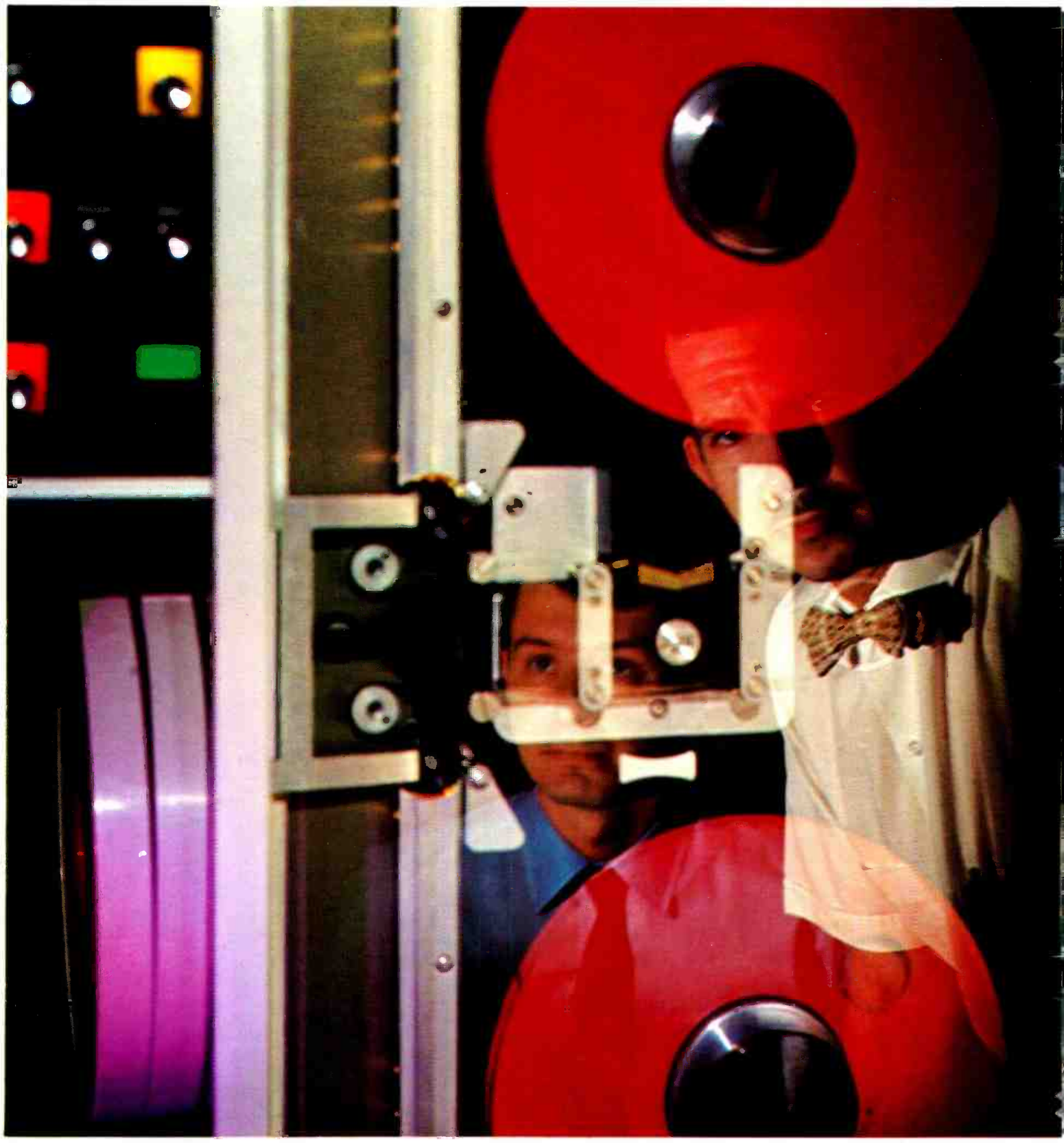
RCA Engineer

Volume 14 No. 3 October-November, 1968



# The Tyranny of Numbers

Lately I have become increasingly aware of the impact on many human activities of what I have been calling "the tyranny of numbers versus the constancy of humans and resources." The principle involved is basically quite simple and obvious. It recognizes that, to a first approximation, humans have essentially uniform physical characteristics such as visual and aural acuity, reaction time, and rate of acceptance and dispensing of information. The ability of a population to do work involving such parameters is, therefore, directly proportional to the size of the population.

In our complex society, work is often generated at a rate proportional to a greater population size. Under such circumstances, we can and do find the demand for certain work exceeding the supply of people to do it. A current example is the closing of the New York Stock Exchange on Wednesdays so Wall Street can catch up on its paper work.

We often solve such problems by automation. The change from the manual connecting of telephones to automatic dialing is a classic example.

In RCA's basic business—information handling—we have several parallel problems. The size and complexity of information systems have been growing at a rate which greatly exceeds the rate of growth of our population. Thus, we will soon exceed the supply of scientists, engineers, technicians, and draftsmen needed to conceive, develop, design, and manufacture the information systems of the future. The intimate involvement of computer systems in all aspects of these functions seems to be the main key to the solution of these problems.

It is very appropriate and timely that the **RCA Engineer** should devote an issue to the theme of persuading RCA's scientists and engineers to use the computer more often and more effectively.

To summarize my message I can say simply, "Time's awasting fellows; let's get at it."

John Hillier

**Dr. J. Hillier,** Vice President
Research and Engineering
Princeton, N.J.

**Our Cover**

Reflections in the computer's glass door create the impression of unity between man and machine. In the cover and in this issue, the purpose is to show a harmonious interaction between man and machine—with the computer an extension of, not a replacement for man's intellect. The engineers in the photo are Dave Ressler (left) and J. Rogers Woolston of the RCA Laboratories.

**Photo credit:** Tom Cook, RCA Laboratories.

# RCA Engineer

Vol 14 No 3
Oct|Nov
1968

**A technical journal published by**
**RCA Product Engineering 2-8, Camden, N.J.**

*RCA Engineer* articles are indexed annually in the April-May Issue and in the "Index to RCA Technical Papers."

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

## Table of contents

### Papers

### Notes

### Departments

# editorial input

# Fortran — friend or foe?

Most papers in this issue mention computer time sharing; although not the original plan, this evolved because authors were asked to describe basic applications of the computer in their engineering work. The authors did their work well, and their application papers—supplemented by introductory papers—comprise an issue that will inform and assist both the neophyte and experienced user in applying the computer in their every-day work.

At this point, one might ask, "Why an issue discussing basics; don't most of our engineers already know how to apply the computer to their work?" From what intelligence we have been able to gather, the converse is probably true. Our first requests for this "computer" issue came in the returns of the reader survey, and these were supplemented strongly by discussions with several individuals responsible for encouraging computer use in engineering throughout the Corporation.

It seems paradoxical that many engineers, infused since college with a strong sense of efficiency, are not using computers—even though such time-saving machines have been around for more than two decades.

An excuse often given is that the equipment is not available or that there is too much red tape required to justify computer use. This excuse may have had some merit several years ago; however, computer centers are now available throughout the Corporation to process work on at least a 24-hour turn-around basis (see page 49 of this issue). The introduction of time-sharing systems throughout RCA has further alleviated this problem.

A more tenable reason for the lack of computer/engineer interaction could be the traditional difficulty that people have in learning a new language; this difficulty may be symptomatic of self satisfaction, or an acceptance of the *status quo*.

Many engineers pay dearly for this privilege of not learning the computer's language: they must perform laborious and repetitive hand calculations, use crude numerical analysis techniques, and waste precious engineering hours in testing breadboard models for data that could well have been supplied by a mathematical model.

Just as the enterprising foreign visitor to America learns English in his search of the "yankee dollar," so has the computer gone a long way toward speaking the engineer's language. With the availability of such high-level computer languages as Fortran, the engineer need invest only a few hours of his time to become a computer user. For proof of this statement, study Dr. Gordon's paper, "Fortran Programming is Easy." If this paper gets your interest, as we feel it will, go next to Murray Spencer's paper on using RCA's time-sharing system. Time-sharing consoles are available in most RCA facilities, and they allow you to interact directly with a computer for a very reasonable cost (about $7/hr.). Then read some of the actual engineering applications described in this issue; many of the authors became computer users less than a year ago, and their papers exhibit an impressive amount of understanding for the plight of the reader not yet familiar with the computer.

Without a doubt, the computer can be a useful and powerful ally to the engineer; it can lighten the burden of mundane mechanical steps involved in many engineering tasks and allow the engineer to concentrate on the more creative aspects of his profession.

We thank the authors in this issue who recognized a problem area and did their part to help solve it. Special thanks go also to Dr. Gary Gordon, who wrote two key papers for this issue and served as technical advisor for the entire issue. If the readers dig into this issue with the same diligence and enthusiasm as the authors, we have no doubt that RCA will soon have many more computer users.

---

**Future Issues**

The next issue of the *RCA Engineer* discusses electron tubes and devices designed and produced by engineers in the Lancaster, Pa., facility of Electronic Components. Some of the topics to be covered are:

**Ruggedized ceramic vidicons**

**Recent developments in photoconductors**

**Color picture tube development**

**Phosphors for color picture tubes**

**Camera tubes for space**

**Cermalox tubes for SSB**

**The coaxitron—a 1 megawatt IC**

**Power triode for RF cooking at 915 MHz**

**Colorimetry, brightness, and contrast in color picture tubes**

**Noble gas ion lasers**

**Heat pipes**

Discussion of the following themes are planned for future issues:

**General review of computers**

**Product and system assurance, reliability, value engineering**

**Microwave devices and systems**

**Interdisciplinary aspects of modern engineering**

**Lasers**

**RCA engineering on the West Coast**

**Computerized Educational Systems**

# What a scientific computer center can do for the engineer

**J. B. Vail | J. R. Sandlin**

**The computational services available to the engineer are many and varied; the typical Scientific Computer Center offers many services to aid in project accomplishment. In this paper, the counsel and assistance available for task definition are described and the procedures, capabilities and equipment complement of a Computer Center are explained. Guidelines for efficient use of the services are offered, and a look at future developments of interest to the engineer is presented.**

---

The engineer requiring computational services today is faced with a bewildering array of choices. He has available to him equipment ranging from desk calculators, time sharing consoles, small general purpose computers up through large scale scientific facilities. Selections do not end with the hardware; software requirements must be defined, and computer programs appropriate for the task must be selected, modified, or developed. Finally, efficient use must be made of the services available. Poor choices in any category can result in high expenses and delayed schedules in meeting the project goals; it is clear that an understanding of what is available plays a vital role in the probability of success.

## Computer Center Services

Possibly the most significant services available to the engineer user are those of the systems programming staff and program librarian. Training, data preparation and storage, provision of materials, the dispatching function, job set-up, priority service, and accounting procedures are other important services. Understanding these, and using them to their fullest, will be instrumental in the successful completion of the engineer's task.

### System Programming Staff

The members of the system programming staff are the experts on hardware and software aspects of the computers in the facility. They generally include at least one superior programmer with well-rounded experience in many programming languages and technical applications. His function, in addition to the maintenance and updating of the operating software systems, is to counsel the engineer on the choice of programming languages, use of similar programs/subroutines on file or available, and to suggest applicable programming techniques.

The systems programmer is also available to assist in "debugging" programs when in the checkout stage, and can help in interpreting error messages and suggest appropriate use of the diagnostic aids incorporated in the software systems.

### Program Librarian

The program librarian, usually a member of the systems programming staff, will have access to a large number of "applications programs." These may have been prepared at the center and

---

**John R. Sandlin, Jr., Mgr.**
Data Processing and Analysis
Scientific Information Processing Center
Missile and Surface Radar Division
Moorestown, N.J.
received the BS in Electrical Engineering from the University of Florida in 1953. He spent three years as a USAF pilot and electronics officer, and joined RCA in 1956 at the USAF Eastern Test Range Missile Test Project. He performed computer maintenance, design, and system studies, joining M&SR in 1960 on DAMP (Down-range Anti Missile Program) where he supervised the data reduction effort. From 1963 to 1965, he was supervisor of RCA's Data Processing and Analysis activity for the TRADEX radar at Kwajalein, Marshall Islands. Mr. Sandlin returned to the Missile Test Project for two years, where he was manager of the ARIS Re-entry Ship Test Planning and Data Control activity. He is a member of the IEEE, coholder of a patent for a Radar Video Data Reduction System, and has authored numerous reports on missile re-entry test results.

**Jacque B. Vail, Ldr.**
Computer Operations
Scientific Information Processing Center
Missile and Surface Radar Division
Moorestown, N.J.
received the BS in Civil Engineering from Pennsylvania State University in 1954. He was employed by the Ohio Department of Highway as a design engineer in the Photogrammetric Design group, and in 1957 he became Manager of the department's Electronic Computing Laboratory. Mr. Vail joined the M&SR division in 1961 as Leader, Computer Operations in the Scientific Information Processing Center. He has undertaken further studies in Business Administration and Data Processing at Ohio State University and Temple University. Mr. Vail is a registered Professional Engineer in the states of Ohio and New Jersey.

---

Reprinted RE-14-3-22
Final manuscript received August 21, 1968.

entered in the program library or they may have been obtained through various services and users groups. The center at M&SR, for example, has access to many thousands of programs via RCA's PAL (Program Applications Library), SHARE (Society to Help Avoid Redundant Effort), the University of Georgia's COSMIC service, and Indiana University's Aerospace Research Applications Center. Many NASA and Department of Defense programs can be obtained. The program librarian has abstracts, indexes and program documentation; Fig. 1 is an example of a portion of a typical SHARE keyword index. Some programs are immediately available, while others may be ordered with two or three week delivery. Modifications are sometimes needed to satisfy specific requirements. It is frequently possible to identify programs which may be incorporated as "subroutines"; or portions of larger programs; time spent in researching available programs frequently results in considerable savings of time and money.

The program librarian also keeps various manuals and instructional material on file, these are available on request and will be updated as addendum documentation is received. He is also responsible for maintaining current reference material in the customer service area.

Many general purpose subroutines are available to solve specific functions, manipulate data, generate random numbers, etc. These are, in general, packages that could be inserted in the engineer's program, thus saving him the effort of coding and testing that segment.

### Training

Computer languages are dynamic in nature and enhancements are continuously being made. This prompts many installations to conduct frequent training courses in the currently popular problem-oriented computer languages (such as Fortran, Cobol) and machine-dependent assembly languages. The advent of third generation computers will call for more special training courses, particularly in the area of random-access storage techniques. During or after hours courses are taught on a regular schedule or whenever a sufficient number of individuals are interested; sample problems are included as part of the course. RCA's Continuing Engineering Education program offers four courses of specific value to the engineer user of computer services.

### Data Preparation

Most computer centers maintain large data preparation groups to keypunch production jobs for subsequent computer processing. Additionally, many centers set aside a number of keypunch machines for the exclusive use of the engineer. Large jobs are generally processed and verified by skilled operators, but the engineer will frequently prepare a few cards to minimize turnaround time.

### Information Storage

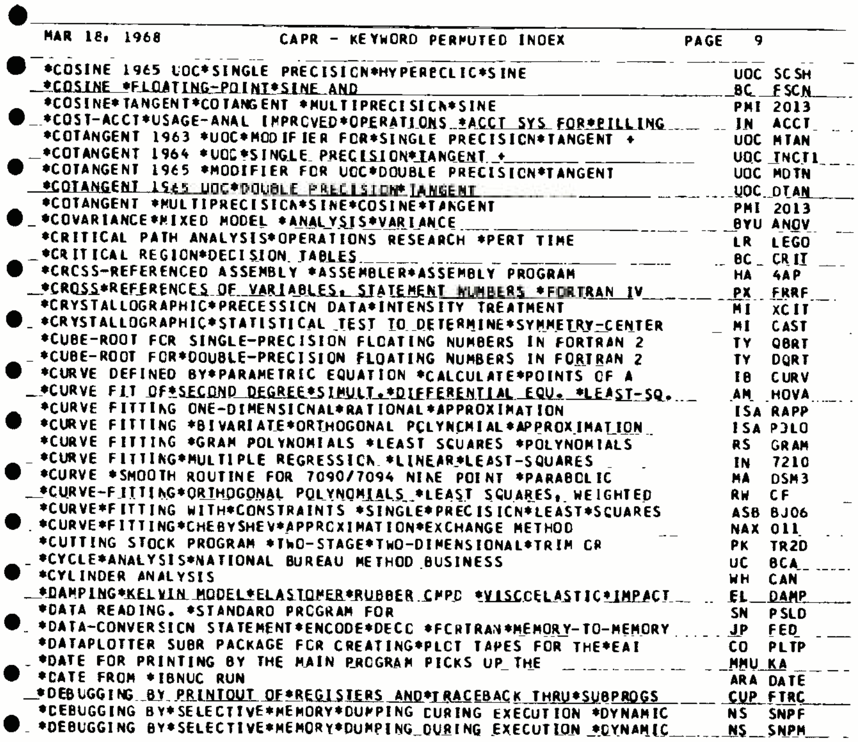
A major service supplied by the Computer Center is temporary use of magnetic storage media such as discs or magnetic tapes. The engineer is allowed to use and save a reasonable number for his job. The Tape Library will allow storing and retrieval of magnetic tape data for a predetermined time. Computer costs and run time may be directly related to the quality of the magnetic tapes used by the facility. The Tape Library will often have the capability to clean, repair, and validate magnetic tapes on site. Cards and other necessary material may be stored in assigned locations.

### Processing the Job

The engineer's program and data will be handled by many people and, most likely, processed by several machines before the final results are returned. The process of controlling the flow of data, collecting and coordinating the various operations and issuing the output is the task of the site dispatcher. Additional dispatcher duties include checking the data for correct control cards, monitoring the progress of the job, and insuring appropriate disposition of the results.

### Computer Applications Group

Once the engineer has developed a working system of programs he may wish to shift the task of collecting data and submitting computer runs to the set-up or applications group. These personnel are familiar with the operation of the data processing activity and can, with the aid of the engineer's instruction, perform the routine data handling functions, telephone coordination, and mail submission/delivery to minimize effort of the engineer.



```
MAR 18, 1968              CAPR - KEYWORD PERMUTED INDEX                    PAGE   9

*COSINE 1965 UOC*SINGLE PRECISICN*HYPERBCLIC*SINE                         UOC SCSH
*COSINE *FLOATING-POINT*SINE AND                                          BC  FSCN
*COSINE*TANGENT*COTANGENT *MULTIPRECISICN*SINE                            PMI 2013
*COST-ACCT*USAGE-ANAL IMPRCVED*OPERATIONS *ACCT SYS FOR*BILLING           IN  ACCT
*COTANGENT 1963 *UOC*MODIFIER FOR*SINGLE PRECISICN*TANGENT +              UOC MTAN
*COTANGENT 1964 *UOC*SINGLE PRECISION*TANGENT +                           UOC TNCT1
*COTANGENT 1965 *MODIFIER FOR UOC*DOUBLE PRECISICN*TANGENT                UOC MDTN
*COTANGENT 1965 UOC*DOUBLE PRECISION*TANGENT                              UOC DTAN
*COTANGENT *MULTIPRECISICN*SINE*COSINE*TANGENT                            PMI 2013
*COVARIANCE*MIXED MODEL *ANALYSIS*VARIANCE                                BYU ANOV
*CRITICAL PATH ANALYSIS*OPERATIONS RESEARCH *PERT TIME                    LR  LEGO
*CRITICAL REGION*DECISION TABLES                                          BC  CRIT
*CRCSS-REFERENCED ASSEMBLY *ASSEMBLER*ASSEMBLY PROGRAM                    HA  4AP
*CROSS*REFERENCES OF VARIABLES, STATEMENT NUMBERS *FORTRAN IV             PX  FRRF
*CRYSTALLOGRAPHIC*PRECESSICN DATA*INTENSITY TREATMENT                     MI  XCIT
*CRYSTALLOGRAPHIC*STATISTICAL TEST TO DETERMINE*SYMMETRY-CENTER           MI  CAST
*CUBE-ROOT FCR SINGLE-PRECISION FLOATING NUMBERS IN FORTRAN 2             TY  QBRT
*CUBE-ROOT FOR*DOUBLE-PRECISION FLOATING NUMBERS IN FORTRAN 2             TY  DQRT
*CURVE DEFINED BY*PARAMETRIC EQUATION *CALCULATE*POINTS OF A              IB  CURV
*CURVE FIT OF*SECOND DEGREE*SIMULT.*DIFFERENTIAL EQU. *LEAST-SQ.          AM  HOVA
*CURVE FITTING ONE-DIMENSICNAL*RATIONAL*APPROXIMATION                     ISA RAPP
*CURVE FITTING *BIVARIATE*ORTHOGONAL PCLYNCMIAL*APPROXIMATION             ISA P3LO
*CURVE FITTING *GRAM POLYNOMIALS *LEAST SCUARES *POLYNOMIALS              RS  GRAM
*CURVE FITTING*MULTIPLE REGRESSICN *LINEAR*LEAST-SQUARES                  IN  7210
*CURVE *SMOOTH ROUTINE FOR 7090/7094 NINE POINT *PARABOLIC                MA  DSM3
*CURVE-FITTING*ORTHOGONAL POLYNOMIALS *LEAST SQUARES, WEIGHTED            RW  CF
*CURVE*FITTING WITH*CONSTRAINTS *SINGLE*PRECISICN*LEAST*SCUARES           ASB BJ06
*CURVE*FITTING*CHEBYSHEV*APPRCXIMATION*EXCHANGE METHOD                    NAX 011
*CUTTING STOCK PROGRAM *TWO-STAGE*TWO-DIMENSIONAL*TRIM CR                 PK  TR2D
*CYCLE*ANALYSIS*NATIONAL BUREAU METHOD BUSINESS                           UC  BCA
*CYLINDER ANALYSIS                                                        WH  CAN
*DAMPING*KELVIN MODEL*ELASTOMER*RUBBER CMPD *VISCCELASTIC*IMPACT          EL  DAMP
*DATA READING. *STANDARD PRCGRAM FOR                                      SN  PSLD
*DATA-CONVERSICN STATEMENT*ENCODE*DECC *FCRTRAN*MEMORY-TO-MEMORY          JP  FED
*DATAPLOTTER SUBR PACKAGE FCR CREATING*PLCT TAPES FOR THE*EAI             CO  PLTP
*DATE FOR PRINTING BY THE MAIN PRCGRAM PICKS UP THE                       MMU KA
*CATE FROM *IBNUC RUN                                                     ARA DATE
*DEBUGGING BY PRINTOUT OF*REGISTERS AND*TRACEBACK THRU*SUBPROGS           CUP FTRC
*CEBUGGING BY*SELECTIVE*MEMORY*DUMPING CURING EXECUTION *DYNAMIC          NS  SNPF
*DEBUGGING BY*SELECTIVE*MEMORY*DUMPING DURING EXECUTION *DYNAMIC          NS  SNPM
```

Fig. 1—Portion of a typical SHARE keyword index.

#### Priority Assignments

Many centers categorize computer runs into groups by language used, computer run times and complexity of equipment configurations. When a large group of runs satisfy a category requirement an "express service" will generally be established to give rapid turnaround and efficient operation. Similar runs are placed in batches for sequential processing; there may be four or five express runs daily and, if properly used, can be very helpful in meeting work schedules. Other runs will be interspersed with express service, although long runs, or those requiring extensive tape mounting and removal, will be held for night shift processing. The center will usually recognize reasonable justification and take appropriate steps to insure priority processing when needed; this is normally coordinated through the computer operations supervision.

#### Accounting

Every engineer is, as a rule, working with a budgeted amount of money; thus it is important that he be aware how much has been spent and what balance remains. When an account is opened, the current billing rates for equipment usage are given and an estimate of services to be provided made. Every job that is processed requires a job submittal form which is returned to the user. This form contains, in addition to operator comments, the time actually used to process. Charging practices vary between centers; some include job set-up as well as actual running time and may charge full rates for "peripheral" operations (card to tape transcription, printing of output). Most centers bill only for program running time and have proportionally lower rates for peripheral operations. Many centers prepare a computerized weekly report that presents a complete history of each task: who worked on the submissions, how long, what machines were used, how many minutes, what were the costs for each category, what was the total cost, etc. A similar report is processed monthly and used for customer billing purposes. These reports are all available to the engineer for review, question, or comment.

### Computer Center Equipments

#### Large Computers

The large scale computers have high-speed, large capacity data storage, many medium-speed (magnetic tapes and discs) units, and sophisticated software operating systems. This type of computer is extremely desirable for the solution of scientific and engineering problems. Large matrices, complex formulae, intricate data manipulations, and various simulation techniques are particularly adapted to the large computer.

#### Small Computers

Most small computers have limited high-speed memory capability but do have high-speed card readers/punches and printers. The small computers are extremely useful for performing utility functions, such as transcribing cards to tape, printing, duplicating magnetic tapes, and editing data, at lower cost.

#### Auxiliary Equipments

Auxiliary equipments include devices in the category of data preparation. Keypunch and paper-tape punches are a way of transferring data from a source document to a media acceptable to a computer-reading device. This media may be punched card, punched paper tape or coded magnetic tape.

Some examples of special purpose data preparation devices would include the Oscillograph Trace Reader (OSCAR) which generates punched cards from analog charts under operator control; the Digital Film Reader (BOSCAR) generates punched cards of film-coordinate data under operator control; and the Programmable Film Reader (PFR) which produces magnetic tape under automatic control of a programmed computer film scanner. Special purpose devices are a part of many centers, and include those capable of digitizing radar video data, speech data, etc.

There is a class of data handling equipment commonly referred to as EAM (Electronics Accounting Machines) designed to process punched cards. These devices, although significantly slower than computers, perform the task of sorting, merging, reproducing, or printing small volumes of data at a reasonable cost. They are especially useful in duplicating program decks, listing programs for modification and manipulating data on cards.

Many times the engineer will develop computed data in tabular form whereas it might be more meaningful to subsequent use in some pictorial form, such as a plot. Most large computer centers have digital plotters on-site which will accept magnetic tape directly, thus generating the final desired output. These plotters operate in conjunction with program subroutines to allow scale changes, axis rotation, three-dimensional effects and automatic pen selections.

Center supporting equipment will include decollators (for separating multiple-part printed output), bursters (for separating pages), microfilm viewers and other miscellaneous equipment.

### Software Selection

Software (the instructions to the computer) is generally the major expense item, as well as the pacing schedule factor in accomplishing engineering computational tasks. The computer center can provide a wide spectrum of support to minimize cost and schedule impact; to do so the following actions must be accomplished by the engineer and center personnel:

1) Adequately define computational requirements.
2) Survey available programs for one suitable in its original form or adaptable with modifications.
3) Determine if the engineer will prepare all or part of the required software. Select programming language, obtain necessary instruction and literature.
4) Define software requiring development by a professional programmer.
5) Determine software documentation required.

Definition of computational requirements is best done at the beginning level on a "question and answer" basis with center personnel. The three basic questions are:

> What is the input data format and characteristics?
> What are the logic and equations needed for computation?
> What is the required output format and method of presentation?

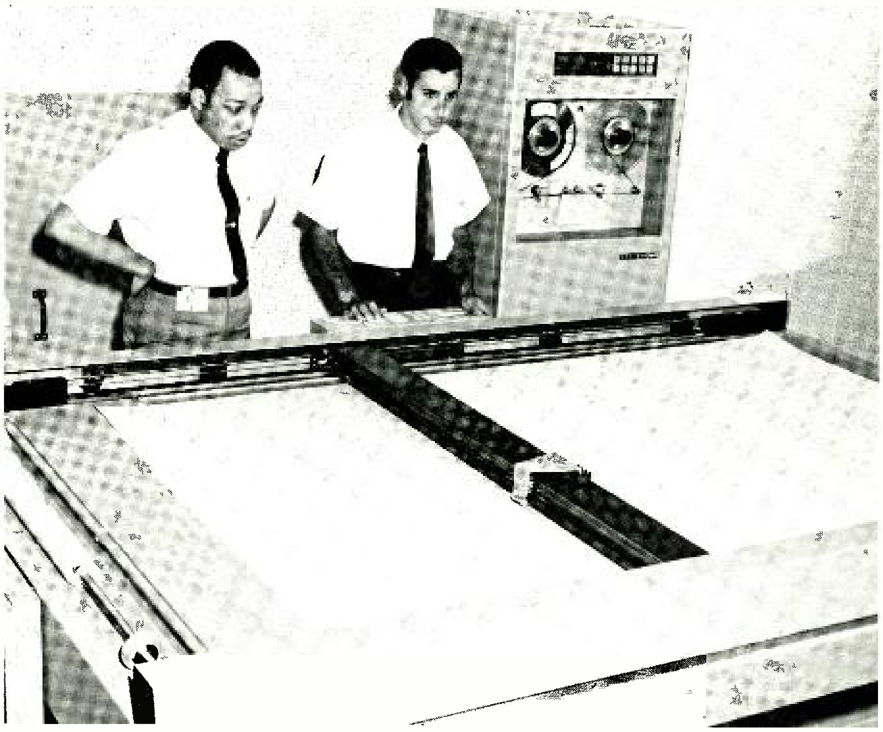
These must be answered precisely to allow proper software support. Following definition of the required software, the program library is checked for the programs or subroutines which could be utilized advantageously.

Many engineering problems can best be solved by the engineer himself, particularly if he possesses some skill in FORTRAN or similar language. It is relatively easy to break out well defined portions for a professional programmer. This is advantageous if machine-level language is necessary for involved logical data manipulation or if complex computer-oriented numerical analysis methods are needed. Present operating systems have features which allow relatively straight-forward interface between program segments, even if written in different languages.

Center personnel can arrange for programming instruction and supporting manuals. Should the application warrant the learning process, there are a number of "problem-oriented" application systems languages available which are sometimes simpler and more applicable to specific tasks. These include ECAP and SCEPTRE (for electronic circuit analysis), LOGSIM, GPSS and SIMSCRIPT (for systems simulation studies), APT (for numerically controlled machine tool tape preparation), ROCKET (for space trajectory studies), and others. Well written user manuals are usually available for them.

When professional programming is required, specific definition is necessary. In most Centers, a "specification" is prepared by the user, a "program plan" by the programmer and necessary reviews are held by the user and Center supervision to ensure satisfactory development and schedules. The depth of detail included is in proportion to task complexity; effort spent in complete definition to the programmer is directly relatable to the success of the overall effort.

The type and extent of software documentation required should be established early in the task. It may range from only the information necessary for the computer run submittal form to detailed operating procedures, data preparation instructions, flow charts, and details of the logic and mathematics used in the program. Center personnel will assist in defining and selecting documentation required. Documentation criteria are based on the anticipated life of the program, who will accomplish data preparation and run submission, the possibility of later modification and utility of the program for other users. Professional recognition and savings for others can be obtained by submittal of the finished program and documentation to the center program library or the many program exchange services.


The CAL-COMP plotter at the M&SR Division computational center.

The software support available from a scientific computer center is extensive and one may realize large cost and schedule savings through efficient use of the services which can be provided.

### Task Initiation

The engineer's first action in initiation of his task should be the seeking of advice from qualified individuals. Each computer facility has one or more "contact men" who are willing and capable of providing information on equipment, software, procedures and cost. Of great benefit to the engineer is counsel on how best to approach his problem. A key element in arriving at the ultimate approach is task definition; the format and characteristics of input data, the logic and equations of computations required, and the presentation of output. Having surveyed the task definition, the "contact man" will refer the engineer to specialists for information on available software, estimates of programming effort, data translation services, applicable manuals, etc. At this point, it is imperative that the engineer answer several questions:

1) Is this the proper facility to use for my job?
2) Has all currently available software of potential use been surveyed?
3) Is it advantageous to use professional programming assistance for all or part of my job?
4) Should I prepare and submit computer runs myself or use facility personnel for assistance?
5) What is the cost and schedule I can reasonably expect for accomplishing my task?

Definitive answers to the above questions should be obtained before proceeding. The task complexity obviously affects the time and effort required; it is all to easy for the beginner (as well as experienced) user to embark on a major task with inadequate definition. The invariable result is an expensive and time-consuming iteration to recover.

Following complete task definition and after the decision to proceed has been made, necessary funding and scope of work are authorized to the computer center and detailed arrangements are made (it is useful that the engineer obtain and study copies of the center's user procedures for job submittal, data retention, priority service, etc. before job submittal begins). Proper task initiation is the major factor in successful accomplishment; the benefit from expenditure of modest effort in this aspect cannot be overemphasized.

### Using the Facility

Following task definition and software selection, use of the physical facilities begins. Many centers assign a user code or number, both for data identification and charging purposes. The services of the center are then at the engineer's disposal. Card or tape punching involved in data or program deck preparation may be performed by the engineer or, at his option, submitted to the center's data-preparation staff. Translation of data (extraction of information from charts, worksheets, etc.) may also be accomplished by center personnel.

Submittal of computer runs is performed at the dispatch desk, where the user code is verified, control cards are checked, and the job logged. The job is then assigned to one of three general categories: *express, normal production,* or *night production.*

The job submittal form requires information necessary to establish the category (estimated running time, program input/output unit assignments, language, operating system, and any special operator instructions) as well as user code and "force time". This latter entry is to indicate the maximum running time, and is for the protection of the user. It is particularly valuable during debugging of a program when inadvertent "loops" may be entered which could cycle program steps indefinitely without normal job termination.

