should begin now to transition their strategies toward the ever-changing technological marketplace.



Bob Blasewitz is Unit Manager, Software System Design in MSR's Computer Control and Software Systems Department. He initially joined RCA in 1969 upon receiving the BSEE degree from the University of Maryland. His specialties have been real-time software systems, languages, and microcomputers, with major emphasis over the past six years on Ada. His involvement with Ada dates from its original definition, and includes work on programming support environment evaluation, interface definition, and Ada training courses and consultation. He is IEEE Software Engineering Standards chairperson for the Working Group on Ada as a Program Design Language. He has published a number of technical papers and is co-author of *Microcomputer Systems—Hardware/Software Design*, published by Hayden in 1982.

Contact him at:
Missile and Surface Radar
Moorestown, N.J.
TACNET: 224-3955

References

1. "Interim List of DoD Approved High Order Languages (HOL)," DoD Directive 5000.31 (November 24, 1976).
2. Ada Programming Language, ANSI/MIL-STD 1815A (1983).
3. R.E. Crafts, "Ada Debuts in Application Program," *Defense Electronics*, p. 67 (July 1982).
4. P.Freeman and A.I. Wasserman, "Ada Methodologies: Concepts and Requirements," *ACM Software Engineering Notes*, Vol. 8, No. 1 (January 1983).
5. Minutes of the IEEE Working Group on Ada as a Program Design Language (PDL), R.M. Blasewitz, chairperson. (1982-83)
6. "Software Technology for Adaptable, Reliable Systems (STARS) Program Strategy", Department of Defense, National Technical Information Service, Stock No. AD A128981 (March 1983).
7. E.W. Martin, "Strategy for a DoD Software Initiative", *Computer*, Vol. 16, No. 3, pp. 52-59 (March 1983).
8. J.D. Ichbiah, "Ada and the Development of Software Components," *Proceedings of the 4th International Conference on Software Engineering* (September 1979).
9. R.F. Brender and I.R. Nassi, "What is Ada?", *Computer*, Vol. 14, No. 6, pp. 17-24 (June 1981).
10. Brender and Nassi, p. 18.
11. Brender and Nassi, p. 19.
12. G. Booch, *Software Engineering With Ada*, Benjamin/Cummings Publishing Co., Menlo Park, California, p. 234 (1983).
13. C.A.R. Hoare, "The Emperor's Old Clothes," *Communications of the ACM*, Vol. 24, No. 2, pp. 75-83 (February 1981).
14. D.A. Fisher, "A Common Programming Language for the Department of Defense—Background and Technical Requirements," Institute for Defense Analysis Report p. 1191 (June 1976).
15. Booch, p. 369.
16. Booch, p. 9.
17. *Computer*, Vol. 16, No. 11, (entire issue devoted to STARS) (November 1983).
18. *IEEE Spectrum*, Vol. 20, No. 11, (entire issue devoted to "the new generation in computing") (November 1983).
19. E.A. Feigenbaum and P. McCorduck, *The Fifth Generation: Japan's Computer Challenge to the World*, Addison-Wesley Publishing Co., p. 14 (1983).

Your *RCA Engineer* Article . . .

Is in the Cards

Tips on technical article writing offered

If you are writing an article for the *RCA Engineer's* multidisciplinary engineering audience, the 48-page "Guide for *RCA Engineer* Authors" can show you practical ways to attract and maintain reader interest. The material specifically applies to RCA users. Experienced and novice authors can use many of the universally applicable principles, methods, and examples given here. Each chapter presents a series of Premises, Goals, and Methods that lead the author through the writing effort. Essays on side topics embellish the main text.

The first chapter, on "Article Mechanics & Specifications," defines the parts of a complete article package. The second chapter, on "Article Content," contains idea starters that will help you to gather the right information and artwork for a multi-disciplinary audience, organize the material, establish and keep audience interest, and write for the reader. The third chapter, on "Writing Style," illustrates by example the five major ways to improve written expressions, by making them active, lean, clearly qualified, symmetric, and specific.

To order, send your request for a free copy of the "Guide for *RCA Engineer* Authors" to the RCA Technical Excellence Center, 13 Roszel Road, P.O. Box 432, Princeton, NJ 08540. Call TACNET: 226-3090.

Meet your Editorial Representatives

The editorial contacts at your location are available to help you share the valuable information you have with the rest of the engineering community.

These Editorial Representatives (Ed Reps) can help you present that project or idea to a very important group—engineers and engineering management at RCA. Editorial representatives (for each major activity) are appointed, usually by the chief engineer. Basically, their objectives are to assist authors by stimulating, planning, and coordinating appropriate papers for the *RCA Engineer* and to keep the editors informed of new developments. In addition, they inform the editors of professional activities, awards, publications, and promotions. On these four pages you will find a photo, location, and phone number of your Ed Rep. They are alphabetically listed by division for your convenience. Get to know your Ed Rep, and develop your own contribution to the professional literature.

Broadcast Systems Division (BSD)

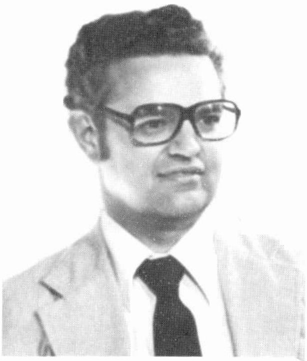
Jack Dearing*
Camden, N.J.
TACNET: 222-4688

Consumer Electronics (CE)

Francis Holt
Indianapolis, Ind.
TACNET: 422-5217

Eugene Janson*
Indianapolis, Ind.
TACNET: 422-5208

Larry Olson
Indianapolis, Ind.
TACNET: 422-5117



Byron Taylor
Indianapolis, Ind.
TACNET: 426-3367

Don Willis
Indianapolis, Ind.
TACNET: 422-5883

Corporate Technology

Tony Bianculli*
Princeton, N.J.
TACNET: 226-2111

*** Technical Publications Administrators**

Government Systems Division (GSD)

Merle Pietz*
Camden, N.J.
TACNET: 222-2161

Ed Master
Camden, N.J.
TACNET: 222-2731

Carol Coleman
Princeton, N.J.
TACNET: 229-2919

Frank Yannotti*
Princeton, N.J.
TACNET: 229-2544

Paul Seeley*
Burlington, Mass.
TACNET: 326-3095

Dale Sherman
Burlington, Mass.
TACNET: 326-3403

Dan Tannenbaum*
Camden, N.J.
TACNET: 222-3081

Thomas Altgilbers
Camden, N.J.
TACNET: 222-3351

Peter Hahn
Cherry Hill, N.J.
TACNET: 222-5319

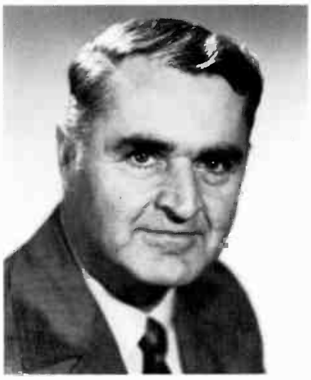
Don Higgs*
Moorestown, N.J.
TACNET: 224-2836

Graham Boose
Moorestown, N.J.
TACNET: 253-6062

Jack Friedman
Moorestown, N.J.
TACNET: 224-2112

*** Technical Publications Administrators**

National Broadcasting Company (NBC)



Bob Mausler*
New York, N.Y.
TACNET: 324-4869

New Products Division

R.E. Reed
Lancaster, Pa.
TACNET: 227-2485

Bob McIntyre
Ste Anne de Bellevue
TACNET: 514-457-9000

John Ovnick*
Van Nuys, Calif.
TACNET: 534-3011

Patent Operations

George Haas
Princeton, N.J.
TACNET: 226-2888

RCA Communications

Murray Rosenthal*
Princeton, N.J.
TACNET: 258-4192

Carolyn Powell
Princeton, N.J.
TACNET: 258-4194

Dorothy Unger*
Piscataway, N.J.
TACNET: 335-4358

RCA Laboratories

Eva Dukes
Princeton, N.J.
TACNET: 226-2882

RCA Records

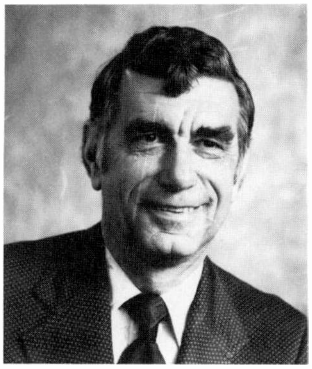
Greg Bogantz*
Indianapolis, Ind.
TACNET: 424-6141

RCA Service Company

Murray Kaminsky*
Cherry Hill, N.J.
TACNET: 222-6247

Ray MacWilliams
Cherry Hill, N.J.
TACNET: 222-5986

*** Technical Publications Administrators**

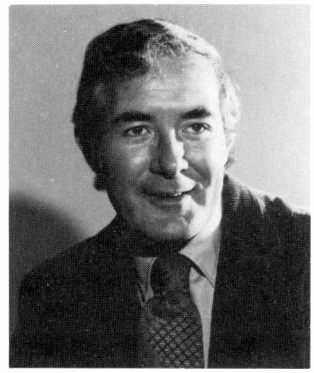


Dick Dombrosky
Cherry Hill, N.J.
TACNET: 222-4414

"SelectaVision" VideoDisc Operations

Nelson Crooks*
Indianapolis, Ind.
TACNET: 426-3164

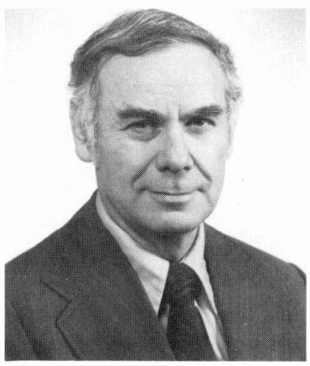
Solid State Division (SSD)



John Schoen*
Somerville, N.J.
TACNET: 325-6467

Harold Ronan
Mountaintop, Pa.
TACNET: 327-1473 or 327-1264

Dick Morey
Palm Beach Gardens, Fla.
TACNET: 722-1262

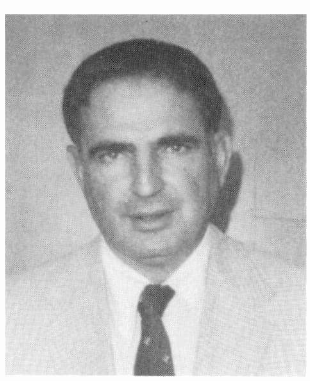


Sy Silverstein
Somerville, N.J.
TACNET: 325-6168

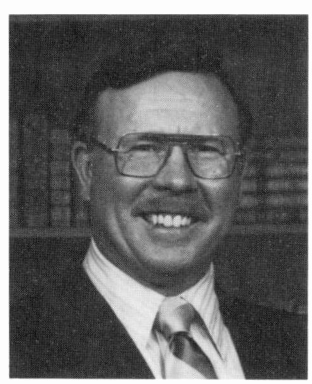
John Young
Findlay, Ohio
TACNET: 325-6168

Video Component and Display Division

Ed Madenford*
Lancaster, Pa.
TACNET: 227-3657



Nick Meena
Circleville, Ohio
TACNET: 432-1228

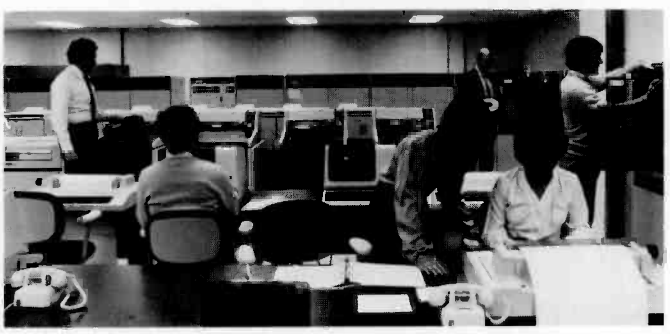


J.R. Reece
Marion, Ind.
TACNET: 427-5566

*** Technical Publications Administrators**

C.K. Liggett, Jr.

MSR centralizes software development activities in new computer center



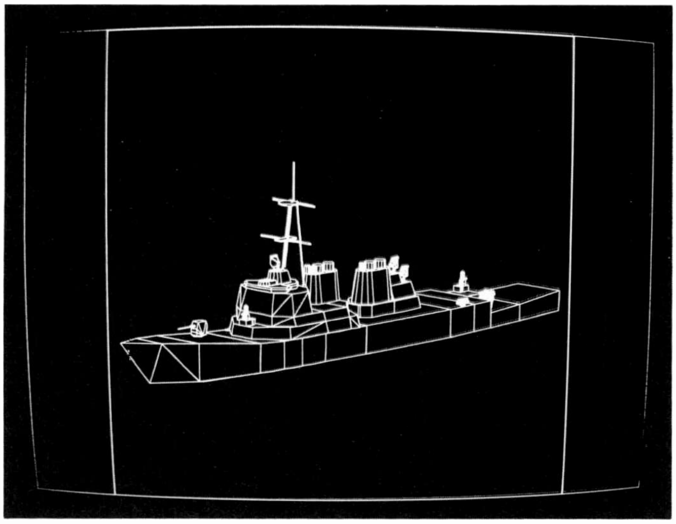
Overview of the Naval System Department's Program Generation, Analysis and Data Reduction Center, showing three DECSYSTEM-2060s behind the disk drives. The two VAX 11/782s are behind the DECSYSTEM-2060s.

In January 1984 a major new computer center became operational in Moorestown, New Jersey, to serve Missile and Surface Radar's rapidly expanding real-time computer software development requirements. The center occupies the entire ground floor of a newly acquired three-story, 90,000 square-foot office building on Route 38 in Moorestown, bringing together essentially all of MSR's computer programming functions under one roof.

In addition to expanded versions of two facilities previously located in the main Moorestown plant (the Naval Systems Department's Analysis and Data Reduction Center and the Engineering Department's Software Development Center), the center features two major new facilities. These are the AEGIS DDG Combat System Program Generation Center and Computer Program Test Site, established to support the development and test of the entire AN/UYK-43-based computer program complex for the AEGIS DDG 51 Combat System. These two special resources are closely patterned after the highly successful AN/UYK-7-based AEGIS CG 47 Combat System Program Generation Center and Program Test Site located in a nearby Computer Sciences Corporation facility.

The terminal room provides access to the Program Generation Center for compilation, test, and debug of computer programs as well as graphics terminals for simulation and design work.

Final manuscript received January 17, 1984.
Reprint RE-29-1-5



Three-dimensional graphics capability supports operations analysis, system simulation, and ship design applications.

The center consists of three major equipment rooms, together with a display console room, tape vaults, programmers' terminal rooms, a conference room, and work areas for the convenience of visiting subcontractor software personnel.

Program Generation Center and Analysis and Data Reduction Center

These colocated facilities support data reduction and performance analysis for the AEGIS program, simulation efforts for operations analysis of system design and development, and development and production of AEGIS computer programs to be hosted on the AN/UYK-43 tactical computers. Computer equipment includes three DECSYSTEM-2060s and two VAX II/782s, all manufactured by the Digital Equipment Corporation. Table I shows the equipment complement.

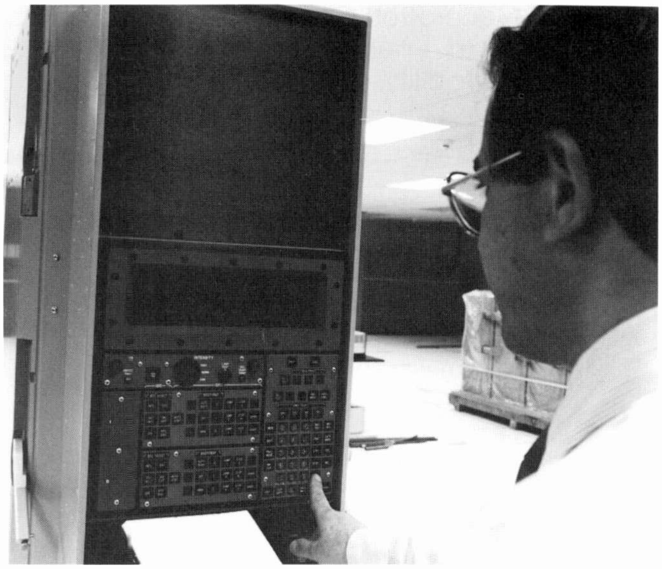
Table I. *Equipment complement for the Program Generation/Analysis and Data Reduction Center.*

Type (and Number) of Computers	Main Memory per Computer (Mb)	Disk Storage (Mb)
DECSYSTEM – 2060 (3)	*2	**(5) 440 **(4) 160
VAX11/782 (2)	8	(2) 516 (1) 256 †(1) 1400

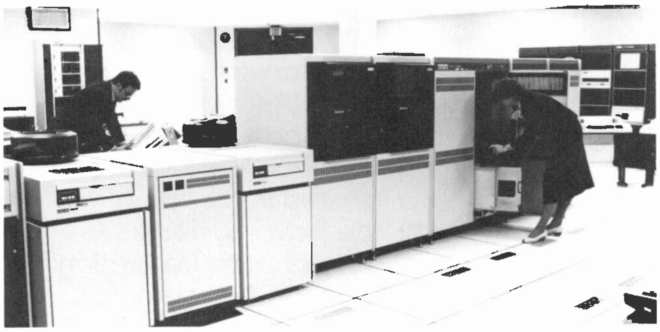
* 36-bit words
** All nine disk drives are shared by the three DEC 2060s
† Shared by the two VAXs

The DECSYSTEM-2060s support operations analysis and data reduction efforts. Simulation applications, including the use of 3-D color graphics, are also hosted on these systems. Languages used are FORTRAN, Pascal, and Basic. The Navy's Machine Transportable Support Software for 32-bit computers (MTASS/L) is available, including the CMS-2L compiler. The Accent R relational data base management system is used for information processing.

Program generation is accomplished using the VAX systems. These VAX 11/782s are the largest of the VAX series of computers. Each system has two Central Processing Units (CPUs) and 8 Mbytes of main memory. The languages available are FORTRAN and CMS-2L. The latter is provided with the Navy's MTASS/L environment, which also enables complete software



The Computer Program Test Site contains one of the first of the Navy's new AN/UYK-43 computers. This machine is being used to host the tactical computer programs for the AEGIS DDG 51 Combat System. The AN/UYK-43 provides a user-friendly maintenance panel for testing and debugging programs.



Overview of Software Development Facility showing VAX 11/780 in foreground and PDP-11s in background.