The engineer may obtain estimated completion time from the dispatcher when his job is submitted or request that he be called when his output is ready. Requests for priority treatment are normally directed to computer operations supervision and will be honored whenever possible without causing undue impact on other users. At job completion, input and output data are placed in the issuing area, generally alphabetically or by user code, and "save" tapes are placed in the tape library. The job submittal form is returned to the engineer with the running time and operator comments noted.

Problems are frequently found in the results, particularly in the beginning stages of a task. These should always be resolved, and extensive help is available to do so. The systems programmer is normally the starting point and he will determine whether a programming, operations, or data error occurred. He will give counsel to help resolve programming problems, and offer suggestions to improve efficiency. His explanation of system diagnostics, reading program dumps, and alternate debugging aids will be of value. Indicative of non-programming errors to be expected are those noted at M&SR's center on a routine data reduction task involving some 1500 runs over a six-month period (stated as percentage of total runs):

| | |
|---|---|
| Operations (operator, machine, software systems) | 7.1% |
| Job submittal (parameter card, erroneous instruction) | 6.5% |
| Input data (data identification, format, recording errors) | 9.8% |

Each of the above large-scale computer runs involved card-to-tape and multiple-output printing. Some 400 to 600 individual operations per day are typical in a large-scale computer center and problems are not unknown.

Time lost due to operations errors is normally not charged to the user and most centers employ a "problem report" to assist in resolving troubles. Center supervision should be contacted when unresolved problems occur; this is a vital part of the feedback process to improve service.

## Observations on Serving the Engineer User

A wide spectrum of use and success is noted in serving the engineer at scientific computer centers. The following observations are derived from mistakes commonly made by less successful users, as well as the characteristics displayed by those who consistently make efficient and profitable use of the services available:

1) The initial task definition is assuredly the most important step in the use of a computer. The full magnitude of the problem must be clearly defined at the start to avoid the wasting of time, effort, and money.
2) Training in a common problem-oriented language (such as FORTRAN) is important to an engineer, not only as a tool to develop his own programs but also to aid in communicating with professional programmers and in the use of available applications programs.
3) The program library should be thoroughly investigated before embarking on the long, tedious, and expensive task of developing new software.
4) The applicable scientific computer center services, hardware, software, and procedures should be thoroughly understood.
5) Some engineers are reluctant to ask for help; any unexplainable problem should be directed to the system programmer or center supervision.
6) Operating instructions should be written clearly; personal communications with operations personnel is generally not satisfactory in a high volume installation.
7) The engineer should apply the same standards of definition, planning, and understanding in his approach to a computational task as would be expected in a hardware design or development task.

## Future Developments

There are two relatively recent developments which are having a major impact on engineering users of scientific computer centers: availability of time-sharing terminals and the capability of third-generation computer hardware/software. Time-sharing terminals have resulted in readily available computational capabilities which require limited familiarity at the beginning level. The learning process is rapid because of the interactive mode of operation; also, exposure to problem-oriented languages and applications systems via time sharing terminals provides a broad familiarity on an economical basis. The result is that when the engineer's task outgrows the time-sharing terminal capability, he is prepared to smoothly transition to a scientific computer center.

Third generation hardware/software, as typified by the RCA Spectra systems, offers expanded storage, faster speeds, multi-programming, as well as a wide variety of languages and applications programs. A large number of specialized and easy to use application systems will evolve. The trend is toward more rapid turnaround times (a benefit primarily of the multiprogramming capability) and accordingly more economical services. There will be a merging of the strengths of time-sharing terminals (availability, fast turnaround, interactive operation) with those of the scientific computer center (large storage, fast input/output, expanded languages, and application systems) in the future as software systems allow program preparation through time-sharing terminals, with "background" processing done on the same large scale machines. The engineer-user of computational services will witness a steady increase in the capabilities available to him; his utilization of these capabilities is largely dependent upon his maintaining pace in understanding and organizing their use.

# Fortran programming is easy

**Dr. G. D. Gordon**

**The basic elements of Fortran IV can be learned in a few hours, and are described in this paper. With this knowledge, an engineer can write programs to generate almost any mathematical table or tabulate any analytical function. Arithmetic operations, a simple output statement, and the use of a special DO statement to generate tables are described. After learning these elements, an engineer can progress in easy stages, learning more Fortran statements and programming techniques as he writes additional Fortran programs.**

**Dr. Gary D. Gordon**
Administrator, Course Development,
CEE Program
Corporate Engineering Services
Research and Engineering
Camden, N.J.

received the BA from Wesleyan University in 1950, and the MA and PhD in Physics from Harvard University in 1951 and 1954 respectively. For two years, Dr. Gordon worked for the U. S. Army at Fort Detrick, Md., and spent another two years as a physicist at Operations Research, Inc. in Silver Spring, Md. Starting in 1958 as a senior engineer at the Astro-Electronics Division of RCA, Dr. Gordon worked five years in the analyses of problems relating to spacecraft. He contributed to the thermal design of the TIROS weather satellite and was responsible for the thermal design of the RELAY communication satellite; he has presented a series of lectures on Spacecraft Thermal Control to other RCA engineers. He has written technical papers on plasma acceleration by RF, the geodetic use of satellites, spacecraft thermal properties, and a proposed 30kw solar cell power plant. In his present staff position with Corporate Engineering Services, he participated in the program for the continuing education of RCA's engineering managers and leaders (CCSE), and is now active in the program for the continuing education of RCA engineers (CEE). In these programs he taught FORTRAN programming to 21 CCSE classes, or a total of 500 managers, and 3 CEE classes, or a total of 80 engineers. Dr. Gordon is a member of American Physical Society, American Institute of Aeronautics and Astronautics, American Association of Physics Teachers, Sigma Xi, and Phi Beta Kappa.

Reprint RE-14-3-4
Final manuscript received July 25, 1968.

PROGRAMS IN FORTRAN are normally written on a special FORTRAN coding form, using a simple language similar to ordinary English and mathematical expressions. A coding form and a sample program are shown in Fig. 1; the actual program will be explained later. Each statement is written on a separate line. For the present, most statements will be written to the right of the two vertical lines, that is, the first 6 spaces on each line are skipped. [Not necessary for RCA Basic Time Sharing System.] Each symbol goes into a separate space, including symbols such as commas and decimal points. Letters will always be capitalized. Blank spaces and blank lines are ignored by the computer but are often included for the convenience of the programmer.

## Arithmetic Statements

### Numbers

Numbers are written with a decimal point, and are known as floating-point constants. *Each number must have a decimal point.* Other symbols are not allowed in a number, such as commas or dollar signs. Extra spaces can be used for large numbers, such as writing 3 000 000. for three million. (Numbers without a decimal point are handled differently by a computer, and should not be used until the programmer learns the difference.)

### Addition and Subtraction

The symbol + represents addition and — represents subtraction. The addition of two numbers, such as 4.7 + 15.8 would be written:

SUM = 4.7 + 15.8

The result can be represented by any short word desired, written in capital letters to the left of the equal sign; these variables will be discussed later.

Additional examples:

TOTAL = 235. + 8400. — 2.5
DIF = 0.05728 — .0000721
Y = — 3.14 + 6.75 — 3. + 140.

### Multiplication and Division

The symbol * represents multiplication and / represents division. Thus, to multiply 9530 by 7 the statement would be written

PROD = 9530. * 7.

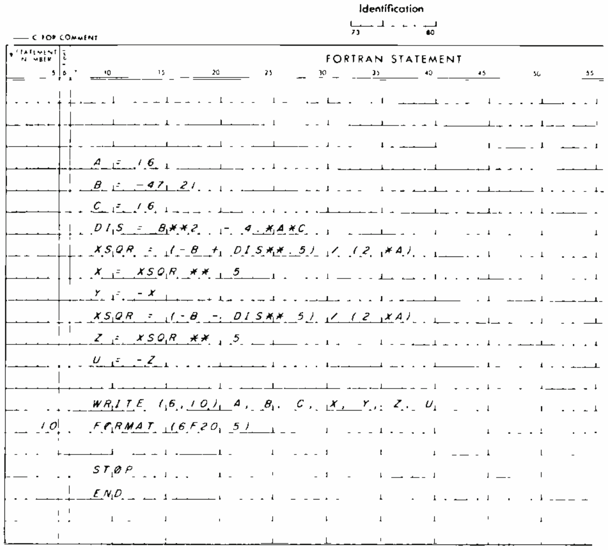
To divide 9530 by 7, the statement would be

QUOT = 9530. / 7.

Note that decimal points *must* be used.

As another example, a rectangle is 7.5 inches long and 4 inches wide. The area and the perimeter can be calculated by:

AREA = 7.5 * .4
PER = 2. * 7.5 + 2. * 4.



```
      A = 1.6
      B = -47.21
      C = 1.6
      DIS = B**2 - 4.*A*C
      XSQR = (-B + DIS**.5) / (2.*A)
      X = XSQR **.5
      Y = -X
      XSQR = (-B - DIS**.5) / (2.*A)
      Z = XSQR **.5
      U = -Z

      WRITE (6,10) A, B, C, X, Y, Z, U
   10 FØRMAT (6F20.5)

      STØP
      END
```

Fig. 1—A sample program written on a Fortran coding form.

Multiplication and division take precedence over addition and subtraction; thus the two multiplications above are performed first, and then 15 and 8 are added.

**Parentheses**

If there is any doubt, parentheses should be used to indicate the order of operations. For example:

```
RES = 120. * 2. / (7.5 * 3.)
```

If parentheses had not been used, multiplication and division are done in sequence and the computer would divide 240 by 7.5 and then multiply by 3. Written with the parentheses, the 3. is included in the denominator. Note that parentheses are not operators, and do not indicate multiplication. For example, the perimeter cannot be written as 2. (7.5 + 4.), but can be written

```
PER = 2. * (7.5 + 4.)
```

with the asterisk indicating the multiplication.

**Raising to a Power**

The symbol ** is used to represent exponentiation. There is no danger of confusing this with two multiplication operators, since two operators are never written side by side. As an example, the cube root of 0.37 would be written

```
CUBRT = .37** (1. / 3.)
```

The exponent can be a fraction or an expression. The diagonal of the rectangle mentioned earlier can also be calculated

```
DIAG = (4.**2 + 7.5**2) **.5
```

which is the square root of the sum of the squares.

*Special case: the decimal point on integer exponents should be omitted.* Only positive numbers can be raised to a fractional power; if the power is an integer, it should be written without a decimal point, and then it can operate on either positive or negative numbers. Exponentials take precedence over all other operations and are performed first, unless parentheses are used to indicate a different order.

**Variables**

A variable is a short word of one to six letters, used to denote any quantity. (Actually only the first character *must* be a letter; the others may be digits.) FORTRAN places no significance on the name, and it should be chosen to help the programmer remember the significance of the variable. Once chosen, the variable must be used in exactly the same form. *Special restriction: first letter cannot be* I, J, K, L, M, or N.

The format of each arithmetic statement is $a = b$, where $a$ is a variable, and $b$ is any expression. The variable is given the calculated numerical value, and this can be used in a subsequent statement. For example, after the area, AREA, and perimeter, PER, of a rectangle are calculated, the ratio of those two can be calculated by

```
R = AREA / PER
```

The numerical value of a variable must always be determined by the computer before it can be used in an expression. A polynomial, $2x^4 + 6x^2 - 8x + 7$, can be evaluated for a specific value of $x$ by:

```
X   = 5.794763
POL = 2.*X**4 + 6.*X**2.—8*X + 7.
```

**Equal Sign**

The case of the equal sign in FORTRAN programs illustrates one of the important differences between FORTRAN and ordinary mathematical statements. In FORTRAN, the precise meaning of the equal sign is: replace the value of the variable named on the left with the value of the expression on the right.

The statement X = 5.7 is an order to replace the value of the variable X with 5.7; the previous value of X is lost. Another example brings out forcefully the special meaning of the equal sign. A statement such as

```
X = X + 1.
```

has the meaning: replace the value of the variable X with its old value plus 1. This statement is clearly not an equation.

## FORTRAN Statements

**WRITE Statement**

To write the results of any computation, two statements are required. Thus, if values of PER, AREA, and R have previously been calculated, they can be written by:

```
   | |  WRITE (6, 10) PER, AREA, R
10 | |  FORMAT (6F20.5)
```

With these two statements, any number of variables can be written; they merely have to be listed after WRITE (6, 10). Note particularly the positions of decimal points and commas; *variables are separated by commas, but there is none before the first variable or after the last.* In the second statement, the 10 is written to the left of the two vertical lines, without a decimal point.

*Note:* For the RCA BTSS this must be modified, and can be written as

```
   | |  PRINT 10, PER, AREA, R
10 | |  FORMAT (6F12.5)
```

Output statements can be quite complicated, and are an important part of programming. At this point a complete understanding is not necessary, and only a brief description is given. The first statement is the actual order to write. The 6 is always 6, and refers to tape station 6, now standard for the usual numerical output. The 10 can be any number, and is a statement number (explained later) which couples the WRITE with the FORMAT statement (while these two are often written consecutively, this is not necessary). The FORMAT specifies the way in which the numbers will be written. In 6F20.5, the 6 indicates that up to six numbers will be written on each line. The F refers to the floating-point number notation, and prints the number with a fixed decimal point. The 20 specifies a total of 20 spaces allowed for the number, and the 5 determines that the number will be written with 5 decimal places.

**STOP and END Statements**

The STOP statement is used to halt computation. The END statement must be the last statement in every program. At present, each program will be terminated (on two separate lines) by the STOP and END statements.

## Example of Complete FORTRAN Program

With the information presented so far, complete FORTRAN programs can be written. Suppose the problem is to solve the following equation (all roots are real):

$$16x^4 - 47.21x^2 + 16 = 0$$

The procedure will be to use the quadratic equation to obtain the two values of $x^2$, and then take the square root. The complete program is shown in Fig. 1. Note that while identical formulas are used to compute X and Z, the results will be different because the value of XSOR has been changed by an intermediate statement.

There are many ways of writing a program to solve the same problem. Instead of calculating intermediate results, the first root, X, could be calculated directly by

```
X = ((47.21 + (47.21 ** 2 — 4. * 16. *
        16.) ** .5) / (2. * 16.)) ** .5
```

but often it is simpler to write shorter statements, and fewer mistakes result. When this program is run on the computer, the results will be written on one line. The last four numbers are:

```
1.60000   —1.60000   0.62500   —0.62500
```

In this case there would be no labeling of the numbers, and only by reading the program can one tell that the first number is X, the second Y, etc. Many other FORMAT statements are available, so that the results can be written in different ways.

## Exercises

1) A soup can has a height of four inches and a diameter of 2½ inches. Write a program to calculate and print: area of one end, area of side, total external surface area, and total volume.

2) A Nichrome wire is 15 meters long, with a 0.0254 cm diameter. Write a program to calculate the cross section, volume, weight, and resistance. The density of Nichrome is 8.2 g/cm$^3$, and the resistivity is 0.0001 ohm-cm.

3) An inductance of 2 H has an internal resistance of 500 ohms. Write a complete FORTRAN program to calculate the following quantities at a frequency of 400 Hz:

Angular frequency $= 2\pi f$
Reactance of the inductance $= 2\pi fL$
The $Q$ of the circuit $= 2\pi fL/R$
Total impedance $= \sqrt{R^2 + (2\pi fL)^2}$
Power factor $= R / \sqrt{R^2 + (2\pi fL)^2}$

4) An airplane in level flight drops a bomb from 10,000 feet. Write a program to calculate the time of fall ($t = 2s/g$), the average velocity ($s/t$), and the final velocity ($v^2 = 2gs$).

## Computation Tables

The objective of this section is to enable the student to write a program to compute a table of any analytic function, over any desired range and any interval. The DO statement is used to repeat a section of a program, and *library functions* are used to compute common mathematical functions.

### Repetition of Instructions

To compute a mathematical table, the computer must be programmed to:

1) Compute the function for one value of the variable;
2) Write the result;
3) Increase the variable to the next value; and
4) Repeat the process.

The flow chart is shown in Fig. 2.

As an example, suppose it is desired to compute the power radiated by a black surface at various temperatures.

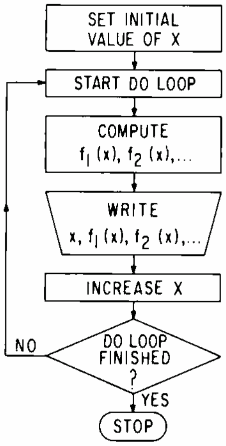


Fig. 2—Flow chart for the computation of tables.

That is, we wish to calculate $\sigma(T + 273.)^4$ for temperatures from —100 to +200°C at 5° increments. The first two statements would provide the initial value of $T$, and the value of $\sigma$.

```
T = —100.
S = 3.6572 * 10.**(—11)
```

These statements will never be repeated. The second step is the computation of the function, denoted by R. Any number of functions of T can be calculated in the same program. For example, since R is in watts/in$^2$, we could also have the power P in watts/cm$^2$.

```
R = S * (T + 273.) ** 4
P = R / (2.54**2)
```

The third step would be to have the computer write the results. Since this is going to be repeated, it is useful to print T as well as R and P:

```
      WRITE (6, 10) T, R, P
10    FORMAT (6F20.5)
```

The calculation and writing are now complete for the first value of T, of —100°C, and the value of T must now be increased to the next value of the table:

```
T = T + 5.
```

Note again the special meaning of the equal sign in FORTRAN: take the old value of T, calculate the expression on the right, and store this as a new value of T.

To calculate the desired table, it is only necessary to repeat the last three steps, that is, compute R and P for the new T, write them, increase T by 5., and so on. This repetition can be done with the DO statement, which is explained below.

### DO Statement

A most powerful tool of FORTRAN is the DO statement, which is used to repeat a section of the program any number of times. Two statements will be used in the process. The principal DO statement is used where the repetition begins:

DO *n* I = 1, *m*

and another statement is used where the repetition ends:

*n*     CONTINUE

In the statements above, *n* denotes a statement number, and *m* is the number of times the repetition is desired (written without a decimal point).

### Statement Number

A statement number is any desired number used to label a specific statement. The statement number is written to the left of the two vertical lines in the FORTRAN coding form, in columns 1 through 5. In the DO statement, *n* is a number that ties the start of the repetition to the end of it; any number can be chosen. Frequently, the first statement number chosen is 10, the next one 20, then 30, and so on. However, the numbers do not have to be in sequence. The only requirement is that there be one, and only one, statement labeled by a specific number.

All the numbers in the DO statement are written without decimal points. These are integers, usually called fixed-point constants, that are used in a number of special ways in FORTRAN programs. Note also the position of the comma in the DO statement, which must be written as shown. The I represents a fixed-point variable, which actually takes on different integral values as the program is repeated. (Other letters—such as J, K, L, M, or N—can be used, and should be used when more than one "DO loop" is in a program.)

For the particular example of the radiation table, 61 repetitions are desired, and the two statements would be written:

```
      DO 20 I = 1, 61
20    CONTINUE
```

The statements to be repeated are written between these two statements. The DO statement means: execute the statements down to the statement labeled 20; execute these statements 61 times: then proceed. Since the temperature, T, starts at —100. and increases 5 each time, the table will have 61 lines, and the last temperature printed is 200. After the last statement, the program can be continued with additional statements. This particular example would be terminated by the STOP and END statements. The complete program is shown in Fig. 3.

### Library Functions

In addition to the arithmetic statements (+, —, *, /, **) described previously, certain common mathematical functions can be used. These are listed in the table:

| *Mathematical Function* | FORTRAN *name* |
|---|---|
| Square root function | SQRT( ) |
| Exponential function | EXP( ) |
| Sine of an angle, in radians | SIN( ) |
| Cosine of an angle, in radians | COS( ) |
| Arctangent, principal value given in radians | ATAN( ) |
| Natural logarithum | ALOG( ) |
| Absolute value | ABS( ) |

To use a mathematical function, write the FORTRAN name of the function, followed by an expression enclosed in parentheses. For example, to compute the sine of an angle named x write:

```
Y = SIN(X)
```

In the above example the argument is the single variable x, but this is not necessary; the argument can be any expression. As another illustration, the square root of $b^2 - 4ac$ could be written:

```
DIS = SQRT (B**2 —4. * A * C)
```

This is an alternate way of raising to the ½ power, which was done in the last section by **.5, but SQRT ( ) is usually preferred.

It is permissible for the argument of a function to contain another function. This can be illustrated by the calculation of the following expression:

$$\log_e \left| \tan \frac{x}{2} \right|$$

The statement to compute this value could be written:

```
VAL = ALOG(ABS(SIN(X/2.)/COS(X/2.)))
```

With complicated mathematical functions it may be easier to start writing from the inside, such as SIN(X / 2.), and work out. An alternative way to simplify the many parentheses is to compute an intermediate value, such as the tangent, and use another statement to compute the final value.

### Exercises

In the following problems, pick an appropriate initial value and increment for the independent variable. The number of lines should be no more than 200, due to computer time limitations.

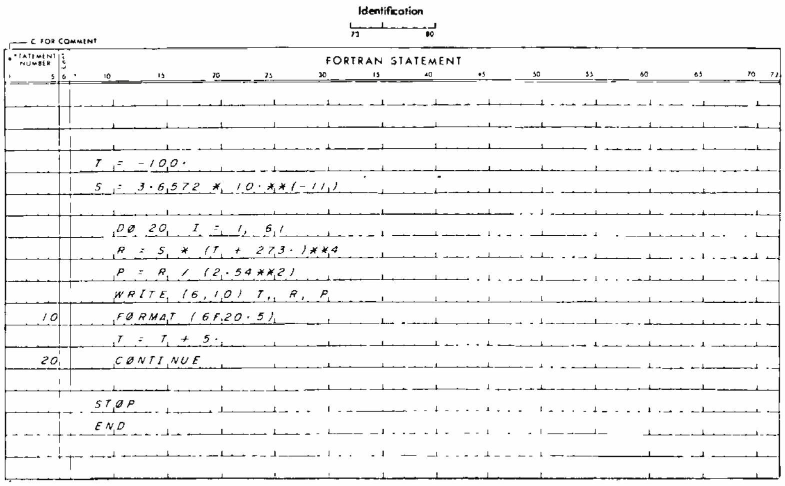
1) Write a program to compute and write a table of the square, the cube, and the fourth power of various numbers.
2) For various temperatures in °F, calculate the corresponding temperature in °C, °K, and °R, where

$$T_C = \frac{5}{9}(T_F - 32),\ T_K = T_C + 273.16,$$
$$T_R = T_F + 459.7$$

3) For a given angle, in degrees, compute and write the angle in radians, the sine, the cosine, and the tangent.
4) For copper wire of various diameters (in), a table is desired that provides the cross section (in$^2$), the weight of the wire (lb/ft), and the resistance (ohm/ft). The density of copper is 0.323 lb/in$^3$, and the resistance per length is equal to the resistivity (.68 x $10^{-6}$ ohm-inch) divided by the cross section.

The exercises above are for simple tables, similar to the program in Fig. 3. Many variations are possible, and some are suggested below as exercises.

1) For compactness, it may be desirable to print the second half of a table alongside the first half. One way to do this is with two separate computations; for the first part $x$ and $f(x)$ are computed, and for the second part $y$ and $f(y)$ are computed. A single WRITE statement then writes a line that includes both sides of the table.
2) It may be desirable not to have a constant increment for a table, but to have the independent variable increase by a fixed percentage. Write a table in which for each line the independent variable increases by some percentage.
3) In the previous variation, the numbers often are not round figures. A preferable method is to increase by a constant increment (DEL) for a while, then change the increment. After setting the initial values of both X and DEL, the basic principle of the method is sketched in Fig. 4. If X and DEL are initially set to 0.1, then ten iterations of the inner DO loop bring X up to 1.4. The value of the increment DEL is then increased to 1.0, and the next ten values will be 1, 2, 3, · · ·, 10 After those ten steps, the inner DO loop is completed once more, the value of DEL is increased to 10., and the next values are 10, 20. 30, · · ·. 100. The number of decades depends on the number of iterations of the outer DO loop. The two DO loops



```
      T = -100.
      S = 3.6572 * 10.**(-11)
         DO 20 I = 1, 61
         R = S * (T + 273.)**4
         P = R / (2.54**2)
         WRITE (6,10) T, R, P
   10    FORMAT (6F20.5)
         T = T + 5.
   20    CONTINUE
      STOP
      END
```

Fig. 3—Complete FORTRAN program.

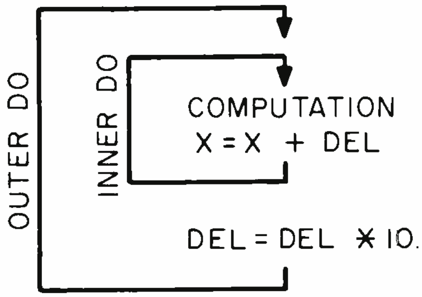


Fig. 4—Increment change by varying the DO loop.

must have different letters (I, J, K, L, M, N).

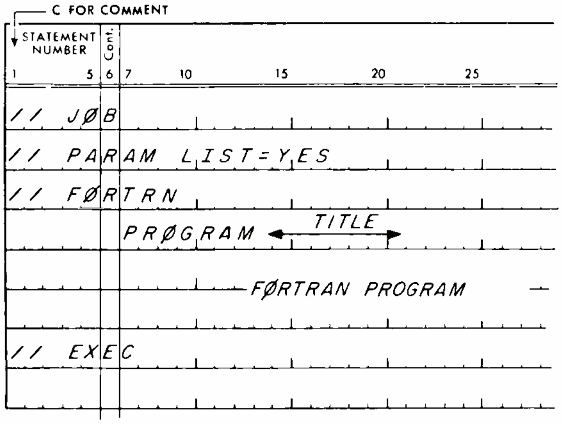
4) A function may depend on two variables. By setting one variable fixed, we can calculate a simple table by varying the second with a DO loop (inner loop). After this is done, we can change the first variable, and repeat the previous table by varying the second variable again, this is done through another DO loop (outer loop). First draw a flow chart, being careful where variables are initialized, and where they are incremented.

## Running the Program

### Control Commands

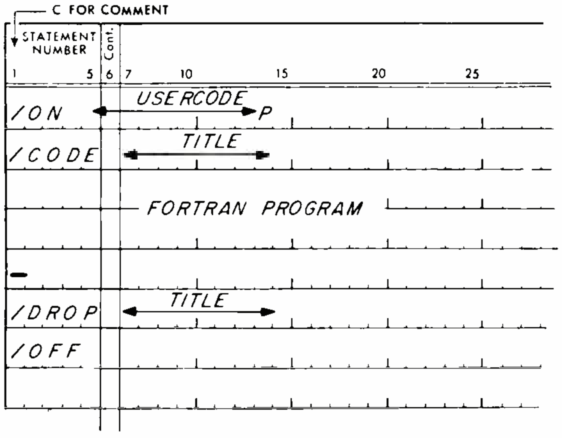
In most computer installations, a variety of computer programs can be run without operator intervention. These include different languages, such as FORTRAN and COBOL, and many options available to the professional programmer. *Control statements* are used to tell the computer how to handle each program. We are interested in running a small FORTRAN program—without complicated options. To do this, the user must find out the necessary control statements for the particular computer and location.

Figs. 5 and 6 show typical control statements which will run a small FORTRAN program. In each case, there is a space for the program. The position of each letter is critical, and extra blanks are not allowed in control statements; these are not FORTRAN statements. In most cases an additional blank space will mean that the computer will reject the entire program.



```
// JØB
// PARAM LIST=YES
// FØRTRN
      PRØGRAM ← TITLE →
      FØRTRAN PROGRAM —
// EXEC
```

Fig. 5—Spectra 70 control statements.



```
/ON   ← USERCODE → P
/CODE ← TITLE →
      FORTRAN PROGRAM
—
/DROP ← TITLE →
/OFF
```

Fig. 6—RCA BTSS control statements.

A possible set of control statements for a Spectra 70 computer is shown in Fig. 5. The name of the programmer and the title of the program can be included on the //JOB card, following a space. On the fourth card a 1-to-6-letter title of the program must be included, following the word PROGRAM.

For the RCA Basic Time Sharing System,[1] a minimum set of control commands is shown in Fig. 6. The first command, /ON, must include an eight-character user code for financial accounting purposes. With the second command /CODE a 1-to-8-letter title of the program, called a filename, must be included; the same is used in the /DROP command.

### Information Flow

After the program and control commands are written on the FORTRAN coding form, the rest of the process is routine. In batch processing, a keypunch operator transfers the information to punch cards, each vertical column of holes corresponding to a symbol, and each card corresponding to one line on the FORTRAN coding form. The cards are read on a card reader, and the information transferred to magnetic tape, which it used as the main input to the computer. The computer then makes the indicated calculations, and prints the desired results.

In time sharing systems, the programmer sits at a console and types the program on a keyboard. Often this is a standard teletype terminal connected by a telephone line to the computer. The results come back from the computer to the terminal and are typed on paper (or displayed on a video terminal). Punched paper tape can be used for intermediate storage of program or data.

## Conclusions

And that's all you need to know. If you study the above carefully, and do some of the exercises, you can write FORTRAN programs. Hundreds of students have proved that the majority of these programs will run on the computer. And once you have written and run a FORTRAN program, you will know that you have mastered the computer, and it can be made to do your bidding.

A word of caution is necessary. Absolute precision in conforming to the specified rules is essential for success. The presence of a single extraneous decimal point, comma, or parenthesis (or the omission thereof when required) will lead to an error and hence a program which will not run. While powerful debugging aids have been developed, a careful scrutiny is usually sufficient for a short program. Make sure that every number has a decimal point, that the first letter of variables is not I, J, K, L, M, or N, and that there is one, and only one, operator between any two quantities.

After you have mastered this lesson, you may want to go on. If you want to print many tables, you will want to learn about INPUT and OUTPUT statements, and corresponding FORMAT statements. For more powerful FORTRAN programs you will need to learn about IF statements and subscripted variable. For some programs, such as solving transcendental equations or differential equations, numerical analysis is needed. For the use of time sharing systems, you will want to learn many other commands that are available. But all this can be learned in small steps, and many programs can be written along the way.

## Further Reading

The best book on FORTRAN is *A Guide to FORTRAN IV Programming*, by D. D. McCracken (Wiley, 1965). This can be used both as a textbook for learning, and as a reference book. Other books by McCracken are also useful, but this should be the first one studied. After progressing beyond the first stages, a useful reference is *The Programmer's FORTRAN II and IV*, by C. P. Lecht (McGraw-Hill, 1966). FORTRAN statements are arranged in alphabetical order, so that questions on specific statements can be quickly settled.

## Reference

1. Spencer, J. M., "Introduction to the RCA Basic Time Sharing System, *RCA reprint* RE-14-3-11.

# What language!!

## M. Pietz
## G. Boose

**"This program bombed out three minutes into the run. It won't Link Edit on TDOS 8." "FILENO specifies which file on the tape the programmer wishes to be positioned at. It is only used if FILABL = STD and OPENPOS = RWD, in which case if FILENO = $\phi$,3* (FILENO-1), tape marks are skipped." These two quotations are honest-to-goodness *typical* statements from the neighborhood computer center. One was a remark by a programmer overheard in passing, and the other was picked at random from the glossary in a Spectra 70 reference manual. The specialization of language in the computer area presents a unique problem to the general engineering community in that engineers from all disciplines are becoming computer users. In this short article, we won't attempt to go so far as defining FILABL and OPENPOS, but we will attempt to give enough of the basic and more common terminology to allow you to hold your own in discussions with computer specialists.**

---

THE FIRST PROBLEM you run into in the computer world is trying to find the computer—and that's not as ridiculous as it sounds. You can read an entire Spectra 70 manual and never see the word *computer* mentioned. If you walk into a data processing center and ask someone to show you the computer, he can't really do it. The basis of the problem is that the term *computer* is very loosely used, and is usually meant to encompass an entire system of various devices. It soon becomes necessary to rid one's vocabulary of this term for the sake of becoming precise.

### Hardware Versus Software

All things associated with data processing divide into two broad categories—hardware and software. As in other types of systems, *hardware* refers to equipment and devices. *Software*, on the other hand, refers to the programs that are written to control the hardware. The insiders make a finer distinction between the software package and application programs, but we will get into that later.

### Equipment Components

#### Central Processor

In a typical configuration, the hardware consists of the central processor and input and output devices. The *central processor,* or CPU (Central Processing Unit), is the heart of the system, and is closest to being what might be called the computer. It is here that the logic circuitry that performs all the data manipulations is contained, along with memories that store programs and data awaiting manipulation.

The RCA Spectra 70/45-55 central processors consist of a program control unit, an arithmetic unit, an input/output interface, the main memory, a nonadressable main memory, and a scratch-pad memory. The *program control unit* interprets program instructions, moves data into and out of storage, determines which stored instruction should be exercised next, performs various logic functions, and activates the *arithmetic unit* for the execution of any instruction requiring an arithmetic operation. The *input/output interface* is the communication link between the central processor and all the external devices in the system.

#### Memories are Made of This

The *main memory* of a Spectra 70 is made up of planes of magnetic cores. It is the storage unit for both data to be processed and controlling instructions. It is by far the largest memory in the system. A Spectra 70/45G main memory can store about 2.4 million bits of information. The *nonaddressable main memory* is an addition to the main memory used to control the operation of the input/output devices.

---

**Merle G. Pietz, Ldr.**
Information Communications
Advanced Technology
Defense Electronic Products
Camden, N.J.
received BS in physics from the University of Florida in 1960. He has completed 60 quarter hours toward the MBA at Drexel Institute of Technology. He joined RCA with the Advanced Technology Department as a Publications Engineer in 1960, and was promoted to Leader of the Technical Publications group in 1963. In 1967, Mr. Pietz was appointed Technical Publications Administrator for Defense Engineering. In May of 1968, he assumed responsibility for the Advanced Technology Computation Center, in addition to the Information Communications group. The Advanced Technology Computation Center, which comprises a Spectra 70/45 computing system, specializes in the processing of Scientific and Engineering problems. The center serves the Camden engineering community. The Information Communications group specializes in the written communication of technical information, serving the engineering groups in Defense Engineering. Mr. Pietz is a member of Sigma Pi Sigma, the America Management Association, and the American Ordnance Association. He is on the DEP MIS committee and the DEP publications managers committee.

**Graham Boose**
Information Communications
Advanced Technology
Defense Electronic Products
Camden, N.J.
received the BA in mathematics in 1962 from Randolph-Macon Woman's College and the MS in Applied Statistics in 1966 from Villanova University. She joined Smith, Kline, and French Laboratories in 1962 as a statistical programmer in the areas of bioassay and quantitative pharmacology. As a technical writer for Computer Usage Company, she was responsible for the entire documentation effort and standards review for two major commercial computing systems. Mrs. Boose joined RCA in 1966 as a Publications Engineer in the Information Communications group of DEP Advanced Technology. The major portion of her work is in the areas of computer hardware and software. In addition to her publications duties, she is presently serving as liaison between the Advanced Technology Computation Center and its engineer-users.

---

Reprint RE-14-3-13
Final manuscript received August 19, 1968.

*Nonaddressable* in this case means that the contents of this memory cannot be altered by any instructions in the program being executed.

The *scratch-pad memory* is a micromagnetic device with very fast access time (300 nanoseconds). It can hold only 4,600 bits. It is used for special control functions, such as interrupt control, execution sequence control, and other program control.

#### Input/Output Devices

All of the many and various equipments used to get instructions and data into the central processor and processed data out in comprehensible form are nicely referred to as *input/output devices*, or more often as I/O devices. A typical array of such equipment is shown in Fig. 1.

Since the name given each device pretty well implies what its function is, we won't dwell on descriptions in this area. Random access deserves some discussion, however. In the recent past, data was almost always stored on magnetic tape, arriving there from punched cards via the card reader and the processor. If a particular piece of data was called for, the tape would have to be run past the read head until the data's storage location was encountered. You can see the time that was consumed when two subsequently required pieces of data were at opposite ends of the tape. The answer to this relatively long search time was solved with the advent of the disk pack. The disk packs used with the Spectra 70 systems consist of 6 platters, providing a total of 10 usable surfaces. Each surface has 203 tracks. A read/write head is provided for each platter surface. The head can be directed instantaneously to any track on that surface, and a piece of data can be picked out within one revolution of the disk. Since the disk pack revolves at 2,400 r/min, the speed gained by using disk storage is quite apparent. Because any piece of data can be picked up without searching sequentially through the data file, this technique is referred to as *random access*.

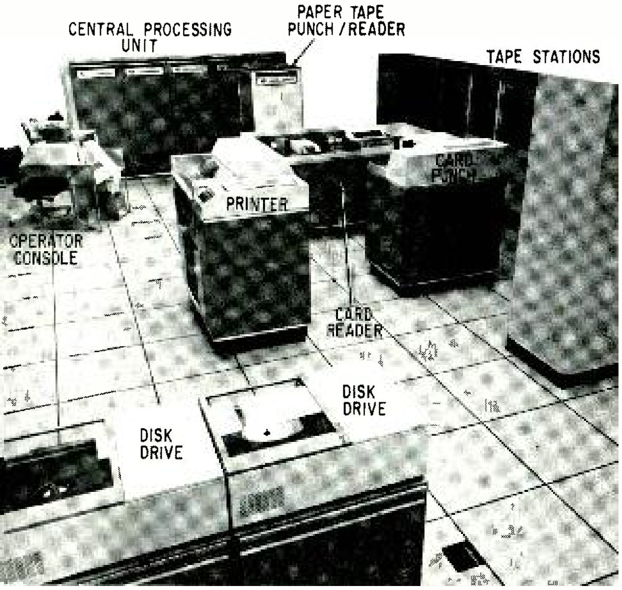


Fig. 1—Typical arrangement of equipment in a scientific computer center.

### Software and Programs

*Software* is a term that is often used in the broad sense to denote any and all kinds of computer programs. Data processing specialists make a finer distinction, however, in that they use *software* to refer only to those programs that are part of the standard operating systems. These programs are supplied by the equipment manufacturers. Programs developed by local programmers or engineers to compute specific problems are called *application programs*, rather than software programs.

#### The Software Package

The package of software programs provided by the equipment manufacturer includes *executive programs*, *subroutines*, *macro instructions*, the *assembler*, and *compilers*.

The *executive* is a remarkably busy program that resides in main memory on a rather permanent basis. It is the job of the executive to schedule the use of the CPU and the I/O devices in a way that permits several programs to be run simultaneously. The executive also monitors program execution and causes messages to be typed out on the operator console to inform the operator of detected errors, special equipment assignment requirements, and program status.

Many programs require the calculation of simple mathematical functions, such as trigonometric values, logarithms, and square roots. Sets of instructions for computing these frequently used functions are called *subroutines*. A library of subroutines that have already been programmed and tested is included in the software package. Several individual arithmetic operations or several subroutines can be called by the programmer with one *macro instruction*. As you can see, subroutines and macro instructions offer the programmer many short cuts in writing application programs.

Further aid to the programmer is provided by the *assembler*, or assembly language. Using assembly language, the programmer can write instructions with alphabetic or alphanumeric symbols. An additional aid is provided when using a *program library*. When a library contains many subroutines, it is likely that two or more subroutines have been written to occupy the same memory locations. These subroutines cannot be used together in the same program, since their storage requirements conflict. A programmer requiring two subroutines having conflicting storage requirements could recode one of them and use a nonconflicting set of memory addresses, but this is a laborious job and one in which mistakes are easily made. It is much simpler to write the subroutine instructions using arbitrarily chosen labels, rather than actual memory location designations. Here the *assembler* again offers considerable aid in assigning nonconflicting memory locations for the subroutines and converting the labels to the newly assigned locations.

A program written in assembly language is called the *source program*, while the assembler-translated program is called the *object program*. (The object program is said to be written in *machine language*.)

Other software programs exist for translating source programs into machine language. Many of them are "more powerful" than the assembler. They are called *compilers*. By "more powerful", we mean that one instruction in the source program is equivalent to many instructions in the object program. The more powerful languages associated with the compilers are called *problem oriented* languages because their instructions are written in terms familiar to practicioners dealing with certain types of problems. These languages are commonly known as *compiler* languages.

The compiler language most widely used for engineering applications is

FORTRAN. FORTRAN is an acronym of FORmula TRANslator, which signifies that this language is especially adapted to the solution of scientific formulas and equations. An example of a FORTRAN instruction is:

$$Y = A + B + C/D$$

This example shows how close the language can be to everyday engineering usage. Another language created for engineering is ALGOL (ALGOrithmic Language). The main language for business applications is COBOL (COmmon Business Oriented Language).

The latest trend is to develop languages that are even closer to "working" English, and that are broad enough in scope to cover both business and scientific applications. BASIC is a language recently written for use with time-sharing terminals. Instructions written in BASIC are so similar to ordinary engineering instructions that an engineer can become proficient in the use of BASIC in a few hours. PL1 (Programming Language, version 1) and BEEF (Business Enriched Fortran) are two of the new broader-scope languages. There are many other languages written for certain types of applications, such as information retrieval and natural language translations (Russian to English, for example).

The software packages supplied with the Spectra 70 systems are referred to as TOS and TDOS. The TOS (Tape Operating System) compiles and executes programs, the data instructions of which are all contained on magnetic tape. A faster operating system is TDOS (Tape-Disk Operating System). The TDOS makes use of both magnetic tapes and disk packs to store and retrieve data and instructions. The DOS is expected to be available in the fall. This will be a much faster system employing only disk packs. The Information Systems Division is continuously improving the TOS and TDOS systems. The versions are designated by a number immediately following the acronym letters. Operating systems currently being used are TOS14 and TDOS10.

#### Application Programs

Application programs are as numerous as the energy units of the programmers who create them. Like the compilers, they are irritatingly named by acronyms. Many of the more useful application programs are listed in the table on pages 48 and 49 of this issue.

### Data Storage and Program Execution

#### Bits, Bytes and Words

Since almost all engineers are familiar with the concept of *bits* as they are used for coding digital information, that concept will not be discussed here. The basic building block for data and instructions in the data processing languages is the *byte*. For Spectra 70 machines, a *byte* consists of 8 bits of information and one parity bit. The *parity bit* is used for accuracy control.

Two bytes grouped together consecutively are called a *halfword*, four consecutive bytes are called a *word*, and eight consecutive bytes are a *doubleword*. Memory access is in units of two bytes, or a halfword, for the Spectra 70/35 and 70/45, and in units of four bytes, or a word, for the Spectra 70/55.

#### Instructions

Programs are actually made up of sets of instructions. Instructions for the Spectra 70 processors may be in lengths of a halfword, a word, or six bytes. Each instruction consists of two parts: an *operation code* that defines the elementary operation to be performed (such as add, subtract, multiply), and a *memory address* that indicates the location of the word of data on which that operation is to be performed. Generally, an instruction does not specify the numbers to use in a calculation, but only their locations in the memory.

#### Link Editing

Very large programs are often written in sections, either to save time by having more than one programmer work on them or because the program is too large to fit into the memory in one section. Also, it is often desirable to combine subroutines that were assembled separately into one program. If the separate sections or subroutines are to be run as one program, there must be communication between the two. An ENTRY code is used to identify a symbol in one program that will be an entry point from a separately assembled program. An EXTRN code identifies a symbol in a program at which point operation is to be assumed by another program. The EXTRN and ENTRY operations are controlled by a software program called the *Linkage Editor.*

#### Multiprogramming

Many of today's computer systems can be run in a *multiprogramming* mode. *Multiprogramming* means that more than one program is run at the same time. Such operation is possible because of the large memory sizes of many modern processors. In the multiprogramming mode, the *executive* software program allocates available memory and input/output devices to the programs sharing the system. The executive also allocates the available computing time between programs according to some priority schedule.

#### Time-Sharing

Multiprogramming is being extended in many places to time-sharing. In time-sharing operations, the computer system users communicate directly with the system from remote terminals. The terminals may simply be an electric typewriter for output and a keyboard for manual input. Larger terminals may include card punches and readers, high speed printers, and CRT displays. One large computer system can handle as many as 100 remote terminals.

#### Bombing Out

*Bombing out* is the action that programmers hate to see happen. It means that their program didn't execute to completion. Programs may *bomb out* because the programmer made an error, because there is a *bug* in the software, or because a piece of hardware is malfunctioning.

### Concluding Remarks

So now you know what the man meant when he said "This program bombed out three minutes into the run (not a description of a military mission). It won't Link Edit on TDOS 8." Start working with the data processing people and systems, and someday you'll know what FILABL and OPENPOS means. When you find out, tell us.

# Introduction to the RCA basic time-sharing system

**J. M. Spencer**

**This article explains the fundamentals of using RCA Basic Time Sharing System II (BTSS II). Since this article is intended to be an introduction for engineers who have not used time sharing, some of the more powerful features of this system, useful to the experienced user, are not described.**

TIME SHARING provides the engineer with a computational tool that will reduce the time required to complete some of the larger calculations presently being done with slide rule and desk calculator. In addition, he will be able to solve larger problems which require the computational methods and speed available only in a digital computer. Mr. Norman Freedman* states: "We look at BTSS as a necessary tool which today's engineer must efficiently utilize if he expects to maintain his value as a professional engineer. If he is to keep up with the pace of current engineering developments, the engineer must know when and how to use the computer for modeling and solving many of his daily problems."

The problem-solving language in BTSS-II is FORTRAN; readers not familiar with FORTRAN will find Dr. Gary Gordon's introductory article in this issue very helpful. Since the intent of this article is to illustrate the simple procedures used to input, debug, and execute programs, very brief FORTRAN programs are used as examples. The computation in these examples converts temperatures from Fahrenheit to centigrade. All of the examples in this article were run using a Model-35 Teletype terminal. In each of the Figures, the input typing done by the user is shaded. The time sharing system typed out the rest of the typing in the figures. [*Editor's Note:* A photograph and keyboard layout of the Model 35 teletype unit is shown in the paper by F. Brill in this issue.]

The first example, in Fig. 1, is a simple, brief terminal session. Before the first line was typed to the computer the following steps were performed at the teletype:

1) Press ORIG button to originate a call, turn up the speaker volume control, listen for dial tone;
2) Dial the computer telephone number, get high-pitch answer tone and BEGIN XX message;
3) Press K button to select the keyboard.

At line 1 in Fig. 1, the user signed on the time sharing system using the command: /ON EPAOOABCJMS®

Any line typed to the computer which has a / (solidus) as its first character is a system command. There are twenty-four system commands—many of them not described in this article. Every line the user types to the computer is followed by the ® (carriage return) character. [A line may be terminated with the escape character instead of the carriage return character if the user wishes the line to be ignored.] The /ON command identifies the user to the computer system by giving the system a valid, preestablished usercode (in the example EPAOOABCJMS). This command must be successfully completed before any other transactions will be allowed. If the user makes a typing error, he must input this command again correctly. When the usercode is accepted, the system will respond with the date and time of day and the message READY.

## Inputting and Executing a FORTRAN Program

At line 2 in Fig. 1, the user began inputting and compiling a program using the command: /CODE TEMPCONV®

The system command /CODE followed by a space and name-of-file selects the part of the system which compiles and executes FORTRAN programs. In this example the file name TEMPCONV is a new file name in this usercode, so the system asks for records to be put into this file. It does this by typing out to the terminal the number of the record to be input into this file: 10. As each record is typed in, it is compiled and checked for errors. This process of inputting and compiling a line at a time continues until the user gives a command to do something else.

To start execution of a FORTRAN program, type in a - ® (hyphen, then carriage return). At line 3, the system typed out the number 40 requesting input for the fourth line of this program. The user desired to begin execution of the three line program so - ® was typed. The single character - (hyphen) is the command which begins execution of a FORTRAN program at the first executable statement.

When the read statement in line 10 of the FORTRAN program is executed, the system types out: FTEMP=. This tells the user that the FORTRAN program wants to read data for the variable FTEMP. At line 4, the user typed 32 ®. This data was used by the FORTRAN program, and the result computed and printed out by the print statement in line 30 of the program. The number, .000000E+00 is the answer for the input

**J. Murray Spencer**
Time Sharing Project
Information Systems
Cherry Hill, N.J.

received the BS in Applied Mathematics, Computer Sciences Option, from Clemson University in 1965. There he was a member of Phi Kappa Phi. He has taken graduate courses at Stevens Institute of Technology in Mathematics and programming. He has worked with small and large digital computers programming applications and as a systems programmer since 1962 at Clemson University, Bell Telephone Laboratories, and, since 1967, at RCA. He is now an EDP Product Planner, and is responsible for the technical aspect of running the RCA Corporate Time Sharing System.

---

*Mr. Freedman is Administrator, Special Engineering Projects, Microwave Devices Operations Department, Industrial Tube Division, Electronic Components, Harrison, N.J.

Reprint RE-14-3-11
Final manuscript received August 22, 1968.

32. The next line shows that the program finished its execution after line 30.

The format ±.XXXXXXE±YY is called floating-point format, or scientific notation. If the number is positive, the first + is usually omitted. The fraction ±XXXXXX is multiplied by 10 raised to the ±YY power. Thus, .579999E+01 = 5.79999. Digital computers which operate with binary arithmetic internally often cannot represent a decimal number exactly. (If the input data for a program which output .579999E+01 were only precise to 4 significant figures, then this representation is quite accurate; it should be considered 5.800). Floating point notation is quite useful for representing with very large or very small numbers: e.g. —.712600E-11 is shorter than its equivalent —.000000000007126, and numbers of greatly differing magnitude can be expressed in the same number of characters.

At lines 5, each of the four pairs of input - ® and a data number ® causes the program to execute again. The - ® starts execution of the program and is required because the program stopped (typed out STOP AFTER * 30) after each execution. If the program had been written with a GO TO statement which caused it to recycle, only the initial - ® at the line noted 3 would have been necessary.

## Disconnecting from the Computer (signing off)

To delete a program file from the usercode type in /DROP TEMPCONV ®. Before this /DROP command was executed at line 6, the three-line FORTRAN program existed in memory and could have been called again without being typed in. This program was dropped before the user signed off so that a file in the usercode would not be used just to store this three-line program. If the /DROP command were not executed, this file would have been available the next time he signed on. The only way to delete a file is by executing the /DROP command. It is not possible to unintentionally delete a file.

To sign off of the time-sharing system, type in /OFF ®. When the user has completed his work for a particular time-sharing session, he notifies the system that he is through by typing in the command /OFF ®. The system closes his usercode, performs an accounting function, and disconnects his terminal. The line which follows the /OFF command tells the user how long he used the time sharing system and approximately how much the session cost him. In this example, the approximate cost SPENT=$001 is high because this cost is rounded off to the nearest dollar and is never rounded to zero. The actual cost of this session (using the rates of the RCA Corporate Time Sharing System) is computed as follows: The amount of time the terminal was signed on the system in hours, printed out as CONSOLE=0.04, is multiplied by the rate of $7 per hour. Thus, the actual cost of this session (not considering telephone cost) was $.28.

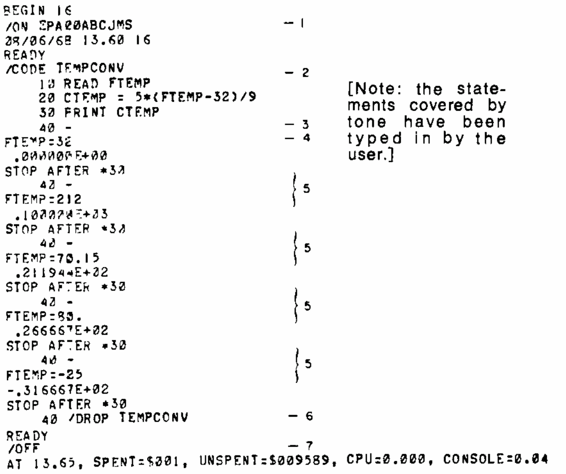
## Compile, Execute, File, and Record Defined

To *compile* a FORTRAN program means to translate the user's FORTRAN statements into code which the computer can execute. No calculations of input data for the program are done during compilation. *Execution* of a program means that the computer is operating as directed by the code generated during the compilation. It is during execution that the program generates answers by computing results based on the input data. A program need be compiled only once, it can be executed many times from a single compilation.

Each usercode has allotted to it a fixed number of *files*, either 8, 16, 24 or 32 files. These files are used to store programs and data in the computer. The contents of files are stored as *records*, each record containing 1 to 255 characters. The number of records that a file can contain varies inversely as the size of the record. For example, a file can contain 393 records of 30 characters each. Input to a file is accomplished by using one of several different commands, one of which is /CODE. Records can contain more than one line but often contain only one line.

## Correction of Typing Errors

The discussion of Fig. 1 and the commands and statements typed in by the user made no reference to errors being made by the user. However, users are likely to make minor typographical errors as they input commands and statements to the system. The user has available to him several correction

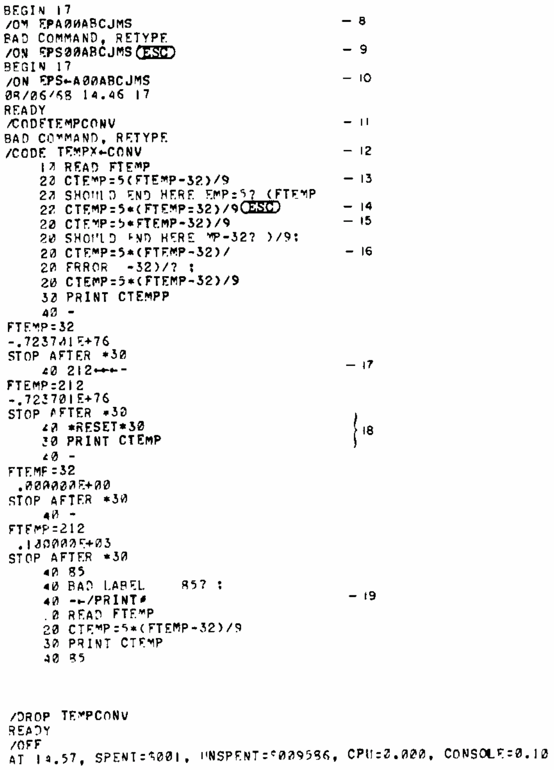


```
BEGIN 16
/ON EPA00ABCJMS                         - 1
08/06/68 13.60 16
READY
/CODE TEMPCONV                          - 2
    10 READ FTEMP
    20 CTEMP = 5*(FTEMP-32)/9
    30 PRINT CTEMP
    40 -                                - 3
FTEMP=32                                - 4
 .000000E+00
STOP AFTER *30
    40 -                                } 5
FTEMP=212
 .100000E+03
STOP AFTER *30
    40 -                                } 5
FTEMP=70.15
 .211944E+02
STOP AFTER *30
    40 -                                } 5
FTEMP=80.
 .266667E+02
STOP AFTER *30
    40 -                                } 5
FTEMP=-25
-.316667E+02
STOP AFTER *30
    40 /DROP TEMPCONV                   - 6
READY
/OFF                                    - 7
AT 13.65, SPENT=$001, UNSPENT=$009589, CPU=0.000, CONSOLE=0.04
```

[Note: the statements covered by tone have been typed in by the user.]

Fig. 1—Time-sharing session to input, compile, and execute a short program.

techniques which make using the time-sharing system less than a nerve racking affair of having to type perfectly. The session in Fig. 2 accomplishes almost exactly the same work as the session in Fig. 1. However, the user made several typing errors. The correction of these errors is, in part, dependent on when the user discovers that he has made an error. If he discovers his error before pressing the carriage return at the end of his input record, he can make the correction with less effort because there will be no action taken on his input. If he does not discover his error until after press-



```
BEGIN 17
/OM EPA00ABCJMS                          - 8
BAD COMMAND, RETYPE
/ON EPS00ABCJMS(ESC)                     - 9
BEGIN 17
/ON EPS←A00ABCJMS                        - 10
08/06/68 14.46 17
READY
/CODETEMPCONV                            - 11
BAD COMMAND, RETYPE
/CODE TEMPX←CONV                         - 12
     10 READ FTEMP
     20 CTEMP=5(FTEMP-32)/9              - 13
     20 SHOULD END HERE EMP:5? (FTEMP
     20 CTEMP=5*(FTEMP=32)/9(ESC)        - 14
     20 CTEMP=5*FTEMP-32)/9              - 15
     20 SHOULD END HERE MP-32? )/9:
     20 CTEMP=5*(FTEMP-32)/              - 16
     20 ERROR  -32)/? :
     20 CTEMP=5*(FTEMP-32)/9
     30 PRINT CTEMPP
     40 -
FTEMP=32
-.723741E+76
STOP AFTER *30
     40 212←←←-                          - 17
FTEMP=212
-.723701E+76
STOP AFTER *30
     40 *RESET*30                        } 18
     30 PRINT CTEMP
     40 -
FTEMP=32
 .000000E+00
STOP AFTER *30
     40 -
FTEMP=212
 .100000E+03
STOP AFTER *30
     40 85
     40 BAD LABEL     85? :
     40 -←/PRINT#                        - 19
     .0 READ FTEMP
     20 CTEMP=5*(FTEMP-32)/9
     30 PRINT CTEMP
     40 85


/DROP TEMPCONV
READY
/OFF
AT 14.57, SPENT=$001, UNSPENT=$009586, CPU=0.000, CONSOLE=0.10
```

Fig. 2—Time-sharing session illustrating correction of typing errors.

ing the carriage return key, the system may have taken action which he did not desire, or as is often the case, the input was not valid for one of several different reasons and the system informed him that his input was not valid. In Fig. 2, the following correction techniques are illustrated:

1) Correcting typing mistakes with the ← (backspace character). If the user discovers that he has typed a character which he did not mean to type, he can type a special character which will cause the system to ignore the previous character. This character is referred to as the backspace character, and is represented in the figures with the ←. More than one of these ← can be used at the same point, e.g. if three of them are typed, the system will ignore the three previous characters.
2) An entire line of input can be discarded by typing the escape character: ESC (no carriage return ® is used after ESC). The action of this character is similar to that of the carriage return in that the system will type out a line feed and carriage return and may type out one of several different messages to indicate that the line was ignored. The different messages occur when the escape character is received when the system is expecting various formats of input. These messages will be explained as they occur.
3) Correcting a line that was incorrectly typed by retyping the line. Many times when a line is incorrectly typed, the system will reject the entire line because the typing error causes that line to not fit the required input format, and the system simply asks for the line to be typed again. As with the escape character, the system has different ways of complaining about input records which are in an invalid format, depending upon which input format it was expecting. For example, an input typing error when typing statements into the FORTRAN compiler may cause the statement to not compile and produce a compilation diagnostic. If a / command is incorrectly typed to the system, only the comment BAD COMMAND, RETYPE will appear.

The following paragraphs explain the use of these techniques as illustrated in Fig. 2. For the purpose of illustrating different possibilities, Fig. 2 contains a higher than normal (even for beginners) number of errors.

To correct a misspelled / command, simply retype the command. At line 8, the /ON command was misspelled as /OM. The system did not recognize this as a valid command even though the usercode was correct. The type out BAD COMMAND, RETYPE informs the user that he has to try again.

To discard an entire line of input, type in an escape: ESC. In line 9, the user typed this line of input: /ON EPSOOABCJMS, but before typing the carriage return, he noticed the misspelled usercode. Instead of backspacing nine characters, he chose to retype the entire line, so he pressed the escape key. The system informed him that it had discarded this input by retyping the BEGIN 17.

To correct a mistyped character, delete it with the backspace character. At line 10 while typing the /ON command, the user realized that he typed an S instead of an A, so the ← was used to delete the S; he typed the rest of the usercode, and followed it by the carriage return. The /ON command was processed normally.

At line 11, the /CODE command was typed without a required space. The comment BAD COMMAND, RETYPE informs the user he must type the command again. At line 12, the user used the backspace character to delete a single mistyped character. At lines 13, 15, and 16, the user made typing errors which changed the input statements to the compiler to ones of invalid syntax. The compiler gave a compilation diagnostic and requested the line to be input again. The compiler indicated that it wished the same record to be input again by typing out the same record number. When the new statement was typed in, the statement formerly in this record was discarded. At line 14, the user noticed that he had a typing error before he pressed the carriage return; he pressed the escape key to discard the entire input. The compiler requested that the record be input again by typing out the same record number, 20.

At line 17, the user noticed that he was typing in the wrong input. The number 212 he intended to be data for the FORTRAN program, but the input which was requested was a statement or command to the compiler. He remembered that he needed to give the - command to begin execution. Three ← were used to delete the 212 and - ® was typed.

To change a statement in a FORTRAN program, use the *RESET*XX command to direct the compiler to request a new statement for record XX. The *RESET*XX command can only be used when the compiler is requesting input for some other record, that is, the compiler has typed out a record number for which it desires input. Note that the *RESET*XX command cannot be used when a FORTRAN program is in execution and the system is asking for data input for a variable. If the system is asking for data, the command /CODE will cause it to request additional statements or commands by typing out the next available record number.

At line 18, the user noticed that the answers typed out by the FORTRAN program seemed incorrect. The user studied the statements in the program and determined that a typing error in line 30 was causing the incorrect answers. The variable CTEMPP in line 30 should have been spelled CTEMP. At lines 18, the compiler typed out 40. It was requesting either a statement or a command. Instead of typing - ® as before, the user typed in the command *RESET*30 ®. This command caused the compiler to request a new record to be input to replace the record number 30 currently in the FORTRAN program. On the following line, 30 was typed out indicating that input was desired for record 30. The user typed in the corrected FORTRAN statement PRINT CTEMP®. The compiler requested input for line 40 and the command -® caused execution of the program to begin. The user typed in the number 32 again and this time received the answer he expected, 000000E+00.

The system command /PRINT# can be used to print the contents of a file. At line 19, the backspace character was used to delete the character when the user changed his mind about what he wanted to do. He typed the system command /PRINT#®. This command causes the system to list the entire contents of a file. No file name was used in this command because it was desired to list the file currently being used. The listing shows that there is a record 40 in this program file containing only 85. This record is not a valid FORTRAN statement. It was input into the file two lines above the /PRINT command. The compiler typed out 40 indicating that it wanted an input of either a FORTRAN statement or a compiler command such as -®, or *RESET*30®. The number 85® was typed in and caused a compilation diagnostic to be printed out on the next line because it was neither a valid FORTRAN statement nor was it a compiler command.

The user decided to end this terminal session at this point. The program file as deleted from the usercode by the command /DROP TEMPCONV®. Note that it is necessary to express the name of the file to be deleted when using the /DROP command. The user signed off with the /OFF® command.

## FLOW and TRACE Program Debug Commands

The time sharing session in Fig. 3 illustrates the interactive debugging aids FLOW and TRACE. This session also shows a simple use of the BTSS editor. At line 20 in Fig. 3, the user typed the command /PRINT# TEMPCONV® to list the contents of a file which was already in the usercode from a previous session. In lines 21 the user typed the command /CODE® to cause the program to be compiled. The - ® command caused the program to begin execution and the following lines show data being supplied the program for the variable FTEMP. At line 22, the user observed that the answer for N printed out above was incorrect. The user typed the command /EDIT ® to call the BTSS Editor. At lines 23, a command was typed into the editor causing compiler commands to be added to the program file. At line 24, the user recompiled the program file and started execution of the program.

The compiler command TRACE causes a line to the output as the program executes each FORTRAN assignment statement. This line indicates which FORTRAN statement is being executed by the AT *xx. The remainder of the line shows the variable and its new value. The FLOW compiler command causes a line to be output when a FORTRAN statement transfers execution to other than the next statement. For example, *50 — > *130 indicates that at line 50 the program transfers to line 130. These two debugging aids give the user an exact description of the execution of his program. They can be turned on or off independently and anywhere in the program. The printouts from these commands during the execution of the program began at line 24 in Fig. 3 and helped the user to discover an error in his program which he corrected beginning at line 25.

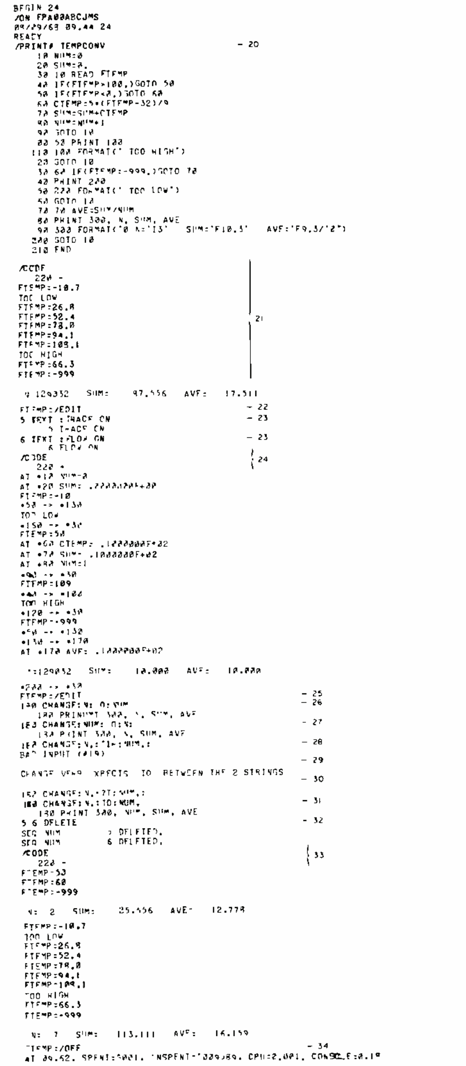
## A Simple Use of the BTSS Editor

At line 25, the /EDIT ® system command called the BTSS editor. At line 26, the user typed in an Editor command to change a variable in record 180 of his FORTRAN program file. The editor typed out a line showing the change that it made, and the user discovered that while the syntax of the editor command was correct, the change was not made as he anticipated. Instead of changing the variable N to the variable NUM, he changed the word PRINT to PRINUMT. At line 27, the user typed in a CHANGE editor command to correct the spelling of the word PRINT. At line 28, the user typed commands to effect the change intended by the first CHANGE command. But, the user misspelled the editor command at line 28, and the command was not in correct format. The editor typed out a line BAD INPUT (#19). The user desired the editor to explain why the previous command was a bad input. At line 29, he typed a single carriage return preceded by no other characters. This is a command to the editor to print a line of explanation. This explanation did not satisfy the user, so at line 30, he again typed only a carriage return. This is a command to the editor to print a line showing where the error in the command was located. The position of the error is indicated by a ? (question mark) which the Editor inserted in the incorrect command. At line 31, the user typed into the editor the correct command. Note that the commas in this editor command isolate the particular occurrence of the letter N in the FORTRAN statement to be changed.