Table II. *Equipment complement for the Computer Program Test Site.*

Type (and Number) of Computers	Main Memory per Computer (Mb)	Disk Storage (Mb)
AN/UYK-7 (V) (3)	1	(*3) 67.4
AN/UYK-43 (V) (5*)	10	(*3) 151.6
AN/UYK-20 (Clock Simulator) (1)		

* One currently operational

systems for the Navy's AN/UYK-7 and AN/UYK-43 computers to be built.

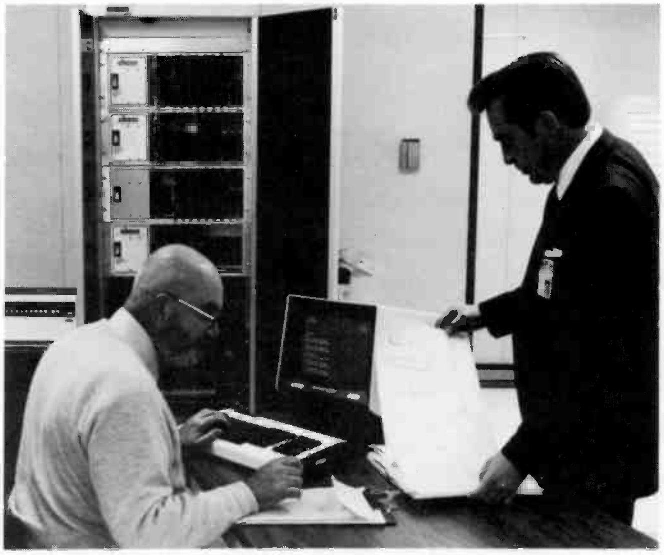
Computer Program Test Site

The Computer Program Test Site is used to test AEGIS tactical computer programs in a simulated environment using tactical computers and displays. Machines ranging from the 16-bit AN/UYK-20 to the 32-bit AN/UYK-7 and 32-bit AN/UYK-43 are available in shipboard configurations. The AEGIS Tactical Executive System (ATES) as well as programs for command, radar, and weapons control functions will be integrated prior to further testing at the Combat System Engineering Development Site or the Production Test Center. The equipment complement is shown in Table II.

Software Development Facility

The Software Development Facility is used for the broad range of software product development and support within MSR. Five Digital Equipment Corporation computers (plus an additional VAX planned for early 1984) are connected in a common network to support more than 100 users located at the Center, at the main Moorestown plant, and at other remote facilities. The network supports software

A direct-access console is provided for each computer in the SDF network to allow system debugging.



The SDF computer network switch allows user access to any of the five machines within the network.

product development, subcontract work, and proposal efforts. Typical examples are support of software/firmware requirements for modular gun fire-control systems and high-technology software development for the Military Computer Family Operating System.

In addition to the basic computer network structure (Table III), three Data General computers are used to support radar software development, maintenance, and simulation (these machines can be accessed from the network if required). Principal languages used within the Software Development Facility are FORTRAN, Basic, Pascal, and Ada.

Conclusion

The realization of this powerful center unifies MSR's computer programming functions under one roof and provides support for the more than 1000 design and development engineers at Moorestown.

Table III. *Equipment complement for the Software Development Facility.*

Type (and Number) of Computers	Main Memory per Computer	Disk Storage (Mb)
VAX 11/780 (1)	4 Mb	(2) 512
		(1) 121
VAX 11/780 (1)	4 Mb	(1) 456
PDP 11/70 (1)	896 Kb	(2) 88
PDP 11/44 (1)	512 Kb	(2) 67
PDP 11/24 (1)	512 Kb	(1) 20.8
Eclipse (1)	128 Kb	(1) 96
Nova 840 (1)	128 Kb	(1) 10
Nova 800 (1)	64 Kb	(1) 2.5

J. Tripoli

Legal protection of computer programs

As the number of profitable software innovations increases, it becomes important to know how the law protects the gains in the field. Here's an algorithm to get you through the legal maze.

Consider for a moment the various ways we encounter computers in our daily lives. We see these machines in businesses, schools, banks, places of employment, and places of amusement. Indeed, it takes an effort to think of classes of people in this country who are not affected in some way or another by computers. But, without the software to make them run and perform the desired tasks, these marvelous machines that can save time, money, and energy would be nothing more than expensive curiosities.

In the United States, the computer software industry has already reached the multibillion dollar sales level. With school districts all over this country emphasizing computer literacy, the software industry will apparently keep on growing for many years to come.

Abstract: *Computer software, an important business asset, should be protected. This paper explores the uses of patent, copyright, and trade secret law as mechanisms for protecting computer software. Some recent court decisions have shed new light on how each of these approaches, and combinations of approaches, can be effectively used. The comparative analysis is discussed in terms of a practical example taken from a landmark Supreme Court decision.*

Final manuscript received October 28, 1983.
Reprint RE-29-1-6

Clearly, computer software can be a valuable businesss asset. Valuable assets should be protected under the law. This article will explore the present-day legal alternatives for protecting computer software. Our point of view will be that of the creator of a particular software package.

A practical example

Jim Diehr and Ted Lutton worked for a Midwest corporation that makes a variety of products molded from rubber compounds. When Jim and Ted first started working for the corporation, these compounds were manually placed in heated (and thermostatically controlled) molding presses. A timer was started when the press was closed. When the rubber-curing process was complete, an alarm went off and the operator manually opened the press.

But, there was a problem.

The press thermostats were only accurate to ±2 percent. In addition, different operators took different amounts of time to load the compound into the press. There was a difference in starting temperature that depended on the amount of time the mold had been cooling before the press was closed. Jim and Ted were told to devise a system that would consistently provide nearly exact cure times for the rubber material being molded.

Eventually, Jim and Ted designed a system that used well-known computers and data storage banks containing the time-temperature cure data for the particular compounds being used. A surveillance system monitored, almost continuously, the mold temperature. The temperature data, along with some other data and the elapsed time, were fed to the computer. The computer was programmed to solve the well-known Arrhenius equation. This computation was continually redone and a new time-temperature cure curve was generated every ten seconds for the particular compound being molded. When the calculated cure time equalled the elapsed time, the computer signalled an electromechanical device, which opened the mold.

Jim and Ted, working together, designed the system and did all of the programming. The Plant Supervisor immediately recognized this work as a significant improvement over the old process and urged Jim and Ted to see the company attorney concerning legal protection for the concept.

The alternatives

Some of you will recognize that this example is taken from the actual facts involved in the United States Supreme Court decision in *Diamond* v. *Diehr and Lutton.*[1] In the actual case, the attorney suggested seeking patent protection. But, there are other ways of protecting the work product. The attorney could have suggested relying on copyright protection for the involved computer programs; or, maintaining the system as a trade secret; or, he might have suggested some combination of protective techniques. We will explore each of the alternatives and try to assess the advantages and disadvantages of each course of action.

Patent protection

When the United States Patent and Trademark Office (PTO) examined the Diehr and Lutton patent application, the claims were rejected as being drawn to nonstatutory subject matter. In essence, the Examiner was saying that the claims in the application, which went to a method of operating a rubber-molding press, were steps that were being carried out by a computer under the control of a stored program. The Examiner reasoned that this was not proper subject matter for a patent under the Supreme Court decision in *Gottschalk* v. *Benson*.[2] In the Benson case, the Supreme Court had said that patent protection was not available for a computer-controlled system that translated binary-coded decimal numbers into pure binary numbers. It was felt that the Benson subject matter was not "any new and useful process, machine, manufacture or composition of matter, or any new or useful improvement thereof . . . "[3] as required by the current Patent Act. In Benson, the Court basically held that it would be wrong to allow patent protection for a mathematical formula or algorithm.

The Diehr and Lutton application went to the PTO Board of Appeals, which upheld the rejection by the Examiner. The case was then appealed to the Court of Customs and Patent Appeals (CCPA), where the rejection was reversed. The PTO then took the case to the Supreme Court, which ruled that the claims were proper for patent protection. This was the first time that the Supreme Court found a computer-program-related invention to fall within the definition of patentable subject matter. The Supreme Court keyed in on the fact that the invention related to a method that transformed raw uncured synthetic rubber into a different state or thing—that is, a cured rubber product—when the press was opened. The Court recognized that the claims involved a well-known mathematical equation or algorithm, but the claims did not seek to preempt the use of that equation. Instead, the equation was used in conjunction with all the other steps in the process, such as continually monitoring the temperature, continually recalculating the cure time, and opening the press at the appropriate time. Of course, once it was determined that the claims fell

United States Patent [19] | [11] **4,344,142**

Diehr, II et al. | [45] **Aug. 10, 1982**

[54] **DIRECT DIGITAL CONTROL OF RUBBER MOLDING PRESSES**

[75] Inventors: **James R. Diehr, II**, Troy; **Theodore A. Lutton**, Birmingham, both of Mich.

[73] Assignee: **Federal-Mogul Corporation**, Southfield, Mich.

[21] Appl. No.: **602,463**

[22] Filed: **Aug. 6, 1975**

Related U.S. Application Data

[63] Continuation of Ser. No. 472,595, May 23, 1974, abandoned, which is a continuation-in-part of Ser. No. 401,127, Sep. 26, 1973, abandoned.

[51] **Int. Cl.**[3] **G06F 15/46; B29H 5/02**

[52] **U.S. Cl.** **364/473**; 264/40.1; 264/325; 364/476; 425/144; 425/156; 374/53; 374/102

[58] **Field of Search** .... 235/151, 151.1, 150.1; 494/1; 264/40, 315, 347, 297, 326, 236, 325; 425/135, 143, 144, 149, 150, DIG. 44, 29, 32, 38, 162, 165, 155, 169, 170

[56] **References Cited**

U.S. PATENT DOCUMENTS

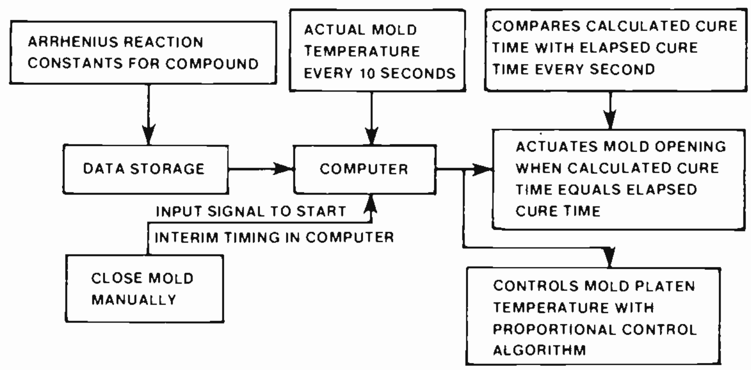
3,579,626	5/1971	Brittain	264/315 X
3,649,729	3/1972	Davis et al.	264/40
3,659,974	5/1972	Neugroschl	264/40 X
3,718,721	2/1973	Gould et al	264/40
3,819,915	6/1974	Smith	235/151 X
3,980,743	9/1976	Smith	264/40.2

Primary Examiner—Joseph F. Ruggiero
Attorney, Agent, or Firm—Owen, Wickersham & Erickson

[57] **ABSTRACT**

Rubber-molding presses, which are closed manually upon installation of pieces of rubber compound. are opened automatically by a system which continuously calculates and recalculates the correct cure time and is actuated when the calculated cure time equals the elapsed cure time. An interval timer starts running from the time of mold closure, and the temperature within the mold cavity is measured often, typically every ten seconds. The temperature is fed to a computer which also is given access to the time-temperature cure data for the compound being molded, and the computer calculates and recalculates every time the data as to temperature is presented, until the total picture of time and temperature presents to the computer the time at which the material is fully cured. Then the computer signals for automatic opening of the mold press. Many presses can be controlled by a single computer, which still operates to recalculate the data about every ten seconds, and the time-temperature cure data for the compound can also be modified by information from a rheometer.

11 Claims, 4 Drawing Figures



This process, run in conjunction with computer programs, is patentable. The program listing, alone, is not patentable.

within the proper subject matter for patent protection, all the other standards for patentability had to be applied. For example, the claims had to be new, useful, and unobvious.

Several software-related CCPA decisions were made after the Supreme Court's Diehr decision. The current test, in view of Diehr, seems to be to look at the subject matter of the claims of a patent application as a whole. If the claimed invention seeks to preempt a mathematical algorithm, then it is not proper subject matter for a patent. If, however, the claimed invention goes to the application of a mathematical algorithm to physical elements or process steps, then it is proper subject matter and we can proceed to consider the other issues of patentability.

At least one "aftermath" case[4] shows that we must distinguish between *mathematical algorithms* and *algorithms that are not mathematical.* If we are dealing with nonmathematical algorithms we are already outside of the Diehr test, and the invention is treated as a process or method. But, we must be careful here. The court can find a *mathematical algorithm* even when such an algorithm is not expressed in mathematical terms in a claim.[5] That is, a claim may include an equation written in words rather than mathematical symbols.

At this point, a clarification should be made. The listing of instructions for execution by a computer does not fall within

1. 450 U.S. 175, 209 USPQ 1 (1981)
2. 409 U.S. 63, 175 USPQ 673 (1972)
3. 35 USC 101
4. *In re* Pardo and Landau, 684 F.2nd 912, 214 USPQ 673 (CCPA 1982)
5. *In re* Meyer and Weissman, 215 USPQ 193 (CCPA 1982)

Resources for inventors

Publications

The following are available from the Commissioner of Patents and Trademarks, Washington, DC 20231:

- *Q & A about patents*—Brief, non-technical answers to frequently asked questions. Free.
- *Q & A about plant patents*—Same as above but for plant patents. Free.
- *Q & A about trademarks*—Same as above but for trademarks. Free.
- Patents—Specifications and drawings of patents cost $1 each. When ordering, give the patent number or full name of inventor and approximate date of issuance.
- Trademarks—Copies cost $1 each. When ordering, give registration number or name of registrant and approximate date of registration.

These publications are available from the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402:

- *General information concerning patents*—Simply written information about applying for and granting of patents. $2.25.
- *General information concerning trademarks*—Same as above but for trademarks. $1.70.
- *Patent official gazette*—Official journal of the Patent Office contains a selected figure of the drawings and an abstract of each patent granted during the previous week. It also lists patents available for license or sale and latest rules changes. Annual subscription: $300. Per copy: $4.00.
- *Trademark official gazette*—Same as above but for trademarks. Annual subscription: $88.40. Per copy: $1.70.
- Annual Indexes—Published for both patents and trademarks. Prices vary, depending upon size of publication.
- *Manual of classification*—Lists numbers and descriptive titles of the more than 300 classes and 100,000 subclasses used in the subject classifications of patents, with an index. Subscription: $60.
- *Manual of patent examining procedures*—Detailed reference work on patent examining practices. Subscription, including basic manual, quarterly revisions, and rules changes: $39.
- *Patents and inventions, an information aid to inventors*—How to decide whether to apply for a patent, how to obtain patent protection, and how to promote inventions. $1.75.
- *Patent laws*—All the statutes. $4.50.
- *Patent attorneys and agents registered to practice before the U.S. Patent and Trademark Office*—A geographical listing of those legally registered to represent inventors before the Patent Office. $8.
- *Trademark rules of practice of the Patent Office with forms and statutes*—Rules, laws, and forms relating to trademarks. $5.
- *37 code of federal regulation*—Rules governing Patent Office procedures. $5.50.

On-line databases

- LEXPAT—Will search and retrieve the full text of patents issued since 1975, plus new patents issued yearly. Key-word

For more information and assistance in searching on-line databases, see your local RCA Librarian.

the classes of subject matter that can be protected under the patent law. For example, if Diehr and Lutton had written a few hundred lines of code, in, say, Fortran, that Fortran listing would be considered as a collection of printed matter. Printed matter is not included in any of the categories of proper subject matter for patent protection. Therefore, the Fortran listing could not be patented. The methodology or process that Diehr and Lutton used to transform uncured rubber into a cured molded product included steps run in conjunction with computer programs. This process was patentable. Their program listing, alone, could not be patented.

By seeking patent protection for their design, Diehr and Lutton received the right to exclude others from making, using, or selling their automated rubber-curing process for a period of 17 years from the date the patent issued. The protection covers the functional concept expressed by the claims in the issued patent.[6]

Before receiving that protection, Diehr and Lutton had to meet the rigorous standards of patentability and, in this case, several years elapsed between the filing of the application and the issuance of the patent as a result of the litigation. Of course, this was precedent-making litigation. In the usual or ordinary case, it takes one and one-half to three years to issue a patent in the United States.

Copyright protection

Diehr and Lutton could have decided (and perhaps did decide) to seek copyright protection for the program that they wrote to control the rubber-curing process.

Copyright protection, in general, prohibits copying the form of expression of a work or the particular way in which the work is set down in tangible form. It does not protect the underlying abstract ideas in the work. Thus, if two people independently produce similar works on the same subject, without copying from one another, there is no copyright infringement. As a result, anyone in the rubber-curing industry could sit down and independently program a computer to solve the Arrhenius equation without fear of infringing the assumed copyright of Diehr and Lutton. But the Diehr and Lutton software could have taken many man-months or years to produce; this alone would make it desirable to protect the software listings *per se.*

There was a question under the Copyright Act of 1976 as to whether or not certain types of software, specifically machine-readable software, could or should be protected by copyright. That issue was resolved by Congress in the Computer Soft-

6. U.S. Patent No. 4,344,142

search is available. Contact: Mead Data Central, 9333 Springboro Pike, Dayton, OH 45401, (513) 859-1611.

- CLAIMS—Umbrella title for several different patent databases including *CLAIMS/UNITERM,* which gives access to chemical and chemical-related patents issued since 1950. Includes subject indexing for each patent from a controlled vocabulary to speed retrieval of chemical structures and polymers. *CLAIMS/U.S. PATENTS ABSTRACTS WEEKLY* is the latest weekly update and records from the current month. *CLAIMS* is published by IFI/ Plenum, Arlington, Va. Contact: Dialog Information Services, Inc., 3460 Hillview Ave., Palo Alto, CA 94304, (415) 858-3785.

- *Hi tech patents: data communications*—Full text of "Hi Tech Patents: Data Communications" newsletter, which provides bibliographic citations and abstracts of patents in the field. Covers Australia, Canada, U.K., as well as the U.S. Published by Communications Publishing Group, Inc., 101 Verndale St., Brookline, MA 02146, (617) 566-2372. Published on-line by NewsNet, Inc., 945 Haverford Road, Bryn Mawr, PA 19010, (800) 345-1301.

- *Hi tech patents: fiber optics*—Similar to above but for fiber technology.

- *Hi tech patents: laser technology*—Similar to above but for laser technology.

- *Hi tech patents: telephony*—Similar to above but for telephony technology.