At line 32, an editor command was typed in to delete the two records in the FORTRAN program file which contained the FORTRAN compiler commands TRACE ON and FLOW ON because the user believed that he no longer needed the debugging output.

At line 33, the user recompiled and restarted the program, and subsequently received a correct value for the variable N. After rerunning the program with the original set of data and getting the correct results for that data, the user signed off of the system.

Note that since he did not execute a /DROP command, the program file TEMPCONV will be in the usercode the next time that he signs on. Note also that the system commands (identified by a / as the first character) are privileged in that they can be typed in at any time when any part of the system is requesting input. In Fig. 3, a / command was used in lines 22, 25, and 34 when a FORTRAN program was requesting data input. Also, at lines 24 and 33 a / command was input when the Editor was expecting another command.



```
BEGIN 24
/ON FPA00ABCJMS
08/29/68 09.44 24
READY
/PRINT# TEMPCONV                                  - 20
    10 NUM=0
    20 SUM=0.
    30 10 READ FTEMP
    40 IF(FTEMP>100.)GOTO 50
    50 IF(FTEMP<0.)GOTO 60
    60 CTEMP=5*(FTEMP-32)/9
    70 SUM=SUM+CTEMP
    80 NUM=NUM+1
    90 GOTO 10
   100 50 PRINT 100
   110 100 FORMAT(' TOO HIGH')
   120 GOTO 10
   130 60 IF(FTEMP=-999.)GOTO 70
   140 PRINT 200
   150 200 FORMAT(' TOO LOW')
   160 GOTO 10
   170 70 AVE=SUM/NUM
   180 PRINT 300, N, SUM, AVE
   190 300 FORMAT('0 N='I3'   SUM='F10.3'   AVE='F9.3/'2')
   200 GOTO 10
   210 END

/CODE                                            |
   220 -                                         |
FTEMP:-10.7                                      |
TOO LOW                                          |
FTEMP:26.8                                       |
FTEMP:52.4                                       | 21
FTEMP:78.0                                       |
FTEMP:94.1                                       |
FTEMP:109.1                                      |
TOO HIGH                                         |
FTEMP:66.3                                       |
FTEMP:-999                                       |

 N: 120352   SUM:    97.556   AVE:    17.511

FTEMP:/EDIT                                      - 22
5 TEXT :TRACE ON                                 - 23
     5 TRACE ON
6 TEXT :FLOW ON                                  - 23
     6 FLOW ON
/CODE                                            { 24
   220 -
AT *10 NUM=0
AT *20 SUM= .0000000E+00
FTEMP:-10
*50 -> *130
TOO LOW
*150 -> *30
FTEMP:50
AT *60 CTEMP= .1000000E+02
AT *70 SUM= .1000000E+02
AT *80 NUM=1
*90 -> *30
FTEMP:109
*40 -> *100
TOO HIGH
*120 -> *30
FTEMP:-999
*50 -> *130
*130 -> *170
AT *170 AVE= .1000000E+02

 N:129052   SUM:    10.000   AVE:    10.000

*200 -> *30
FTEMP:/EDIT                                            - 25
180 CHANGE:N: 0:NUM                                    - 26
   180 PRINUMT 300, N, SUM, AVE
180 CHANGE:NUM: 0:N:                                   - 27
   180 PRINT 300, N, SUM, AVE
180 CHANGE:N,:"1":NUM,:                                - 28
BAD INPUT (#19)
                                                       - 29
CHANGE VERB EXPECTS TO BETWEEN THE 2 STRINGS
                                                       - 30
180 CHANGE:N,:?T:NUM,:
180 CHANGE:N,:TO:NUM,                                  - 31
   180 PRINT 300, NUM, SUM, AVE
5 6 DELETE                                             - 32
SEQ NUM        5 DELETED.
SEQ NUM        6 DELETED.
/CODE                                                  { 33
   220 -
FTEMP:50
FTEMP:60
FTEMP:-999

 N:  2   SUM:    25.556   AVE:    12.778

FTEMP:-10.7
TOO LOW
FTEMP:26.8
FTEMP:52.4
FTEMP:78.0
FTEMP:94.1
FTEMP:109.1
TOO HIGH
FTEMP:66.3
FTEMP:-999

 N:  7   SUM:   113.111   AVE:    16.159

FTEMP:/OFF                                             - 34
AT 09.52. SPENT:0001. UNSPENT:0009989. CPU=2.001. CONSOLE=0.19
```

Fig. 3—Time-sharing session showing use of interactive debugging aids FLOW and TRACE, and showing simple use of BTSS editor.

## Conclusions

With the information provided in this article, the engineer can use BTSS-II for a wide range of applications. As more knowledge is gained with this system, it becomes a proportionately more useful computing tool. Several other articles in this issue, including those by Mr. A. Morgan, Dr. G. Gordon, Dr. R. Schilling, Mr. W. Pratt, and Mr. D. Persyk, illustrate the computing power available to the engineer at his time-sharing terminal.

# Efficient use of your computer time-sharing terminal

**F. W. Brill**

**The computers used in the time-sharing system usually handle a large number of simultaneous inputs. Because of this heavy load, these systems demand an efficient programming input. The most direct input is obtained with the use of teletype, such as the Model 35 Teletype Unit. However, using paper tape input to the teletype increases the rate of data transmission and reduces the need for editing and reediting the programs and, thus, significantly cuts the cost of the computer use. This paper describes the methods of producing tapes off line for entry into the RCA time-sharing system.**

---

THE TELETYPE CONSOLE can be used in two different ways. One way is the on-line method, i.e., connected through telephone line with the Spectra 70/45 computer. Charges are based upon the duration of this connection. The second way is the off-line method, i.e., not connected to the computer, but only for paper-tape punching. Information punched on this tape can be entered later during on-line operation.

Before the computer can correctly execute a program, the typing must be accurate in every detail. The computer, for instance, will never correct the typed in "1" when it was meant to be an "I", or the "0" instead of the alphabetic letter "O".

## System Inputs

The RCA Time-Sharing System accepts two kinds of inputs: keyboard and tape. The rate at which information is accepted from the keyboard is limited by the rate at which it is typed. The rate at which information is accepted from the tape is limited only by the maximum reading speed of the teletypewriter, which is 7 characters a second, or 420 characters a minute.

Fig. 1a is a photograph of the Model 35 teletype unit; Figs. 1b and 1c show the diagrams of the keyboard and associated controls. Information in the form of letters, numbers, symbols, and special control characters is controlled by the keyboard. The routing of this information is controlled by the buttons on either side of the keyboard. The manual[1] supplied with the teletype describes the function of these buttons. The button marked LCL, which stands for local, places the terminal in position to do off-line work such as preparing tape. The button marked CLR, which stands for clear, turns the terminal off. The buttons TD ON and TD OFF turn the tape reader on and off, respectively. Buttons K, KT, and T route the flow of information from the keyboard, from the tape reader, and from the telephone link, respectively, to the tape perforator, to the link, and to the typer which produces the printed copy. The route of the information depends not only on the selected button, but also on any of the three possible sources feeding the system. This interaction is shown in Table I.

Programs are punched on tape with the use of these controls. The teletype is prepared for punching when the buttons marked LCL and KT are depressed. Next, the button marked HERE IS or the keys marked CTRL, SHIFT, P, and REPEAT are depressed simultaneously and held to produce blank tape to serve as leader. The first FORTRAN statement then can be typed. (FORTRAN is the language of the RCA Basic Time-Sharing System). At the end of the statement, the key marked CTRL is depressed. While this key is held, the key marked X OFF is depressed. Then, both of these keys are released and the keys marked RETURN and LINE FEED are depressed.

This procedure can also be used to prepare input data for entry. Exactly how the data is entered depends upon the program requirements. If a simple case is assumed in which a list of numbers is to be read, the same procedure

**F. W. Brill**
Equipment Development
Electronic Components
Lancaster, Pa.
received the BS in Mathematics in 1950 from Western Maryland College, and the MS in Chemistry in 1965 from Franklin & Marshall College. In 1950 he joined the Lansdale Tube Company where he worked in the production engineering department of the B&W television tube line. He transferred to the newly formed transistor manufacturing facility in 1952. In 1954, Mr. Brill joined the Brown Instrument Division of Minneapolis Honeywell, Inc. There he worked in the electromechanical activity responsible for improvements in the performance of the null balance temperature recorders and indicators. In 1957, he started his own business which reclaimed germanium from the wastes of transistor manufacture. In 1959 he sold the businss and joined the Equipment Development Department EC&D, RCA Lancaster, Pa. where he is involved in the developing of new production processes for the manufacture of electron tubes. Mr. Brill is a registered Professional Engineer in Pennsylvania, a member of ASTME, PSPE, and has filed two Patent Disclosures.

Table I

| *Buttons Depressed* | *Result* |
|---|---|
| K only | Keyboard connects to the typer and to the link; link connects to typer. |
| KT only | Keyboard connects to the typer and to tape perforator. Link connects to typer and to perforator. |
| KT, TD ON | Tape reader connects to the typer, to the tape perforator, and to the link. Link connects to typer and to perforator. |
| T only | Keyboard connects to tape perforator; link connects to typer. |
| T, TD ON | Tape reader connects to typer and to the link; link connects to the typer. |

Reprint RE-14-3-17
Final manuscript received September 6, 1968.

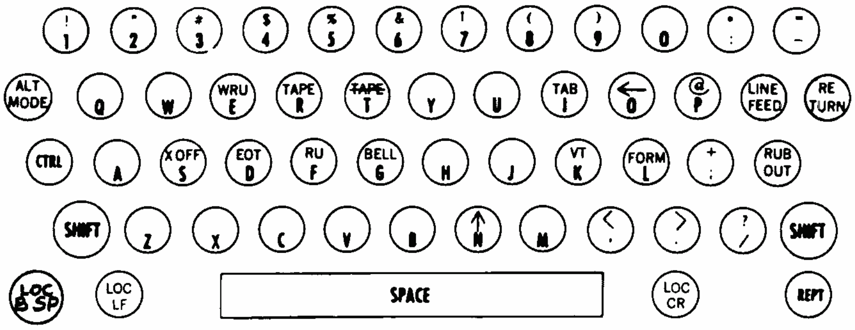
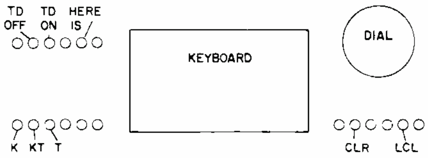
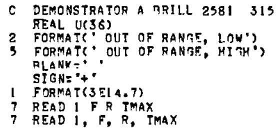


Fig. 1a—Photo at left shows the author at the Model 35 teletype console.

Fig. 1b—Above is a diagram of the keyboard.

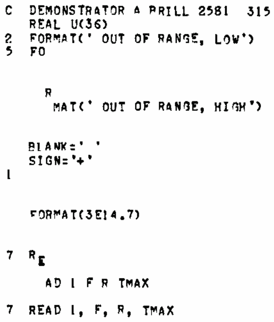
Fig. 1c—Below is a diagram of the controls associated with the keyboard.





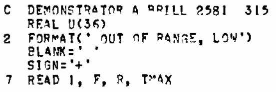
```
C  DEMONSTRATOR A BRILL 2581  315
   REAL U(36)
2  FORMAT(' OUT OF RANGE, LOW')
5  FORMAT(' OUT OF RANGE, HIGH')
   BLANK=' '
   SIGN='+'
1  FORMAT(3E14.7)
7  READ 1 F R TMAX
7  READ 1, F, R, TMAX
```

Fig. 2a—Printed record before editing.



```
C  DEMONSTRATOR A BRILL 2581  315
   REAL U(36)
2  FORMAT(' OUT OF RANGE, LOW')
5  FO


     R
      MAT(' OUT OF RANGE, HIGH')


   BLANK=' '
   SIGN='+'
1


   FORMAT(3E14.7)


7  R
    E
     AD 1 F R TMAX

7  READ 1, F, R, TMAX
```

Fig. 2b—Printed record with edit instructions added.



```
C  DEMONSTRATOR A BRILL 2581  315
   REAL U(36)
2  FORMAT(' OUT OF RANGE, LOW')
   BLANK=' '
   SIGN='+'
7  READ 1, F, R, TMAX
```

Fig. 2c—Printed record after editing.

is followed as for program preparation except that a number is typed instead of a FORTRAN statement. After each number, CTRL, X OFF, RETURN, and LINE FEED are typed. The technique described overcomes speed problems but is not very accurate.

### Off-line Editing

With the use of off-line editing techniques, the record on the tape can be added to, removed from, or changed. Teletype tape and the Model 35 teletype constitute a very versatile combination. In the case of a program error, which requires an entire FORTRAN statement to be deleted from a punched tape, it is necessary to place the tape in the tape reader. The tape is correctly positioned if the row of central holes in the leader fits over the cogged wheel in the tape reader with the rough edges of the holes up. The button marked KT is then depressed, followed by the button marked TD ON. This procedure causes the teletype to print what is on the tape and, also, to make a copy of the tape being read. At the end of a FORTRAN statement, the tape reader turns off automatically. Button TD ON must be depressed to read the next statement. Each time button TD ON is depressed, one FORTRAN statement is read, printed, and put on tape. When it is necessary to delete the next FORTRAN statement, key T is depressed before TD ON. In this case, the tape is read and the record is printed, but no copy is made. It is important to remember, of course, to press KT before the next FORTRAN statement is read so that it will be correctly copied on tape. This technique may be used to eliminate unwanted or incorrectly typed FORTRAN statements.

Figs. 2a, 2b, and 2c show the printed record before, during, and after editing. The 4th, 7th, and 9th FORTRAN statements have been deleted. As an aid to the programmer, the key marked LOC LF (local line feed) was actuated during the printing of the deleted edit instruction. This procedure clearly illustrates the deleted instruction in Fig. 2b. Local line feed causes the paper in the typer to advance without punching of the new tape. This editing helps during preparation of the original tape. If a mistake is made during the formation of a FORTRAN statement, the statement is terminated when CTRL and X OFF, LINE FEED and RETURN are typed. The correct FORTRAN statement is then retyped, and, during off-line editing, the incorrect FORTRAN statement is dropped. In Fig. 2a, the last statement deleted was of this type. Fig. 2c shows the output from the new tape.

Gross additions to the program are also possible. In this case, the original tape is read and copied, while KT is depressed initially followed by TD ON for each new FORTRAN statement. The tape is copied to the point where the addition is to be made. The additions are then typed on the keyboard. After the additions are included, copying of the original tape resumes with TD ON depressed again. This technique is often useful in conjunction with the aforementioned method for deletion. In cases where a mistake is not detected until after the program is prepared, a new tape can be generated with new typing where the old instruction is deleted. Figs. 3a, b and c show the records before, during, and after the off-line editing process, respectively. The sixth instruction in the original record has been deleted, and the fourth instruction has been substituted.

Errors within a FORTRAN statement can also be corrected. Such a technique is particularly useful if the statement is very long, as in the case of a format statement with Hollerith or other lengthy statements. With this technique, however, it is possible to salvage the first half of such a statement where the error is in the latter half. In this procedure, the original tape is placed in the tape reader, as already described. The KT and TD ON buttons are depressed. Copying is continued until the FORTRAN statement containing the error is reached. Just before the error is reached in the statement, the TD OFF button is quickly depressed. This button stops the tape being read and the new tape being generated. At this point, the rest of the statement is typed at the keyboard. This statement is ended with CTRL and X OFF, LINE FEED, RETURN depressed. However, a slight problem then exists. If the TD ON button is depressed again, the remainder of the statement containing the error will be punched on the new tape. This copying is prevented when the T button and then TD ON button are depressed. During this step, the original tape is read, a printed record is made, but no tape copy is made. After the printing stops, buttons KT and then TD ON are depressed. The next FORTRAN statement is then correctly copied on the tape. It is possible to repeat this process as often as necessary for accurate recording of long statements. The main advantage of this technique is that once the beginning of the statement is typed correctly it is no longer possible to make an error in that part. The chance that the program will be ruined just as it is being completed does not occur with this technique. Fig. 4 shows the progress of ridding a long statement of errors with the use of off-line editing. In the second entry, the copying was not halted soon enough; therefore, the step was repeated.

```
C  DEMONSTRATOR  Z BRILL 2581 315
   REAL B(100)
   DO 10 I=1.100
5  READ 7, X, Y
7  FORMAT (2E14.7)
   B(3)=(X*Y+3.145926/X)-Y
10 B(I)=(X+4)/Y-X)+Y*X
```

Fig. 3a—Record of tape before editing.

```
C  DEMONSTRATOR  Z BRILL 2581 315
   REAL B(100)
   DO 10 I=1,100
5  RE

         7, X, Y
3  READ 7, X, Y
7  FORMAT (2E14.7)
   B(3

        =(X*Y+3.145926/X)-Y
10 B(I)=(X+4)/Y-X)+Y*X
```

Fig. 3b—Record of tape during off-line editing.

```
C  DEMONSTRATOR  Z BRILL 2581 315
   REAL B(100)
   DO 10 I=1,100
3  READ 7, X, Y
7  FORMAT (2F14.7)
10 B(I)=(X+4)/Y-X)+Y*X
```

Fig. 3c—Record of tape after off-line editing.

```
5  Y32NX=RAD(((EXP(DIA?DIA
5  Y32NX=RAD(((EXP(DIA?_
5  Y32NX=RAD(((EXP(DIA/3.14159+342)+AR TAN
5  Y32NX=RAD(((FXP(DIA/3.14159+342)+ARCTAN(SUMY-IXTAX))-LOG(IRAD?Y32NX))
5  Y32NX=RAD(((EXP(DIA/3.14159+342)+ARCTAN(SUMY-IXTAX))-LOG(IRAD/Y32NX))
```

Fig. 4—Editing of a lengthy statement.

```
C  MRPHS LAS SDSTES  TAT WEN SOMTING CANGO WRONG ID USALY DOES>
C  MURPHY'S LASW SDTASTES  THAT WHEN SOMTHING CAN GO WRONG IDT USUALLY


C  MURPHY'S LAW STASTES THAT WHEN SOMTHING CAN GO WRONG IT USUALLY DOES>
C  MURPHY'S LAW STASTES THAT WHEN SOMTHING CAN GO WRONG IT USUALLY DOES>
C  MURPHY'S LAW STATES THAT WHEN SOMT
C  MURPHY'S LAW STATES THAT WHEN SOMETHING CAN GO WRONG IT USUALLY DOES.
```

Fig. 5—Single character editing.

Very small changes consisting of a single character change are also possible. For such a change, it is necessary to depress TD OFF at or before the character to be changed. This procedure requires some practice. In one approach, TD OFF and TD ON are depressed in quick succession so that the record advances one character at a time until the error is reached. Once the error is reached, the correction is typed on the keyboard. As before, it is necessary to prevent this error from being copied on the new tape. For this procedure, button T is depressed and again, TD ON and TD OFF are depressed in quick succession to print the error without copying. If a few extra characters, besides the one being changed, get printed during this stage, no great harm is done. It is merely necessary to depress KT ON and then to enter the accidentally deleted characters at the keyboard onto the new tape. The copying is then resumed when TD ON is depressed. Fig. 5 shows the single-character editing in progress where an English sentence has been used as a simplified example. The underlined characters have not been copied on the latest tape; therefore, these characters are missing from the next printing as shown. The characters designated with a dot have been added to the latest tape from the keyboard. The next to the last line was not completed because the copying was not stopped soon enough for the "E" to be added. The "E" was added to the subsequent record, and the ">" was replaced by the period to produce the final record.

### Editing with Tape

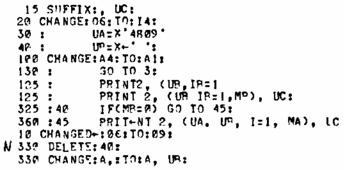
Great savings in time can also be obtained by a combination of off-line tape punching with on-line editing of the conventional kind. Often, serious errors in logic during program preparation are exposed during initial testing. In many cases these errors will not be simple, one-statement changes, but a whole series of these. The user can correct these errors by making a copy of the current program, writing out the instructions for the editor, and then punching a tape of these instructions off-line. After each instruction, CTRL and X OFF, RETURN, and LINE FEED must be entered. It is surprising how often it is necessary to edit the editing tape, that is, to recopy the editing instructions and to make off-line corrections. The chance to check and correct editing instructions before the actual program is edited can save much time. When a bad edit instruction is entered, a moderately lengthy error message is fed back by the computer. Even in cases of on-line editing where the error is detected before the RETURN key is pressed, the message DELETED from the computer is somewhat time-consuming. All such wasted steps are eliminated when editing is done with the use of a tape. With this method, major program changes can be executed in just a few minutes of computer time. The experienced user can employ the /NO CHECK feature of the system to further increase the editing speed. This system command entered before /EDIT inhibits the printing of the corrected statement.

### Examples

Now that the methods of off-line editing have been discussed, their actual application is presented. Realistic examples from actual computer sessions have been selected to illustrate the normal confusion that occurs. Figs. 6a through 6f constitute such an example. During preparation of Fig. 6a, it was decided to eliminate the comma after UB in the edit instruction which will produce seq #125. The presence of this comma was not noticed until four more characters had been typed and punched. At this point, the editing instruction was aborted and the desired editing instruction was retyped on the next line. Fig. 6b shows the record produced during off-line editing of the editing instructions. As previously described, line 7 was printed but not copied. LOC LF (local line feed) was used to identify the statement being deleted. In the 12th edit instruction, the letter N was typed and prefixed to the seq #330 before the editing instruction was copied from the previous tape. Fig. 6c is a printout of the resulting taped editing instructions. Fig. 6d is a copy of the program to be edited. Fig. 6e is the printout produced as the taped edit instruction operated on the program. [*Note:* Model 35 Teletype Units use two different fonts. Therefore, symbols such as ←, ', and " of one font are equivalent to —, , and " on the other.]

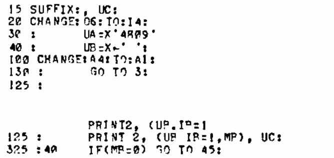
After the computer responded READY, /EDIT was entered from the keyboard followed by the prepared taped edit instructions. These instructions are read in with the button T depressed, followed by the button marked TD ON. On the first line after /EDIT, the request to suffix , UC to seq #15 is recorded. On the next printed line, the results of this editing step are recorded along with the next editing instruction for seq #20. When these changes are read in from tape, the system gives no signal that separates the results of the last editing change and the instructions for the next one. On the next line, the new FORTRAN statement for seq #20 is recorded. The next editing instruction should have followed this one immediately. Instead, the system acted as though it had received a blank line and printed the current delimiter symbol and the verification state. For verification that seq #15 and 20 had been edited properly, the system was switched to keyboard control when button K was depressed, and these sequence numbers were read when 15 and 20 were typed with the RETURN after each. After seq #15 and 20 indicated correct editing, more instructions were read in from the tape. For this procedure, T was depressed followed by TD ON. On the next line, the changes for seq #30 were copied. On the line following, the new FORTRAN statement for seq #30 was recorded followed immediately with the edit instructions for seq #40. In this fashion, seq #100, 130, 125, 325, 360, and 10 were edited. The presence of the prefixed N in front of the edit instruction for seq #330 caused a change. This N is used when verification of the edit instruction is not required. In this circumstance, the system merely read the next edit instruction and printed it on the next line. The result of this latter instruction was printed on the next line.

A confusion of errors plagued the next change for seq #325. The first time the tape was read into the system, it was not accepted. The tape was repositioned in the tape reader to read the instruction in again. This time the system accepted the instruction, but could not execute the change because the instruction requested that UB be changed to UA when seq #325 did not contain UB. At this point, the use of keyboard editing was required. With the K depressed, the 325 typed, and the RETURN depressed, the nature of the current FORTRAN expression at seq #325 was determined. The change that was really intended was entered, namely: a change of variable name MB to MA. After this editing command, the system responded with the new FORTRAN statement at seq #325. With the exception of the reinserted comma after UB at seq #125 and the addition of a comma at seq #330 in accordance with FORTRAN IV rules, the above corrections were the only ones that were needed. Fig. 6f is a copy of the resulting program.



```
 15 SUFFIX:, UC:
20 CHANGE:06:TO:I4:
30 :        UA=X'4809'
40 :        UB=X←' ':
100 CHANGE:A4:TO:A1:
130 :        GO TO 3:
125 :        PRINT2, (UB,IB=1
125 :        PRINT 2, (UB IB=1,MB), UC:
325 :40     IF(MB=0) GO TO 45:
360 :45     PRIT←NT 2, (UA, UB, I=1, MA), UC
10 CHANGED←:06:TO:09:
N 330 DELETE:40:
330 CHANGE:A,:TO:A, UB:
```

Fig. 6a—Original copy being typed.



```
15 SUFFIX:, UC:
20 CHANGE:06:TO:I4:
30 :        UA=X'4809'
40 :        UB=X←' ':
100 CHANGE:A4:TO:A1:
130 :        GO TO 3:
125 :


             PRINT2, (UB,IB=1
125 :        PRINT 2, (UB IB=1,MB), UC:
325 :40     IF(MB=0) GO TO 45:
360 :45     PRIT←NT 2, (UA, UB, I=1, MA), UC
10 CHANGED←:06:TO:09:
N330 DELETE:40:
330 CHANGE:A,:TO:A, UB:
```

Fig. 6b—Off-line editing corrections to remove a comma from seq #125.

## Summary

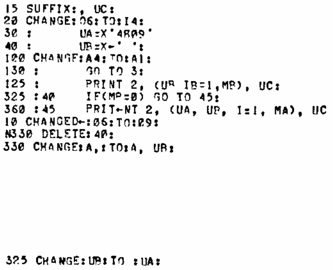
Off-line editing is a low cost way to correct errors and edit instructions. These off-line techniques include:

1) Deletion, addition, or both of an entire FORTRAN statement, an edit instruction or data entry.
2) Replacement of the latter portion of a FORTRAN statement, edit instruction, or data entry.
3) Deletion or replacement of as little as one character within a message.

With proper use of off- and on-line editing, the edit features of the system can be used at the maximum rate.

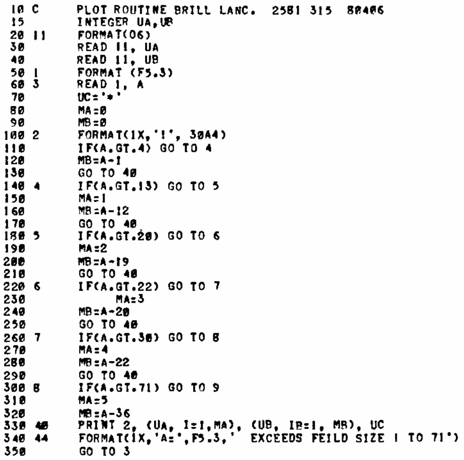
## References

1. The Bell System, "Operator Instructions for the #35 teletypewriter TWX".



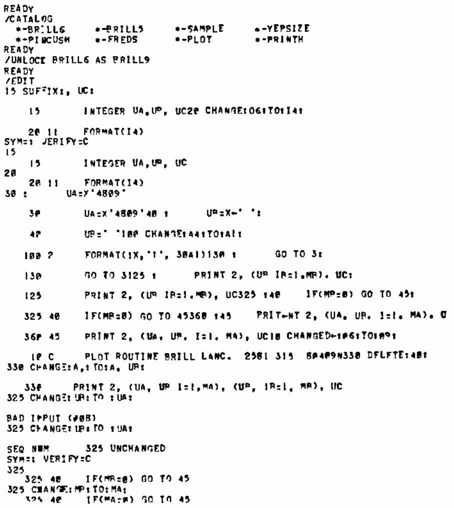
```
15 SUFFIX:, UC:
20 CHANGE:06:TO:I4:
30 :        UA=X'4809'
40 :        UB=X←' ':
100 CHANGE:A4:TO:A1:
130 :        GO TO 3:
125 :        PRINT 2, (UB IB=1,MB), UC:
325 :40     IF(MB=0) GO TO 45:
360 :45     PRIT←NT 2, (UA, UB, I=1, MA), UC
10 CHANGED←:06:TO:09:
N330 DELETE:40:
330 CHANGE:A,:TO:A, UB:







325 CHANGE:UB:TO :UA:
```

Fig. 6c—Result of changes shown in Fig. 6b.



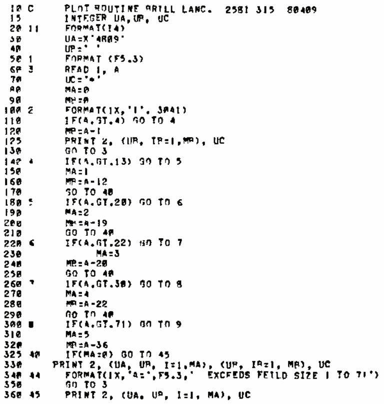
```
 10 C      PLOT ROUTINE BRILL LANC.  2581 315  80406
 15        INTEGER UA,UB
 20 11     FORMAT(O6)
 30        READ 11, UA
 40        READ 11, UB
 50 1      FORMAT (F5.3)
 60 3      READ 1, A
 70        UC='*'
 80        MA=0
 90        MB=0
100 2      FORMAT(1X,'!', 30A4)
110        IF(A.GT.4) GO TO 4
120        MB=A-1
130        GO TO 40
140 4      IF(A.GT.13) GO TO 5
150        MA=1
160        MB=A-12
170        GO TO 40
180 5      IF(A.GT.20) GO TO 6
190        MA=2
200        MB=A-19
210        GO TO 40
220 6      IF(A.GT.22) GO TO 7
230              MA=3
240        MB=A-20
250        GO TO 40
260 7      IF(A.GT.30) GO TO 8
270        MA=4
280        MB=A-22
290        GO TO 40
300 8      IF(A.GT.71) GO TO 9
310        MA=5
320        MB=A-36
330 40     PRINT 2, (UA, I=1,MA), (UB, IB=1, MB), UC
340 44     FORMAT(1X,'A=',F5.3,'  EXCEEDS FEILD SIZE ! TO 71')
350        GO TO 3
```

Fig. 6d—Complete program to be edited.



```
READY
/CATALOG
  *-BRILL6    *-BRILL5    *-SAMPLE    *-YEPSIZE
  *-PINCUSH   *-FREDS     *-PLOT      *-PRINTH
READY
/UNLOCE BRILL6 AS BRILL9
READY
/EDIT
15 SUFFIX:, UC:

    15       INTEGER UA,UB, UC20 CHANGE:06:TO:I4:

    20 11    FORMAT(I4)
SYM=: VERIFY=C
15
    15       INTEGER UA,UB, UC
20
    20 11    FORMAT(I4)
30 :       UA=X'4809'

    30       UA=X'4809'40 :      UB=X←' ':

    40       UB=' '100 CHANGE:A4:TO:A1:

   100 2     FORMAT(1X,'I', 30A1)130 :      GO TO 3:

   130       GO TO 3125 :      PRINT 2, (UB IB=1,MB), UC:

   125       PRINT 2, (UB IB=1,MB), UC325 :40    IF(MB=0) GO TO 45:

   325 40    IF(MB=0) GO TO 45360 :45    PRIT←NT 2, (UA, UB, I=1, MA), U

   360 45    PRINT 2, (UA, UB, I=1, MA), UC10 CHANGED←:06:TO:09:

    10 C     PLOT ROUTINE BRILL LANC.  2581 315  80409N330 DELETE:40:
330 CHANGE:A,:TO:A, UB:

   330     PRINT 2, (UA, UB I=1,MA), (UB, IB=1, MB), UC
325 CHANGE: UB:TO :UA:

BAD IPPUT (#0B)
325 CHANGE: UB:TO :UA:

SEQ NUM      325 UNCHANGED
SYM=: VERIFY=C
325
   325 40    IF(MB=0) GO TO 45
325 CHANGE:MB:TO:MA:
   325 40    IF(MA=0) GO TO 45
```

Fig. 6e—Printed copy of edit instructions operating on the program of Fig. 6d.



```
 10 C      PLOT ROUTINE BRILL LANC.  2581 315  80409
 15        INTEGER UA,UB, UC
 20 11     FORMAT(I4)
 30        UA=X'4809'
 40        UB=' '
 50 1      FORMAT (F5.3)
 60 3      READ 1, A
 70        UC='*'
 80        MA=0
 90        MB=0
100 2      FORMAT(1X,'I', 30A1)
110        IF(A.GT.4) GO TO 4
120        MB=A-1
125        PRINT 2, (UB, IB=1,MB), UC
130        GO TO 3
140 4      IF(A.GT.13) GO TO 5
150        MA=1
160        MB=A-12
170        GO TO 40
180 5      IF(A.GT.20) GO TO 6
190        MA=2
200        MB=A-19
210        GO TO 40
220 6      IF(A.GT.22) GO TO 7
230              MA=3
240        MB=A-20
250        GO TO 40
260 7      IF(A.GT.30) GO TO 8
270        MA=4
280        MB=A-22
290        GO TO 40
300 8      IF(A.GT.71) GO TO 9
310        MA=5
320        MB=A-36
325 40     IF(MA=0) GO TO 45
330     PRINT 2, (UA, UB, I=1,MA), (UB, IB=1, MB), UC
340 44     FORMAT(1X,'A=',F5.3,'  EXCEEDS FEILD SIZE ! TO 71')
350        GO TO 3
360 45     PRINT 2, (UA, UB, I=1, MA), UC
```

Fig. 6f—The edited program.

# The computer as a tool in acoustical research

### A. R. Morgan

**The scientist and engineer should be aware that the modern computer, with FORTRAN programming, can be a valuable tool in their routine activities. In particular, the RCA Basic Time Sharing System allows a centrally located computer to be operated from a remote Teletype console. This combination is as easy to use as a modern desk calculator, but is considerably more powerful. The following account demonstrates an application of the computer to a problem in acoustical research.**

**Adolph R. Morgan**
Consumer Electronics Research Laboratory
Laboratories
Princeton, N.J.
received the BSEE in 1931 from the University of California. Following graduation, he did research work at RCA, and from the opening of the RCA Laboratories in Princeton in 1942 he has been a member of its Acoustical and Electromechanical Laboratory. Mr. Morgan has been active in every phase of audio research and has made many basic contributions to the fields of photographic recording, mechanical and magnetic recording, and electro-acoustic transducers. He is a member of Phi Beta Kappa, Tau Beta Pi, Eta Kappa Nu and Sigma Xi, fellow of the Audio Engineering Society and member of the Acoustical Society of America.

APPROXIMATELY NINE MONTHS before this writing, the author was invited to attend an RCA Laboratories sponsored course in FORTRAN programming of the current models of RCA computers. [The course presented by Dr. G. D. Gordon, consisted of four three-hour sessions. The author's attendance, accompanied by a like amount of time spent on homework and practice with the computer, provided the necessary proficiency to perform the type of programming shown herein.] Before that invitation, the author's concept of the modern computer was such that he wouldn't dare approach the computer camp without a PhD in mathematics and a problem as complex as sending a man to the moon. The course proved the fallacy of this concept; in fact, the author now believes that many simple problems may well be considered for computer aid in their solution.

There is no intention, in this paper, to belittle the expert programmers with their sophisticated problems. In truth, the author envies their abilities and hopes they will write more about their programming techniques.

It is the purpose of this paper to show the ease and speed with which relatively simple problems may be processed through RCA's most recently developed aspect of modern computers: The RCA Basic Time Sharing System (BTSS).

The ultimate in simplicity might well be illustrated by "asking" the computer to solve:

$$\theta = \tan^{-1} 2.0$$

and to provide the answer in radians. The complete session with the BTSS Teletype terminal to achieve the result is shown in Fig. 1. The typed lines marked with an arrow (←) were entered by the author. The remaining lines are computer response. The first five lines constitute the standard procedure for establishing a connection with the computer and setting it in a mode to receive the program. The sixth line, in this example, is the complete program.[1] The asterisk (*) directs the computer to execute the line without further instruction. The seventh line is the desired answer with seven significant figures. (The quantity, E+01, indicates that the answer is to be multiplied by 10 to the +1 power, i.e., the answer is 1.107148). The eighth and tenth lines are the procedure for erasing the program from the computer and terminating the Teletype session.

The last line is the computer report of the time and charges for the session. In this case the charge was $1 (a minimum charge) and the session time on the console was 0.01 hour.

For comparison, the above problem was solved manually by reference to an eight-place tangent table. For this solution, interpolation was required and a conversion to radians was necessary. The manual operation consumed approximately 10 minutes and resulted in the same answer.

## Acoustical Applications

To illustrate how the BTSS can be useful in acoustical research, consider a preliminary examination of a higher-order unidirectional gradient microphone. The distinguishing characteristic of a unidirectional gradient microphone is the discrimination against sound waves which come from directions other than the forward axis of the microphone. The primary interest, therefore, will be in the directional characteristic and in the resulting directional efficiency.[2]

Suppose there is a need to estimate the performance of a combination of two first-order gradient microphones, a delay network, and a summing network arranged as shown in Fig. 2. The response of the combination, referenced to the response on the forward axis, may be written:

$$R = \frac{\alpha + \cos\theta)\cos\theta}{\alpha + 1} \qquad (1)$$

where $\alpha = D_2/D_1$; $D_1$ is the distance between the first-order gradient elements, $D_2$ is the path length of the delay; and $\theta$ is the angle of incidence of the sound wave.

The directional efficiency of a directional microphone may be written:

$$\text{EFF} = ½ \int_0^{\pi} R^2 \sin\theta \, d\theta \qquad (2)$$

where $\theta$ is the angle of incidence of a sound wave; and $R$ is the ratio of response at angle $\theta$ to that for $\theta = 0$.

---

Reprint RE-14-3-3
Final manuscript received July 1, 1968.

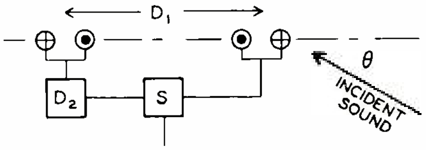


Fig. 2—Schematic arrangement of a higher-order unidirectional microphone; consisting of two first-order gradient microphones, a delay system (D₂) and summing network (S).

Substituting Eq. 1 in Eq. 2 and performing the integration, the directional efficiency becomes:

$$\text{EFF} = \frac{\dfrac{\alpha^2}{3} + \dfrac{1}{5}}{(\alpha + 1)^2}$$

Since the directional efficiency is a measure of performance, it will be instructive to examine Eq. 3 for varying values of the ratio, $\alpha$. It will be even more instructive to try the computer again.

The author has found it expeditious to plan and prepare a rough-draft of the necessary program before starting the computer session. In this case, the rough-draft was prepared in two minutes. It was arbitrarily decided to let the ratio, $\alpha$, range from 0.1 to 1.5 in 0.1 increments. The complete Teletype session is shown in Fig. 3. [In this and the following programs, the procedure for making connection with the computer and terminating the session will be deleted in reproduction of the Teletype copy.] The six statements (lines) following the /CODE command constitute the planned program. The hyphen (-) in the line after the end of the program is the command to execute the program. The computer then automatically calculates and prints corresponding values of $\alpha$ and EFF (the data column headings were supplied later, manually). The session time was 0.05 hour and the cost was within the minimum charge.

It is observed that the directional efficiency has a broad minimum, but before concluding that $\alpha = 0.6$ is the desired adjustment, it may be well to examine the effect of the ratio, $\alpha$, on the directional characteristic as determined from Eq. 1.

Let us arbitrarily assign the values, 0.4, 0.6, 0.8 and 1.0 to $\alpha$. Since the response is symmetrical about the microphone axis, it is only necessary to assign values for $\theta$ between 0 and 180 degrees. For a preliminary examination let us increment $\theta$ in 10-degree steps. Turning again to the computer, it required approximately 10 minutes to plan the program and a 0.12-hour session at the Teletype terminal to produce the program and data shown in Fig. 4, again for the minimum charge.

For some conditions, the response data is negative, indicating a phase reversal. Negative or positive, it is still response and should be so treated. It would have been in order to "ask" the computer to print-out absolute values of the response. If the response is examined for the different values of $\alpha$, it is seen that, in addition to the desirable peak of response in the direction of the forward axis, there are undesirable peaks of response (back-lobes) which are dependent on $\alpha$. For example, there is a back-lobe at 180° which varies from 0.428 for $\alpha = 0.4$ to 0.00 for $\alpha = 1.0$. There is a second back-lobe which ranges from 0.028 at 100° to 0.125 at 120°, corresponding to $\alpha$ ranging from 0.4 to 1.0. One may also deduce that the two lobes would be equal for $\alpha$ somewhat larger than 0.8. The lobe equality may be desirable since, for that condition, the overall back-lobe response will be a minimum.

If we return to Eq. 1, it may readily be seen that the response at 180° is:

$$R_1 = \frac{1 - \alpha}{1 + \alpha} \qquad (4)$$

If the derivative of Eq. 1, with respect to $\theta$, is equated to zero, it may be shown that the absolute response at the peak of the second lobe is:

$$R_2 = \frac{\alpha^2}{4(1 + \alpha)}$$

It remains then, to find the value of $\alpha$ which makes $R_1 = R_2$. [The desired value of $\alpha$ could be determined by a manual solution of the quadratic resulting from equating $R_1$ and $R_2$; however, the following procedure was selected to serve as an example of computer use.]

```
BEGIN 24
/ON 99999ZZZ  <-
05/15/68 15.68 24
READY
/CODE ARM  <-
    10 *THETA=ATAN(2.0)  <-
THETA= .1107148E+01
    10 /DROP ARM  <-
READY
/OFF  <-
AT 15.69, SPENT=$001, CONSOLE=0.01
```

Fig. 1—BTSS teletype session for computation of $\theta = \tan^{-1}(2.0)$.

```
/CODE EFF
10      DO 20 I=1,15
20      A=0.1*I
30      EFF=(A*A/3.+0.2)/(A+1.)**2
40 10   FORMAT(' ',F4.1,F10.5)
50 20   PRINT 10,A,EFF
60      STOP
70 -

α       EFF
0.1     0.15804
0.2     0.14815
0.3     0.13609
0.4     0.12925
0.5     0.12593
0.6     0.12500
0.7     0.12572
0.8     0.12757
0.9     0.13019
1.0     0.13333
1.1     0.13681
1.2     0.14050
1.3     0.14430
1.4     0.14815
1.5     0.15200
```

Fig. 3—Computer program and computed data for microphone directional efficiency.

```
 10      REAL R(4)
 20      THETA=0.0
 30      DO 30 I=1,19
 40      C=COS(THETA*.PI/180.)
 50      A=0.4
 60      DO 10 J=1,4
 70      R(J)=(A+C)*C/(A+1.)
 80 10   A=A+0.2
 90 20   FORMAT(' ',F8.1,4F9.5)
100      PRINT 20,THETA,R(1),R(2),R(3),R(4)
110 30   THETA=THETA+10.
120      STOP
130 -

   θ        R         R         R         R
          α=0.4     α=0.6     α=0.8     α=1.0
   0.0   1.00000   1.00000   1.00000   1.00000
  10.0   0.97412   0.97546   0.97650   0.97733
  20.0   0.89921   0.90427   0.90821   0.91136
  30.0   0.78315   0.79351   0.80157   0.80801
  40.0   0.63803   0.65403   0.66648   0.67643
  50.0   0.47873   0.49928   0.51523   0.52798
  60.0   0.32143   0.34375   0.36111   0.37500
  70.0   0.18128   0.20137   0.21700   0.22950
  80.0   0.07115   0.08396   0.09393   0.10190
  90.0   0.00000   0.00000   0.00000   0.00000
 100.0  -0.02808  -0.04627  -0.06042  -0.07175
 110.0  -0.01416  -0.05515  -0.08702  -0.11252
 120.0   0.03571  -0.03125  -0.08333  -0.12500
 130.0   0.11147   0.01719  -0.05614  -0.11481
 140.0   0.20029   0.07950  -0.01445  -0.08961
 150.0   0.28828   0.14399   0.03177  -0.05801
 160.0   0.36225   0.19950   0.07293  -0.02834
 170.0   0.41137   0.23685   0.10111  -0.00748
 180.0   0.42857   0.25000   0.11111   0.00000
```

Fig. 4—Computer program and computed data for preliminary examination of microphone directional characteristics.

Let us, therefore, program the computer as follows:

1) Calculate $R_1$ and $R_2$ for a group of closely spaced values of $a$, starting at 0.8 and ranging upward,
2) When the difference between $R_1$ and $R_2$ becomes less than 0.0001, print-out $\alpha$, $R_1$, and $R_2$.

The rough-draft of the program was prepared in 7 minutes. The session with the computer, as shown in Fig. 5, required 0.06 hour. It is seen that for $\alpha = 0.828$, the two lobes will be substantially equal.

Now we have two values for $\alpha$ (0.6 or 0.828), either of which provides a desirable adjustment of the microphone. To make the final choice, one could wish for a detailed representation of the directional characteristic for either adjustment. We could, of course, return to the program shown in Fig. 4, which had been retained by temporary storage in the computer files. However, from the standpoint of demonstrating the use of the computer, let us examine a more involved program; one which will provide the desired data in a more elegant form and, possibly, more informative.

```
10      A=.7999
20 10   A=A+.0001
30      R1=(1.-A)/(1.+A)
40      R2=A*A/(4.*(1.+A))
50      IF(ABS(R1-R2)-.0001)20,10,10
60 30   FORMAT(' ',3F10.6)
70 20   PRINT 30,A,R1,R2
80      STOP
90 -

   a          R1         R2
0.828388   0.093860   0.093829
```

Fig. 5—Computer program and computed data for determining value of which produces equal back-lobes of response.

```
 10      REAL RX(37),RY(37)
 20 10   FORMAT(3F7.3,I1)
 30 20   FORMAT('1'3X'ANGLE'2X'RESPONSE'/
               5X'DEG.'3X'A='F5.3/I1)
 40 30   FORMAT('  'F8.1,F10.5)
 50 40   FORMAT('1   A='F5.3)
 60 C    DATA,M=1;CURVE,M=2;BOTH,M=3
 70      READ 10,A,ANG,AINC,M
 80      IF(M.NE.2)PRINT 20,A
 90      DO 50 I=1,37
100      THETA=ANG*.PI/180.
110      C=COS(THETA)
120      R=ABS((A+C)*C/(A+1.))
130      IF(M.NE.2)PRINT 30,ANG,R
140      IF(M.NE.1)RX(I)=R*C
150      IF(M.NE.1)RY(I)=R*SIN(THETA)
160 50   ANG=ANG+AINC
170      PRINT 40,A
180      IF(M.NE.1)CALL XYPLOT(RX,RY)
190      STOP
200 /DO M XYPLOT
530 -
A=0.600
ANG=0.0
AINC=5.0
M=3
```

Fig. 6—Computer program for detailed examination for microphone directional characteristic.

We will plan a program which, when initiated, will permit a choice of:

1) the values of $\alpha$,
2) the range and incrementing of $\theta$, and
3) printing the calculated data, plotting a curve of the data, or both.

The preparation required approximately 40 minutes and resulted in the program shown in Fig. 6. Note the term XYPLOT in statements 180 and 200. XYPLOT is a curve-plotting program which had been prepared and stored in one of the computer files several months prior to this writing. The word CALL, in statement 180, transfers the appropriate data to XYPLOT. The command /DO M, in statement 200, causes the program XYPLOT to be associated with the current program.

After the command to execute the program (-), the computer "asks" for the choices as shown and supplied in the final four lines of Fig. 6.

The first two runs of the program were for choices of $\alpha = 0.6$ and $\alpha = 0.828$; otherwise, the choices were for $\theta$ to range from 0° to 180° in 5° steps and for print-out of data *and* curves. The results are shown in Fig. 7. [The program, XYPLOT, automatically adjusts the scale of the plotted curve so that the curve "fills up the sheet", so to speak. In addition, the $X$ and $Y$ axes are supplied ($\theta = 0°$, 90° and 180° for Eq. 1)]. The RESPONSE data are presented in absolute value versus angle, $\theta$, in degrees.

In the curves, it is noted that the smaller back-lobes are rather poorly resolved, particularly for $\alpha = 0.828$. Let us therefore, try again with parameter choices that will produce curves for one quadrant or the other. It is only necessary to repeat the command for execution and supply the choices for the desired curve. There is no need to repeat the data, so a choice for curve *only* is made. For the first quadrant, $\theta$ is started at 0°. For the second quadrant, $\theta$ is started at 90°. For either quadrant, $\theta$ is incremented in 2.5° steps. First and second quadrant portions of curves for $\alpha = 0.6$ and $\alpha = 0.828$ are shown in Fig. 8.

The computer session time for the entry of the program in Fig. 6 and the print-out of data and curves shown in Figs. 7 and 8 was 0.51 hour, at a cost of $6.

One can conclude that the subject microphone may be adjusted to an optimum directional efficiency of 0.125 by setting $\alpha = 0.6$. However, with a small sacrifice in directional efficiency (to approximately 0.128), by setting $\alpha = 0.828$, the maximum back-lobe response may be reduced from 0.25 to 0.094. From the curves, or the data, it may be seen that there is very little difference in the forward response (first quadrant) for either value of $\alpha$. Under practical conditions, the microphone with the latter adjustment would be preferred.

## Concluding Comments

Summing the individual times quoted for the programs in Figs. 2 through 6, it is found that a total of 59 minutes was spent planning and preparing the programs; the total session time with the computer was 0.74 hour (45 minutes), giving a grand total of approximately 1¾ hours.

From the author's experience, production of the above data and curves manually would require from 8 to 12 hours. In that case, the use of the computer would have saved 6 to 10 hours—a sizable repayment of the time spent on the FORTRAN course.

The total computer cost was $9. It should be noted that this cost is the result of the prevailing rate of charge for use of the RCA Laboratories Basic Time Sharing System. The RCA Corporate Time Sharing System at Cherry Hill has a comparable rate of charge.

It is recommended that the reader assume that FORTRAN programming is a relatively easy field to enter. The seeming complexity of the program language (such as shown in Fig. 6) is easily learned from a short course or, even self teaching with available textbooks.[3] One needs to learn the ground rules[1,5] peculiar to the computer that will be used. Then he is ready to prepare and use programs such as illustrated above.

Although the capabilities of FORTRAN programming are extensive and can be very sophisticated, it is only necessary to acquire a level of technique that fits the complexity of one's problems.

## Appendix

For the reader who may be interested, the program XYPLOT is reproduced in Fig. A-1. The program has been arranged as a SUBROUTINE and may be made a part of a current program as discussed in connection with Fig. 6, above. For the reader having experience with BTSS, it should be noted that XYPLOT takes advantage of the enhanced version of BTSS which is now available, and designated as BTSS-II.

```
    SUBR. XYPLOT(X,Y)
    REAL X(37),Y(37)
    COMMON INTEGER L(71,43)
    XMIN=X(1)
    XMAX=X(1)
    YMAX=Y(1)
    DO 10 I=1,37
    IF(X(I).LT.XMIN)XMIN=X(I)
    IF(X(I).GT.XMAX)XMAX=X(I)
10  IF(Y(I).GT.YMAX)YMAX=Y(I)
    XYMAX=XMAX-XMIN
    IF(YMAX.GT.XYMAX)XYMAX=YMAX
    NX0=-XMIN*42./XYMAX+1.5
    DO 30 IX=1,43
    DO 20 IY=1,71
20  L(IY,IX)=' '
30  L(1,IX)='+'
    LYS=YMAX*70./XYMAX+1.5
    DO 40 IY=1,LYS
40  L(IY,NX0)='+'
    DO 50 I=1,37
    NX=(X(I)-XMIN)*42./XYMAX+1.5
    NY=Y(I)*70./XYMAX+1.5
50  L(NY,NX)='*'
    DO 80 I=1,43
    DO 60J=1,71
60  IF(L(J,I).NE.' ')K=J
70  FORMAT(' ',71A1)
80  PRINT 70 (L(N,I),N=1,K)
90  FORMAT('1')
    PRINT 90
    RETURN
    END
```

Fig. A-1—Subroutine XYPLOT.

## References

1. Gordon, G. D., "An Introduction to FORTRAN Programming", RCA reprint RE-14-3-4.
2. Olson, H. F., *Acoustical Engineering* (D. Van Nostrand Co., Princeton, N.J., 1957) p. 316 & 331.
3. McCracken, D. D., *A Guide to* FORTRAN IV *Programming* (John Wiley & Sons, Inc., New York, N.Y.) 1965.
4. FORTRAN IV *Reference Manual*, 70-06-604, EDP Publications, RCA Cherry Hill.
5. BTSS-II *Users Manual* (This manual is a preliminary publication by Systems Programming—For information, refer to Murray Spencer, EDP, RCA Cherry Hill.)

```
ANGLE   RESPONSE   RESPONSE
 DEG.    A= .600    A= .828

  0.0   1.00000    1.00000
  5.0   0.99382    0.99412
 10.0   0.97546    0.97662
 15.0   0.94535    0.94792
 20.0   0.90427    0.90869
 25.0   0.85324    0.85986
 30.0   0.79351    0.80255
 35.0   0.72656    0.73811
 40.0   0.65403    0.66800
 45.0   0.57766    0.59381
 50.0   0.49928    0.51718
 55.0   0.42071    0.43978
 60.0   0.34375    0.36324
 65.0   0.27011    0.28913
 70.0   0.20137    0.21891
 75.0   0.13892    0.15388
 80.0   0.08396    0.09515
 85.0   0.03743    0.04363
 90.0   0.00000    0.00000
 95.0   0.02794    0.03532
100.0   0.04627    0.06216
105.0   0.05519    0.08059
110.0   0.05515    0.09093
115.0   0.04685    0.09372
120.0   0.03125    0.08972
125.0   0.00947    0.07983
130.0   0.01719    0.06513
135.0   0.04733    0.04676
140.0   0.07950    0.02596
145.0   0.11220    0.00396
150.0   0.14399    0.01801
155.0   0.17351    0.03882
160.0   0.19950    0.05742
165.0   0.22091    0.07288
170.0   0.23685    0.08448
175.0   0.24668    0.09166
180.0   0.25000    0.09409
```

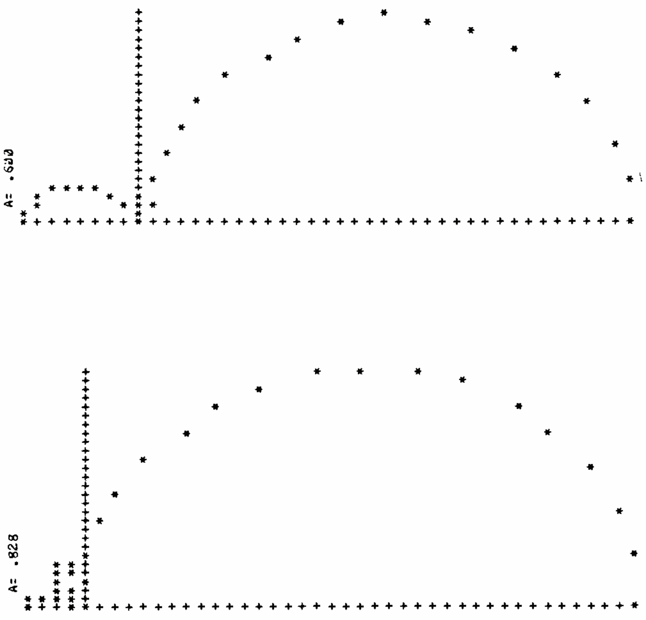


Fig. 7—Detailed directional characteristics for ranging from 0° to 180°: (a) data for $\alpha = 0.600$ and $\alpha = 0.828$; (b) curve for $\alpha = 0.600$; (c) curve for $\alpha = 0.828$. (Curves reproduced approx. ½ size).

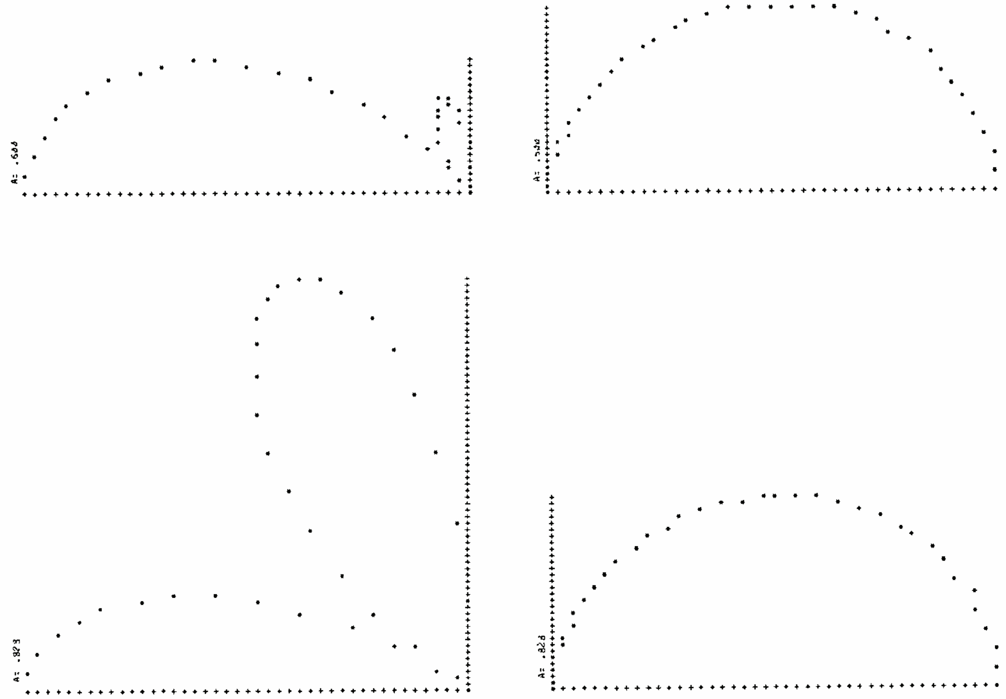


Fig. 8—First and second quadrant representations of directional characteristics: (a) $\alpha = 0.600$; (b) $\alpha = 0.828$. (Reproduced approx. ½ size).

# Computer programming—a design aid

**A. Feller**

**Although engineers should exert every effort to utilize existing computer-aided design programs, they should not automatically reject the possibility of formulating their own. By combining his own special technical skills with available "building-block" programs, the engineers can develop custom programs which will serve as valuable design tools. This paper describes the development of such custom programs for use in computer-aided circuit design; however, the principles are applicable to many other applications.**

FEW PEOPLE will question the concept that the full potential of computers should be used by engineers to help solve their various engineering problems. However, many people limit their thinking to mean that an engineer should use available programs in which he merely supplies the input data in a rather inflexible format. And of course he should . . . provided an available program meets his needs.

But what if such a program does not exist? As an example, consider what an engineer can do when an oscilloscope will not meet all of his needs. Usually, he will not have expert knowledge of every detail of the oscilloscope. However, if he has a good working knowledge of the oscilloscope and its capabilities, he will be able to build an external circuit which will extend the range of the test equipment to meet his requirements. So it is with the use of computers in design. The engineer should be able to modify or make additions to certain special purpose computer programs to increase their usefulness to him. To do this he needs to develop a familiarity with Fortran and he must not hesitate to call upon professional programmers for initial guidance.

## Special Programs Can Be Developed

If the engineer finds that no available program can be easily modified to satisfy his needs, he can revert to using programming subroutines. These are virtually complete programs which can be used as building blocks to build a larger comprehensive program. For example, if a special-purpose program is required, the circuit engineer can combine his circuit knowledge and analytical ability with existing programs that perform numerical integrations, solve simultaneous equations, and provide graphical printouts. Although this approach is not being recommended as a normal engineering procedure for the design engineer, there are situations when it should not be automatically rejected.

During the design of some P-MOS integrated circuit arrays, for example, a situation arose in which a special purpose program had to be developed. Breadboarding with discrete transistors appeared to provide only second order accuracy in predicting array performance, while designing the circuits and then waiting several months for delivery of arrays involved too much delay time. A computer-aided design program with a built-in model for the MOS devices was needed that could be used to analyze a general P-MOS circuit.

Initially, the various programs available were checked to determine if one were suitable for the P-MOS analysis. The ECAP program was rejected as unsuitable for large signal transient analysis, because it is not capable of handling nonlinear components such as voltage dependent capacitors. Other programs that were designed for transient analysis, such as Net 1 and Circus, contain built-in bipolar models and are therefore not suitable for MOS circuit analysis. A remaining possibility was a program such as Sceptre which has transient analysis capability and can accept any type of model. However, a program like this, which even if it were capable of running on the Spectra machine, and presently it is not, requires a great deal of housekeeping even for small problems. For example, compiling time may take from five to seven minutes on the IBM 7090, independent of the problem size. In addition, programs of this size discourage making any modifications to introduce capabilities and features that may be useful or even required for certain problems.

Because none of the available programs appeared to meet our requirements, the only alternative was to generate a special program. First to be determined was the overall capability required of a program of this type. Since, in general, we were interested in a range of circuit configurations, it was desirable that the program be capable of accepting the circuit in the normal topological form that the engineer uses and be capable of generating the appropriate equations. The program had to have the capability of solving these equations and performing a transient solution based on the equations.

**A. Feller**
Computer Systems Research and Applications
Advanced Technology, Camden, N.J.

received the BSEE from the University of Pennsylvania in 1951 and the MSEE from the same university in 1957. He has earned 12 credits toward a PhD in Electrical Engineering. With the RCA Broadcast Television Group since 1951, Mr. Feller worked as a design and development engineer. In 1958, Mr. Feller transferred to EDP advanced development engineering and became a member of Advanced Technology in 1963. In this capacity he developed high speed circuits and interconnection techniques for proposed new computers. Mr. Feller has been awarded three patents in high speed switching circuits. Recently, Mr. Feller has been applying design automation techniques for bipolar integrated circuit-mask artwork generation. In the area of MOS ARRAY, Mr. Feller has been involved in MOS MODEL specifications and in MOS CIRCUIT DESIGN. In this connection he has written a program for transient analysis of P, N or C MOS CIRCUITS.

---

Reprint RE-14-3-12
Final manuscript received August 26, 1968.

## Using Program Building Blocks

The generalized building blocks for the circuit analysis program used in our P-MOS design effort are shown in Fig. 1. Block A represents the various steps that an engineer will ordinarily take to define his problem, whether or not computer techniques are used. Blocks B and C are mathematical techniques not ordinarily performed manually by an engineer (except for relatively simple problems); however, the blocks can be solved by computer, using debugged programs that are generally available. The desired output of the computer is represented by block D.

Because the programming requirements for blocks B and C can be implemented by available subroutines, the engineer's primary concern is implementing blocks A and D. A natural approach would be to implement block A such that the computer program will accept any general circuit topology containing MOS active devices and generate the appropriate set of differential equations that describe the circuit. One of the many ways to do this is to mechanize in the minutest detail the individual steps that the engineer follows in writing the equations, except, in this case, the equations themselves are written in the most general terms.

The key portions of the program written for the transient analysis of MOS integrated circuits, using the basic format described in Fig. 1, are shown in Fig. 2. The lettered blocks in Fig. 2 correspond to the blocks in Fig. 1. Fig. 2 also contains a main-program block. This main program treats all the previously written programs, as well as the new ones, as subroutines and effectively integrates them into a single comprehensive program. Use of this format facilitates adding new subroutines as well as modifying the present ones.

As noted, block A, which generates the circuit equations, is the portion of the program that the engineer has to implement in detail. This block recognizes the various circuit configurations and devices and generates a set of first order differential equations describing the circuit. A detailed flowchart of this program is included in Reference 1. It shows how the general equations were formulated for the P-MOS design effort by making a detailed analysis of each node, determining the components attached to each node, and then writing a set of differential equations. The format of these equations must be compatible with the input requirements of the numerical integration routine.

The Node subroutine in Fig. 2, block B, is a complete program written by the Laboratories for the solution of a set of first order differential equations.[2] It was incorporated into the overall P-MOS program without any modification. The matrix inversion routine in block C was taken from a standard text book and used in the program without modification.[3]

## Checking Program Accuracy

Once the mechanics of assembling the program are completed, the accuracy of the program is considered. Program accuracy is usually limited by the ability of block A (in Fig. 2) to faithfully describe the circuit in two fundamental ways. First, the various active and passive devices must be characterized so that their properties can be incorporated into the differential equations. This characterization usually involves a model that contains lumped or distributed parameters, or is described by a set of mathematical equations.

The use of a model becomes increasingly difficult for large-signal nonlinear devices, especially in an integrated circuit array environment where process techniques influence the characterization of passive devices as well as active devices. For example, consider the voltage dependency of the nonlinear junction capacitances, one of the most important passive devices that must be considered in both bipolar and MOS modeling. If the junction is an abrupt one, as in the case of an alloyed junction, the capacitance varies as

$$\frac{1}{V^{1/2}},$$

where V is the net applied voltage. In modern silicon planar technology, where junctions are formed by diffusion, the exponent of V can vary considerably, although ⅓ is the most common.

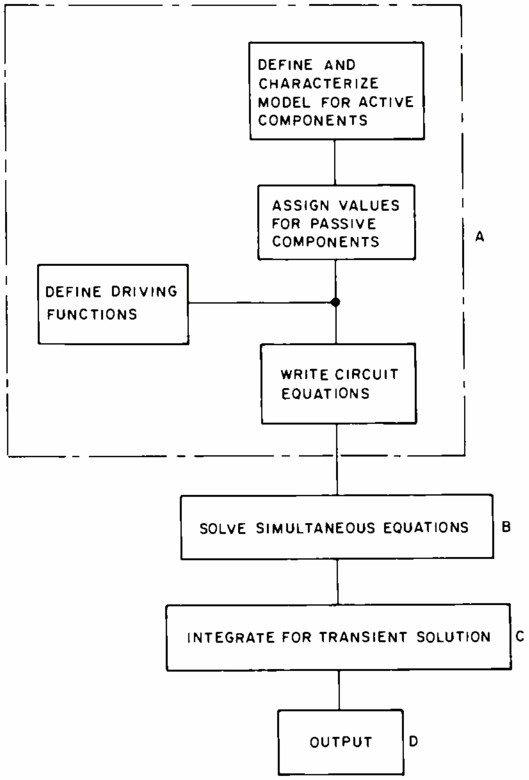


Fig. 1—Generalized building blocks for P-MOS circuit analysis program.