- National Technical Information Service—Current patent full-text and bibliographic files produced weekly; patents issued since January 1969 by title or by company; patents issued by inventor since 1975; and more. On magnetic media. Contact NTIS, 5285 Port Royal Rd., Springfield, VA 22151.

Inventors' associations

- Intellectual Property Owners, 1000 M St., NW, Washington, DC 20005, (202) 466-2396. Group acts as lobbyist for corporate and individual members to convince Congress and the Administration of the importance of intellectual property protection such as copyrights, patents, and trademarks.

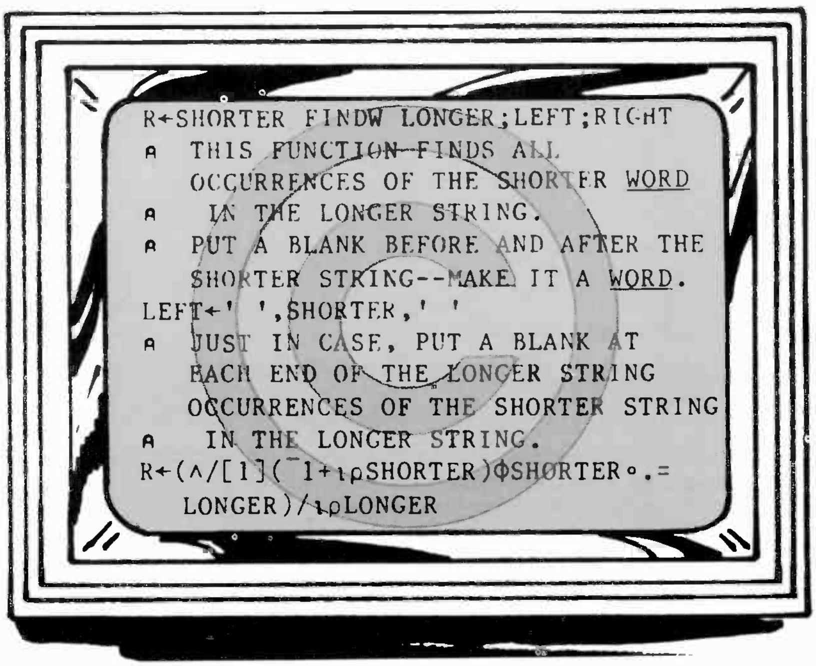
- National Congress of Inventor Organizations, 345 E. 47th St., New York, NY 10017, (212) 705-7943. Coalition of inventors' groups to further creativity and innovation. Among its 25 members are California Inventors Council, Inventors Workshop International, Inventors Clubs of America, National Inventors Cooperative Association, Affiliated Inventors Foundation, Inc., and International Association of Professional Inventors.

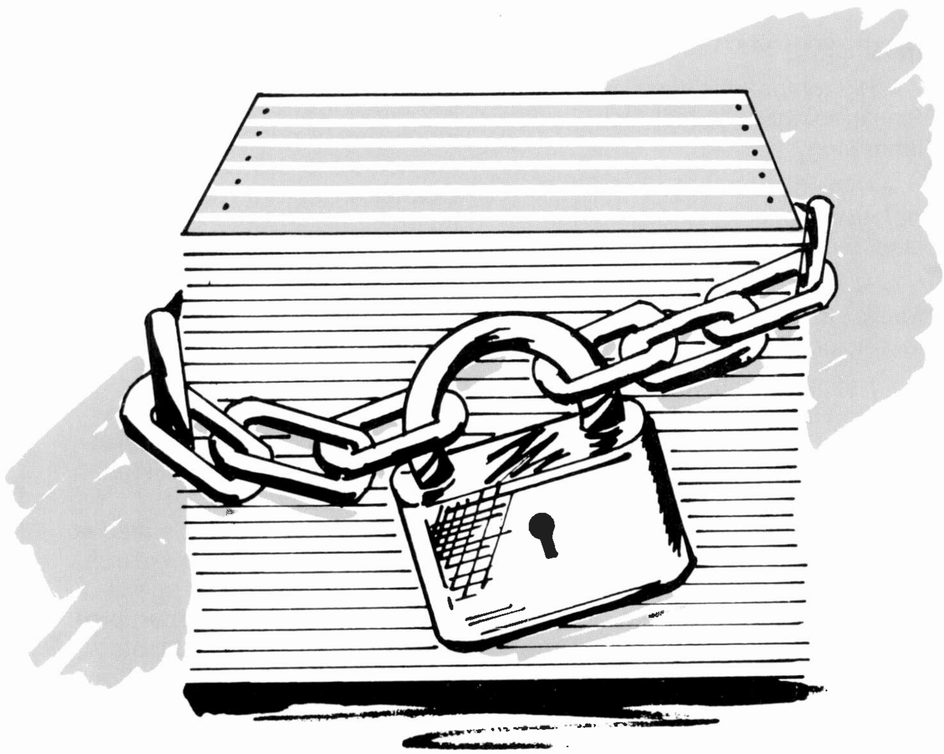
- Small Business Innovation Research Program. Established in 1977, this program provides phased grants to support advanced research. It is aimed at small firms with strong research capabilities, rather than at individuals, and will not support product development, pilot plants efforts, or clinical research. Contact: National Science Foundation, 1800 G St., NW, Washington, D.C.

- Technology Commercialization Program. Established in 1976, it is aimed at anyone with a technology that can be commercialized provided that a minority inventor, investor, or innovator benefits from the project. Contact: Department of Commerce, Washington, DC 20230.

ware Copyright Act of 1980. The Act states that "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result . . . " is copyrightable subject matter. There is no real question any longer that the assumed Diehr and Lutton software could be protected under the Federal Copyright Act whether in source or object code.

Under the present statute, the copyright comes into existence immediately when the work is fixed in tangible form. Thus, the Diehr and Lutton copyright came into existence the moment they finished writing the program regardless of the language used in making the listings. A copyright gives the owner the exclusive right to make copies, to prepare derivative works, to distribute the works by sale, by other transfer of ownership, or by rental, lease, or lending. The copyright owner also gets the





exclusive right to perform and display the work publicly.

Copyright protection comes into being automatically under the Federal Copyright Act for unpublished works. When the work is published, a copyright notice on the work is required and registration of the copyright in the Copyright Office of the Library of Congress should be made. The copyright notice consists of three elements: (a) the letter *c* in a circle, or the word "Copyright," or the abbreviation "Copr."; (b) the name of the copyright owner; and (c) the year of first publication of the work. Registration is required as a prerequisite to a suit for infringement and for recovery of certain statutory damages. The Copyright Act is forgiving in the sense that publication without notice is not an immediate and fatal bar to protection under the law, but care must be taken to rectify such publication without notice.

Among the works generally accepted by the Copyright Office for registration are programs in human-readable form, that is, source programs, user manuals, and flow charts. In addition, the Copyright Office will accept, under the "Rule of Doubt," programs in machine-readable form—for example, programs expressed in object code. In the "Rule of Doubt" cases, the copyright owner may have to shoulder an additional burden of proof to demonstrate the existence of a valid copyright.

There is a current split of opinion as to whether or not a program encoded in a read-only memory (ROM) is proper subject matter for copyright protection. One lower court[7] held that ROMs become part of the computer hardware and, therefore, become part of the mechanical process. Mechanical processes are not proper subject matter for copyright protection. But other courts, expressing what appears to be the majority view, have held that ROMs represent another form of the source code, and therefore, the object code burned into a ROM should be covered by the copyright on the source code. The latest decision in *Apple* v. *Franklin*[8] held that operating programs encoded in ROMs are proper for copyright protection. Incidentally, the artwork used to make integrated circuits, including ROMs, is proper subject matter for copyright protection as pictorial or graphic works.

The formalities for copyright registration of programs are relatively straightforward and inexpensive. A simple form is completed and sent to the Copyright Office with a ten dollar fee and the required number of copies of the work. When a program is first published in machine-readable form—for example, on tape or disc—the Copyright Office will accept one copy of the identifying portion of the work in human-readable form. The portion must include the copyright notice. If a source-code listing is very lengthy, then the Copyright Office will accept the first and last 25 pages of the listing. These deposit copies are public records and are available for inspection in the Copyright Office. The duration of the copyright is typically for the life of the author plus 50 years. When a work, such as a program, is created by an employee within the scope of his or her employment, it is termed a "work made for hire." The Copyright Act provides that title to the copyright in a work made for hire resides in the employer, not the employee. The duration of this type of copyright is 75 years from the date of publication or 100 years from the date of creation, whichever occurs first.

Trade secret protection

As a third option, Diehr and Lutton could have decided to maintain their work as a trade secret. The law concerning trade secrets is based on state statutes and the common law resulting from court decisions. There is no federal law concerning trade secrets.

The Uniform Trade Secrets Act is a model statute that some states have used as a basis for enacting their own trade secret statutes. The Uniform Act defines a trade secret as follows:

> "Trade Secret" means information, including a formula, pattern, compilation, program, device, method, technique, or process that:
>
> (i) derives independent economic value, actual or potential, from not being generally known to, and not being readily ascertainable by proper means to, other persons who can obtain economic value from its disclosure or use, and
>
> (ii) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy.

Trade secret protection can go to almost anything that gives a competitive edge and is not known to others (or easily reverse-engineered) and is kept secret by the owners. There is no government agency involvement in securing the protection.

It can be assumed that the overall design of the Diehr and Lutton system, including the application software, had value and gave the owner of the information a competitive edge over others. To acquire the protection of the trade secret law, the owner

7. Data Cash Systems, Inc. v. JS&A Group, Inc. 480 F. Supp. 1063 (N.D. ILL. 1979), affirmed on other grounds 628 F.2nd 1038 (7th Cir. 1980)

8. Apple Computer Inc. v. Franklin Computer Corporation, 219 USPQ 113 (3rd Cir. 1983).

Table I. *Brief comparison of software-protection techniques.*

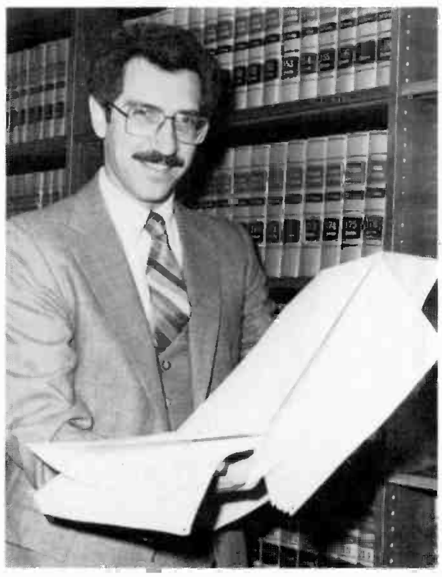
Parameter	*Patents*	*Copyrights*	*Trade Secrets*
What is protected?	Underlying ideas, concepts, or methods used in the system.	Original works of ownership fixed in any tangible medium of expression.	Formulas, ideas, devices, compilations, programs, techniques.
How is protection obtained?	File an application in USPTO.	For published work: Place copyright notice on work. For unpublished work: Automatic upon completion.	Maintain secrecy.
When does protection start?	Issuance of patent.	Upon completion of the work.	At creation of secret information.
How long does protection last?	17 years from issue date of patent.	Life of author plus 50 years. Work for hire: 75 years from publication, or, 100 years from creation, whichever comes first.	As long as information is kept secret.
Criteria for providing protection	Must fall within statutory subject matter and must be new, useful and unobvious.	Any original work of authorship.	Must have value and not be generally known to others.
Who infringes?	Anyone who, without authorization, makes, uses, or sells claimed invention.	Anyone who, without authorization, copies, makes derivative works, publicly performs or distributes copyrighted work by sale, lease, rental (subject to statutory exceptions).	Anyone who violates a confidential relationship or uses unlawful means to gain access to trade secret information.
Available remedies for infringement	Injunction, reasonable royalties.	Injunction, statutory damages, royalties	Injunction, damages, criminal sanctions for theft.
Applicable Law	Federal Patent Act.	Federal Copyright Act.	State Trade Secret Statutes.

of the information must do more than merely declare an intent to treat the information as a trade secret. Positive steps must be taken to maintain security. Is there a stated company policy in respect to keeping proprietary information secret? How many people in the company have access to the information? Is the information kept in a safe or otherwise guarded location? Are copies of the information numbered and controlled? Is there an agreement with employees that prevents them from using this type of information when they are no longer employed by the company? Are disclosures of the information to outsiders made only pursuant to confidential disclosure agreements? These are some of the factors a court might consider in determining whether or not reasonable efforts had been made to maintain secrecy.

If the court finds that all the elements of the trade secret are present, it can award damages and/or stop the unauthorized use of the information. The duration of the trade secret can last indefinitely, as long as the information is kept secret and does not otherwise become available to the public.

To demonstrate infringement of a trade secret, one would have to show that there was a breach of a confidential relationship or that unfair means were used to discover or disclose the information. Very often, trade secret litigation arises where a former employee takes trade secret information with him and undertakes to use it in competition with the former employer. To a lesser extent, the litigation in this area involves the theft of information in the form of industrial espionage.

Joseph Tripoli is the Director of the Electronic Systems Group of RCA Patent Operations. In 1965, he joined RCA MSR at Moorestown, as a microwave engineer. In 1969, he transferred to Patent Operations, Princeton, as a Member of the Patent Staff. After he was graduated from the Temple University School of Law with the J.D. degree, he was appointed Patent Counsel. He served as Resident Patent Counsel in Camden from 1972 through 1974. In December 1974, he was appointed Staff Patent Counsel. He served in this post until February 1978 when he was promoted to Managing Patent Attorney in the Consumer Products Group of Patent Operations. In October 1982, he was appointed to his present position where he coordinates the Patent activities in support of the systems work at RCA.
Contact him at:
RCA Patent Operations
Princeton, N.J.
TACNET: 226-2992



Which form of protection is best?

By now it should be clear that the selection of the form of protection for software depends on the facts of a given case. Patent protection does not cover the software listing, per se, but can be used to protect the system in which the software is used. Since the patent covers abstract ideas, it can be used to exclude or control a variety of approaches to the solution of a given problem.

Copyright protection is relatively simple to secure compared with patents. Even though the copyright covers only the form of the expression of a program, this can be a very important protective mechanism. A copyright notice (followed by registration) should be placed on all programs that appear to have general commercial value or widespread appeal.

Since patents and copyrights protect two distinctly different areas, that is, ideas on the one hand and forms of expression on the other, it probably would be wise to seek both forms of protection in the appropriate case. For example, where a program is created to perform in a new and unobvious system and it is likely that others will work toward obtaining the inventive system, it would be desirable to protect both the system and the program to run the system.

The trade secret approach should probably be reserved for those situations where the program has an extremely important intrinsic value to the organization. For example, in cases where the very life or well-being of the organization is riding on a particular program, then all efforts should be directed toward preserving the secrecy of the program. The number of disclosures made, even pursuant to confidential disclosure agreements, should be carefully controlled. It becomes difficult to maintain a trade secret when a large number of people have been given the information.

The question is, "Which protective measure, or combination of measures, should be taken to protect computer software?" The answer, as in so many legal issues, is "It depends." The best course of action is to seek advice from your attorney before charting a course and before taking actions that could impair the ability to use these various forms of protection.

E.A. Knapp

WNBC-TV expands live news coverage

To achieve competitive news ratings, stations need live electronic journalism. The behind-the-scenes engineering provides the technological foundation for these visible journalistic achievements.

WNBC-TV, the NBC Owned and Operated flagship station in New York City, shares one of the major challenges faced by every television station in the United States. The challenge is to increase news coverage for WNBC-TV viewers while at the same time remaining cost efficient.

Pioneering live TV news

WNBC-TV, one of the pioneer broadcasters in the use of microwave transmission of live television news, started over 30 years ago with portable (for those days) receivers and transmitters and large parabolic antennas mounted on tripods. Eventually, large plastic domes were installed on the four corners of the 66th floor of the RCA building in the heart of New York City, the location of the Channel 4 studios. These domes were manned by a technician who would manually move the portable dish to receive the incoming microwave signals. Although this system required sending people to the roof for each microwave remote, it worked well as long as only one or two microwave shots were scheduled a month.

At that time, all of WNBC-TV's local news was covered with film cameras. With the introduction of smaller portable television cameras and videotape machines, WNBC-TV News wanted to provide more live coverage to bring Channel 4 viewers news as it was actually happening. The objective was to improve the manned domes on the roof or to develop a high-tech replacement.

Abstract: *WNBC-TV provides news to over 20-million people in the tri-state area of New York, New Jersey, and Connecticut. Expansion of the existing system of microwave pickups of live news was necessary to better serve the Channel 4 viewers.*

This expansion included the establishment of news bureaus with live picture transmission, microwave relay facilities to extend range, new microwave-equipped mobile vans, and diversification of receive sites. Presented in this paper is the development of this expanded microwave system.

Final manuscript received October 17, 1983.
Reprint RE-29-1-7

The beginnings of electronic journalism

Because the WNBC-TV transmitter was located on the 85th floor of the Empire State Building (the tallest building in New York City at that time), it was decided to install receiving antennas that could be controlled by the transmitter engineers from inside the transmitter room. This evolved into four custom-built dual-band quad horn antennas mounted on the four corners of the 85th floor parapet. One of these antennas is shown in Fig. 1. There was a switching system to allow for horizontal, vertical, or left- or right-circular polarization selection. The antennas had 18-dB gain and 18-degree half-power beam width (HPBW) on 2 GHz, and 29 dB gain and 5.2-degree HPBW on 7 GHz. Each horn was panned and tilted by a multiwire remote-control system and could cover more than a 90-degree sweep with some overlap between antennas.

This was truly a microwave system designed for Electronic Journalism (EJ), where a crew with a portable television camera and portable microwave transmitter could send live pictures back to our studios with minimal set-up time. Four 13-GHz microwave circuits were established between the Empire State Building and the RCA building so that each of the four antennas with their own receivers could simultaneously be fed to our studios.