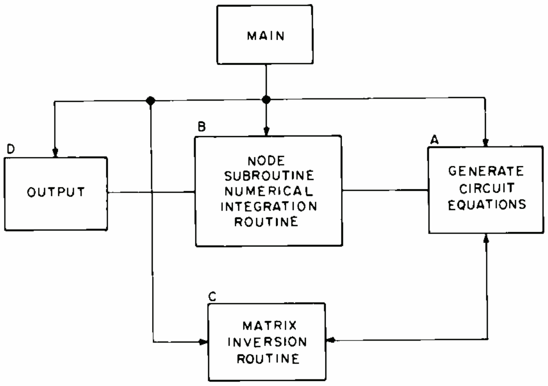


Fig. 2—Key portions of transient analysis program for P-MOS integrated circuits.

The second fundamental way in which the fidelity of the circuit representation must be ensured is in a functional sense. In developing block A, the engineer must include any interactive effects between the components or any special properties or peculiarities of the devices, components or circuits. For example, most P-type MOS devices are relatively symmetrical in their construction and have bidirectional properties. The program, and more specifically block A, must be implemented so as to detect the direction of transistor conduction and to account for the reversal of the current that results from the changing of device terminal voltages. Of course, the corresponding mathematical formulation must reflect these changing conditions.

How accurately the program and, more importantly, the models for the active and passive devices represent the physical circuit must be determined by laboratory correlation. This phase of the development provides the engineer with the opportunity of not only optimizing his representation of the physical problem, but, equally important, of improving his basic understanding of the active and passive devices. It allows him, for example, to ascertain the dependence of device performance on variations in certain characteristics. As an illustration, consider the $g_m$ (transconductance) of a MOS device, a parameter usually assumed to be a linear function of the net applied voltage. This implies that the mobility of the majority carrier is a constant, independent of the applied voltage and the resulting gate field. Laboratory observations will quickly show that the measured current does not follow this ideal relationship over a wide range of applied voltages. A series of computer runs in which variations are introduced in the model to reflect these observations will not only improve the model but also the engineer's understanding of the physical mechanism which determines device action. Even if ordinarily remote from process parameters, an engineer who is reasonably flexible in the use of the computer-aided circuit design program can gain increased insight into the functional dependence of device characteristics on such properties as dopant level by a properly selected set of runs.

## Other Advantages of Computer Usage

After the comprehensive program has been developed and verified, its scope can be extended to permit the engineer to determine the various functional relationships between individual components—active and passive—and overall circuit performance. The functional dependence of a particular performance criteria, such as propagation delay, on a single parameter of a single transistor or the variation of all the parameters in the network can be observed quickly and completely. Even with imperfect models for the active and distributed passive devices, a great deal of useful information can be observed. A wide variety of alternative design methods, which otherwise might be completely ignored because of a seemingly low probability of success, can be investigated. In short, the program can be used as an exploratory tool in research and development.

With programs that contain models whose accuracy and reliability can be specified at least over certain ranges, the resourceful engineer can extend his computer-aided design to computer-aided breadboarding. The objective here is to reduce the time and manpower spent on experimental breadboards by quickly converging on the optimum parameter values, before evaluating their performance in the laboratory. Integrated circuits, designed with the use of circuit analysis programs, have been fabricated without intermediate laboratory breadboarding and have produced performances in good agreement with the computer prediction. As an example, the Computer System Research and Application Group of Advanced Technology has recently completed a government contract for the design and layout of MOS integrated circuits. Excellent results were obtained using this computer-aided-breadboarding approach. This approach is also being used in the area of bipolar circuits, where computer programming is being used extensively to analyze analog circuits with small-signal models. In addition, large-signal models are now being developed by RCA. Another company reports the optimization of their TTL (transistor-transistor logic) computer circuits using computer-aided design techniques.[4]

The flexibility of the computer also provides the advantage of permitting problems to be attacked in ways not previously possible. For example, the effect of radiation on a circuit might be analyzed by applying the simultaneous impulse driving functions at every voltage node in the circuit and computing the response. Similarly, cooling requirements might be more precisely defined by more accurate thermal distribution analyses, using computer simulation techniques to incorporate as many local heat sources as required.

## Concluding Remarks

Whether the application is research and development, circuit analysis, device evaluation and representations, computer breadboarding, design automation, statistical analysis, or worst case analysis, the computer is a tool which all engineers should learn to use profitably. It is a tool that he can take advantage of either through the use of readily available general-purpose programs, or if necessary, through the use of special-purpose custom programs in which he incorporates his special skills and knowledge.

In order to efficiently generate these special purpose design programs, the engineer must have at least a working knowledge of Fortran. He should also seek the assistance of experienced programmers in order to save time in combining the various subroutines into an overall complete program under control of the main program. After the initial effort of setting up the main program is completed, the engineer will generally find that he can be self-sufficient in handling the remainder of the program tasks.

Finally, the amount of time involved in writing a scientific program should not be a deterrent in increasing the use of the computer-tool by engineers. The debugging phase of generating a program usually consumes the most time. Even in that phase, however, most of the time is consumed in the turnaround period, i.e., the period from the submission of the program until it has been run and returned. Further, when the program is returned, more frequently than not, only a few moments will be required to determine what the next steps in the debugging process will be. Thus, the total programming demand on the engineer's time is small, and he is able to devote most of his attention to his other assignments.

## References

1. A. Feller, private communication.
2. Private communication with R. Klopfenstein, the Laboratories.
3. S. Skuo, Numerical Methods and Computers, Addison Wesley, 1963, or C. R. Pendred, private communications.
4. P. Spiegel and P. Economopoules, "Computer Programs for Designing Large-Scale Arrays," Electro-Technology, January, 1968.

# Self-consistent regional approach to computer-aided transistor design

**Dr. R. B. Schilling**

**Present-day transistor structures operated at high currents are characterized by electric-field and charge-density profiles that are extremely complex. As a result, the validity of standard rules governing transistor design that were developed for classical diffusion theory become questionable. The regional approach to transistor design presented in this paper is an outgrowth of earlier work on insulator problems,[1,2] and is developed in terms of electron and hole concentrations and electric field. It makes use of physical as well as mathematical factors in the definition of a final design. The regional method makes possible the determination of the transistor gain $h_{FE}$ for any given transistor (doping profile specified) operating at any given values of emitter current $I_E$ and collector-to-emitter voltage $V_{CE}$. The primary advantage of the regional approach to computer-aided transistor design is that the regional approximations used give rise to a set of equations of the first order. First-order equations can be solved within the limited storage of the time-sharing system. Because the time-sharing system can be used, the design equations can be utilized simultaneously by a number of design groups. Details of the computer program used with the regional approach are described elsewhere in this issue.[3]**

**Dr. Ronald B. Schilling, Ldr.**
Microwave Module Design
Electronic Components
Lancaster, Pa.

received the BS in Electrical Engineering from the City College of New York in 1961. The same year, he joined RCA Laboratories to work in the areas of semiconductor-device theory and fabrication. He attended Princeton University under the RCA Laboratories Graduate Study Program and received the MS in Electrical Engineering in 1963. Under the RCA Laboratories Doctoral Study Award, he attended the Polytechnic Institute of Brooklyn and received the PhD in Electrical Engineering in 1966. In December of 1966, he transferred to the Technical Programs Laboratory of Electronic Components in Somerville as a Project Engineer involved in computer-aided transistor design and semiconductor-device design and fabrication. Dr. Schilling has lectured at Brooklyn Polytechnic Institute, The City College of New York, and in the RCA Somerville CCSE and CEE programs. He is author or coauthor of 6 published papers and of a chapter in Vol. 8 of *Semiconductors and Semimetals*. He has presented 5 papers, served on a panel on "Computer-Aided Bipolar Design," and is co-chairman of the 1969 IEEE-NUTSAC seminar on "Computer-Aided Analysis and Design of Semiconductor Devices." He is a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and IEEE, and is active in the Basic Sciences Committee of NUTSAC.

THE GENERALIZED REGIONAL APPROACH used in this paper extends the classical theory of Shockley[4] and Webster[5] to include high current effects. In addition, the regional approach allows for base widening and provides the means of determining charge density, electric field, and voltage as functions of position within a one-dimensional transistor structure such as that shown in Fig. 1. The regional approach described is governed by the self-consistent use of physical approximations. An example of a physical approximation is charge neutrality in the base region. Although this approximation is in general use in transistor theory, its validity was not adequately demonstrated until the development of the regional approach. The technique explained below shows that charge neutrality controls the extent of the base width during base widening.

## Mathematical Formulation of the Regional Approach

### Base Region

The boundaries of the regions within a transistor and the associated parameters are identified in Fig. 1. As shown in the Figure, the base region extends from the edge of the emitter at $x = 0$ to the boundary of the transition region at $x = X_B$. The position of $X_B$ is determined from charge-neutrality conditions in the base. Poisson's equation relating the net charge to its components is given by

$$\frac{\epsilon}{q}\frac{dE}{dx} = p - n - N \qquad (1)$$

where $\epsilon$ is the dielectric constant, $q$ is the electronic charge, $E$ is the electric field, $x$ is position, $p$ is hole density, $n$ is electron density, and $N$ is the doping profile.

Rearranging Eq. 1, a charge neutrality factor $R_t$ is given by

$$R_t \equiv \frac{\frac{\epsilon}{q}\frac{dE}{dx}}{p} = 1 - \frac{(n+N)}{p} \qquad (2)$$

When $|R_t| << 1$, the *net* charge at a given position in the device is small compared to either the positive or negative charge and therefore charge neutrality given by

$$p \simeq n + N \qquad (3)$$

is a reasonable approximation.

The value of $x$ at which $R_t$ is no longer sufficiently less than 1 is designated $X_B$. Determination of the effective base $0 \leqq x \leqq X_B$, is therefore arbitrary. It has been found that values of $R_t$ between 0.1 and 1 result in negligible variation in the position of the effective base and in the resulting voltage $V_{CE}$ because the electric field increases sharply with the position at the junction of the base and the transistion region. The fact that $X_B$ and $V_{CE}$ are not sensitive functions of $R_t$ lends credence to the regional approach because it demonstrates that physical rather than mathematical processes are governing the design. In the work that follows, the effective base was determined with $|R_t|$ equal to 0.5.

In analyzing an N-P-N transistor, the net hole current can be taken as zero[4,5,6]

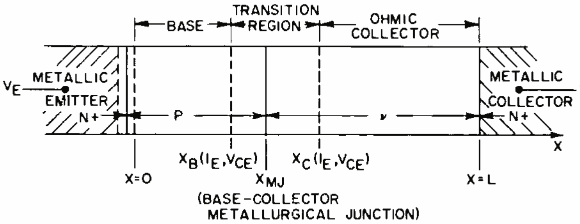


Fig. 1—One-dimensional transistor structure showing regions and parameters.

Reprint RE-14-3-16
Final manuscript received September 4, 1968.

# Self-consistent regional approach to computer-aided transistor design

**Dr. R. B. Schilling**

**Present-day transistor structures operated at high currents are characterized by electric-field and charge-density profiles that are extremely complex. As a result, the validity of standard rules governing transistor design that were developed for classical diffusion theory become questionable. The regional approach to transistor design presented in this paper is an outgrowth of earlier work on insulator problems,[1,2] and is developed in terms of electron and hole concentrations and electric field. It makes use of physical as well as mathematical factors in the definition of a final design. The regional method makes possible the determination of the transistor gain $h_{FE}$ for any given transistor (doping profile specified) operating at any given values of emitter current $I_E$ and collector-to-emitter voltage $V_{CE}$. The primary advantage of the regional approach to computer-aided transistor design is that the regional approximations used give rise to a set of equations of the first order. First-order equations can be solved within the limited storage of the time-sharing system. Because the time-sharing system can be used, the design equations can be utilized simultaneously by a number of design groups. Details of the computer program used with the regional approach are described elsewhere in this issue.[3]**

**Dr. Ronald B. Schilling, Ldr.**
Microwave Module Design
Electronic Components
Lancaster, Pa.

received the BS in Electrical Engineering from the City College of New York in 1961. The same year, he joined RCA Laboratories to work in the areas of semiconductor-device theory and fabrication. He attended Princeton University under the RCA Laboratories Graduate Study Program and received the MS in Electrical Engineering in 1963. Under the RCA Laboratories Doctoral Study Award, he attended the Polytechnic Institute of Brooklyn and received the PhD in Electrical Engineering in 1966. In December of 1966, he transferred to the Technical Programs Laboratory of Electronic Components in Somerville as a Project Engineer involved in computer-aided transistor design and semiconductor-device design and fabrication. Dr. Schilling has lectured at Brooklyn Polytechnic Institute, The City College of New York, and in the RCA Somerville CCSE and CEE programs. He is author or co-author of 6 published papers and of a chapter in Vol. 8 of *Semiconductors and Semimetals*. He has presented 5 papers, served on a panel on "Computer-Aided Bipolar Design," and is co-chairman of the 1969 IEEE-NUTSAC seminar on "Computer-Aided Analysis and Design of Semiconductor Devices." He is a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and IEEE, and is active in the Basic Sciences Committee of NUTSAC.

THE GENERALIZED REGIONAL APPROACH used in this paper extends the classical theory of Shockley[4] and Webster[5] to include high current effects. In addition, the regional approach allows for base widening and provides the means of determining charge density, electric field, and voltage as functions of position within a one-dimensional transistor structure such as that shown in Fig. 1. The regional approach described is governed by the self-consistent use of physical approximations. An example of a physical approximation is charge neutrality in the base region. Although this approximation is in general use in transistor theory, its validity was not adequately demonstrated until the development of the regional approach. The technique explained below shows that charge neutrality controls the extent of the base width during base widening.

## Mathematical Formulation of the Regional Approach

### Base Region

The boundaries of the regions within a transistor and the associated parameters are identified in Fig. 1. As shown in the Figure, the base region extends from the edge of the emitter at $x = 0$ to the boundary of the transition region at $x = X_B$. The position of $X_B$ is determined from charge-neutrality conditions in the base. Poisson's equation relating the net charge to its components is given by

$$\frac{\epsilon}{q}\frac{dE}{dx} = p - n - N \qquad (1)$$

where $\epsilon$ is the dielectric constant, $q$ is the electronic charge, $E$ is the electric field, $x$ is position, $p$ is hole density, $n$ is electron density, and $N$ is the doping profile.

Rearranging Eq. 1, a charge neutrality factor $R_1$ is given by

$$R_1 \equiv \frac{\frac{\epsilon}{q}\frac{dE}{dx}}{p} = 1 - \frac{(n+N)}{p} \qquad (2)$$

When $|R_1| << 1$, the *net* charge at a given position in the device is small compared to either the positive or negative charge and therefore charge neutrality given by

$$p \simeq n + N \qquad (3)$$

is a reasonable approximation.

The value of $x$ at which $R_1$ is no longer sufficiently less than 1 is designated $X_B$. Determination of the effective base $0 \leqq x \leqq X_B$, is therefore arbitrary. It has been found that values of $R_1$ between 0.1 and 1 result in negligible variation in the position of the effective base and in the resulting voltage $V_{CE}$ because the electric field increases sharply with the position at the junction of the base and the transistion region. The fact that $X_B$ and $V_{CE}$ are not sensitive functions of $R_1$ lends credence to the regional approach because it demonstrates that physical rather than mathematical processes are governing the design. In the work that follows, the effective base was determined with $|R_1|$ equal to 0.5.

In analyzing an N-P-N transistor, the net hole current can be taken as zero[4,5,6]

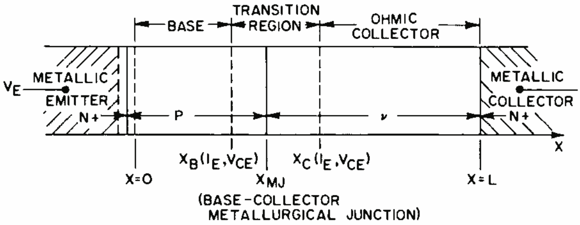


Fig. 1—One-dimensional transistor structure showing regions and parameters.

Reprint RE-14-3-16
Final manuscript received September 4, 1968.

because of the high injection efficiency, high transport factor and the action of the collector-base junction in keeping holes from leaving the base. The general equation for hole-current density $J_p$ is given by

$$J_p = q\mu_p pE - q\,D_p\,\frac{dp}{dx} \qquad (4)$$

where $\mu_p$ is hole mobility and $D_p$ is the hole diffusion coefficient. This equation can be modified on the basis of the above considerations to obtain

$$J_p << q\mu_p pE \simeq q\,D_p\,\frac{dp}{dx} \qquad (5)$$

Using Eq. 5 the electric-field equation can be written as follows:

$$E = V_T\,\frac{dp/dx}{p} \qquad (6)$$

where $V_T = D_p/\mu_p$

The electron current density is given by

$$J_n = q\mu_n nE + q\,D_n\,dn/dx \qquad (7)$$

where $D_n$ is the electron diffusion coefficient, $\mu_n$ is electron mobility, and where $D_n/\mu_n = D_p/\mu_p = V_T$. Because $J_p << J_n$, $J$, the total current, is approximately equal to the electron current, $J_n$ ($J = J_n + J_p = I_E/A_{emitter}$ where $A$ is area). This assumes no current crowding.

In the base region, a first order nonlinear equation for the hole density $p$ is given by

$$\frac{dp}{dx} = \frac{p[J_n/q\,D_n + dN/dx]}{2\,p - N} \qquad (8)$$

Eq. 8 is obtained by substitution of Eqs. 3 and 6 into 7. Eq. 8 is an Abel equation of the second kind and is not analytically tractable for other than homogeneous background densities ($N$ is a constant).[7] The equation is utilized in the base region, $0 \leq x \leq X_B$, and is constrained by the value of the hole density at $x = 0$, that is

$$p(0) = n(0) + N(0) \qquad (9)$$

Because $N(0)$ is determined from the doping profile, the value for hole density given by Eq. 9 is largely dependent on $n(0)$; Eq. 8 and the gain $h_{FE}$ are also critically dependent on $n(0)$ and, in addition, $J_n$. The quantities $n(0)$ and $J_n$ are key quantities in transistor operation. The value of hole density $p$, found from Eq. 8, is used in Eq. 6 to find the electric field, $E$, and in Eq. 3 to the electron density, $n$.

### Transition Region

The transition region extends from $x = X_B$ to $x = X_C$, where $X_C$ is the boundary of the ohmic collector. In the ohmic collector, the electron density is equal to the background doping. In this paper, the boundary $X_C$ is defined as that point at which $n = 0.99N$.

It is assumed that in the transition region, the drift current outweighs the diffusion current. Therefore, in the transition region, Eq. 7 becomes

$$J_n \simeq nq\mu_n E \qquad (10)$$

A self-consistent check showing that drift current is very much greater than diffusion current consists of monitoring the following ratio:

$$\frac{nq\mu_n E}{q\,D_n\,dn/dx} = \frac{nE}{V_T\,dn/dx} \equiv R_2 \qquad (11)$$

The approximation used in this paper is $R_2 \geq 10$. However, the quantity $R_2$ can in some instances be of the order of 0.1 at $X_C$. With a value of 0.1 for $R_2$, the approximation given by Eq. 10 is not self-consistent at point $X_C$ and a more useful self-consistency criterion must be developed. This criterion is found by determining the percentage distance, starting at $X_C$, over which the approximation $R_2 \geq 10$ is invalid. Percentage distance is defined as $100(X_{crit} - X_C)/(X_L - X_C)$ where $X_{crit}$ is located at $R_2 = 10$. If this percentage distance is less than 5 it can be assumed that the approximation made is valid wherever used. For this analysis, percentage distances of less than 2 were typical. The above self consistency criterion has been shown to be satisfactory for space-charge-limited-current problems.[8]

In the transition region, the differential equation for the electric field is given by

$$\frac{dE}{dx} = -\frac{q}{\varepsilon}\,N - \frac{J_n}{\varepsilon\mu_n E} \qquad (12)$$

Eq. 12 is obtained from Eq. 1 with $p = 0$ and Eq. 10. It can be shown that $p$ is negligible in Eq. 1 within the transition and ohmic regions. Eq. 12, like Eq. 8, is an Abel equation of the second kind and requires computer solution. The value of electric field, $E$, found from the solution of Eq. 12 is used in Eq. 10 to find electron density, $n$.

### Ohmic Region

The ohmic region extends from $x = X_C$ to $x = L$, the position of the metallic collector. Because $n \simeq N$ in this region, Ohm's law is used.

## Computational Procedure

When a specific doping profile has been chosen (e.g., ERFC, Gaussian, exponential) and an emitter area, $A$, has been specified, operation at a given $I_E$ and $V_{CE}$ is assured by use of an electron current density $J_n$ equal to $I_E/A_{emitter}$; the value of the electron density at the edge of the base, $n(0)$, is chosen arbitrarily and later adjusted for the desired $V_{CE}$. As described in the section on mathematical formulation, $J_n$ and $n$ (0) are key parameters in obtaining the output variables; for a given $J_n$, the voltage $V_{CE}$ is uniquely determined by $n(0)$.

Computer solution of the base-region equations subject to $J_n$ and $n(0)$ yields the values of $n$, $p$, $R_1$, $E$, and $V$ as functions of position within the transistor structure. When the charge-neutrality factor $R_1$ becomes equal to 0.5, the boundary $X_B$ separating the effective base region from the transition region is reached and the base region portion of the program is complete. The boundary values at $X_B$ which are used for solving the transition-region equations are $E(X_B)$ and $X_B$.

The transition-region equations are next solved from $x = X_B$ to $x = L$. In certain cases it is useful to end the transition-region program at $X_C$, the start of the ohmic region, and to specify a resistive contribution to the electric field from $X_C$ to $L$. The transition-region computation yields $n$, $R_2$, $E$ and $V$ as functions of position within this region.

The total voltage $V_{CE}$ is obtained by use of the following equation:

$$V_{CE} = \int_0^L |E|\,dx - \int_0^L |E_0|\,dx + V^j{}_{BE} \qquad (13)$$

where $E_0$ is the electric field under equilibrium conditions and $V^j{}_{BE}$ represents the terminal voltage across the emitter-base junction (i.e., from the emitter edge of the base at $x = 0$ to the emitter). Absolute value signs are used because the bulk field between $x = 0$ and $x = L$ is directed toward $x = 0$. If the value of total voltage obtained is not close enough to the desired operating voltage, a different value for $n(0)$ is chosen and the procedure is repeated. Iteration is only required when operation must take place at a specific voltage. For example, if a comparison of the current-handling capabilities ($h_{FE}$ specified at a given current) of several devices is desired, iteration is not required. Only $h_{FE}$ [which determines $n(0)$] and $J_n$ need be specified; the device requiring the lowest $V_{CE}$ would then be the best device.

## Practical Application of the Regional Method

Several types of design problems have been solved by using the regional method on the time-sharing system. A typical example is a comparison of several structures with different doping profiles to determine which structure is most capable of preventing base widening and therefore fall off of current gain $h_{FE}$ at a given $I_E$ — $V_{CE}$ operating point. Details of the computations used for design purposes are given below, first for fixed current and varying voltage and then for fixed voltage and varying current.

### Fixed Current and Varying Voltage

For purposes of illustration, it is assumed that a high voltage N-P-N transistor with a doping profile such as that shown in Fig. 2 has a fixed emitter current $I_E$ of 200 mA and operates at a voltage $V_{CE}$ ranging from 1.5 to 10 volts. The area of this device is 8000 square mils so that the operating current of 200 mA corresponds to a current density $J_n$ of 4 A/cm². This value is generally considered to be a low value of current density for power transistors. However, even at this low current density, there is significant base widening when the operating voltage is low enough. Computed data of electric field as a function of position are shown in Fig. 3. All of the curves have approximately the same value of electric field at $x = 0$ (the base emitter edge of the base region). The electric field at $x = 0$ is a "built in" field associated with the background doping profile. The strength of the field is found from the approximate prevailing conditions when hole current is zero under low-injection operation. Using Eq. 3 and $n < p$

$$p = n + N \simeq N \qquad (14)$$

The doping profile is approximated by the following exponential equation derived from Fig. 2.

$$N(x) = 5 \times 10^{16} \exp(-x/\lambda) - 6.5 \times 10^{13} [1 - \exp(-x/\lambda)] \qquad (15)$$

where $\lambda = 1.1 \times 10^{-4}$ cm. The electric field $E$ at $x = 0$ is then found from Eqs. 6, 14, and 15 to be

$$E(0) = \frac{-V_T}{\lambda} = \frac{0.026}{1.1 \times 10^{-4}} = -225 \text{ volts/cm} \qquad (16)$$

The constant electric field on the right side of each curve in Fig. 2 represents the ohmic region of the collector. In the ohmic region, $n = |N|$, and $J_n$ from Eq. 10 is given by

$$J_n = Nq\mu_n E = \sigma E \qquad (17)$$

where $\sigma$ is the conductivity. Therefore,

$$E = \frac{J_n}{\sigma} \qquad (18)$$

which is constant for a given current.

At a fixed current, the length of the ohmic region is critically affected by the operating voltage; as the voltage decreases, the base edge of the ohmic region moves toward the collector.

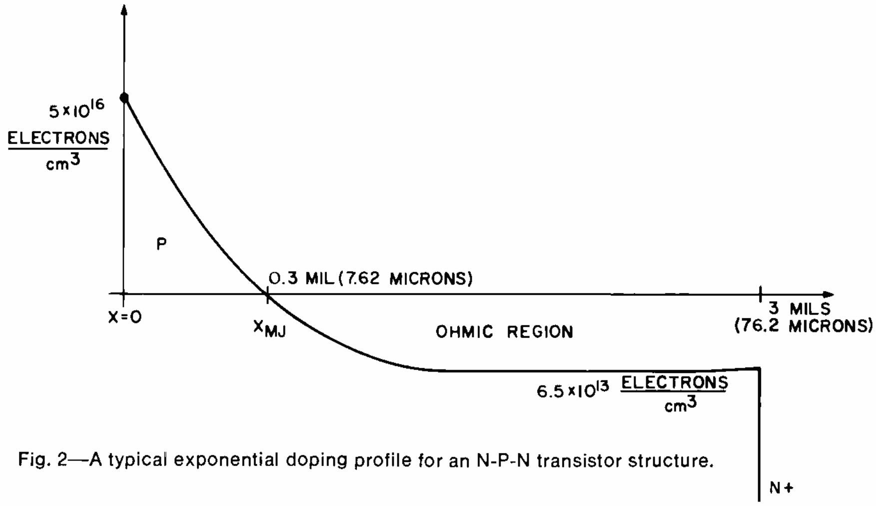


Fig. 2—A typical exponential doping profile for an N-P-N transistor structure.

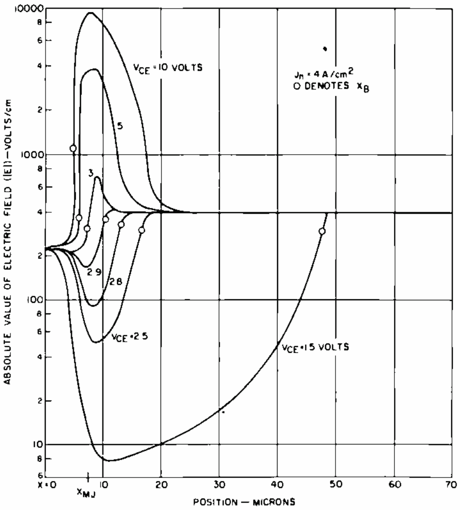


Fig. 3—Electric field as a function of position at fixed current and variable voltage.

Shrinkage of the ohmic region occurs because low values of voltage cause a weak sink condition for electrons in the region near the metallurgical junction. [A sink is a condition caused by the presence of a high voltage and thus a high electric field which forces electrons away from a given region, not allowing their build up.] Eventually, as the voltage continues to drop, the base widens into the N⁺ collector.

The effective base was arbitrarily terminated where $R_t$ equals 0.5. Actually $R_t$ could be chosen between 0.1 and 1 with negligible effect on both the shape of the curves shown in Fig. 3 and the computed values of the defect factors. This negligible effect is the result of two factors. First, $R_t$ increases rapidly with position in the range from 0.1 to 1. Therefore, variation in $X_B$ for different values of $R_t$ is small. Second, the electric-field at the end of the base matches closely the electric-field at the beginning of the collector. Any slight shift in the point joining the two solutions would therefore result in negligible change in the shape of the curve. Insensitivity of the solution to the value of the $R_t$ factors is a significant benefit in the regional approach.[9]

The shapes of the curves in Fig. 3 are noteworthy. For $V_{CE} = 10$ volts, the effective base extends from $x = 0$ to $X_B = 5$ microns. The electric field then rises sharply, peaking at approximately the metallurgical junction $X_{MJ}$. The electric field then decreases, reaching the value of the ohmic field at approximately 24 microns. This type of behavior (i.e., high fields peaking at $X_{MJ}$) is referred to as high-voltage or low-current behavior. It is important to emphasize that the 10-volt curve in Fig. 3 is a high-voltage or low-current curve only because of its shape. The name refers only to the relative values, i.e., 10 volts is high voltage at 4 A/cm$^2$ but may be considered low voltage at 40 A/cm$^2$. High or low voltage therefore depends on the current density and vice versa.

At a $V_{CE}$ of 5 volts, the base edge shifts to 6 microns. The position of the peak field also shifts from $X_{MJ} = 7.62$ microns to about 8.7 microns. A further decrease in $V_{CE}$ to 3 volts produces a marked variation in the electric-field profile and the base edge moves to 7.2 microns, a point close to $X_{MJ}$. The significant change in the field profile is in its slope at $x = 0$, $dE/dx(0)$. At 3 volts, $dE/dx(0)$ signifies a net positive charge; for $V_{CE} = 5$ and 10 volts, the net charge at $x = 0$ is negative. A net positive charge front therefore develops at the origin as the base pushes toward the collector. For $V_{CE} = 5$ volts, the net charge goes from negative to positive, being equal to zero at $X_{MJ}$. For $V_{CE} = 2.8$ to 3 volts, the net charge is positive-negative-positive, and for $V_{CE} = 2.5$ volts it is positive-negative. As voltage decreases, a positive front originates at the origin while a second positive front merges with the collector.

It is significant that 0.1-volt change in $V_{CE}$ from 3 to 2.9 volts shifts $X_B$ by 3 microns in the vicinity of $X_{MJ}$, whereas a 2-volt change from 5 to 3 volts is required to produce a 1-micron shift to the left of $X_B$. The increased rate in base widening at and beyond $X_{MJ}$ indicates that base widening becomes significant when the effective base crosses the base-collector metallurgical junction.

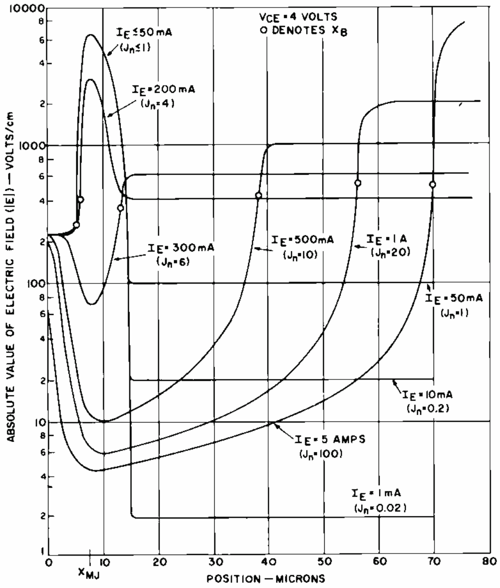


Fig. 4—Electric field as a function of position at fixed voltage and variable current.

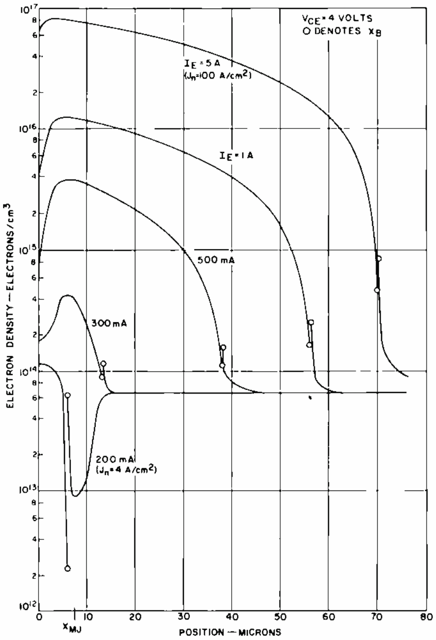


Fig. 5—Electron density as a function of position at fixed voltage and variable current.

### Fixed Voltage and Varying Current

Fig. 4 shows the curves for electric field as a function of position at a fixed voltage of 4 volts with emitter current varying from 200 mA (4 A/cm$^2$) to 5 amperes (100 A/cm$^2$).

As shown in Fig. 4, the electric field at $x = 0$ starts to depart from the built-in field at approximately 1 ampere, (20 A/cm$^2$). The 4-volt, 1-ampere curve corresponds to an electron density at the base edge, $n(0)$, of 4x10$^{15}$, which is roughly an order of magnitude below the background doping level, $N(0) = 5\text{x}10^{16}$. At 4 volts and 5 amperes, $n(0) = 6.7\text{x}10^{16}$ and $E(0)$ decreases from a value of 250 volts/cm at 1 ampere to a value of 65 volts/cm.

The electric field in the ohmic region as a function of current can be determined by use of Eq. 18. The values derived should agree with those shown in Fig. 4.