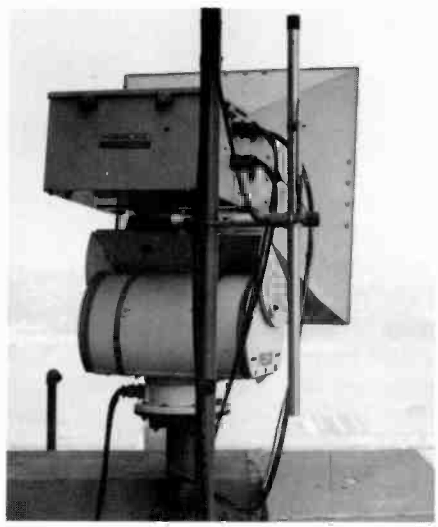


Fig. 1. *Old system's Quad horn, Empire State Building. This horn could only be controlled from the Empire State Building site.*

The system served WNBC-TV well for several years and was also used by NBC Network News and NBC Network shows such as the "Today Show." But, as the news competition in New York City grew, the need for more live coverage grew. Several other stations had installed microwave receiving equipment at the Empire State Building for their news coverage. This caused interference problems when three or four stations would cover the same story. It became apparent that the "old" receive system needed diversification and modernization. The wide acceptance

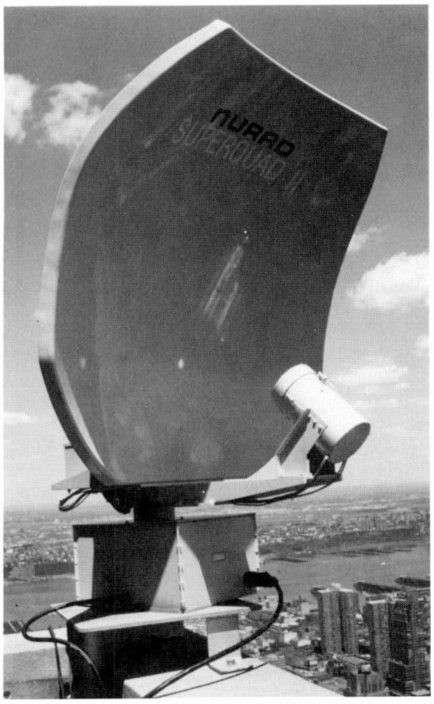


Fig. 2. *New System's Superquad, Empire State Building, fully remote-controllable from the WNBC-TV studios.*

angle of the antennas and high receiver-noise levels needed improvement over a simple diode front-end type receiver. Low noise amplifiers and selective tuned circuits offered by modern receivers were necessary to overcome interference.

Reevaluating electronic journalism needs

An evaluation of the needs and operations was researched by the WNBC-TV Engineering Department, defining special problem areas. There was co-channel interference from other broadcasters. The antenna beam width was such that reflections way off the main lobe were being received along with the desired signals. The receiver sensitivity was poor and the noise level was high by today's standards. The "canyons" of New York City, created by the tall buildings, were making it impossible to cover news without careful positioning of the transmitting antennas. Something better than portable equipment thrown into a truck or car was needed for fast-breaking news. And last, but not least, there was a concern for live pickups from WNBC-TV News bureaus that were to be located in New Jersey and Long Island.

To overcome these problems, WNBC-TV Station Management, News Management, and Engineering Management developed the following goals:

1. Provide "state-of-the-art" receiving equipment.
2. Diversify the receiving locations in New York City.
3. Establish distant microwave-relay facilities.
4. Design small mobile vans with built-in microwave systems.
5. Provide a central control point for all the receive sites.
6. Provide live pickup links from the News Bureau.

The WNBC-TV Engineering Department implemented the project, except for Goal 4. Since the News Department crews would be manning the vans, they were given the responsibility to design and build the units.

The projects

New equipment

All of the latest microwave equipment was studied to determine which devices would best suit our purposes. The choice of receivers and antennas was wide; however, Nurad, Inc., of Baltimore, Maryland, offered a unique antenna that we eventually chose. Their "Superquad" Model 45SQ3 provides simultaneous reception of 2 and 7 GHz, is remote controllable over telephone lines, has built-in low-noise preamplifiers and the reflector offered high gain and narrow beamwidth. The reflector also has a cosecant-squared elevation beam shape that does away with the need to tilt the antenna. Moreover, the sense of polarization can be changed by remote control from clockwise circular to counterclockwise circular to vertical linear or to horizontal linear. Figure 2 shows this new antenna mounted on the Empire State Building.

In addition, Nurad offered the latest 2- and 7-GHz rack-mounted Harris-Farinon frequency-agile receivers with the capability to split each channel into three segments. Frequency change, channel split and fault alarms all tie into the Nurad remote-control system. The final clincher in picking Nurad was that they had the expertise and manpower to do complete "turn-key" installations.

Diversification of sites

Two of the large plastic domes were still available on top of the RCA building for receive sites and were located on the northeast and southwest corners of the building. It was decided to install only two Superquads on the Empire State Building and to place two more Superquads on the RCA building. One of the dome-mounted antennas is shown in Fig. 3.

Fig. 3. *WNBC-TV's Superquad inside plastic dome on the RCA Building. The dome was originally intended for manned operation.*

The diagonally opposite corners from RCA were chosen at Empire (the northwest and southeast corners) for the Superquad positions. This gave us four receiving antennas but better than a half-mile separation between antenna pairs.

Remote relay facilities

WNBC-TV searched for sites that would provide for increased coverage in New Jersey, Connecticut, and Long Island while providing a microwave path back to New York City. A New Jersey Public Broadcasting tower was selected in Warrenville, New Jersey because it gave WNBC-TV the coverage area outlined by the News Department and a path back to the city. A second site was found in Plainview, Long Island, that gave us the expanded Long Island coverage and, due to its proximity to Connecticut, expanded coverage in that state. This tower-mounted antenna is illustrated in Fig. 4.

Investigation of frequencies for the link between the relay sites and the city caused some concern. The paths were both approximately 20-miles long, therefore, 13 GHz would not provide acceptable fade margins. We did not want to use 2 or 7 GHz for these inter-city relays since those frequencies are heavily used for news pickups. A relatively unused band located at

Fig. 4. *Tower mounted Superquad on Long Island. A radome enclosure protects the antenna from the weather.*

Fig. 5. *WNBC-TV mobile microwave vans. Fold-over masts are shown extended and retracted.*

Fig. 6. *Microwave receiver and transmitter control room. The operator controls all microwave receivers along with the main broadcast transmitters.*

2.5 GHz offered four channels for television. The concern was that, because microwave ovens operate in this band, there was a possibility for interference to the New York City receivers. In a careful review of the situation, we concluded that since we were locating the receivers on the 66th floor of the RCA building, we could, in all probability, escape the usually low level of microwave oven radiation. We have encountered no problems in almost two years that we have been running these 2.5 GHz links.

Mobile units

The WNBC-TV News Department designed and built six mobile vans that contained separate antennas for 2 and 7 GHz. The fold-over masts reach 20 feet above the ground and feature a dual "Goldenrod" 2-GHz antenna and a two-foot dish for 7 GHz, both manufactured by Nurad. The vans included video and audio switching, two-way radios, mobile telephone, gasoline generator, and videotape equipment.

The units were designed to permit any of the vans to be used as a microwave-relay system. This was accomplished by providing the necessary radiofrequency shielding and receivers for 2 and 7 GHz. Therefore, by raising both masts, the van can receive on one of the bands and retransmit the received signal on the other band. This allows our units to function even when mountains or other obstructions are in our path and extends our range of coverage for big breaking news stories. Two of these vans are pictured in Fig. 5.

Central control point

WNBC-TV had been installing remote-control equipment to operate the Channel 4 transmitter at the Empire State Building and was searching for a suitable space at our studio location, the RCA building. This space would also house the remote-control units for the microwave equipment along with the necessary switching and monitoring equipment. WNBC-TV News provided a suitable location on the 7th floor, around which they were building a new videotpae editing complex.

We moved the Channel 4 remote-control equipment, off-air monitoring and slow-scan videotape recording equipment for 24-hour recording of WNBC-TV programming into this control room. We then installed the microprocessor-based monitoring and switching units. New audio and video cables were run to the RCA building roof for connection to the various receivers along with the data lines to operate the remote-control units. Additionally, two-way radios were installed. These radios allow the operator to talk directly to the vans. Figure 6 shows the operating position of the control room with the television transmitter remote controls and the microwave-receiver remote controls.

News bureaus

A News Bureau site was selected in Newark, New Jersey, in a 32-story office building. The roof offered a perfect microwave path to the RCA building. The 7.5-mile link, between the RCA transmitter and a four-foot dish mounted on the Newark roof, operates at 13 GHz. Audio, video, and control cables were run from the roof down to the bureau where provisions were made to plug in a camera or videotape machine. The transmitter is remotely controlled from the RCA building 7th-floor control room by rented telephone data lines.

A Long Island News Bureau was established about two miles east of WNBC-TV's microwave-relay facility. Another 13-GHz line was installed between the Bureau and the relay tower with a dc control line rented from the telephone company to provide a control link between a spare relay in the Nurad remote control at the tower and a control unit at the bureau. Suitable remote-controlled switching facilities were built into the relay tower equipment to allow the 2-, 7-, or 13-GHz receiver outputs to be connected to the 2.5-GHz transmitter.

Conclusion

Figure 7 is a line drawing of the entire system. A news microwave-receiving system of this size is not developed and built overnight. It has taken over three years to reach the point that has been described in this paper. But, the development of the system never ends. WNBC-TV is now in the process of moving the present Empire State Building transmitter site to the World Trade Center. When that move is completed, we will add two more microwave-receiving systems to the top of the World Trade Center. We are also looking at new remote receiving sites for further diversity and increased coverage.

One of the benefits of building, operating, and maintaining this complex system is that WNBC-TV has been a leader in local news for several rating periods. Live news coverage using microwave transmissions has played a major part in achieving this competitive position.

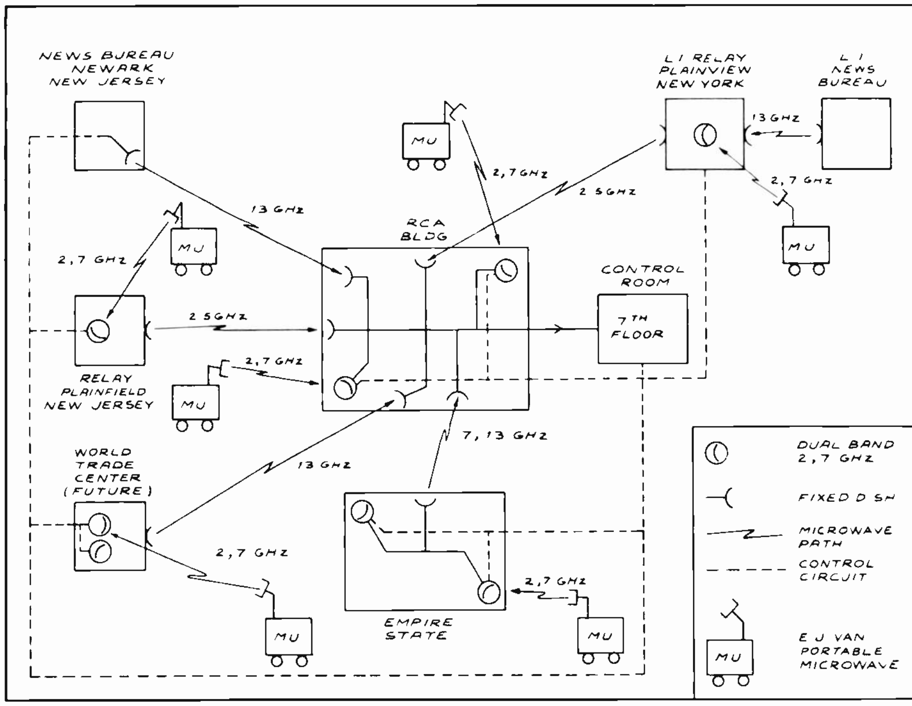
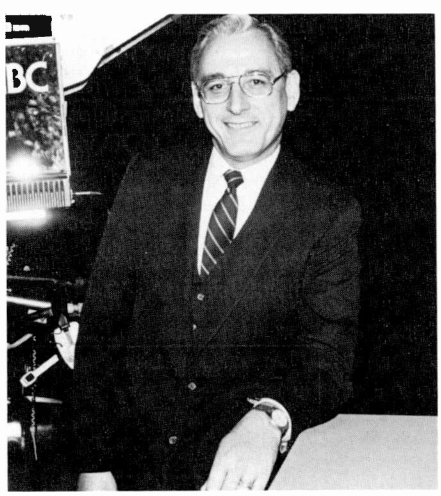


Fig. 7. *WNBC-TV News microwave system. As this line drawing shows, the heart of the system is the seventh-floor control room at the RCA Building.*



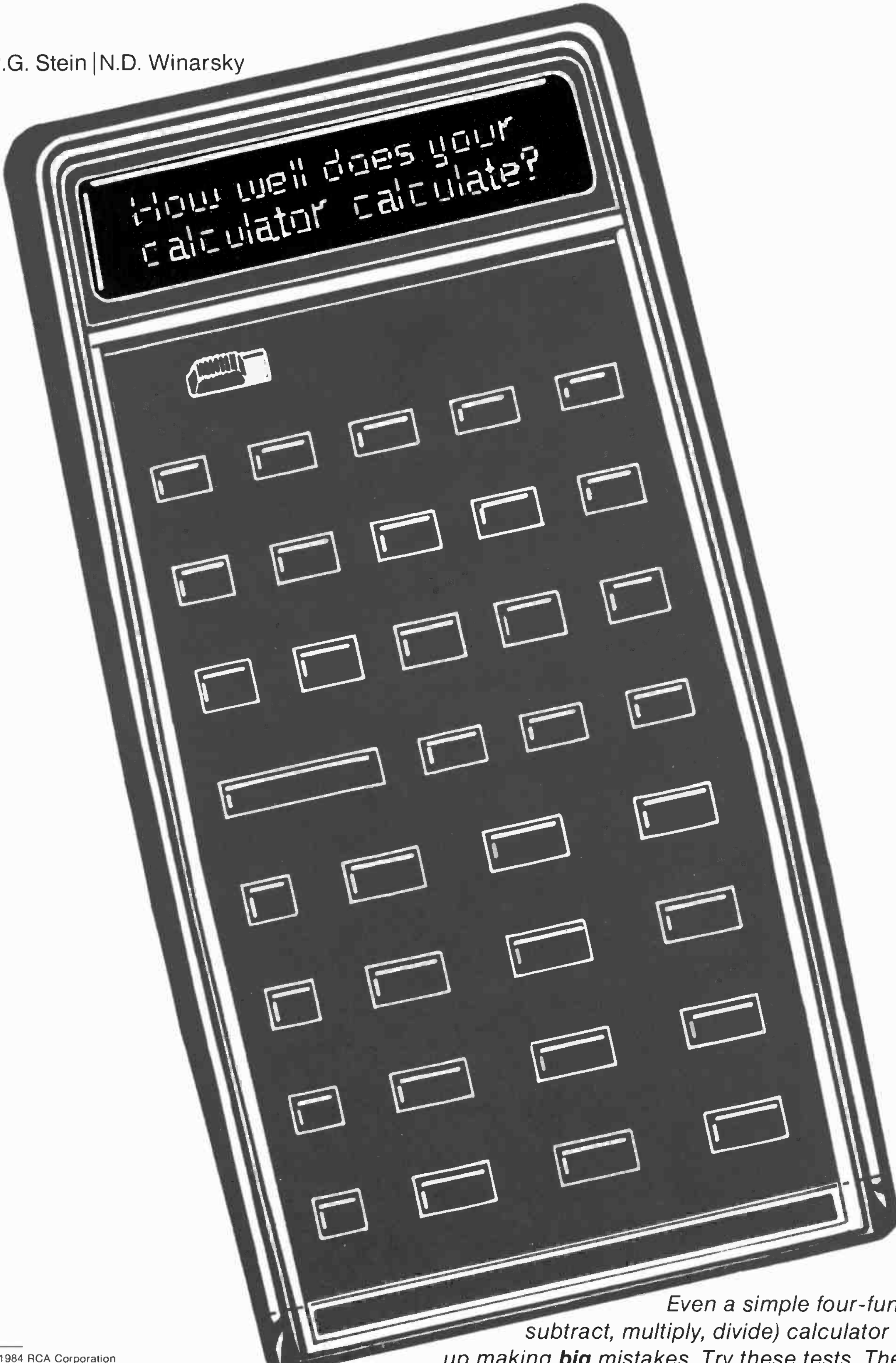
Edward A. Knapp was graduated from the State University of New York at Farmingdale with an Associate in Applied Science degree in 1956. Mr. Knapp was Chief Engineer for several Long Island radio stations before joining the RKO General stations in New York City in 1963. Over the next seventeen years, he worked for RKO General AM, FM, and TV stations in various engineering capacities. Mr. Knapp moved to the NBC Owned and Operated Television Stations Division when he joined WNBC-TV as Manager, Technical Operations in 1980. He is currently Director, Station Operations for WNBC-TV, a position he has held since 1981.

Contact him at:
WNBC-TV
New York, N.Y.
TACNET: 324-2767

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 DM² (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)

P.G. Stein | N.D. Winarsky



*Even a simple four-function (add, subtract, multiply, divide) calculator can wind up making **big** mistakes. Try these tests. The results might amaze you.*

Final manuscript received December 2, 1983.
Reprint RE-29-1-8

Electronic calculators and computers are used extensively both at work and for our personal calculations. Until recently, these machines have been capable of arithmetic errors, large and small, that have been largely unnoticed by their users. Now, a new IEEE standard for floating-point arithmetic offers the possibility of solving these problems.

Computing errors pose a special problem when the calculations are done in conjunction with automatic test and measurement systems and with computer-aided manufacturing. When the computing equipment operates heavy machinery, calculation errors may result in more than defective numbers. They could cause incorrect test results in production, and they can cause improper operation of automatic manufacturing or aligning equipment. The machinery could then damage a product, or, indeed, the equipment itself. One well-known optics company spent nine months and a lot of money looking for mechanical problems in a measuring instrument, only to discover errors in the ability of the machine's controller to compute the sine function accurately.

Where do these errors come from? It's true that add, subtract, multiply, and divide are simple step-by-step procedures that are taught to every schoolchild, and if done correctly, can be made as accurate as necessary for the problem at hand. It's also true that electronic circuits can carry out these operations as correctly as people can and will do so much faster. Once these operations have been "debugged," they continue to operate correctly.

However, people carry out calculations in different ways than machines. They can check the results of each ministep, and will notice intermediate results that look implausible. This is not an easy thing for a computer to do.

Automation issues

Computation errors are especially pernicious when the computer being used is part of a larger system, such as might be used in automated manufacturing. Robots, assembly machines, automatic test and alignment systems, and numerically controlled machine tools are at special risk from computing problems. A mistake in a computation made for an office or business environment and printed in a report has at least the possibility of being discovered and corrected before action is taken. A computer-controlled machine acts immediately. The consequences of these actions might simply be incorrect, such as the rejection of a good part by a tester, but they might also be dangerous, such as a wild motion of a robot arm.

A well-designed system will have limits, checks, and retests built in so that a single error would likely have little or no effect. It is difficult to anticipate and prevent all such errors, though, and it's therefore better to have some guarantees of correctness in the computations themselves.