At high currents and with $V_{CE} = 4$ volts, the base widens towards the collector. The large dip in the electric-field profile occurs because the area under the curve is kept approximately constant ($V_{CE} = 4$ volts) as the right-hand portion of the curve (in the ohmic region) rises with increasing current. At 5 amperes, the ohmic region has almost disappeared. At this current level the base has widened almost to the end of the structure, 76.2 microns.

The electron density as a function of position is shown in Fig. 5 for $V_{CE} = 4$ volts and emitter currents ranging from 200 mA to 5 A. At 200 mA, the electron density decreases monotonically and approaches very low values close to the metallurgical junction. Under this condition, the effective base, $X_B$, is chosen where the electron concentration goes to zero (actually just before $n = 0$). Zero electron density corresponds to a perfect sink condition. When $X_B$ is chosen where $n = 0$, the result is a sharp discontinuity in $n(X_B)$, as shown in the 200-mA curve of Fig. 5. A discontinuity in $n(x)$ results at $X_B$ for all current values because of the assumption that the electric field is continuous at $X_B$. The discontinuity is large only under perfect sink conditions; i.e., when $n = 0$.

For currents of 300 mA to 5A and with $V_{CE} = 4$ volts, the electron density has a positive slope at the origin. For each curve, the electron density reaches a peak and then monotonically decreases toward the value of the electron concentration in the ohmic collector, 6.5x10$^3$ electron/cm$^3$. The peak position shifts towards the origin with increasing current. When the electron concentration at the origin has a positive slope, drift and diffusion currents oppose each other. At the peak, only drift current is present; beyond the peak, both drift and diffusion currents are in the same direction.

Base widening is clearly evident in Fig. 5 as $X_B$ moves closer to $X_{MJ}$ with increasing current. The movement is due chiefly to the large increase in total electron density with increasing current, and the lack of a sufficient sink ($V_{CE}$ of only 4 volts).

### Comparison of Theoretical and Experimental Results

As an initial test of the validity of the regional approach, a comparison of computed and experimentally obtained values of current gain $h_{FE}$ as a function of current, for the device shown schematically in Fig. 2, was performed. Fall off of $h_{FE}$ at high currents (where surface effects can be neglected) was believed to be caused mainly by inadequate injection efficiency rather than loss of carriers through base transport. A study of the computed value of base transit time and injection efficiency defect factor as a function of current, substantiated this premise.

The computed values of transit time as a function of electron current density, at $V_{CE} = 4$ volts, are shown in

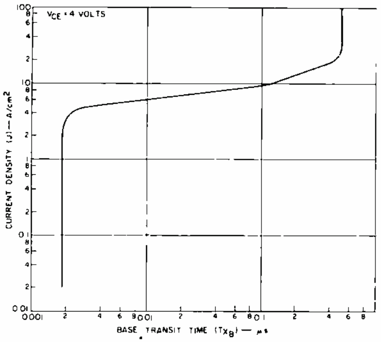


Fig. 6—Base transit time as a function of current density.

Fig. 6. For currents greater than 1 ampere ($J_n = 20$ A/cm²), the transit time reaches a constant value. Any variation in $h_{FE}$ beyond 1 ampere will, therefore, not be caused by transit time, or consequently, base transport factor if it is assumed that lifetime is approximately independent of injection level. Comparison of the computed transit-time data shown in Fig. 6 for any operating point with the position of the effective base at the same operating point as shown in Fig. 4 indicates that the transit time reaches a fixed value, approximately 0.5 microseconds, as the edge of the effective base approaches the boundary of the ohmic collector. The transit time for currents less than 100 mA ($J_n = 2$A/cm²) is independent of current and fixed at approximately 18 nanoseconds. As shown in Fig 4, the end of the base is fixed at approximately 5 microns under low-current operation. The figure also shows that, as the current increases from 200 to 300 mA, the base starts to widen rapidly. As expected, the transit time increases sharply in this current range; the sharp increase can be noted in Fig. 6.

Computed values of emitter-efficiency defect factor as a function of current at $V_{CE} = 4$ volts, are shown in Fig. 7. The scaling factor

$$K \equiv q\, D_{pe}/L_{pe} n_{eo} \qquad (19)$$

where $D_{pe}$ is emitter hole-diffusion coefficient; $L_{pe}$ is hole diffusion length in the emitter; and $n_{eo}$ is the equlibrium value of emitter electron density. $K$ is chosen so that there is a reasonable fit between measured and computed data. Thus, the values for defect factor are arbitrary. The equation for emitter-efficiency defect factor is given by

$$\delta_E = \frac{J_p}{J_n} = \left\{ \frac{qD_{pe}\, n(0)\, [n(0) + N(0)]}{L_{pe}\, n_{eo}} \right\} \frac{1}{J_n} = \frac{K\, n(0) [n(0) + N(0)]}{J_n} \qquad (20)$$

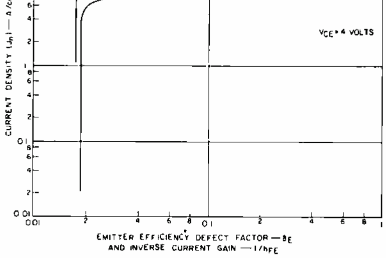


Fig. 7—Emitter-efficiency defect factor.

Measured values of inverse current gain $1/h_{FE}$ as a function of current density are also shown in Fig. 7. The inverse current gain can be computed under conditions of high current fall-off (when surface recombination effects can be ignored) in terms of the injection-efficiency defect factor $\delta_E$, and the base-transport defect factor $\delta_B$, as follows:

$$\frac{1}{h_{FE}} = \frac{\delta_E + \delta_B}{1 - \delta_B} \qquad (21)$$

Because $1/h_{FE} \leq 0.2$ in Fig. 7 and because the transit time clearly has little influence on $1/h_{FE}$ at high currents, as shown in a comparison of Figs. 6 and 7, $\delta_B << 1$ and can be neglected in the denominator of Eq. 21; thus the equation can be simplified to the following form:

$$\frac{1}{h_{FE}} \simeq \delta_E + \delta_B \qquad (22)$$

Eq. 22 re-emphasizes the point that the high current fall-off in the device shown in Fig. 2 is governed by the emitter-base junction injection efficiency and that the base transport factor has little affect.

The close agreement of the slopes of the curves shown in Fig. 7 demonstrates that the regional approach can adequately predict high current fall-off. Fitting of theoretical and experimental data of $h_{FE}$ as a function of current over the complete current range is presently being studied. This work is complicated by the fact that the expressions for surface recombination velocity and lifetime are complex[10,11,12] and unknown functions of the carrier densities. In addition, effects of generation-recombination currents in space-charge regions must be considered[11,12]. However, these effects play only a minor role under high-current, base-widening conditions.

Transistor designs have been performed by carrying out the above analyses for different doping profiles on the time-sharing system. More complex design problems, however, such as optimization of doping profiles through the use of iteration schemes, require a computer storage capability not available on the time-sharing system.

Present efforts in computer-aided design using the approach described include development of a two-dimensional analytical system for studying both base widening and current crowding (which allows for the study of thermal effects and second breakdown), iteration schemes for optimization of doping profiles when operation at a given point is desired, and the study of multilayer structures (e.g., $N^+ PP^- N^- N^+$) for achieving both high-voltage breakdown and high gain under base-widening conditions.

## Acknowledgment

The author thanks Prof. A. Lampert, Dr. F. Heiman, Dr. C. Meuller, J. Olmstead, W. Einthoven, and R. Denning for their helpful discussions; Dr. F. Schneider for his assistance in operating the SNODE routine on BTSS; S. Ponczak for his significant contributions to the computations; and Dr. A. Blicher and Dr. R. Janes for their interest in this work.

## References

1. Patrick , L., *J. Appl. Phys.* Vol. 28 (1957) p. 765.
2. *Report on Progress in Physics.* Vol. 27 Institute of Physics and the Physical Society (1964) p. 329.
3. Gordon, G. D., "Differential Equations and Integration with BTSS," *Reprint RE-14-3-18.*
4. Shockley, W., *Electrons and Holes in Semiconductors* (D. Van Nostrand Co., Princeton, N.J. 1950).
5. Webster, W. M., "On the Variation of Junction-Transistor Current-Amplification Factor with Emitter Current," *Proc. IRE* (Jun 1954) p. 914-920.
6. Lindmayer, J. and Wrigley, C., "The High-Injection-Level Operation of Drift Transistors," *Solid-State Electronics* Vol. 2 (1961) p. 79-84.
7. Murphy, G. M., *Ordinary Differential Equations* (D. Van Nostrand Co., Princeton, N.J. 1960).
8. Schilling, R. B., and Schachter, H., "Neglecting Diffusion in Space-Charge-Limited Currents," *J. Appl. Phys.* Vol. 38, No. 2 (Feb 1967) p. 841-844.
9. Lambert, M. A. and Schilling, R. B., "Injection Currents in Insulators," Chapter 8 of *Physics of Insulators and Semimetals,* to be published.
10. Shockley, W. and Read, W. T., "Statistics of the Recombination of Holes and Electrons," *Physical Review,* Vol. 87 (Sept 1952) p. 835-842.
11. Grove, A. S., *Physics and Technology of Semiconductor Devices* (John Wiley and Sons, New York, N.Y., 1967) pp. 136-140, 172-191, 298-305.
12. Hauser, J. R., *Bipolar and Unipolar Transistors* (Prentice Hall, Englewood Cliffs, N.J., 1968).

# Differential equations and integration with BTSS

**Dr. G. D. Gordon**

**The RCA Basic Time Sharing System (BTSS) contains a stored program to solve differential equations, known as SNODE. The objective of this article is to teach the reader to use SNODE to compute any definite integral or to solve any set of simultaneous first-order differential equations. The details of how these calculations are performed by the computer are not included. Three detailed examples demonstrate how to integrate, how to solve a single first-order differential equation, and how to solve a set of three simultaneous first-order differential equations. Familiarity with these programs, plus a knowledge of numerical analysis, will enable an engineer to use SNODE to solve multiple integrals, higher order differential equations, and partial differential equations.**

FOR SOLVING DIFFERENTIAL EQUATIONS, there is a useful computer program, known as SNODE, (Spectra Numerical Ordinary Differential Equations). This program can be run on the Spectra 70/35, 70/45, or 70/55 computer and can be used in performing integrations or solving differential equations. The SNODE program is particularly convenient for users of the RCA Basic Time-Sharing System, because it is stored in the computer and can be called from any FORTRAN PI compiled program.

An engineer can write his own program to solve a differential equation without using the SNODE subroutine. However, the advantages of using SNODE are:

1) Computer memory space is saved,
2) Programming time is saved, and
3) Powerful numerical analysis techniques are used to save computing time and yield more accurate results.

The format of SNODE is for the solution of a set of simultaneous first-order differential equations. However, this can be simplified to evaluate a simple definite integral or to solve a single first-order differential equation; on the other hand, SNODE can be extended to perform multiple integration, solve higher-order differential equations, and even partial differential equations.

## Integration

### Basic Principles

Before getting into the details of SNODE, let us examine the few basic steps necessary to evaluate any integral by computer. The function to be integrated is shown in Fig. 1, and the integration corresponds to calculating the area under the curve. After some integration the area under the curve on the left, $Y$, has been calculated. We calculate the area of a narrow rectangle (in this case 0.05 wide), with a height equal to the value of the function $YP$. We add the area of this strip to $Y$, then we increment the value of $X$, and calculate the area of the next strip. These basic steps are illustrated in the FORTRAN program on the left of Fig. 1:

1) The initial area $Y$ is set to zero, and an initial value of $X$ is set (*Steps 10 and 20*).
2) The value of the function, $YP$, is calculated (*Step 30*).
3) For each value of $X$, the area of a strip is calculated and is added to the total area (*Step 40*).
4) The independent variable, $X$, is incremented (*Step 50*).
5) Steps 30, 40, and 50 are repeated (*Step 60*).

In more sophisticated programs, such as SNODE, instead of a simple rectangle, a curve is fitted to the function so that the top of the strip closely approximates the curve. The efficiency of the calculation is also improved by adjusting the step size to the maximum permissible for a specified accuracy.

### Constants Needed by SNODE

Now let's see how this integral would be calculated using the SNODE program. The general format of variables used in SNODE for a simple integration is shown in Fig. 2. The function to be integrated is $YP$, which is a function of $X$. The limits of integration are from an initial $X$ to a final $P(6)$; this means that the initial value of $X$ must be stored in location $X$ and that the final value of $X$ must be stored in location $P(6)$. ($P$ is a subscripted array described later).

An initial STEP SIZE must be specified for the initial calculations, and this is stored in location $P(2)$. This choice is not critical, because the step size is adjusted by the program to maintain the desired accuracy. The $P(3)$ location stores the number of significant figures desired for the calculations; initially, a rough estimate may be desired, two or three significant figures may be designated, and the computing time will be short. After the program has been debugged, and a rough value is known, the accuracy can be improved by increasing the number in $P(3)$.

A few other constants must be set, and these would be the same for any definite integral: $P(1)$ and $P(5)$ are set to unity, and $Y$, the initial value of the integral, is set equal to zero. All the other values of $P$ should be set equal to zero.

### Main Program

The FORTRAN program to perform this integration using SNODE is shown in Fig. 3. Lines 10 and 20 can always be written as shown, and will be discussed later. Lines 30 through 110 are used to set the various values of the $P$ array and initial values of $X$ and $Y$. In this case the initial step size, $P(2)$, is equal to 0.001, and five significant figures of accuracy are requested in $P(3)$. The integration is performed over $X$ from 0.0005 to 50 which is a reasonable approximation of 0 to $\infty$. This avoids division by zero at $X = 0$; beyond $X = 50$, $YP < 10^{-16}$ and the area is insignificant.

To start the SNODE program, the CALL ODESTA statement (line 120, Fig. 3) is used. In this statement $X$, $Y$, and $YP$ are variables, and $P$ is a storage array. The next six labels are subroutines supplied by the programmer; the first two are labeled DER and the next four

$$\int_0^\infty \frac{x^3dx}{e^x - 1}$$

```
10      X=0.025
20      Y=0.
30  10  YP=X**3/(EXP(X)-1.)
40      Y=Y+YP*0.05
50      X=X+0.05
60      IF(X-20.)10,30,30
70  30  DUMP Y
80      END
```

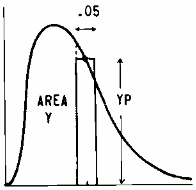


Fig. 1—Evaluation of a definite integral with a simple program.

---

Dr. Gordon's photograph and biography appear with his other paper on page 8 of this issue.

Reprint RE-14-3-18
Final manuscript received September 24, 1968

are labeled RTN; so only two subroutines must be written (these are covered in the next paragraph). The last four labels in the CALL statement are the names of four subroutines stored in the computer. Line 130 in the FORTRAN program refers to $P(11)$, which is the location of an error signal. If $P(11)$ is non-zero, trouble has occurred in the program, so it should be printed to give the user a clue to what type of error is involved. Finally, DUMP Y will print the value of $Y$, the value of the desired integral.

### Subroutines

The necessary subroutines shown in Fig. 3 are quite simple. The subroutine DER has the three variables ($X$, $Y$, $YP$) in its call statement, which are defined as double precision in the type statement. The value of the function to be integrated, $YP$, is calculated in terms of the variable $X$. In the example, the function is $YP = x^3/(e^x - 1)$. The second subroutine is simpler yet. Essentially no action is necessary in the other subroutines; the subroutine RTN is just a dummy subroutine which returns control immediately to the calling program. It contains a type statement, defining the four variables, $X$, $Y$, $YP$, and $P$, as double precision, and a RETURN statement.

### Execution

Execution of the program is initiated with a hyphen, the computer calculates the value of the integral, and prints out the answer, which is 6.494···. The execution in this case takes only 5 or 10 seconds. If no doubling had occurred and the step size had remained at 0.001, the 50,000 steps necessary from 0 to 50 would take considerably longer. Actually the computer doubled the step size for the larger values of $X$. Thirteen doublings occurred, so the next to the last step size was 8.192, rather than 0.001.

The program above illustrates how a definite integral can be evaluated with the SNODE program. If a different integral is desired, it would only be necessary to change the function to be integrated in the subroutine DER, and change the limits in the main program. Now that we've seen how the SNODE program works, let's go back and define each of the quantities more precisely.

$$Y = \int_{x}^{P(6)} YP\, dx$$

Step Size: $P(2)$ = *initial* $\Delta x$

Accuracy: $P(3)$ = *number of significant figures*

Constants: $P(1) = 1$; $P(5) = 1$; $Y = 0$; *all other P's to zero.*

Fig. 2—Performing integration with the SNODE program.

## Description of SNODE

### Basic Features

The SNODE program is modular, and is divided into five component subroutines: ODESTA, ODENPT, ODENDR, ODEERR, ODERUN, each of which can be called separately or replaced by a programmer's subroutine. The easiest way to use it is to call the subroutine ODESTA, which will automatically call the other four subroutines. The call for ODESTA is typically:

```
CALL ODESTA (X, Y, YP, P, DERX, DERY,
        OUT, HALF, DUB, STOP, ODENPT,
        ODENDR, ODEERR, ODERUN)
```

There are four variables or arrays, and ten subroutine names in the call statement. The actual names used for the variables and first six subroutines can be changed by the programmer; it is their location in the call statement that is important. (If necessary, storage requirements can be reduced by setting $P(12) = 1$, calling ODESTA, resetting $P(12) = 0$, and calling ODERUN, as shown in Fig. 8)

### Variables

The three variables used—$X$, $Y$, and $YP$—are all double precision storage locations supplied by the user, and contain the current values of these variables; $X$ is the independent variable and $Y$ is the dependent variable. If there is only one equation to solve, then $Y$ is a single number. But if there is a set of $n$ simultaneous differential equations to solve, then there will be $n$ different $Y$'s, and $Y$ will be a subscripted variable. $YP$ is the derivative of $Y$, and if there is more than one differential equation, $YP$ will also be a subscripted variable with $n$ subscripts. $X$ and $Y$ are initially set by the programmer, and $YP$ is calculated in the subroutine DER.

```
  10 DOUBLE PRECISION X,Y,YP,P(21)
  20 EXTERNAL DER,RTN
  30 P(1)=1
  40 P(2)=.001
  50 P(3)=5
  60 DO 10 I=4,14
  70 10 P(I)=0.0
  80 P(5)=1.
  90 P(6)=50.
 100 X=0.0005
 110 Y=0.
 120 CALL ODESTA (X,Y,YP,P,
     DER,DER,RTN,RTN,RTN,RTN,
     ODENPT,ODENDR,ODEERR,ODERUN)
 130 IF (P(11).NE.0) PDUMP P(11)
 140 DUMP Y
 150 END

 160 SUBR. DER (X,Y,YP)
 170 D.P. X,Y,YP
 180 YP=X**3/(EXP(X)-1.)
 190 RETURN
 200 END

 210 SUBR. RTN (X,Y,YP,P)
 220 D.P. X,Y,YP,P(21)
 230 RETURN
 240 END

 250 -
Y =  .64939582560654800E+01
STOP EXECUTED AFTER *140
 250
```

Fig. 3—Example of integration program using SNODE.

### The P Storage Array

The subscripted array $P$ contains options set by the programmer, information available to the programmer, and working storage. $P(1)$ through $P(14)$ are used for specific parameters; beyond $P(14)$ seven additional storage locations are required for each of $n$ different equations. If $n = 1$, there will be twenty-one $P$ locations, and in general the number of locations in the $P$ array is $14 + 7n$. The first nine $P$ locations must be initially set by the programmer. Following is a list of the first eleven $P$ parameters:

$P(1)$ is the number of equations; if there is only one differential equation, then $P(1) = 1$.

$P(2)$ is a step size, or interval size, equal to the increment of the independent variable $X$. The value of $P(2)$ is set initially by the programmer, but the program will adjust $P(2)$ to an optimum value.

$P(3)$ and $P(4)$ refer to the accuracy desired by the programmer. If $P(4)$ is set to zero, then $P(3)$ specifies relative accuracy and should be set to the number of significant figures desired at each step in the calculation. If $P(4)$ is not zero, then $P(3)$ contains the number of decimal places of accuracy desired, that is, it specifies absolute accuracy. These are the accuracies at each step, and not necessarily the accuracy in the final result.

$P(5)$ and $P(6)$ refer to the endpoint, the upper limit in the integration, or the last point desired in a differential equation. If $P(5)$ is set to zero, the program will continue to run, and no endpoint is defined (the value of $P(6)$ is then immaterial). If $P(5)$ is set to a non-zero value, the endpoint is defined, and the value of the endpoint must be stored in $P(6)$.

$P(7)$ refers to the printing of points. In some cases the computer calculates a point, finds that the error is excessive, and recalculates the point. The routine also recalculates points in the iterative starting procedure. Usually the programmer only wants good points printed, so $P(7)$ should be set to zero. If $P(7)$ is not zero, the subroutine OUT will be called for every point—good or bad.

$P(8)$ and $P(9)$ refer to the halving or doubling, respectively, of the step size by the computer. If these are set to zero, the step size will be adjusted; if not, the change is suppressed, and the step size specified in $P(2)$ is maintained.

$P(10)$ contains the value of the maximum local error estimate, and this value can be printed in one of the subroutines, if the programmer so desires.

$P(11)$ contains various error signals. If the computer sets this equal to one, it means relative accuracy could not be calculated because the value of $Y$ was zero. If the computer sets $P(11) = 2$, the step size was too small; that is, increasing the independent variable by the step size did not change the independent variable numerically. A $P(11) = 3$ indicates the step size was halved for thirty consecutive times and still did not achieve the desired accuracy. If $P(11) = 4$, then the value of $P(1)$ was zero or negative, which is not allowed.

$P(12)$ and the remaining $P$ parameters are beyond the scope of this article.

### User-Supplied Subroutines

The first two subroutines in the call statement are used in the calculation of the derivatives, $YP$. Frequently, only one subroutine is written, and in it $YP$ is calculated as a function of $X$ and $Y$. The third subroutine, OUT, will be called each time a new point is calculated in the solution. If the programmer is interested in the different values of $Y$, the form in which the printing is desired can be specified in this subroutine. At each step the subroutine will be executed, and the programmer will get his desired print out. The values of $X$ and $Y$ are usually printed, and if desired, the values of $YP$ and any of the $P$ parameters can be printed.

The next two subroutines, HALF and DUB, are called when the step size is about to be halved or doubled, respectively. If the programmer wants an indication that this has occurred, then appropriate print instructions can be put in these subroutines. Finally, the subroutine STOP is called whenever an error occurs, and the programmer can put in appropriate print instructions, usually with reference to $P(11)$ which will give the clue as to the type of error. (If long execution times are expected, there should be an executable stop in subroutine STOP, as shown in Fig. 8, to prevent serious looping.)

### Computer-Supplied Subroutines

In each of the five SNODE subroutine names, ODE stands for Ordinary Differential Equations. The ODESTA subroutine STArts computation, and calculates various initial values; ODENPT calculates a New PoinT as the calculations proceed; ODENDR extrapolates New back DeRivative values when the step size is changed; ODEERR computes an estimate of the ERRor involved in each calculation. Finally, ODERUN obtains successive points in the main calculation.

## Differential Equation

### Basic Principles

The basic procedure in any numerical solution of a first-order differential equation is shown in Fig. 4. It is always assumed that one point on the curve is known; this point may be the initial point, or a boundary condition for the problem. Starting at this known point, the slope is calculated from the differential equation. By extrapolating along a straight line with that slope, a new point can be found. This point may not be exactly on the curve, but if the step size is small, the new point will be very close to the curve. Using this new point, another slope can be calculated from the differential equation, and another extrapolation made. Thus the solution is obtained by generating successive points along the curve.

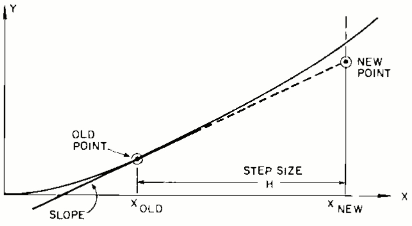
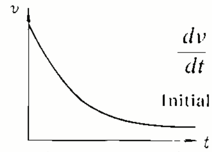


Fig. 4—Illustration of the basic principles in the numerical solution of differential equations.

$$YP = dY/dx = f(x,y)$$

Step Size: $P(2)$ = *initial* $\Delta x$
Accuracy: $P(3)$ = *number of significant figures*
Constants: $x$ = *initial* $x$; $Y$ = *initial* $Y$; $P(1)$ = 1; *all other P's to zero.*

Fig. 5—Solving a first-order differential equation with SNODE.



$$\frac{dv}{dt} = g\left(1 - \frac{v^2}{256}\right)$$

Initially, $v = 176$ and $t = 0$.

Fig. 6—Example of a first-order differential equation—an opening parachute.

For any numerical solution of a first-order differential equation, a few basic steps are always necessary.

1) The initial values of $X$ and $Y$ must be set.
2) The derivative, or slope, $YP$, must be calculated.
3) New values of $X$ and $Y$ are calculated by extrapolation.
4) Steps 2) and 3) are repeated, and the curve is generated.

In more powerful numerical methods, such as SNODE, extrapolation is not along a simple straight line, but several slopes are calculated with the result that higher-order derivatives are taken into consideration.

### Constants Needed by SNODE

The general form of a first-order differential equation is shown in Fig. 5, with the variables that must be set. The differential equation is solved for the first derivative, $YP$, and this function of $X$ and $Y$ is put into the subroutine DER. Again, the initial step size is inserted in $P(2)$ and the accuracy desired in $P(3)$. The value of $P(1)$ is set to unity (representing only one equation); the initial values of $X$ and $Y$ are set; and all other $P$'s are set equal to zero.

The specific problem to be used for illustration is shown in Fig. 6. A parachutist is falling with a downward velocity of 176 ft/sec; when the parachute opens, the change in velocity is given by the differential equation. The parachutist gradually slows to a terminal velocity as shown in Fig. 6. The

```
 10 D.P.X,Y,YP,P(21)
 20 EXTERNAL DER, RTN,OUT
 30 P(1)=1
 40 P(2)=.005
 50 P(3)=2
 60 DO10 I=4,14
 70 10 P(I)=0.0
 80 X=0.
 90 Y=176.
100 CALL ODESTA (X,Y,YP,P,
    DER,DER,OUT,RTN,RTN,RTN,
    ODENPT,ODENDR,ODEERR,ODERUN)
110 END

120 SUBR.DER(X,Y,YP)
130 D.P.X,Y,YP
140 YP=32.-32.*Y**2/256.
150 RETURN
160 END

170 SUBR.OUT(X,Y,YP,P)
180 D.P.X,Y,YP,P(21)
190 8 FORMAT (1X
190 PRINT 5,X,Y
200 5 FORMAT(1X,F7.3,F6.0)
210 RETURN
220 END

230 SUBR.RTN(X,Y,YP,P)
240 D.P.X,Y,YP,P(21)
250 RETURN
260 END
```

```
                                          0.435   21.
 270                   0.085   63.        0.475   21.
0.000  176.            0.095   58.        0.515   20.
0.005  159.            0.115   51.        0.595   19.
0.010  145.            0.135   46.        0.675   18.
0.015  133.            0.155   42.        0.755   17.
0.020  123.            0.175   39.        0.835   17.
0.025  114.            0.195   36.        0.915   17.
0.030  107.            0.215   33.        0.995   16.
0.035  100.            0.235   31.        1.155   16.
0.045   99.            0.275   28.        1.315   16.
0.055   81.            0.315   26.        1.475   16.
0.065   73.            0.355   24.        1.635   16.
0.075   68.            0.395   23.        1.795   16.
```

Fig. 7—Solution of a first-order differential equation using SNODE.

FORTRAN program could be written in terms of $v$ and $t$, but to be consistent with the rest of the paper, time will be denoted by $X$ and the velocity by $Y$.

### FORTRAN Program

The program is shown in Fig. 7. There is only one differential equation, so $Y$ and $YP$ are not subscripted and the $P$ array will have twenty-one locations. These quantities, as well as $X$, must be double precision, and this is stated in line 10 of Fig. 7. We will be using the names of three subroutines in the CALL ODESTA statement, and these names must be designated in an EXTERNAL statement (line 20). Lines 30 through 90 set the values of the $P$ parameters and the initial values of $X$ and $Y$; an initial step size of 0.005 seconds is chosen, and two-significant-figure accuracy is requested.

The CALL ODESTA is like the previous program, except that the third subroutine is designated OUT, instead of the dummy subroutine RTN. A specific endpoint is not desired, because we are not looking for the final velocity, but are interested in how the velocity changes as a function of time. So the computer should print the values of the velocity as they are calculated, and the subroutine OUT specifies the details of the printing. After the CALL statement, no other statements are necessary in the main program, because we do not expect this call actually to terminate.

The subroutine DER (lines 120 to 160) provides the derivative $YP$ as a function of $Y$ (in this case, $YP$ is not a function of $X$). The subroutine OUT (lines 170 to 220) provides the printing of the solution. The values of $X$ and $Y$ are printed whenever OUT is called, which occurs after each step has been computed. (The deceleration, $YP$, could also be printed if desired). The last subroutine, RTN (lines 230 to 260), is just a dummy subroutine as before.

### Execution

Execution is initiated with a hyphen, and the corresponding values of time and velocity are printed. Initially, the velocity is changing by a significant amount, and the step size is maintained at the programmer's value of five milliseconds. After thirty-five milliseconds, the change in velocity is smaller, and the computer doubles the step size to ten milliseconds. As the changes in velocity become less significant, the step sizes are doubled successively. At one second, the increment is 0.160 sec., and the velocity has decreased to its terminal value of 16 ft/sec.

The main limitation in time for this program is not the computation, but the speed of the teletype printer. When the velocity has reached the terminal value of 16 ft/sec, its value no longer changes, and execution is terminated by pressing the ESCAPE button.

## Simultaneous Differential Equations

Solving a set of $n$ simultaneous equations is similar to solving one equation. The difference is that everything done to the dependent variable $Y$ must now be done for $n$ dependent variables ranging from $Y(1)$ to $Y(n)$. The value of $P(1)$ is set to the number of equations to be solved. The size of the $P$ parameter array must be increased by seven working locations for each additional differential equation (a total of $14 + 7n$). The dependent variable $Y$ will be subscripted, with a different Y for each dependent variable. Similarly, $YP$, the derivative of $Y$, will be subscripted, ranging from $YP(1)$ to $YP(n)$. In the main program, each dependent variable $Y(I)$ must be set to an initial value. The differential equations must be in the subroutine DER, so that each of the derivatives $YP(I)$ can be calculated in terms of the variables $X$ and $Y$.

### Transistor Design Problem

To illustrate the solution of simultaneous differential equations, an example will be taken from a transistor design program, described by Dr. R. B. Schilling in this issue. While the complete transistor design program involves a base region, a transition region, and an ohmic region in the transistor, only the base region will be considered here. The transition region would require similar treatment, but the ohmic region does not require the solution of a differential equation.

For the base region, Schilling assumes an exponential doping profile, given by

$$N = 5 \times 10^{16} \exp(—x/\lambda) \\ — 6.5 \times 10^{13} [1 — \exp(—x/\lambda)] \quad (1)$$

donor atoms/cm$^3$, where $\lambda = 0.11 \times 10^{-3}$ cm. (See Eq. 15 in Schilling's paper).

The three variables we are concerned with are:

1) the hole density $p$,
2) the voltage V, and
3) the transit time T.

Each of these quantities varies along the distance $X$ in the transistor. The differential equations that define the variations in these quantities are:

$$\frac{dp}{dx} = \frac{p[J_n/qD_n — dN/dx]}{2p — N} \quad (2)$$

$$\frac{dV}{dx} = —V_T \frac{dp/dx}{p} \quad (3)$$

$$\frac{dT}{dx} = q(p — N)/(—J_n) \quad (4)$$

The notation used above, and the corresponding notation used in the Fortran program are defined in Table I. Eqs. 2 and 3 are Eqs. 8 and 6 in Schilling's paper, respectively. The last equation is derived from the fundamental equation $J_n = —qnv$, that is, the current density $J_n$ must be equal to the charge on each electron, $—q$, times the free electron density $(n = p — N)$, times the velocity $(v = dx/dT)$.