We live in an analog domain

A measurement such as your height can have any value and is not limited to, say, multiples of one-half inch. Differences in people's heights of less than a half inch are not usually interesting, though, and we usually truncate or round-off answers with more significant figures. People also truncate or round-off in the middle of calculations. For example, in dividing 1 by 3, you would sort of automatically stop using the 0.3333333333333333-33333333 when you knew you had enough digits. Although people sometimes make mistakes in rounding or truncation, the act of keeping in touch with each step of the calculation is likely to keep them close to the right answer.

Word lengths are limited

Computers and calculators, the ones we commonly use today, are all digital. They *must* function with a limited number of digits, though sometimes this number is large. Computers can only approximate the "real" number system, since they are machines with finite storage. Nearly all computers today approximate real numbers with "floating-point" numbers.

A floating-point number is characterized by three quantities:

- The number base β
- The precision t
- The exponent range $[L, U]$

Each floating-point number x has the value

$$x = \pm\left[\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_t}{\beta^t}\right]\beta^e$$

where the integers d_i satisfy $0 \leq d_i < \beta$
for $1 \leq i \leq t$
and $L \leq e \leq U$.

The integer e is called the exponent, and the number

$$\left[\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_t}{\beta^t}\right]$$

is called the fraction. The set of floating-point numbers is, therefore, finite, with exactly $2(\beta-1)\,\beta^{t-1}\,(U-L+1) + 1$ numbers.[1]

It is the limited precision of the floating-point number system that allows the computer to violate the rules of arithmetic, such as closure, associativity, and distributivity. Since some familiar properties of numbers in fact do not hold for computer arithmetic, our intuitive grasp of how these things behave will repeatedly fail us. For example, in the floating-point domain,

- There are positive numbers η such that $1 + \eta = 1$.
- There is a largest number.
- There is a smallest number.
- Situations such as $\frac{1}{3} \neq \frac{9}{27}$ can occur.

In fact, working with floating-point numbers is a little like working with piles of sand. Each time you move one, you lose a bit of sand and pick up a bit of dirt. Modeling the "continuous" real world with a finite, discrete world also means that concepts involving limits, such as integration and differentiation, must be reanalyzed to have a meaning.

For example, let **f** be a function differentiable at a point x. Then

$$\mathbf{f}'(x) = \lim_{h \to 0} \frac{\mathbf{f}(x+h) - \mathbf{f}(x)}{h}.$$

With the floating-point number system, we can choose an h_0

Abstract: *We have come to trust that our calculators and computers do arithmetic correctly, yet this is far from true. Here are some examples of calculator errors, along with a discussion of a proposed IEEE standard which should offer some relief. A sidebar gives some simple methods for testing the accuracy and behavior of your own calculator.*

1. This is an oversimplification in some ways. For example, in some computers, the fraction is binary, $\beta = 2$, yet the base for the exponent portion is 16, not 2.

Test your calculator

The following test is for four-function calculators. The purpose is to see what happens to significant figures when the arithmetic forces the calculator to not display them all. Some calculators pass this test with no trouble; others fail miserably.

This test is written for algebraic logic calculators, by far the most common four-function setup. Each keystroke sequence is shown here, followed by what should show in the display. *Don't* try to shortcut or change the key sequence, it will likely change the results.

The keystroke sequence	*What the display shows*
1 ÷ 11 ÷	0.090909
100 ÷	0.000909
100 ×	0.000009
100 ×	0.000909 (pass) 0.0009 (fail)
100 ×	0.090909 (pass) 0.09 (fail)
100 =	9.0909 (pass) 9.090909 (outstanding) 9 (fail)

As you can see, the calculators that fail actually truncate small numbers rather than just moving them to the right, off the display. The best ones, such as the one labeled outstanding, actually carry more digits than shown in the original calculation.

Scientific calculators should pass this test handily. Because they have to use floating-point notation internally anyway, they should carry a separate fraction and exponent, and all this multiplication and division by 100 should not affect the precision of the fraction.

The following test works for scientific calculators. We have not specified a keystroke sequence since there are so many possibilities. What you do is to take the $\sqrt{2}$, then take the $\sqrt{}$ of the answer, then take the $\sqrt{}$ of the answer again, to a total of ten or twenty times. When finished, square the answer the same number of times and see how close you get back to 2.

One calculator we tried wasn't bad for ten times (2.0005832 came back), but for twenty times, the returned answer was total nonsense. Another, one that was expected to conform closely to the IEEE standard, refused to be fooled at all!

You might also try the test shown in the text for argument reduction. Take the sine of π in radians. It should be near zero. Then try the sine of $\pi \times 10$, $\pi \times 100$, and so on, until you find trouble or get tired. Trouble in poorly designed calculators will likely come when the multiplier for π approaches the word size of the calculator, eight digits or whatever else it is.

such that $x + h_0 = x$. Hence, $\mathbf{f}(x+h) = \mathbf{f}(x)$ for any $h \leq h_0$. From this, we can mistakenly conclude that $\mathbf{f}'(x) = 0$.

By naively applying a definition of limit appropriate only for the analog domain, we have shown that the derivative of any function at any point is always zero. This was an interesting exercise, but it didn't take place in a vacuum. Computational errors based on this type of problem take place every day, and it's necessary to be aware of them before writing programs that make mathematical computations.

Rounding and truncation

Actually, there are two types of errors to be confronted in computing: round-off error and truncation error. Round-off error is simply the error introduced when numbers are rounded off to the precision of the computer. Truncation error is the error committed when a limiting process is stopped before reaching its limiting value. For example, approximating a function by the first few terms of its Taylor series results in a truncation error.

In the example above, where the derivative of a function was found, the key to calculating an accurate derivative is "balancing" the two errors. We commit a truncation error by approximating

$$\mathbf{f}'(x) \text{ by } \frac{\mathbf{f}(x+h) - \mathbf{f}(x)}{h}.$$

and we commit a round-off error in rounding x, $x+h$, $\mathbf{f}(x)$, and $\mathbf{f}(x+h)$.

If we pick h too large, we will have a large truncation error, since the difference quotient will not necessarily be close to the true tangent of $\mathbf{f}$ at x. If h is too small, round-off error will, as we have seen, destroy the accuracy. Reference 1 describes a technique for selecting h.

Choosing the right algorithm

Even some of the simplest formulas for solving equations can lead to difficulty. For example, if we are asked to solve

$$x^2 - 10^6 x + 1 = 0 \text{ for the smallest root,}$$

our natural inclination might be to apply the quadratic formula,

$$x = \frac{10^6 \pm \sqrt{(10^6)^2 - 4(1)}}{2} \quad \text{or} \quad x = \frac{10^6 \pm \sqrt{10^{12} - 4(1)}}{2}$$

At this point, most calculators will return 0 for the smaller root of x.

The difficulty here is that 10^6 and $\sqrt{10^{12} - 4}$ are *nearly equal large numbers*, and we lose all accuracy by performing the subtraction. We can't even tell you what your calculator will do with this problem. There are many possible responses, and this is one difficulty addressed by the IEEE floating-point standard, which we will discuss later in this article.

There are several good examples of how professional mathematicians specializing in computation (called numerical analysts) might go about avoiding this particular error. The first way is to try another algorithm. This may seem inappropriate; after all, the quadratic formula is an exact mathematical relationship. It's hard to conceive of it as being the "wrong" algorithm for solving a quadratic equation, yet it just got us into a lot of trouble. The trouble came not from the quadratic formula itself, but from the fact that our calculator couldn't handle all the digits of precision necessary for the quadratic formula to come out right. At this point, we could choose another algorithm, one that didn't require as much precision. For example, we could handle this particular quadratic as a linear equation with a small perturbation term, the x^2.

Building this technique into a general equation solver, we would have the calculator or computer test the sizes of the incoming data (called *arguments*), and choose to use either the plain quadratic formula or something else, according to which is most likely to give the best results. Points at which the algorithms may switch are called *breakpoints*.

Another approach to our quadratic solution is to avoid the subtraction entirely using the following trick:

$$\frac{10^6 - \sqrt{10^{12}-4}}{2} = \frac{(10^6 - \sqrt{10^{12}-4})\quad(10^6 + \sqrt{10^{12}-4})}{2(10^6 + \sqrt{10^{12}-4})}$$

$$= \frac{10^{12} - 10^{12} + 4}{2(10^6 + \sqrt{10^{12}-4})} = \frac{2}{(10^6 + \sqrt{10^{12}-4})}$$

Now, there is no subtraction and we see that $x \approx 10^{-6}$.

Another example of the need to understand the problem and to choose the correct algorithm is a simple calculation: $\mathbf{f}(z) = \ln(1 + z)$. If z is large, this formula is fine. If z is close to zero, the addition to 1 will cause a loss of some (or all) of the digits of z, resulting in an incorrect value for the natural log of $(1 + z)$.

For small z, a better (but not obvious) way[4] to calculate f would be

$$\mathbf{g}(z) = \frac{\bar{z}}{\overline{1+z} - 1} \ln \quad \overline{(1 + z)},$$

where $\bar{z}$ means the computer-generated value for z. $\mathbf{g}(z)$ is accurate to the precision of the machine.

Argument reduction

As a final example of loss of accuracy, let's look at one example of an all-too-common problem. When we calculate sin (x), we expect the accuracy to be the same for all values of x. Unfortunately, this is rarely true. We tried sin $(n\pi)$ in radians on a typical handheld scientific calculator. The answer should always be zero, of course. Table I shows what we got:

Table I. *Typical calculator makes errors in calculating sin (nπ).*

n	Sin *(nπ)*
1	10^{-7}
100	10^{-5}
10^4	0.999×10^{-3}
10^6	0.09998
10^8	0.554
	(Should Always Be Zero)

Error analysis for ln (1 + z)

Let $\overline{1 + z}$ be the computer-generated value of $1 + z$. We can set $\overline{1 + z} = (1 + z) + \epsilon$, where ϵ is the difference between the computer-generated argument and the actual argument. Then,

$$\ln(\overline{1 + z}) = \ln[(1 + z) + \epsilon].$$

Now $\ln(1 + z) = 0 + z - (z^2/2) + (z^3/3) \ldots$

and so

$$\ln[1 + (z + \epsilon)] = z + \epsilon - [(z + \epsilon)^2/2] + \ldots.$$

Let the absolute error E be the difference between the computer answer and the actual answer.

$E = \ln(\overline{1 + z}) - \ln(1 + z)$

$$\approx (z + \epsilon) - z - (z + \epsilon)^2/2 + z^2/2 \ldots \approx \epsilon.$$

The *relative* error is then $\approx \epsilon/\ln(1 + z)$, which is $\approx \epsilon/z$ for small z. This blows up for small z!

What's going wrong here? Most trigonometric algorithms have several breakpoints and work best within limited argument ranges, usually

$$-\frac{\pi}{4} \leq x \leq \frac{\pi}{4}$$

To use these algorithms over very wide ranges, techniques of *argument reduction* are employed. To go from an angle near $10^6\pi$ to the corresponding angle near π, it is only necessary to divide the argument by π and save the remainder. If you think about it, though, to get a remainder from such a division and retain the full precision inherent in the original number, you have to handle twice the number of digits normally used, and your internal value of π must also be of double precision. In this way, when you throw away the full precision of the quotient, the part you weren't interested in, you are still left with full precision in the remainder.

These problems and others are computer dependent—that is, the accuracy depends on the internal programming of the calculator used. Also, as with the previous examples, how you must use your calculator to get the most accuracy can also depend on which calculator it is.

Each computer or calculator can have its own floating-point numbering system, with its own number base, precision, exponent range, argument reduction, algorithm breakpoints, decimal-to-binary conversions, overflow and underflow handling, and so on. This presents an enormous challenge to the programmer and even to the casual user. How can we find our way out of this jungle?

The proposed IEEE floating-point standard[2] describes *in detail* an "arithmetic engine." In particular, specifications are laid down for

- Single- and double-precision format representations for numbers.
- Overflow and underflow thresholds.
- Bit patterns for $\pm\infty$, and other "exceptional numbers." These cases arise, for example, from division by zero.
- Standardized responses for exceptions such as division by zero, invalid operations, and so on.
- Rules for the algebraic operations $+, -, \times, \div, \sqrt{}$.

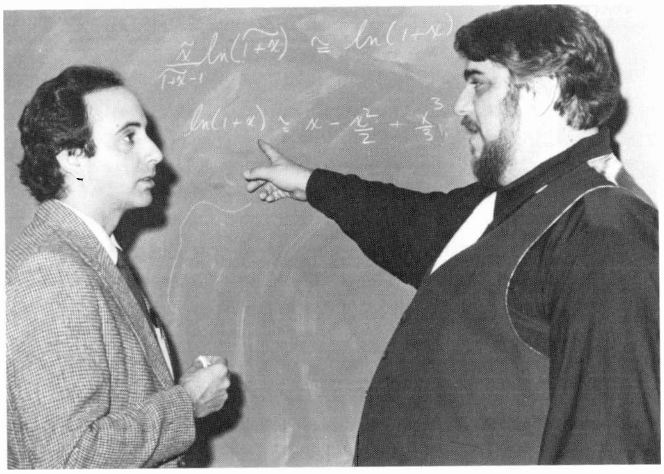
Without the IEEE standard, a programmer can either customize a version of his or her procedure to the arithmetic idiosyncrasies of each calculator or computer, or can just ignore the potential problems shown here and take a chance.

The standard will not guarantee correct results when performing numerical calculations. The fact that the floating-point numbering system is finite is fundamentally inconsistent with the notion of always getting the "right" answer. The standard will, however, make it much easier to understand how the floating-point system models the real-number system, and, therefore, make it easier to understand the consequences of each operation and to estimate the resulting accuracy.

As a further consequence, programs and procedures can be moved from one machine to another without creating unexpected inconsistencies in the answers. One manufacturer has already produced an integrated-circuit arithmetic engine to this standard, and it is, therefore, possible for the first time to do well-understood, portable floating-point computations. This should prove of immense benefit not only to the mathematicians and computer scientists, but also to the rest of us who use computers and calculators in our everyday work.

References

1. R.S. Stepleman and N.D. Winarsky, "Adaptive Numerical Differentiation," *Mathematics of Computation*, Vol. 33, No. 148, pp. 1257-1264 (October 1979).
2. W.J. Cody and D. Hough, "A Proposed Standard for Binary Floating-Point Arithmetic," *Computer*, Vol. 14, No. 3, pp. 51-62 (March 1981).
3. W. Kahan, "Why Do We Need A Floating-Point Arithmetic Standard?" lecture notes, University of California at Berkeley (March 1981).
4. R.W. Klopfenstein, personal communication (1981).



Authors Winarsky (left) and Stein.

Philip Stein is a Member, Technical Staff, in the Productivity and Quality Assurance Research group at Princeton. His interests include statistics, measurement science, and automated measurements, and their application to manufacturing and automatic testing.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2858

Norman Winarsky is Head of the Electron Optics and Applied Mathematics Research group at Princeton. His interests include numerical algorithms and their applications to physical models.
Contact him at:
RCA Laboratories
Princeton, N.J.
TACNET: 226-2872

Contents, September 1983

Patents

Advanced Technology Laboratories

Pryor, R.L.
Protection and anti-floating network for insulated-gate field-effect circuitry—4408245

Astro-Electronics

Fox, S.M.
Attitude control system for spacecraft utilizing the thruster plume—4407469

Automated Systems

Sutphin, E.M., Jr.
Apparatus for testing linearity of an FM transmitter—4412348

Commercial Communications Systems

Pastore, D.C.
Multichannel magnetic head for a recorder-reproducer system—4408240

Consumer Electronics

Griffis, P.D.
Matching volume control characteristics for two channels—4404429

Griffis, P.D.
Volume control signal coupling circuit in an audio signal processing system—4405948

Muterspaugh, M.W. | Whitledge, G.A.
Impedance transformation network for a saw filter—4410864

O'Leary, D.B. | Ross, G.D., Jr.
Video disc player—270442

Thornberry, G.E.
Vertical detail enhancement on/off switch—4403246

Government Communications Systems

Nossen, E.J.
Digital phase demodulation and correlation—4412302

Thomas, C.H., Jr.
Digital sequence detector—4404542

Laboratories

Avery, L.R.
Protective integrated circuit device utilizing back-to-back diodes—4405933

Bell, A.E.
Reinforced bubble recording medium and information record—4408213

Blackstone, S.C.
Method of forming closely spaced lines or contacts in semiconductor devices—4402128

Buzgo, A.W.
Spacer for use in testing information recorded discs—4408320

Chin, D. | Maturo, R.J.
Tuning display for a television receiver—4410913

Coutts, M.D. | Matthies, D.L.
Video disc processing—4405541

Datta, P. | Poliniak, E.S.
High density information disc lubricants—4410748

Fairbanks, D.W.
Stylus dispensing apparatus and method—4406381

Ferguson, J.M. | Chen, T.Y. | Gibson, W.G.
Video disc player having vertical timing signal generator—4409626

Flory, R.E.
Simplified transmission system for sequential time-compression of two signals—4410981

Gorog, I. | Fox, L.P.
Method for the manufacture of a metallic recording substrate for a capacitance electronic disc and the recording substrate detained thereby—4402798

Hernqvist, K.G.
Processing the mount assembly of a CRT to suppress afterglow—4406637

Kipp, R.W. | Johnson, H.C.
Frequency synthesizer with learning circuit—4410860

Kleinknecht, H.P. | Ham, W.E. | Meier, H.
Optical measurements of fine line parameters in integrated circuit processes—4408884

Knight, S.P.
Television signal converting apparatus providing an on-screen tuning display—4405946

Kosonocky, W.F.
Charge transfer circuits with dark current compensation—4412343

Labib, M.E.
Capacitance electronic disc stamper having improved stain resistance and method for the manufacture thereof—4405670

Lang, F.B.
Television receiver high voltage protection circuit—4412254

Levine, P.A.
Electrical compensation for misregistration of striped color filter in a color imager with discrete sampling elements—4404587

Levine, P.A.
Apparatus for processing CCD output signals—4412190

Lin, P.T.
Internal caddy cleaning apparatus—4403369

Maa, J.
Etching of tantalum silicide/doped poly-silicon structures—4411734

Marcantonio, A.R.
Interrupt signal generating means for data processor—4404627

Miller, A.
Array positioning system—4404465

Pampoline, T.R. | Kilichowski, K.B.
Positive radiation sensitive resist terpolymer from omega alkynoic acid—4405776

Pritchard, D.H.
Signal processing apparatus effecting asymmetrical vertical peaking—4404584

Robbi, A.D. | Sinniger, J.D.
Digital timing system for spark advance—4408296

Ruhland, K.
Pseudo random number generator apparatus—4408298

Schiff, L.N.
FM/TV automatic gain control system—4403255

Sechi, F.N.
Bandpass filter with an active element—4409557

Shanley, R.L., 2nd. | Harwood, L.A.
Brightness control circuit—4404593

Shanley, R.L., 2nd
Switching circuit for television receiver on-screen display—4412244

Shefer, J. | Catanese, C.A.
Electron gun with improved beam forming region—4409514

Smith, T.R.
Memory-type sync generator with reduced memory requirements—4412250

Tarng, M.L. | Hicinbothem, W.A., Jr.
Method for improving adhesion of metal film on a dielectric surface—4404235

Theriault, G.E.
Multiband tuning system for a television receiver—4408348

Tosima, S. | Harada, S.
Method for the evaluation of solderability—4409333

Tults, J. | French, M.P.
Dual search mode type tuning system—4405947

Vinekar, S.R. | Hettiger, J. | Friedline, K.L.
Burst gate keying and back porch clamp pulse generator—4410907

Wine, C.M.
Servo system for a video disc player carriage assembly—4406002

Missile and Surface Radar

Breese, M.E. | Robinson, A.S.
Controllable phase shifter comprising gyromagnetic and non-gyromagnetic sections—4405907

Martinson, L.W.
Digital dual half word or single word position scaler—4411009

Patton, W.T. | Mason, R.J.
Coaxial line to waveguide coupler—4409566

Pruit, D.L.
Dc-to-dc switching converters—4408267

Solid State Division

Atherton, J.H.
Input buffer circuit—4406957

Butterwick, G.N.
Method for processing a lithium-sodium-antimony photocathode—4407857

Delrio, E.H.
Ion implanter end processing station—4405864

Harford, J.R.
Noise sensitivity reduction apparatus for a TV receiver AGC system—4408229

Olmstead, J.A.
Operational amplifier with feed-forward compensation circuit—4410857

Solid State Technology Center

Dingwall, A.G.
Active load pulse generating circuit—4404474

Stewart, R.G.
Precharge with power conservation—4405996

Video Component and Display Division

Chase, T.L. | Ehemann, G.M., Jr.
Photographic method for printing a viewing-screen structure using a light-transmission filter—4408851

D'Amato, R.J.
Electrode for an electron gun—4409513

DeAngelis, M.E.
System for removing shadow mask assemblies from kinescope panels of varying sizes—4406638

Deyer, C.E.
System and method for use with apparatus for sensing bare metal on a moving strip of insulatively coated conductive material—4404515

Forberger, S.C.
Method of printing intelligible information—4403547

Halbrook, J.C.
Degassing a CRT with modified RF heating of the mount assembly thereof—4410310

Hale, J.R.
Glass support rod for use in electron-gun mount assemblies—4409279

Hughes, R.H.
Color picture tube having an expanded focus lens type inline electron gun with an improved stigmator—4406970

Raush, R.G. | Alleman, R.A.
Externally adjustable seal and bearing structure—4407531

Wardell, M.H., Jr.
Television display system handling and adjustment apparatus—4405950

Williams, R.H.
Wet processing of electrodes of a CRT to suppress afterglow—4406639

VideoDisc Division

Cave, E.F. | Cowden, J.J.
Stylus coning fixture—4403453

Mehrotra, G.N. | Vanarsdall, G.
Video disc player having stylus cleaning apparatus—4408315

Miller, M.E.
Stylus assembly—4404669

Prusak, J.J.
Apparatus for producing disc records with a molded-in center hole—4402660

Taylor, B.K.
Pickup arm retainer for video disc cartridge—4404670

Taylor, B.K. | Riddle, G.H.
Video disc player with self calibrating stylus translator—4412319

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

G. Ammon
An Optical Disk Jukebox Mass Memory System—Presented at Large Scale Memories Symposium, sponsored by AFCEA and NADC, Willow Grove, Pa. (9/29-30/83)

A. Feller
Custom LSI Design Using Computer-Aided Design Techniques—Presented at a Seminar at Drexel University, Phila., Pa. (10/25/83)

J. Saultz
RCA/GSD Perspective on VHSIC—Presented at 3rd Annual VHSIC Conference, Gaithersburg, Md. (9/19-21/83)

J. Saultz
RCA as a VHSIC Second Source—Presented at 3rd Annual VHSIC Conference, Gaithersburg, Md. (9/19-21/83)

J. Tower | J. Moffa
High-Density Schottky-Barrier IRCCD Sensors for Remote-Sensing Applications—Presented at IEEE IGARSS Conference, San Francisco, Calif. and published in the *Proceedings* (8/31 - 9/2/83)

Astro-Electronics

J. Baldo
Thermovision Used for Non-Destructive Test of Solar Panels—Presented at SPIE Thermosense VI, Brook, Ill. (10/2/83)

G. Brucker
Soft Error Impact on Computer Devices in Space Applications—Presented at AIAA Computers in Aerospace, Hartford, Conn. (10/24/83)

G.J. Brucker
Ground Test of Devices for Ionization Damage Testing & Failure Analysis—Presented at '83 Intl. Soc. Test & Fail. Anal., Marriott, La. (10/17/83)

L. Bulwer
Performance Measurement Using Value of Work—Presented at the Artemis Users Assoc. (11/15/83)

R. Buntschuh
Direct Broadcast TV—Economical TV Distribution System—Presented at the Nat'l. Telsys. Conf., San Francisco, Calif. (11/14/83)

C. Chao | F. Chu | J. Yang
New Algorithm Force Orthogonality of Measured Modes—Presented at Shock & Vib. Sym., Pasadena, Calif. (10/28/83)

F.H. Chu
Modal Testing and Model Refinement (Book Published by ASME)—Published in *ASME AMD-Journal* (12/83)

F. Chu | J. Yang
Heuristic Opt Balancing in High Speed Rotors—Presented at the ASME Winter Mtg., Boston, Mass. (11/13/83)

S. Gaston
The GSTAR & Spacenet Nickel Hydrogen Batteries for Geosynchronous Applications—Presented at IECEC (Intersociety Energy Conversion Engineering Conf.), San Francisco, Calif. (8/19//83)

S. Gaston | S. Schiffer
RCA Satcom Battery in Orbit Performance Update—Presented at NASA GSFC WORKSHOP, Greenbelt, Md. (11/15/83)

P. Goodwin
Autonomous Redundancy Mgmt. on DMSP Spacecraft—Presented at AIAA Computers in Aerospace Conf., Hartford, Conn. (10/24/83)

J. Graebner
EHF Interoperability for DMSP—Presented at MILCOM 83, Washington, D.C. (11/1/83)

D. Gross | K. Johnson
E. Bair | N. Samhammer
Development of Computer-Aided Mechanical Engineering System at Astro—Status of Experiment—Presented at Mechanical Eng. Sym., RCA, Moorestown, N.J. (10/19/83)

K. Johnson | D. Gross
Development of Computer-Aided Mechanical Engineering System at Astro—Retrospective—Presented at Mechanical Eng. Sym., RCA, Moorestown, N.J. (10/19/83)

J. Keigler | B. Stewart | J.F. Clark
Advanced Satcom: State of the Art Tech. Operational Service—Presented at the 34th Int'l. Astronaut Congress, Budapest, Hungary (10/9/83)

J.N. LaPrade
Solid-State C-Band Power Amplifier for Communications Satellites—Presented at Int'l. Telemetering Conf., San Diego, Calif. (10/27/83)

J. Petheram
A Raman Laser for Temperature/Humidity Profiling—Presented at Optical Soc. of America, New Orleans, La. (10/17/83)

W. Proscia
Multiple Step Global Mechanisms for Oxidizing Higher Carbon Number Alkanes—Presented at Brown University, Providence, R.I. (11/9/83)

C. Voorhees | G. Clark
Experimental Modal Analysis Applied to Lightweight Composite Structures—Presented at ASME Annual Winter Mtg., Orlando, Fla. (11/13/83)

Automated Systems

L.B. Blundell | T.J. Plunkett
Implementing Microprocessor Test Strategies for the Army Aquila RPV Program—AUTOTESTCON '83, Forth Worth, Texas (11/83)

D.A. Gore
Simplified Test Equipment Using Microprocessors for Maintenance of Diesel and Turbine Engines—IEEE Student Chapter Meeting, Northeastern University, Boston, Mass. (10/83)

S. Imhoff
Applications of a Microprocessor to a Dedicated ATE System—IEEE Student Chapter Meeting, Northeastern University, Boston, Mass. (11/83)

J.F. Martin | J. Craven (Termiflex)
Front-line Automated Diagnostics Facilitated by Handheld Terminals—Mini/Micro West Symposium, San Francisco, Calif. (11/83)

R.J. McLaughlin
Recent Product Developments at Automated Systems—Meeting of Town Government Executives, Town of Burlington, Mass. (10/6/83)

W.T. Meyer
Electronic Templating Techniques—Tactical SIGINT & EW Officers Course, U.S. Army Intelligence Center & School, Ft. Huachuca, Ariz. (11/83)

S.P. Patrakis
Computer-Aided Electronic Rack Design—Computer-Aided Mechanical Engineering Symposium, RCA MSR, Moorestown, N.J. (10/19/83)

J.C. Tranfaglia
Profile of Two CAD/CAM Operators—Computer-Aided Mechanical Engineering Symposium, RCA MSR, Moorestown, N.J. (10/19/83)

Consumer Electronics

Basab B. Dasgupta
Harmonic Analysis of a Planar-Wound Toroidal Deflection Coil—Published in *IEEE Transactions on Consumer Electronics*, CE-29, p. 508 (1983)

Missile and Surface Radar

J.A. Bauer
Chairman of CAD/CAM Session—ISHM Annual Meeting, Philadelphia, Pa. (11/2/83)

R.M. Blasewitz
Status of the IEEE Working Group on Ada as a Program Design Language—IEEE 2nd Software Engineering Standards Applications Workshop (SESAW-II), San Francisco, Calif. (5/19/83)

F.J. Buckley
Software Quality Assurance—IEEE Software Quality Assurance Seminar, Paris, France (10/12/83); Los Angeles, Calif. (11/30 - 12/2/83)

F.J. Buckley
Software Quality Assurance and the IEEE Standards Process—AFCIQ Congress, Toulouse, France (11/20/83)

A.S. Cooper
Computer-Aided Engineering for Thick Film Multilayer Hybrids—ISHM, Philadelphia, Pa. (11/2/83); Applicon Fall Users Meeting, Detroit, Mich. (11/9/83)

L.P. Dorsett
Survivability in Design—4th Annual CS Symposium, ASNE, Washington, D.C. (10/13/83)

J.W. Douglas
Applicon Disk RSX File Structure—Applicon Fall Users Meeting, Detroit, Mich. (11/83)

W.A. Harmening
Advanced Near-Field Scanning for Ultra-Precise Phased Array Antenna Alignment—Published in *Microwave Journal* (10/83)

W.C. Grubb, Jr.
Minicomputers and Microcomputers for Non-Electrical Engineers—George Washington University, presented at Naval Air Test Center, Patuxent, Md. (7/27/83); George Washington University, Washington, D.C. (10/26/83)

W.C. Grubb, Jr.
Solid State Electronics for Non-Electrical Engineers—Drexel University, presented at Western Electric, Allentown, Pa. (6/3/83); George Washington University, presented at Naval Air Test Center, Patuxent, Md. (7/13/83); Drexel University, presented at Lakehurst Naval Air Station, Lakehurst, N.J. (7/22/83); George Washington University, Washington, D.C. (7/9/83)

W.C. Grubb, Jr.
Electro-Optics for Non-Electrical Engineers—George Washington University, Washington, D.C. (11/30/83)

D. Lesser
Quality Control: A Primer for the Manager—Published in *Machine Design* (9/22/83)

F.E. Oliveto
The Multipurpose Aspect of the Failure Mode and Effects Analysis (FMEA)—Published in *NAVMAT 06 - OVERVIEW*, Issue No. 5 (10/83)

J.W. Smiley
Wirewrap Backplane Design Automation System—Applicon Fall Users Meeting, Detroit, Mich. (11/6-11/83)

W.A. Soper
Panelist for Discussion on Level of Model Detail—Winter Simulation Conference WSC 83, Arlington, Va. (12/14/83)

M. Trachtenberg
Order and Difficulty of Debugging—*IEEE Transactions on Software Engineering* (11/83)

M.L. Weisbein
I'd Rather Be Programming, or A View from the Quality End of Software Engineering—ASQC 27th Annual Symposium, Horsham, Pa. (11/17/83)

B.A. Wiegand
Three-Dimensional Computer-Aided Design and Manufacturing—Numerical Control Society, Delaware Valley Chapter, MSR, Moorestown, N.J. (6/14/83)

Engineering News and Highlights

Lohman named Staff Vice-President for Solid State Research

Appointment of **Robert D. Lohman** as Staff Vice-President, Solid State Research, was announced by **Dr. William M. Webster,** Vice-President, RCA Laboratories, in Princeton, N.J. In his new position, Mr. Lohman is responsible for RCA Laboratories integrated circuit technology research.

A native of Chicago, Illinois, Mr. Lohman received a B.S. degree in Electrical Engineering from Norwich University in 1949 and an M.S. degree in Electrical Engineering from North Carolina State College in 1951.

Mr. Lohman joined RCA Laboratories as a Member of the Technical Staff in 1951 and engaged in research in the areas of basic semiconductor noise phenomena, transistor circuit development, color television display systems, and information theory. In 1956, he transferred to the RCA Semiconductor and Materials Division in Somerville, N.J., as an Applications Engineer. The next year he was appointed Manager of Applications for computer devices. In 1960, Mr. Lohman was promoted to Engineering Manager, Computer Products, and in 1963 was named Manager, Integrated Circuit Engineering. He returned to RCA Laboratories in 1966 as Head of Integrated Electronics Research and, in 1968, was appointed Head of Optical Data Storage Research.

Mr. Lohman became a Laboratory Director in 1977 and has headed the Display Systems Research Laboratory, the VideoDisc Systems Laboratory, and the Display Processing and Manufacturing Research Laboratory. Earlier this year, he was named Director of the Integrated Circuit Technology Research Laboratory, the position he held until his new assignment.

Mr. Lohman has been issued 14 U.S. patents and has published 15 technical papers. In 1973, he was elected a Fellow of the Institute of Electrical and Electronics Engineers "for contributions to transistor technology applied to television and computers."

Smylie named Americom Vice-President, Government Communications Services

Dr. James J. Tietjen announced the election of **R.E. Smylie** as Vice-President, Government Communications Services, at RCA American Communications, Inc. In his new position, Mr. Smylie is responsible for directing the marketing of RCA Americom's satellite communications services to various agencies of the Federal government. He replaces **John Boning,** who has retired.

A 21-year NASA veteran, Mr. Smylie most recently served as Associate Administrator for Space Tracking and Data Systems at NASA headquarters in Washington. In announcing the appointment, Dr. Tietjen said, "Mr. Smylie brings to RCA Americom an outstanding record of accomplishment in both the technical and administrative aspects of a highly complex business. We look forward to his contributions in expanding the Satcom satellite system's role in providing voice, data, and video services to our various customers in the Federal government."

Reed is New Products Division Ed Rep

Robert E. Reed, Manager, Neutral Beam Source Project, has been appointed the *RCA Engineer* Editorial Representative at the New Products Division. Mr. Reed has been employed continuously by RCA-Lan caster since June 1952. He received his B.S. degree in Electrical Engineering with Distinction from the University of Oklahoma in 1952, an M.S. dgree in Physics from Franklin & Marshall College in 1971, and has continued his graduate studies towards a PhD degree with Solid-State Physics courses at F & M College.

Mr. Reed has a diversified background in high-power electronic devices and circuits, including the development of structural methods, evaluation of environmental capabilities, improvements of processing, design of circuits, computer-aided design techniques, and applications engineering. X-ray shielding, high voltage hazard, and rf radiation control have all been adjunct responsibilities in these endeavors. He has also served on the Product Safety Committee for the Lancaster plant. He pioneered many developments of ceramic-to-metal sealing techniques, carburizing and coating techniques for thermionic emitting cathodes, tuned broadband rf circuits, and spurious mode suppression, as well as improved vacuum processing and testing methods. High power devices required the development of high intensity cooling for the high dissipation densities encountered.

Contact him at:
RCA New Products Division
Lancaster, Pa.
TACNET: 227-2485

Braverman named Marketing Manager at RCA Advanced Technology Laboratories

Dr. Leonard W. Braverman has been named Manager, Marketing and Planning at RCA Government Systems Division's Advanced Technology Laboratories (ATL). A resident of Moorestown, N.J., he is responsible for planning ATL's technical efforts and assisting in the achievement of business goals.

Dr. Braverman previously served as Manager of ATL's Systems and Electromagnetics Laboratory, a position he had held since 1982. Formerly he was a staff scientist at Bechtel Group Inc., San Francisco, Calif., and before that, manager of the laser sources section at GTE, Sylvania Systems Group, Mountain View, Calif.

Born in Philadelphia, Dr. Braverman received bachelor's, master's, and doctor's degrees from the University of California at Berkeley. He holds four patents relating to gas lasers. A member of the Institute of Electrical and Electronics Engineers, Dr. Braverman has published articles on gas lasers, molecular spectroscopy and ultra-short pulsed lasers.

GCS becomes Camden landlord

John D. Rittenhouse, Group Vice-President, announced that effective December 1, 1983, the Government Communications Systems Business Unit of Government Systems Division assumed landlord responsibility for the Camden location.

In addition to his current responsibilities, **Lawrence J. Schipper,** Division Vice-President and General Manager, Government Communications Systems, directs Industrial Relations and all Manufacturing Activities including Materials, Plant Engineering, and other related functions in the location.

Additionally, the location Accounting and Management Information Systems responsibility is assigned to the Government Communications Systems' Finance Department. Mr. Schipper will continue to report to **Paul E. Wright,** Division Vice-President and General Manager, Government Systems Division.

Staff announcements

Consumer Electronics

D. Joseph Donahue, Vice-President and General Manager, Consumer Electronics Division, announces the appointment of **Bruce M. Allan** as Director, Strategic Planning.

Larry A. Olson, Manager, Manufacturing Technology Center, announces the appointment of **D. Terry Stephens** as Administrator, Production Systems.

William A. Lagoni, Manager, Signal Processing, announces the appointment of **Robert L. Shanley, II,** as Manager, Signal Processing-Chassis.

Gary A. Gerhold, Plant Manager, Indianapolis Components Plant, announces the appointment of **Henry L. Slusher** as Manager, Plant Engineering Projects.

Gary A. Gerhold, Plant Manager, Indianapolis Components Plant, announces the appointment of **Walter E. Todd** as Manager, Facilities Services-CED Indianapolis.

Robert R. Beasley, Manager, Plant Engineering, announces the appointment of **Donald R. Stapert** as Manager, Engineering and Services.

Government Communications Systems

Lawrence J. Schipper, Division Vice-President and General Manager, Government Communications Systems, announces his organization as follows: **John R. Allen,** Principal Scientist; **Arthur J. Barrett,** Director, Manufacturing; **Donald D. Miller,** Chief Engineer; **Marvin Gelman,** Manager, Special Programs; **Alan E. Matt,** Manager, Industrial Relations; **Donald J. Parker,** Director, Digital Communications and Recording Systems; **Morton Raphelson,** Manager, Product Assurance; **Lawrence J. Schipper,** Acting, Advanced Missions Group; **Lawrence J. Schipper,** Acting, Radio Systems; **John F. Serafin,** Director, Integrated Communications Systems; **Guy H. Shaffer,** Director, Information Processing Systems; **John C. Shannon,** Director, Transmission Systems; **Alfred C. Thompson,** Director, Marketing and Advanced Programs; **Edward W. Williams,** Director, Finance; and **George P. Williams,** Manager, Business Planning and Plant Security.

Arthur J. Barrett, Director, Manufacturing, announces his organization as follows: **H. Stanley Barr,** Manager, Manufacturing Administration; **Joseph D. Borucki,** Manager, Plant Engineering; **Richard J. Grimm,** Manager, Materials; **George H. Lines,** Manager, Fabricated Products; **Fred Pfifferling,** Manager, Technical Operations; **L. Clair Phillips,** Manager, Government Manufacturing; **Vincent J. Renna,** Manager, Commercial Manufacturing; and **Nathan Shectman,** Manager, GWEN Manufacturing.

Morton Raphelson, Manager, Product Assurance, announces his organization as follows: **Smith A. Cochrane,** Manager, Quality Assurance; **Meyer R. Greenberg,** Manager, Quality Assurance, Quality Control; **Myles J. Burke,** Manager, Government Quality Control; **Thomas E. Hassett,** Manager, Government Quality Control; **Frederick P. Molden,** Administrator, Quality Assurance; **Virl E. Haas,** Manager, Quality Assurance; **R. David Houck,** Manager, Quality Assurance; **James J. Brennan,** Administrator, Quality Support; **Nicholas B. Sher,** Manager, Quality Assurance; **Judith C. Dodd,** Administrator, Quality Support; **Anthony J. LaRocca,** Manager, Assembly Quality Control; **J. Douglas Logan,** Manager, Materials Quality Assurance; **John P. Baleria,** Manager, Field QA and Administration; **Edward J. Horner,** Leader, Material Quality Assurance; **Iliana Okum,** Manager, Training and Statistics; **Walter A. Sawyer,** Manager, PMI; **Michael J. Yanky,** Manager, Assembly Quality Control; **Richard J. Noch,** Administrator, Quality Assurance; **Joanna M. Shukal,** Manager, Quality Assurance; **Jerry L. Lenk,** Administrator, Quality Assurance; **Robert W. Moore,** Manager, Government Quality Control; **James B. Reid,** Manager, Government Quality Control; **Thomas R. Yahraes,** Manager, Assembly Quality Control; **David I. Troxel,** Manager, Systems Effectiveness; **Harvey R. Barton,** Unit Manager, Engineering Staff; **John J. Davaro,** Unit Manager, Engineering Staff; **Ronald M. Gould,** Administrator, Quality Assurance; and **Frank M. Winton,** Unit Manager, Engineering Staff. Miss Shukal and Messrs. Cochrane, Greenberg, Haas, Houck, Sher, and Troxel will report to the Manager, Product Assurance.

Laboratories

William M. Webster, Vice-President, RCA Laboratories, announces the following appointments: **Bernard J. Lechner** is Staff Vice-President, Advanced Video Systems Research; and **Robert D. Lohman** is Staff Vice-President, Solid State Research.

William M. Webster, Vice-President, RCA Laboratories, announces the organization of the RCA Laboratories as follows: **Jon K. Clemens,** Staff Vice-President, Consumer Electronics Research; **Bernard J. Lechner,** Staff Vice-President, Advanced Video Systems Research; **Robert D. Lohman,** Staff Vice-President, Solid State Research; **James L. Miller,** Staff Vice-President, Manufacturing and Materials Research; **Kerns H. Powers,** Staff Vice-President, Government Systems and Communications Research; **Richard E. Quinn,** Staff Vice-President, Administration; **Brown F Williams,** Staff Vice-President, Display and Optical Systems Research; and **Dominick A. Zurlo,** Staff Vice-President, Industrial Relations.

Jon K. Clemens, Staff Vice-President, Consumer Electronics Research, announces his organization as follows: **Arch C. Luther,** Senior Staff Scientist; **David D. Holmes,**

Director, Television Research Laboratory; **Arthur Kaiman,** Director, Digital Products Research Laboratory; **Diane P. Smook,** Director, Special Programs; and **John A. vanRaalte,** Director, VideoDisc Systems Research Laboratory.

Bernard J. Lechner, Staff Vice-President, Advanced Video Systems Research, announces his organization as follows: **Curtis R. Carlson,** Head, Image Quality and Human Perception Research; **Charles H. Anderson,** Fellow, Techical Staff; **Frank J. Marlowe,** Head, Digital Video Research; **Paul Schnitzler,** Head, Broadcast Systems Research; and **Robert E. Flory,** Fellow, Technical Staff. Drs. Carlson, Marlowe, and Schnitzler will report to the Staff Vice-President, Advanced Video Systems Research.

Robert D. Lohman, Staff Vice-President, Solid State Research, announces his organization as follows: **Walter J. Merz,** Director, Laboratories RCA, Ltd. (Zurich); **Norman Goldsmith,** Head, Lithography and IC Processing Research; **Gary W. Hughes,** Head, LSI Imagers and Special Devices Research; **Walter F. Kosonocky,** Fellow, Technical Staff; **Louis S. Napoli,** Head, LSI Memories and Devices Research; **Charles J. Nuese,** Head, Silicon Device Research; **Jacques I. Pankove,** Fellow, Technical Staff; **George L. Schnable,** Head, Device Physics and Reliability Research; and **David E. O'Connor,** Senior Project Manager. Messrs. Goldsmith, Hughes, Merz, Napoli, Nuese, O'Connor, and Schnable will report to the Staff Vice-President, Solid State Research.

Kerns H. Powers, Staff Vice-President, Government Systems and Communications Research, announces the appointment of **Leonard Schiff** as Director, Communications Research Laboratory.

Kerns H. Powers, Staff Vice-President, Government Systems and Communications Research, announces his organization as follows: **Leonard Schiff,** Director, Communications Research Laboratory; **Paul Hashfield,** Head, Communications Technology Research; **Allen H. Simon,** Fellow, Technical Staff; **Charles B. Oakley,** Head, Satellite Transmission Technology Research; **Leonard Schiff,** Acting, Communications Analysis Research; **Harold Staras,** Fellow, Technical Staff; **Fred Sterzer,** Director, Microwave Technology Center; **Erwin F. Belohoubek,** Head, Microwave Circuits Research; **Ho-Chung Huang,** Head, Microwave Process Research; **Yegna S. Narayan,** Head, Microwave Device Research; **Markus Nowogrodzki,** Head, Subsystems and Special Projects; and **Herbert J. Wolkstein,** Manager, Space and Countermeasure Programs. Drs. Schiff and Sterzer will report to the Staff Vice-President, Government Systems and Communications Research.

Carmen A. Catanese, Director, Picture Tube Systems Research Laboratory, announces his organization as follows: **David L. Staebler,** Head, Magnetic Deflection Research; **William H. Barkow,** Fellow, Technical Staff; **Norman D. Winarsky,** Head, Electron Optics and Applied Mathematics Research; **Roger L. Crane,** Fellow, Technical Staff; and **Ralph W. Klopfenstein,** Fellow, Technical Staff. Drs. Staebler and Winarsky will continue to report to the Director, Picture Tube Systems Research Laboratory.

David D. Holmes, Director, Television Research Laboratory, announces the appointment of **Edgar J. Denlinger** as Head, Signal Conversion Systems Research.

New Products Division

Erich Burlefinger, Division Vice-President and General Manager, New Products Division, announces his organization as follows: **Harold R. Krall,** Division Vice-President, New Product Development; **Don R. Carter,** Director, Tube Operations; **Reginald R. Pattey,** Director, Strategic Planning and Services; **Ronald G. Power,** Director, Solid State Emitters and Detectors; **Carl L. Rintz,** Director, Closed Circuit Video Equipment; **Randolph C. Rose,** Director, Finance; and **Eugene D. Savoye,** Director, CCD and Silicon Target Technology. Messrs. Krall, Carter, Pattey, Rintz, Rose, and Savoye will report to the Division Vice-President and General Manager.

Solid State Division

Robert S. Pepper, Vice-President and General Manager, announces his organization as follows: **Herbert V. Criscito,** Division Vice-President, Marketing; **Peter A. Friederich,** Division Vice-President, Industrial Relations; **Larry J. Gallace,** Director, Product Assurance; **Donald W. Gangaware,** Director, Strategic Planning and Services; **Walter J. Glowczynski,** Division Vice-President, Finance; **Robert P. Jones,** Director, Power Operations; **Heshmat Khajezadeh,** Division Vice-President, Standard Integrated Circuit Products; and **Jon A. Shroyer,** Division Vice-President, LSI Products and Technology Development.

Heshmat Khajezadeh, Division Vice-President, Standard Integrated Circuit Products, announces his organization as follows: **Stephen C. Ahrens,** Director, Engineering – Standard Integrated Circuit Products; **Richard E. Davey,** Director, Manufacturing – Standard Integrated Circuit Products; **John R. Kowalak,** Administrator, Standard Integrated Circuit Products Administration; **Arthur M. Liebschutz,** Manager, Long Range Product Planning; and **James L. Magos,** Director, Product Marketing – Standard Integrated Circuit Products.

Stephen C. Ahrens, Director, Engineering – Standard Integrated Circuit Products, announces his organization as follows: **Charles Engelberg,** Manager, Test Engineering; **Merle V. Hoover,** Manager, Engineering, Computer, Telecom and Industrial Products; **Lewis A. Jacobus, Jr.,** Manager, Engineering – Logic Products; **Sterling H. Middings,** Section Manager, Layout Services; and **Bruno J. Walmsley,** Manager, Engineering – Automotive and Consumer Products. Messrs. Engelberg, Hoover, Jacobus, Middings, and Walmsley will report to the Director, Engineering – Standard Integrated Circuit Products.

James L. Magos, Director, Product Marketing – Standard Integrated Circuit Products, announces his organization as follows: **Richard E. Funk,** Manager, Applications Engineering; **James L. Magos,** Acting Manager, Product Marketing – Distribution; **James L. Magos,** Acting Manager, Product Marketing – Computer & Industrial Products; **James L. Magos,** Acting Manager, Product Marketing – QMOS & Telecommunications Products; and **Seymour Reich,** Manager, Product Marketing – Automotive & Consumer Products. Messrs. Funk and Reich will report to the Director.

N.C. Turner, Manager, Trident and High Reliability Operations, and **L.J. Gallace,** Director, Product Assurance, announce that the Materials and Processing Laboratory formerly reporting to **Norman C. Turner,** Manager, Trident and High Reliability Operations is transferred to the Product Assurance organization. This function will report to **Leonard H. Gibbons, Jr.,** Manager, Reliability Engineering Laboratory.

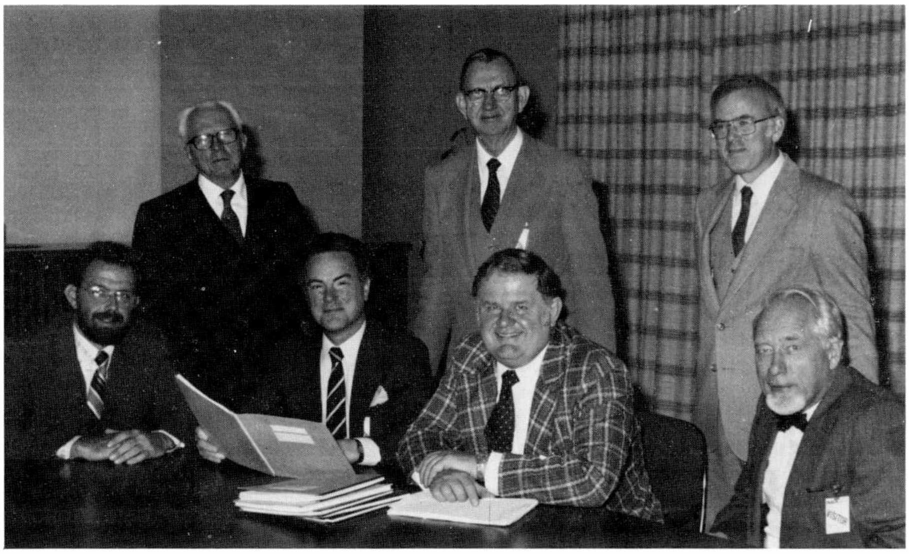
VideoDisc Division

Arnold T. Valencia, Division Vice-President and General Manager, VideoDisc Division, announces his organization as follows: **David M. Arganbright,** Division Vice-President, Marketing Operations; **Bruce G. Babcock,** Division Vice-President, Custom Pressing Marketing; **Jay J. Brandinger,** Division Vice-President, Disc Operations; **Joseph P. Clayton,** Division Vice-President, Consumer Sales; **Mark L. Frankel,** Division Vice-President, Finance; and **Herbert J. Mendelsohn,** Division Vice-President, Advertising and Merchandising.

Eta Kappa Nu Jury of Award meets at RCA

The 1983 Eta Kappa Nu (Honorary Electrical Engineering Society) Jury of Award convened at RCA "SelectaVision" VideoDisc Operations on November 14, 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:

- **Mr. George L. Benning,** Vice-President for Advanced Technology and Engineering, Collins Avionics Group, Rockwell International Corporation;
- **Dr. J. Robert Betten,** Professor of Electrical Engineering, University of Missouri, (Past President of Eta Kappa Nu);
- **Dr. Edward M. Davis,** President, General Technology Division, IBM Corporation;
- **Mr. Stephen A. Mallard,** Sr. Vice-President, Planning and Research, Public Service Electric & Gas Company;
- **Mr. E.D. Maynard, Jr.,** Director, VHSIC PROGRAM, Office of the Under Secretary of Defense for Research and Engineering; and
- **Dr. George F. Mechlin,** Vice-President, Research and Development, Westinghouse Electric Company.



1983 Jury of Award. Seated (left to right): ***E.D. Maynard, Jr.; Edward M. Davis; J. Robert Betten; George F. Mechlin.*** *Standing (left to right):* ***Stephen A. Mallard; George L. Benning; James A. D'Arcy.***

The purpose of the Eta Kappa Nu Recognition Award is to emphasize 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-seven 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 Labs) who was the 1977 winner. This year, two RCA employees are being recognized as finalists; they are: **Russell R. Barton**, RCA Laboratories, Princeton, N.J.; and **John W. Betz**, RCA Automated Systems Division, Burlington, Mass. The award is given on the basis not only of what success the young Electrical Engineers have had in their vocation, but also what they did to broaden themselves culturally and what they did for others. The 1983 winners will be honored at an award banquet on April 30, 1984, in Philadelphia.

D'Arcy elected Director of Electrical Engineering Honor Society

James A. D'Arcy, Senior Member of the Engineering Staff at RCA VideoDisc Operations in Indianapolis, Indiana, has been elected a Director of the Electrical Engineering Honor Society, Eta Kappa Nu. Founded in 1904 at the University of Illinois, Eta Kappa Nu today has a membership of about 150,000 of the top electrical engineering graduates with student chapters in 168 of the leading universities and colleges in the United States, four alumni chapters and seven foreign branches (Chapters-At-Large).

During his professional career, Mr. D'Arcy has published several technical articles and has been involved in various technical societies and activities. He was active in the Young Engineers' Organization of the Engineers' Club of Philadelphia, serving in various offices, including President (1969-1970), and he received the Club's Outstanding Young Engineers' Award in 1967.

Since 1974, he has been a member of the Eta Kappa Nu Award Organization Committee and has served as Chairman since 1979.

He was a member of the Engineering Excellence Committee of RCA Astro-Electronics for several years, serving as Chairman on two occasions. He was appointed a charter member of the Technical Excellence Committee of RCA VideoDisc Operations when it was being formed in 1980 and served as the first chairman of this committee in 1981.

Prize-winning paper is national contest entry

Rutgers University School of Law has selected a paper written by a Moorestown engineer as its first-prize entry in a national competition. **John Corbett's** paper, entitled "Who Owns the Program?: Adverse Rights of Employees, Consultants, and Independent Contractors in Copyrights of Computer Program Products," was selected by Rutgers as its first-prize entry in the Nathan Burkan annual competition for scholarly legal papers on copyright law. This competition, sponsored by the American Society of Composers, Authors and Publishers (ASCAP), is open to all accredited law schools in the United States and awards prizes for the best five papers entered; the winning papers will then be published in ASCAP Copyright Law Symposium Number Thirty-Three.

John is Unit Manager, Combat System Definition, in the Naval Systems Department's Combat System Design organization. His engineering career dates from 1968 when he received the BSEE degree from Rensselaer Polytechnic Institute. Since joining RCA in 1976, he has been involved in systems engineering tasks on the AEGIS Program. John is in his final year at Rutgers Evening Division law school, scheduled to graduate in May 1984.

Hoffman is president of engineering council

Dorothy M. Hoffman, RCA Laboratories, has been elected president of the Engineering and Technical Societies Council of Delaware Valley, Inc. This non-profit cooperative of over 30 professional societies provides career guidance for young people, public service for the community, and professional betterment for the members. The headquarters is at the Engineers' Club in Philadelphia, Pa.

IEEE Society to meet

The IEEE Professional Communication Society will meet on October 10-12, 1984 in Atlantic City, N.J. The Program Chairman is **Jack Friedman,** Technical Publications, RCA Missile and Surface Radar in Moorestown. Designed as a forum for engineers, managers, professional communicators, educators and others, the program this year emphasizes practical aspects of scientific and technical communication. Prospective authors are invited to get a list of topics from the Program Chairman (TACNET: 224-2112). A 250-word abstract is due on February 29, 1984.

Astro engineers belong to IEEE working groups

Ronald L. Cariola, Manager, Ground Software Development is a member of the IEEE Working Group for a Standard for Software Reviews and Audits. **Raouf H. Farag,** Manager, Ground Software Development, is a member of the IEEE Working Group on a Guide for Software Configuration Management.

Astro scientist chairs robotics sessions

Dr. Kamal Karna, Senior Staff Scientist, RCA Astro-Electronics, chaired two sessions at the Twenty-sixth IEEE Computer Society International Conference (Compcon 83) held in Crystal City, Virginia, on September 26-30, 1983. He chaired a robotics session that included four papers and a computer-vision session that included four papers.

MSR's Bauer honored as Engineer of the Year

Paul E. Wright, *Division Vice-President and General Manager of Government Systems Division, congratulates* ***John Bauer*** *on his selection as Engineer of the Year by the International Electronics Packaging Society.*

The International Electronics Packaging Society has named **John A. Bauer** its Engineer of the Year for 1983. The Society, which makes only one award each year, honored Bauer in its Annual Meeting in Chicago for his long-term contributions to improving the state-of-the-art in microelectronics packaging and to disseminating the information both nationally and internationally.

Bauer, who is Manager, Design Automation at RCA's Missile and Surface Radar operation in Moorestown, N.J., introduced the chip-carrier ceramic module to the industry, demonstrating its potential for high density and high reliability. He was also an early advocate of applying design automation tools to the chip-carrier technology. His contributions in electronic packaging technology are currently being implemented in the Surface Mount Technology, which is providing approaches to bring down the size, weight, and cost of electronics for all applications.

Kant receives Founder's Award

Milton Kant, Senior Member Engineering Staff, Naval Systems Department, Missile and Surface Radar, received the Founder's Award from the Institute of Electrical and Electronics Engineers Electromagnetic Compatibility (EMC) Society at the 1983 IEEE International Electromagnetic Compatibility 25th Anniversary Symposium held in Washington, D.C., on August 23-25, 1983.

The Founder's Award was given in recognition of contributions to the October 10, 1957, founding of the IEEE EMC Society. Mr. Kant was one of six founders of the Institute of Radio Engineers Professional Group on Radio Frequency Interference, forerunner of today's 2000-member IEEE EMC Society. He has served in numerous capacities with the society over the years, including Administrative Committee member, Newsletter Committee member, Secretary of the society, and chairman of the RFI/EMC symposium held in New York City in 1965.

He was awarded the IEEE EMC Society Certificate of Appreciation in 1968. From 1968 to 1977 he served as chairman of the society Information Retrieval Committee, which prepares EMC abstracts to be published in the EMC Society Newsletter, He was chairman of the Philadelphia Chapter from 1978 to 1983.

RCA participates in recent ISHM conference

The annual meeting of the International Society for Hybrid Microelectronics was held in the Philadelphia Civic Center and Franklin Plaza Hotel, Oct. 30 to Nov. 3, 1983. **Dr. Thomas T. Hitch** of the RCA Laboratories' Electronics Packaging Group was the Program Chairman. **Dr. Ashok N. Prabhu,** also of Electronics Packaging, participated in abstract selection and helped to organize the technical paper session on the topic of Soldering. **Richard Brown** of the Microwave Technology Center at the Labs led a short course entitled "Introduction to Microwaves." Missile and Surface Radar employees also participated in the program: **John F. Bauer** and **Mitchell Davis** were session chairmen. In addition, **Alan S. Cooper** gave a paper and **R.L. Schelhorn** spoke in a round table. **Bernard Greenstein** of Consumer Electronics and **Robert R. Bigler** of Government Systems, Camden, were session chairmen.

The conference proceedings include 105 papers. Four ISHM monographs are available describing the short courses. Dr. Hitch gratefully acknowledges the support of RCA for this work and thanks **Lu Onyshkevych, John Vossen,** and **Dorothy Hoffman** for their excellent advice during the organization of the technical program.

Chu is committee chairman

David Chu, RCA Astro-Electronics, served as Chairman for the Modal Testing and Model Refinement Symposium at the ASME Winter Annual Meeting in Boston on November 14-18, 1983. He was elected to be Secretary of the Shock and Vibration Committee Meeting for June 1984 to June 1985, and will become Chairman of the Committee for June 1985 to June 1986.

RCA and NASA receive group achievement award for Shuttle radio

A team of RCA and NASA engineers has received the space administration's group achievement award for producing the space shuttle's UHF radios.

Six RCA and three NASA employees were recognized for "outstanding performance and technical creativity in accomplishing the development and provisioning of the Shuttle Extravehicular Activity/Air Traffic Control Communications System at low cost while meeting all other program requirements."

Named in the citation were: **Sam B. Holt, John O. Sheldahl,** and **Charles T. Shelton** of Government Communications Systems in Camden, N.J.; **John C. Feltus, Robert L. Black,** and **Ronald A. Sykora** of RCA Service Company in Houston; and Richard W. Armstrong, Paul E. Shack, and Gareth H. Nason of NASA in Houston. NASA said the team's significant accomplishments included development of a management and technical concept resulting in a 50-percent cost savings for the program.

The RCA-built radios played an important role in the first shuttle flight when they became the only line of communications during key moments of the launch and landing of shuttle orbiter Columbia. The main, S-band radio failed to provide communications, so the astronauts turned on the UHF radio and kept it going for the rest of the 54-hour flight. The first voice the astronauts heard after they came out of reentry "blackout" was over the UHF radio.

The radios continued to perform successfully in future flights, and RCA-built UHF backpack radios were used by the astronauts during the STS-6 flight when they worked in space outside the shuttle orbiter Challenger.

Technical excellence

Americom presents Technical Excellence Awards

RCA Americom Technical Excellence Award. From left to right: ***Fred Hoedl, Carlton Barnes, David Fremont, Steve Osman, and Dem Ilagan*** *with* ***John Christopher,*** *Vice-President, Technical Operations.*

The Americom Technical Excellence Committee announced that the latest recipients of Americom's Technical Excellence Award are **Fred Hoedl, Carlton Barnes, Steve Osman, Dem Ilagan** (members of the Microwave Engineering Group) and **David Fremont,** Manager, PMO. The team has been recognized for their outstanding accomplishment in implementing the complex Digital Audio Transmission System (DATS).

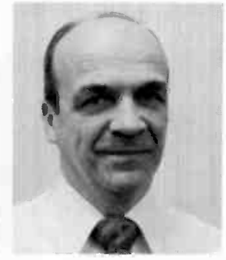
The team's dedication and combined skills—used during the installation, alignment, and debugging of the advanced digital audio equipment—enabled RCA Americom to overcome some of the technical and delivery problems experienced by our hardware subcontractor. The team's concerted effort was key to a successful conclusion to an extremely difficult service offering.

The Digital Audio Service has been accepted by all four networks, ABC, CBS, NBC, and RKO and is now operational across the country, offering quality audio performance in a cost-efficient manner to our customers. In addition to cash awards, the team was honored at a luncheon.

SSD Mountaintop award given

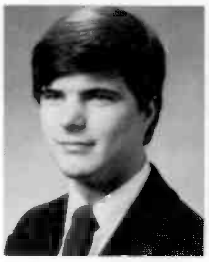
The December, 1983, Mountaintop Technical Excellence Award, designed to recognize and reward members of the technical community who have consistently exhibited qualities of initiative, leadership, technical competence, attitude and follow-up, has been given to **Jim Hoshowsky, Carl Obaza,** and **Frank Zdancewicz.**


Hoshowsky


Obaza


Zdancewicz

Moorestown third-quarter 1983 Technical Excellence Award winners


Clark

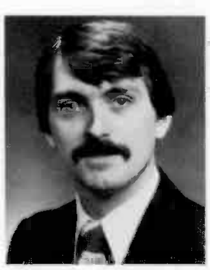

Grandmaison


Drenik


Fanelle


Kent


Link


Matthews


Nesbit

Richard D. Clark—for developing a computer-controlled acceptance testing station for the EDM-4 array column assembly, ensuring proper connections to every wired component and verifying absence of crosstalk. He selected equipment for the test station and developed a flexible software design for rapid isolation and resolution of crosstalk problems. The station, which worked efficiently from the start, is being replicated for future growth test applications.

Nelson De Grandmaison—for systems engineering contributions to the Federal Aviation Agency's Advanced Automated Program host computer system design competition. His development of an accurate method to verify cost proposals proved invaluable to the FAA, as did his approach to contract data item review. The methodologies developed by Mr. De Grandmaison gave the FAA a fundamental structure for coordinated, disciplined reviews of contract documentation and a firm technical basis for cost proposal analysis.

John Drenik—for his outstanding achievement in the design of the EDM-4 antenna assembly, involving modularization of columns and integration of all components in self-sustaining subassemblies independent of the antenna support structure. His unique configuration of the Beam Steering Controller and power supplies in a totally integrated antenna assembly and his use of pre-tested column assemblies permit production of AN/SPY-1B antennas at rates necessary to meet Navy shipbuilding schedules.

John D. Fanelle—for a successful system design and development of an Imbedded Sub-Element Test System (ISETS) for the EDM-4 signal processor. He developed the algorithms and hardware configuration to support fault detection and isolation, plus a microcomputer Maintenance Panel concept design that proved successful in system checkout. His proposal for eliminating a potential data-base memory overflow has been highly effective in actual hardware demonstrations.

Edward J. Kent—for special achievements in stripline component development for the EDM-4 receive-column combiner networks. Combining analytical tools with precise empirical procedures, he developed five different T-junctions as the basis for computer-generation of the 200 specific designs needed for 40 different combiner circuits. He also devised a method of fine phase adjustment of these networks by means of capacitive irises to compensate for minute, but critical, variations in material characteristics.

Alan J. Link—for extensive personal contributions to the design, testing, and integration of the TV-tracker software modification of the NIDIR radar system at Aberporth, Wales. In the process of developing the necessary modifications, he also discovered and fixed several software problems and resolved unexpected instabilities in the radar. His ability to isolate and resolve hardware and software problems was a key to customer acceptance of the modification.

David L. Matthews—for outstanding performance in the development of timing and control application firmware for the microcomputer control network of the EDM-4 signal processor. His technical direction and personal design contributions were key elements in the successful implementation of a set of complex real-time programs for the Executive, Waveform Generator, and Detection microcomputers. The resulting working programs have supported all significant milestones of frame test and signal processor integration.

Gerald Nesbit—for unique contributions to critical microwave component development for an L-band solid-state phased array T/R module. Mr. Nesbit's successful use of GaAs FET devices in his low-noise L-band amplifier design was without precedent at microwave frequencies in Moorestown, and he also developed an innovative RF connector design for the T/R module. His contributions will play a significant role in RCA's entry into the airborne radar market.

Microfilms, Indexes, and Back Issues Put the *RCA Engineer* at Your Fingertips

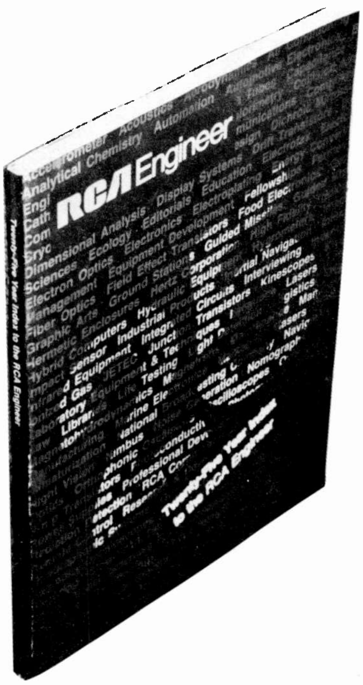
The back issues of the *RCA Engineer*—over 160 of them—record RCA's progress in invention, development, and manufacturing. You can take advantage of this wealth of technical information by using the cumulative *25-Year Index to the RCA Engineer,* published in 1980. We have recently updated this reference tool with a supplementary *Index to Volumes 26, 27 and 28*—covering 1980 through 1983.

Subject and author indexes can help you find specific information that you need for your work. Or the *Index* might provide a vital contact in another RCA business, someone who has related experience.

Free copies available: You can obtain your personal copies of the *25-Year Index* and its *Supplement* at your RCA Technical Library. Or, write to: *RCA Engineer Indexes,* Technical Excellence Center, 13 Roszel Road, Princeton, NJ 08540.

Once you've found the article you want to read, your RCA Librarian has a complete set of back issues for your use—in hard copy or microfilm edition. Moreover, back issues and reprints listed below may be ordered internally from the *RCA Engineer* Editorial Office at "RCA Laboratories—Roszel Road, Princeton."

All back issues are $2.00. (The *Index* and supplement are free.) To order, photocopy this form, and include your name, RCA charge number, and your internal mailing address. Please don't send cash.



THEME-RELATED ISSUES

- ☐ Consumer Electronics Manufacturing
  VOL. 24, NO. 4 Dec 78/Jan 1979
- ☐ Microelectronics
  VOL. 24, NO. 6 Apr/May 1979
- ☐ Microprocessor Applications
  VOL. 25, NO. 3 Oct/Nov 1979
- ☐ Quality (Do it right the first time)
  VOL. 25, NO. 4 Dec 79/Jan 1980
- ☐ RCA International
  VOL. 25, NO. 5 Feb/Mar 1980
- ☐ Color TV Receivers
  VOL. 25, NO. 6 Apr/May/Jun 1980
- ☐ 25th Anniversary Issue
  VOL. 26, NO. 1 Jul/Aug 1980
- ☐ Communications Trends
  VOL. 26, NO. 3 Nov/Dec 1980
- ☐ Mechanical Engineering
  VOL. 26, NO. 4 Jan/Feb 1981
- ☐ Increasing Your Effectiveness
  VOL. 26, NO. 5 Mar/Apr 1981
- ☐ "SelectaVision" VideoDisc, Part II
  VOL. 27, NO. 1 Jan/Feb 1982
- ☐ Manufacturing Engineering
  VOL. 27, NO. 2 Mar/Apr 1982
- ☐ Electro-Optics
  VOL. 27, NO. 3 May/Jun 1982
- ☐ 26th Anniversary Issue
  VOL. 27, NO. 4 Jul/Aug 1982
- ☐ Microwave Technology
  VOL. 27, NO. 5 Sept/Oct 1982
- ☐ Modeling, Simulation, and Analysis
  VOL. 27, NO. 6 Nov/Dec 1982
- ☐ Engineering Productivity
  VOL. 28, NO. 1 Jan/Feb 1983
- ☐ Space Technology
  VOL. 28, NO. 2 Mar/Apr 1983
- ☐ Technology Transfer
  VOL. 28, NO. 3 May/Jun 1983
- ☐ 27th Anniversary Issue: Systems Engineering
  VOL. 28, NO. 4 Jul/Aug 1983
- ☐ Digital Broadcast
  VOL. 28, NO. 5 Sept/Oct 1983
- ☐ Software
  VOL. 28, NO. 6 Nov/Dec 1983

COLLECTIONS OF ARTICLES

- ☐ Consumer Electronics
  PE-515 1971
- ☐ Solid State Technology
  PE-552 1972
- ☐ COS/MOS Technology
  PE-580 1973
- ☐ Automotive Electronics
  PE-681 1976
- ☐ Microprocessor Technology
  PE-702 1976
- ☐ Electro-Optics
  PE-703 1976
- ☐ Picture Tube
  PE-748 1980

CUMULATIVE INDEX & supplement

- ☑ 25-Year Index to *RCA Engineer*
  Aug 1980
- ☑ Index to Volumes 26-28
  Dec 1983

Editorial Representatives

Contact your Editorial Representative at the TACNET numbers listed here to schedule technical papers and announce your professional activities.

Broadcast Systems Division (BSD)

		TACNET
*Jack Dearing	Camden, New Jersey	222-4688

Consumer Electronics (CE)

*Eugene Janson	Indianapolis, Indiana	422-5208
Francis Holt	Indianapolis, Indiana	422-5217
Larry Olson	Indianapolis, Indiana	422-5117
Byron Taylor	Indianapolis, Indiana	426-3367
Don Willis	Indianapolis, Indiana	422-5883

Corporate Technology

*Tony Bianculli	Princeton, New Jersey	226-2111

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 Coleman	Princeton, New Jersey	229-2919

Automated Systems

*Paul Seeley	Burlington, Massachusetts	326-3095
Dale Sherman	Burlington, Massachusetts	326-3403

Government Communications Systems

*Dan Tannenbaum	Camden, New Jersey	222-3081
Thomas Altgilbers	Camden, New Jersey	222-3351

GSD Staff

*Peter Hahn	Cherry Hill, New Jersey	222-5319

Missile and Surface Radar

*Don Higgs	Moorestown, New Jersey	224-2836
Graham Boose	Moorestown, New Jersey	253-6062
Jack Friedman	Moorestown, New Jersey	224-2112

National Broadcasting Company (NBC)

*Bob Mausler	New York, New York	324-4869

New Products Division

R.E. Reed	Lancaster, Pennsylvania	227-2485
Bob McIntyre	Ste Anne de Bellevue	514-457-9000

Cablevision Systems

*John Ovnick	Van Nuys, California	534-3011

Patent Operations

		TACNET
George Haas	Princeton, New Jersey	226-2888

RCA Communications

American Communications

*Murray Rosenthal	Princeton, New Jersey	258-4192
Carolyn Powell	Princeton, New Jersey	258-4194

Global Communications

*Dorothy Unger	Piscataway, New Jersey	335-4358

RCA Laboratories

Eva Dukes	Princeton, New Jersey	226-2882

RCA Records

*Greg Bogantz	Indianapolis, Indiana	424-6141

RCA Service Company

*Murray Kaminsky	Cherry Hill, New Jersey	222-6247
Dick Dombrosky	Cherry Hill, New Jersey	222-4414
Ray MacWilliams	Cherry Hill, New Jersey	222-5986

"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	327-1473 or 327-1264

Integrated Circuits

Dick Morey	Palm Beach Gardens, Florida	722-1262
Sy Silverstein	Somerville, New Jersey	325-6168
John Young	Findlay, Ohio	425-1307

Video Component and Display Division

*Ed Madenford	Lancaster, Pennsylvania	227-3657
Nick Meena	Circleville, Ohio	432-1228
J.R. Reece	Marion, Indiana	427-5566

***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
the RCA Technical Excellence Center
"by and for the RCA engineer"

13 Roszel Road, P.O. Box 432, Princeton, NJ 08540
Address correction requested

BULK RATE
US Postage
PAID
Phila., Pa.
Permit No. 2906

EJ NOSSEN
203 ELKINS RD
CHERRY HILL
NJ 08034
GS DGC

Printed in the U.S.A.
Form No. RE-29-1