Table I—Quantities used in Transistor-Design Program

| *Quantity* | *Notation* *Usual* | *Notation* *Program* | *Value* |
|---|---|---|---|
| Doping at junction | | CJCTN | 5 x $10^{16}$ elec/cm$^3$ |
| Doping in bulk | | CBULK | 6.5 x $10^{13}$ hole/cm$^3$ |
| Electron current density | $-J_n$ | CABS | 20 amp/cm$^2$ |
| Exponential factor | $\lambda$ | DD | 0.11 x $10^{-3}$ cm |
| Exponential | $e^{-x/\lambda}$ | GX | — |
| Doping | $N(x)$ | FX | — |
| Doping gradient | $dN/dx$ | FXP | — |
| Hole density | $p$ | $Y(1)$ | 5.4 x $10^{16}$ holes/cm$^3$ $(x = 0)$ |
| Voltage | $V$ | $Y(2)$ | 0 volts (x = 0) |
| Transit time | $T$ | $Y(3)$ | 0 sec. $(x = 0)$ |
| Electron diffusion constant | $D_n$ | — | 25 cm$^2$/sec |
| Thermal voltage | $V_T$ | — | .026 volts |
| Electronic charge | $q$ | — | 1.6 x $10^{-19}$ coulombs |

Using the numerical value of the constants and a notation suitable for the Fortran program, the three equations can be written:

$$\text{YP}(1) = \frac{—2.5 \times 10^{17}\ \text{CABS} + \text{FXP}}{2Y(1) — \text{FX}} \quad (5)$$

$$YP(2) = —.026\ YP(1)/\text{Y}(1) \quad (6)$$

$$YP(3) = 1.6 \times 10^{-19}\ [Y(1) — \text{FX}]/\text{CABS} \quad (7)$$

### FORTRAN Program

The program is shown in Fig. 8. Rather than calling ODESTA directly, a subroutine SNODE has been written; the main program calls SNODE, which in turn calls ODESTA. The main program and the first two subroutines, DER and OUT, are written specifically for this problem. The last four subroutines are of a general nature, and can be used in many other programs. The subroutine HALF and DUB are written to print messages whenever the interval is halved or doubled respectively. Similarly, the subroutine STOP prints a clear message, if any trouble has occurred.

### Execution

Execution of the program is shown in Fig. 9. The first chosen step size of 0.001 was too large to maintain the desired accuracy of two significant figures, so four successive halvings of the interval occurred, and a step size of 6.25 x $10^{-5}$ was used. Rather than continue with this odd size, execution was terminated and restarted with a step size of 5 x $10^{-5}$.

```
C THREE SIMULTANEOUS EQUATIONS USING SNODE
DOUBLE PRECISION X,Y(3),YP(3),P(35),FIGACC,TYPACC,STPSZE
C P ARRAY CONTAINS (14+3*7) DOUBLE WORDS
C FOLLOWING ARE PHYSICAL CONSTANTS PECULIAR TO THE PROBLEM
COMMON DOUBLE PRECISION CJCTN,CBULK,CABS,W,DD
CJCTN=5E16
CBULK=6.5E13
CABS=20
W=.762E-3
DD=W/LOG(1+CJCTN/CBULK)
EXTERNAL DER,OUT,HALF,DUB
2PRINT3
3FORMAT('1')
READ FIGACC,TYPACC,STPSZE
C SET UP FOR 3 EQUATIONS AND INSERT FIGURES AND TYPE OF ACCURACY
P(1)=3
P(3)=FIGACC
P(4)=TYPACC
C START AT 0 AND INSERT STEP SIZE
X=0
P(2)=STPSZE
C HIT ENDPOINT .0055
P(5)=1
P(6)=5.5E-3
C PRINT GOOD POINTS ONLY, ALLOW HALVING AND DOUBLING
P(7)=0
P(8)=0
P(9)=0
P(10)=0
C INITIALIZE DEPENDENT VARIABLES
Y(1)=5.4E16
Y(2)=0
Y(3)=0
PRINT1
1FORMAT('0',T7,'X',T21,'Y(1)',T35,'Y(2)',T49,'Y(3)')
CALL SNODE (X,Y,YP,P,DER,OUT,HALF,DUB)
GOTO2
C
SUBROUTINE DER (X,Y,YP)
DOUBLE PRECISION X,Y(3),YP(3),GX,FX,FXP
COMMON DOUBLE PRECISION CJCTN,CBULK,CABS,W,DD
GX=EXP(-X/DD)
FX=CJCTN*GX-CBULK*(1-GX)
FXP=-(CJCTN+CBULK)*GX/DD
YP(1)=Y(1)*(-2.5E17*CABS+FXP)/(2*Y(1)-FX)
YP(2)=-.026*YP(1)/Y(1)
YP(3)=1.6E-19*(Y(1)-FX)/CABS
RETURN
END
C
SUBROUTINE OUT (X,Y,YP,P)
DOUBLE PRECISION X,Y(3),YP(3),P(35)
1FORMAT(1X,1P4E14.6)
PRINT1,X,Y(1),Y(2),Y(3)
RETURN
END
SUBROUTINE HALF (X,Y,YP,P)
DOUBLE PRECISION X,Y(3),YP(3),P(35)
1FORMAT(' INTERVAL HALVED')
PRINT1
RETURN
END
C
SUBROUTINE DUB (X,Y,YP,P)
DOUBLE PRECISION X,Y(3),YP(3),P(35)
1FORMAT(' INTERVAL DOUBLED')
PRINT1
RETURN
END
C
SUBROUTINE SNODE (X,Y,YP,P,D,O,H,B)
EXTERNAL D,O,H,B,STOP
DOUBLE PRECISION X,Y(3),YP(3),P(35)
P(11)=0
P(12)=1
P(13)=0
P(14)=0
CALLODESTA(X,Y,YP,P,D,D,O,H,B,STOP,ODENPT,ODENDP,ODEERR,ODERUN)
P(12)=0
CALLODERUN(X,Y,YP,P,D,D,O,H,B,STOP,ODENPT,ODENDP,ODEERR,ODESTA)
RETURN
END
C
SUBROUTINE STOP (X,Y,YP,P)
DOUBLE PRECISION X,Y(3),YP(3),P(35)
N=P(11)
IF(N.EQ.1)PRINT1
IF(N.EQ.2)PRINT2
IF(N.EQ.3)PRINT3
IF(N.EQ.4)PRINT4
PRINT5
PAUSE
1FORMAT('0DEPENDENT VARIABLE ZERO,RELATIVE ACCURACY')
2FORMAT('0INTERVAL TOO SMALL--X+H=X')
3FORMAT('0THIRTY CONSECUTIVE INTERVAL REDUCTIONS')
4FORMAT('0NUMBER OF EQUATIONS ZERO OR NEGATIVE')
5FORMAT(' JOB TERMINATED')
END
START
```

Fig. 8—Program to solve three differential equations—transistor problem.

This last step size was used for the first twelve steps, and later the interval was doubled several times, saving on computation time. At the final distance of 5.5 x $10^{-3}$ cm, the hole density $p$ was 3.14 x $10^{11}$ holes/cm$^3$; the voltage $V$ across the base region was 0.136 volts; and the transit time $T$ across the base region was 3 x $10^{-7}$ sec.

After a program has run successfully, its accuracy can be increased; in Fig. 10 is shown part of the program when five-significant-figure accuracy was requested for each step. The program runs considerably longer, since the step sizes used must be smaller. The results in Figs. 9 and 10 can be compared, and an estimate of the accuracy of one significant figure is obtained in the first table. The best way to check the accuracy obtained in the second table (which had 5-figure accuracy at each step), would be to rerun the program with six or seven-figure accuracy, and compare the solutions.

```
FIGACC=2
TYPACC=0
STPSZE=.001

      X            Y(1)          Y(2)          Y(3)
  0.000000E+00  5.400000E+16  0.000000E+00  0.000000E+00
INTERVAL HALVED
INTERVAL HALVED
INTERVAL HALVED
INTERVAL HALVED
  6.250000E-05  3.475446E+16  1.145096E-02  2.427083E-09
  1.250000E-04  2.459395E+16  2.045576E-02  5.034931E-09
  1.875000E-04  1.929858E+16  2.673706E-02  1.021504E-08
  2.500000E-04  1.649040E+16  3.070458E-02  1.540476E-08
  3.125000E-04

  5.000000E-05  3.771...       9.332010E-0.   ..871636E-09
  1.000000E-04  2.789870E+16  1.717311E-02  4.359615E-09
  1.500000E-04  2.207144E+16  2.326131E-02  7.494042E-09
  2.000000E-04  1.861166E+16  2.767722E-02  1.122305E-08
  2.500000E-04  1.651031E+16  3.077428E-02  1.541607E-08
  3.000000E-04  1.519170E+16  3.292615E-02  1.994111E-08
  3.500000E-04  1.433275E+16  3.443303E-02  2.468320E-08
  4.000000E-04  1.374975E+16  3.550984E-02  2.955691E-08
  4.500000E-04  1.333577E+16  3.630342E-02  3.450256E-08
  5.000000E-04  1.302703E+16  3.691188E-02  3.947926E-08
  5.500000E-04  1.278460E+16  3.740003E-02  4.445937E-08
  6.000000E-04  1.258433E+16  3.781043E-02  4.942440E-08
INTERVAL DOUBLED
  7.000000E-04  1.225253E+16  3.850329E-02  5.926718E-08
  8.000000E-04  1.196876E+16  3.911146E-02  6.895419E-08
  9.000000E-04  1.170510E+16  3.969021E-02  7.845365E-08
  1.000000E-03  1.144981E+16  4.026339E-02  8.775830E-08
  1.100000E-03  1.119802E+16  4.084146E-02  9.686552E-08
  1.200000E-03  1.094770E+16  4.142922E-02  1.057742E-07
INTERVAL DOUBLED
  1.400000E-03  1.044811E+16  4.264330E-02  1.229952E-07
  1.600000E-03  9.949467E+15  4.391444E-02  1.394201E-07
  1.800000E-03  9.451097E+15  4.525046E-02  1.550453E-07
INTERVAL DOUBLED
  2.200000E-03  8.454475E+15  4.814752E-02  1.839643E-07
  2.600000E-03  7.458532E+15  5.140621E-02  2.095756E-07
  3.000000E-03  6.463187E+15  5.513090E-02  2.320584E-07
INTERVAL DOUBLED
  3.800000E-03  4.475174E+15  6.472355E-02  2.674750E-07
  4.600000E-03  2.494475E+15  8.016442E-02  2.901857E-07
INTERVAL HALVED
  5.000000E-03  1.510890E+15  9.329688E-02  2.967988E-07
  5.400000E-03  5.465860E+14  1.218533E-01  3.002032E-07
  5.500000E-03  3.146055E+14  1.362690E-01  3.006785E-07
```

Fig. 9—Solution of transistor problem using two-figure accuracy.

```
FIGACC=5
TYPACC=0
STPSZE=.00005

      X            Y(1)          Y(2)          Y(3)
  0.000000E+00  5.400000E+16  0.000000E+00  0.000000E+00
INTERVAL HALVED
INTERVAL HALVED
  1.250000E-05  4.915686E+16  2.443159E-03  4.162554E-10
  2.500000E-05  4.486775E+16  4.815877E-03  8.663721E-10
  3.750000E-05  4.107576E+16  7.112711E-03  1.352243E-09
  5.000000E-05  3.772905E+16  9.322392E-03  1.875571E-09
  6.250000E-05  3.478027E+16  1.   ...75E-02  2.43.....-09
  7.500000E-05

  ...00...E-03  9.418247E+14  1.05276..E-01  ......656E-07
  5.325000E-03  8.212764E+14  1.088373E-01  3.012642E-07
  5.350000E-03  7.014051E+14  1.129403E-01  3.015947E-07
  5.400000E-03  5.824472E+14  1.177739E-01  3.018774E-07
INTERVAL HALVED
  5.425000E-03  5.234198E+14  1.205520E-01  3.020010E-07
  5.450000E-03  4.647815E+14  1.236415E-01  3.021128E-07
  5.475000E-03  4.066264E+14  1.271173E-01  3.022129E-07
  5.500000E-03  3.490859E+14  1.310849E-01  3.023015E-07
  5.500000E-03  3.490859E+14  1.310849E-01  3.023015E-07
```

Fig. 10—Solution of transistor problem using five-figure accuracy.

### Other Types of Problems

Higher-order differential equations can be handled by solving the differential equation for the highest order derivative, and treating the lower orders as distinct variables. For example, in the second-order differential equation

$$y'' = f(x, y, y') \tag{8}$$

the first-order derivative $y'$ can be denoted as $q$. Then the problem can be written as two first-order equations

$$q' = f(x, y, q) \tag{9}$$

$$y' = q \tag{10}$$

which is readily handled by SNODE.

To do multiple integration, it is necessary to do each integration in turn.

For example, consider the following problem, proposed and programmed by Dr. F. W. Schneider:

$$y = \int_0^{\pi/2} \int_0^{\pi/2} \sin(xu)\,du\,dx \tag{11}$$

If the inside integral is denoted as $v$, which is a function of $x$, then the problem is to evaluate

$$y = \int_0^{\pi/2} v\,dx \tag{12}$$

$$v = \int_0^{\pi/2} \sin(xu)\,du \tag{13}$$

The main program will be written to evaluate $y$, and will require a subroutine DER to evaluate the function $v$. But in this subroutine, in order to evaluate the function $v$, SNODE will be called again to perform the second integration. (This is allowed since SNODE is recursive). This second call to SNODE will require a different DER subroutine, which might be labeled DER2, in which the function sin $(xu)$ is written. The two calls to SNODE will have different variables, including two distinct $P$ arrays.

### Conclusions

Engineers encounter differential equations in a variety of fields: transients in a circuit, the motion of an electron, diffusion effects in a transistor, temperature rise in equipment. Using techniques described in this paper, the engineer can use SNODE to handle these problems, obtaining the fast turnaround of time sharing systems, and the efficiency of a library program.

# Computer simulation of photomultiplier-tube operation

**D. E. Persyk**

**Computer simulation can be employed to predict detection efficiency and time response of tentative photomultiplier-tube designs. The simulation discussed in this paper makes use of an Electron-Optics program to acquire electron ballistic data and a Time-Sharing Program to simulate photodiode operation of the photomultiplier tube.**

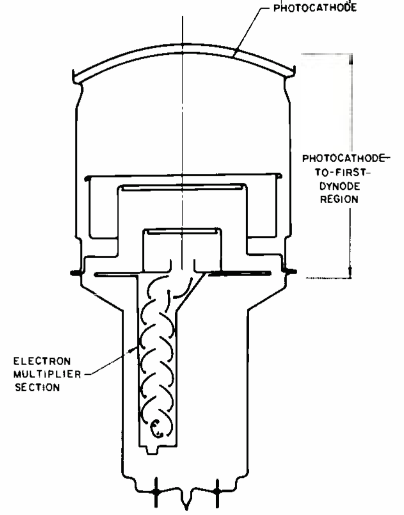


Fig. 1—Photomultiplier tube cross section.

COMPUTER-AIDED PHOTOMULTIPLIER TUBE design has largely replaced traditional design methods because of the time and cost savings realized. Complete computer simulation of a photomultiplier tube is feasible and affords considerable advantages over the conventional approach of building a tube and experimentally measuring its performance. This paper illustrates a method of simulating photodiode operation of a photomultiplier tube.

## The Photomultiplier Tube

A photomultiplier tube consists of two sections as shown in Figs. 1 and 2: the photocathode-to-first-dynode region and the electron-multiplier section. The photocathode-to-first-dynode region contains a light-sensitive photocathode that converts light energy (photons) to photoelectrons. The photoelectrons are accelerated and focused on the first electron-multiplier stage, called the first dynode. The electrons are multiplied by secondary emission by a factor of 4 or 5 upon each jump to a successive dynode of the multiplier; total multiplication factors of $10^8$ are common. Thus, a single photon may give rise to an easily detectable electrical pulse of $10^8$ electrons.

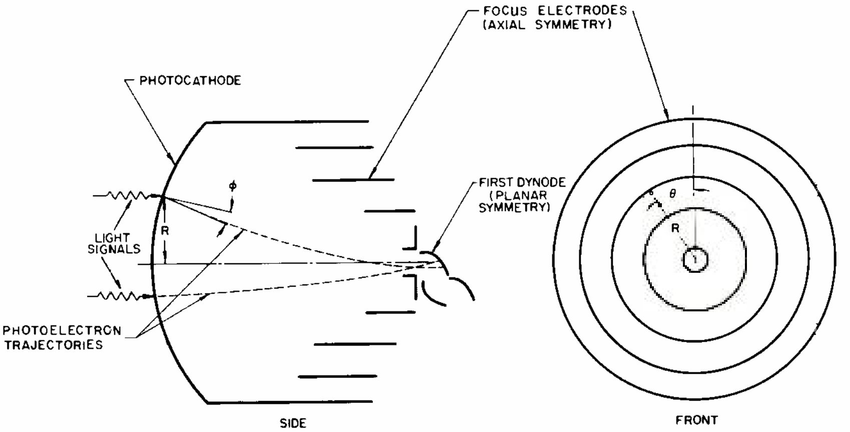


Fig. 2—Side and front views of the photocathode-to-first-dynode region showing parameters used in electron trajectory computations.

When a photomultiplier tube is operated as a photodiode, the electron-multiplier section is not used. In photodiode operation, the first dynode serves as the collector or anode.

### Photocathode-to-First-Dynode Region

The photocathode-to-first-dynode region usually limits the time response and detection efficiency of the entire photomultiplier tube; consequently, design of this section is critical.

The design requirements of the photocathode-to-first-dynode region are twofold: 1) all photoelectrons leaving the photocathode should be collected on the surface of the first dynode, and 2), all electrons should have equal times of flight. The first condition assures high detection efficiency; the second assures optimum time response. Collection efficiency, $\varepsilon$, is defined as the ratio of the number of photoelectrons collected to the number of photoelectrons launched. Time response is defined in terms of the broadening that a delta-function electron packet undergoes in its flight from photocathode to first dynode, as shown in Fig. 3. Ideally an instantaneous light signal incident upon the photocathode at time $t_o$ liberates $N$ photoelectrons, all of which arrive at the first dynode at time, $t$, later. Actually, however, the delta-function electron packet is time-broadened in its flight, i.e., the electrons in

---

Reprint RE-14-3-15
Final manuscript received September 6, 1968.

**Dennis E. Persyk**
Photomultiplier Advanced Development
Electronic Components
Lancaster, Pa.

received the BA in Physics in 1963, and the MA in Physics in 1964, from the University of Wisconsin. He has completed one year of additional graduate study. During his undergraduate and graduate years he worked in the Vacuum Technologies Lab, and served two years as a research assistant. Mr. Persyk has had experience in developmental work on ultra-high vacuum systems, designed and constructed getter-ion pumps, and worked on the development of the Orbitron ionization gauge. He was associated with research and development of the Pelletron, a compact Van de Graaf machine. Joining RCA in June of 1965, he has had the responsibility for establishing and maintaining DT production of the C70133 photomultiplier and investigating the properties of the central wire photomultiplier and variants of the venetian-blind photomultiplier as possible elements in a wideband control system. His present assignments include work on the development of a fast time response planar photocathode photomultiplier, and development of a five inch version of a metal-ceramic photomultiplier. He is a member of the American Association of Physics Teachers and the American Vacuum Society.

the packet have different times of flight, because the $N$ photoelectrons leave the photocathode from different places and with different velocities. Thus, the rate-of-arrival function, $dN/dt$, for electrons arriving at the first dynode has non-zero width, and may be characterized by its 10-to-90-percent amplitude risetime, $T_r$, and its full-width-at-half-maximum amplitude FWHM as shown in Fig. 3. In the design of the photocathode-to-first-dynode section, the geometry of the focus electrode is varied in a manner that minimizes $T_r$ and FWHM while maximizing collection efficiency $\varepsilon$. Values for $T_r$, FWHM, and $\varepsilon$ can be obtained by a computer program that can be run on the RCA Basic Time-Sharing System (BTSS).

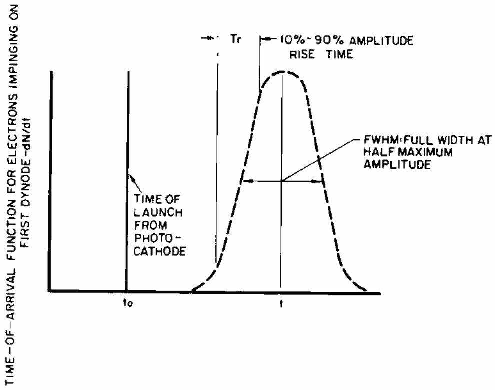


Fig. 3—Time broadening of a delta-function electron packet in its flight from photocathode to first dynode.

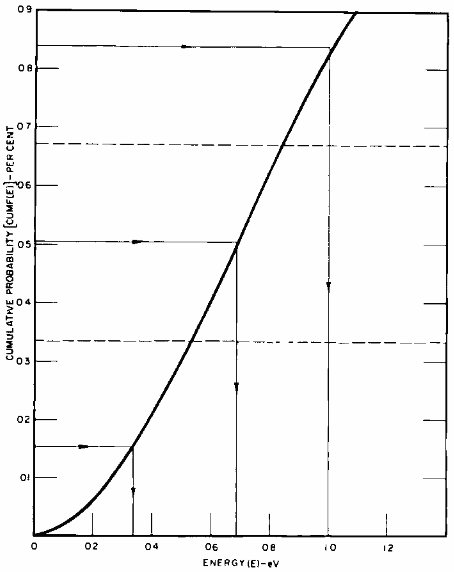


Fig. 4—Photoelectron energy distribution.

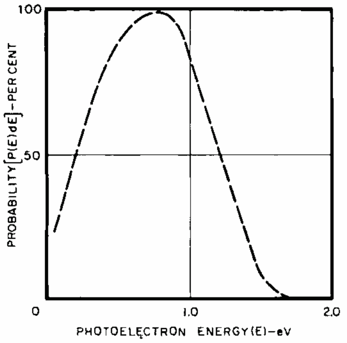


Fig. 5—Method of selecting three equally probable energies from the cumulative probability function.

## Electron Trajectory Computations

Electron trajectories are computed by an Electron-Optics program[1] that provides spatial and temporal information about the electron path through the photocathode-to-first-dynode region. Data are accumulated for transit time (time of flight) as a function of the electron initial conditions at the photocathode: radial position $R$ on the photocathode, energy $E$, and angle of launch $\phi$; these parameters are shown in Fig. 2. Trajectories are usually constrained to lie in a plane containing the symmetry axis so that the required number of trajectory computations is minimized. Values of time as a function of $R, E$, and $\phi [T(R,E,\phi)]$ are tabulated, and a geometrical factor $\theta$ is added to allow for the transition from axial to planar symmetry at the surface of the first dynode. The $\theta$, shown in Fig. 2, represents the rotational launch position or sector on the photocathode from which an electron is launched. Electrons from the axisymmetric region impinge upon a cylindrical-section surface; an extrapolator program handles the three-dimensional problem of determining final landing position and total transit time to the surface of the first dynode for a given $\theta$.

## BTSS Program

### Operation

Data for transit time in the form TIME $= T\ (R,\phi,E,\theta)$ are loaded into the $T$ array using the convention that TIME $= 0$ designates an electron that misses the dynode; this technique allows spatial information to be included in the $T$ array. Photodiode operation is then simulated by assigning weighted values for $R,\phi,E$, and $\theta$. The program then stores the corresponding value of time from the $T$ array. At the completion of the process a printout is taken of 1) the number of misses, an indication of collection efficiency, and 2), the rate-of-arrival statistics $dN/dt$, a parameter that indicates time response.

### Weighting for R, $\phi$, E, and $\theta$

If the photocathode is evenly illuminated, photoelectrons are emitted with constant current density; i.e., the number of photoelectrons per second leaving a unit area of the photocathode is everywhere constant. To simplify calculations, usually only those trajectories contained in a plane through the symmetry axis are considered. Viewed in cross section, the number of photoelectrons leaving an element $\Delta R$ at radius $R$ is proportional to $2\pi R\Delta R$, the area of the differential annulus at radius $R$. To make all $R$ values equally probable, launch radii $R$ are chosen to be representative of equal areas of the projection of the spherical section photocathode upon a plane perpendicular to the symmetry axis. This is done as follows:

Let $N$ equal area annulii be chosen such that

$$\begin{array}{rl} 0 = R_0 \leq R < R_1 & \textit{1st annulus} \\ R_1 \leq R < R_2 & \textit{2nd annulus} \\ \bullet & \bullet \\ \bullet & \bullet \\ R_{N-1} \leq R < R_N & \textit{Nth annulus} \end{array} \quad (1)$$

where $R_N$ is the maximum radius of the photocathode.

For $N$ equal area annuli

$$\frac{\pi R^2_N}{N} = \pi (R^2_K - R^2_{K-1})$$

and

$$R_K = \left( \frac{R^2_N}{N} + R^2_{K-1} \right)^{1/2} \quad (2)$$

for $K = 1,2,\ldots,N$.

The launch position for the $Kth$ annulus, $\bar{R}_K$, is found by setting

$$\pi(R^2_K - \bar{R}^2_K) = \pi(\bar{R}^2_K - R^2_{K-1})$$

This constraint yields

$$\bar{R}_K = \left( \frac{R^2_K + R^2_{K-1}}{2} \right)^{1/2} \quad (3)$$

where $K = 1, 2, \ldots, N$.

The selection of equally probable energies can be understood by reference to Figs. 4 and 5 taken from the data of Philipp and Taft.[2] The probability function $P(E)dE$ shown in Fig. 4 gives the probability that an electron will be emitted within an increment $dE$ of energy $E$. This function is integrated and normalized to obtain the cumulative probability function:

$$CUMF(E) = \frac{\int_0^E P(E)dE}{\int_0^\infty P(E)dE} \qquad (4)$$

The values of 3 equally probable energies are obtained by dividing the probability axis into 3 equal segments, and determining the energy values corresponding to the probabilities at the mid-points of the 3 segments as shown in Fig. 5. The same technique can be used to determine equally probable angles of launch.

## Program Results

The program was tested on RCA's Basic Time-Sharing System (BTSS) because of the relative ease with which program structure can be quickly altered, and because the system produces results in minutes. To stay within the memory capabilities of BTSS and keep execution time short, the input data in the $T$ array were restricted to a small sample size.

The $T$ array contains 152 elements distributed as follows: $T(6,3,3,3)$; i.e.: 6 equally probable launch radii $R$, 3 equally probable launch angles $\phi$, 3 energies $E$ (0, 0.5, and 1.0eV with arbitrary weighting of $\frac{1}{7}$, $\frac{3}{7}$, and $\frac{3}{7}$, respectively), and 3 geometrical factors $\theta$ (arbitrarily weighted ¼, ¼, and ½). A histogram of the rate-of-arrival statistics is shown in Fig. 6 with a smooth curve sketched in to indicate the shape of the function that might be obtained with more input data and finer time increments. The design in this simulation has a predicted risetime $T_r$ of the order of 1 nanosecond, a full-width-at-half-maximum amplitude FWHM of approximately 3.5 nanoseconds, and a collection efficiency $\epsilon$ of 80%.

These results are easily obtained on time sharing; the weighting functions can be varied and a new result produced in minutes. The accuracy of this technique is limited only by the amount of the input data and by present knowledge of the energy distribution function for photoelectrons. In contrast, experimental measurements on these types of experiments are quite time consuming and very limited in scope due to fundamental equipment limitations imposed by state of the art instrumentation for measuring low-level, fast-risetime electrical pulses.

Because of the ease and speed with which different weighting functions can be substituted in the time-sharing simulation program, a number of interesting simulations were performed in addition to the one described above. In one simulation, only low-energy electrons were allowed to leave the photocathode; this condition corresponds physically to illuminating the photocathode with light of very long wavelength so that the photoelectrons which are released have very small initial energies. Fig. 7 shows that the risetime, $T_r$, and the full width at half maximum amplitude, FWHM, are less for the low energy photoelectrons than for the normal energy distribution shown in Fig. 6. Collection efficiency is of the order of 100% for the low energy photoelectrons.

A second simulation restricted photoemission to a small spot on the axis of the photocathode; this restriction is equivalent to illuminating the photocathode with a light beam of small diameter. As shown in Fig. 8, the values of $T_r$ and FWHM are less than the values obtained for the fully illuminated case shown in Fig. 6. As in all other simulations, results were in agreement with experimental observations.

## Conclusions

A method of evaluating tentative designs of photomultiplier tube photocathode-to-first-dynode regions without recourse to tube building has been demonstrated. As the results above show, simulation of a variety of photodiode actions by RCA's Basic Time-Sharing System is easily performed by simple editing of the weighting functions; results are obtainable in minutes. In contrast, experimental measurements of the same type are very limited in scope. Extensions of the computer simulation technique can be employed to simulate an entire photomultiplier tube, and other conversion tube devices.

## Acknowledgment

The author owes the success of this project to the many people who have assisted in all phases of it. Guidance from R. M. Matheson and the collaborative effort of J. F. Parker are gratefully acknowledged.

## References

1. H. E. Kulsrud, "A Programming System for Electron Optical Simulation," *RCA Review*, 28,351.
2. E. R. Taft and H. R. Philipp, Phys. Rev. 115, 1583. Fig. 4 is taken from Fig. 3 of Taft's paper.

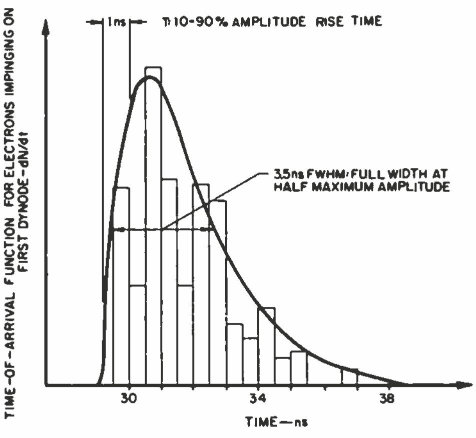


Fig. 6—Predicted time response characteristic for a fully illuminated photocathode excited by a delta-function light impulse at time $t = 0$.

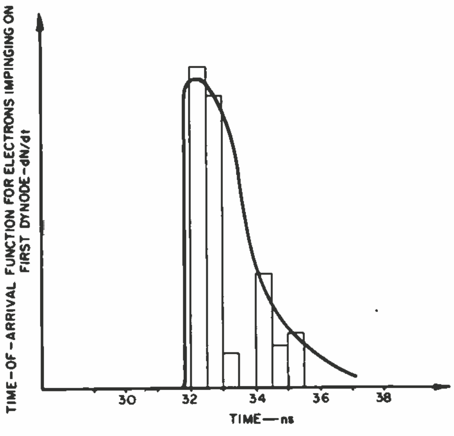


Fig. 7—Predicted time response characteristic for a fully illuminated photocathode considering only low-energy photoelectron emission.

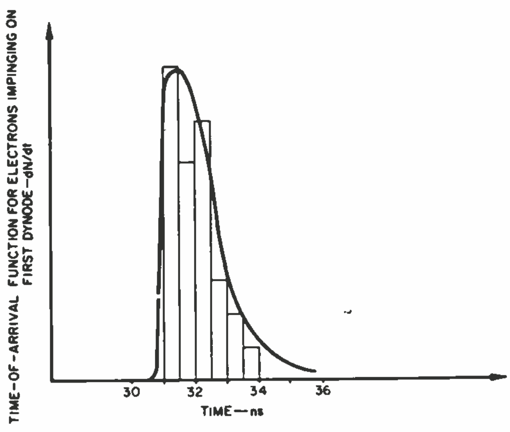


Fig. 8—Predicted time-response characteristic for a photocathode illuminated by a beam of small diameter focused on the axis of symmetry

# Filter network and function analysis by computer

**F. M. Brock**

**This paper describes two computer programs—ANALYS and FUNVAL—that can be used to determine filter network parameters and to evaluate filter network functions along the real axis. Both programs are written in FORTRAN IV for the Spectra 70/45 and can be used for a wide range of filter designs.**

**Frank M. Brock**
Carrier and Data Engineering
Microwave Engineering Dept.
Commercial Electronic Systems Division
Camden, N.J.

received the BSEE from the University of Alberta, Canada, in 1950. He received the MSEE from the University of Pennsylvania in 1962. Following graduation in 1950, he worked on the G.E. Test Course with the Canadian General Electric Co. in Canada, and then joined the Electronics Dept. of C.G.E., working on radar and microwave communications equipment. In 1953 he joined the Microwave Engineering Dept. of RCA in Camden, N.J. to participate in the design and development of frequency division multiplex communications equipment and systems. From 1954 to 1957, Mr. Brock contributed to the original development program for the AN/GRC-50 radio relay equipment, in the areas of local service telephone system practices and circuits, and feedback amplifier design. From 1957 to 1961, he contributed to the design of custom voice and data multiplex equipment, intermodulation noise test sets, and special communications systems wave filters. Since 1961, he has acted as Project Engineer for the development of the CT-42 solid state FSK telegraph data equipment; the development of precision filters and networks for the CV-600 voice multiplex equipment; and for advanced development studies in network synthesis and data communication systems. Mr. Brock is a Registered Professional Engineer of the Province of Ontario, and a member of the IEEE. He holds 3 U.S. patents.

One of the most important and frequently used applications of a computer in the design of filters and networks is the steady-state analysis of a designed filter network from its schematic. The availability of a computer program to quickly compute the pertinent characteristics of filter networks provides assurance that the proposed design has no gross errors, and allows evaluation of the effects of component variations and parasitics prior to model construction.

## Network Analysis

Since most filter networks are of the ladder type, a computer program can advantageously use the repetitive series/shunt branch immittance relations to provide fast computation of the network parameters with high accuracy. During the past two years, such a ladder network analysis computer program has been specifically developed and extensively used for filter designs.

This program, given the name ANALYS, will compute the following output parameters at each given frequency:

1) Insertion loss in dB;
2) Insertion phase shift in degrees;
3) Input resistive impedance in ohms;
4) Input reactive impedance in ohms;
5) Input reflection coefficient magnitude;
6) Input reflection coefficient phase in degrees; and
7) Input return loss in dB.

An added feature is the (optional) ability to use an included subroutine for automatic graph plotting (using the computer page printer) of any or all of output parameters 1), 2), 3), 4), and 7) above.

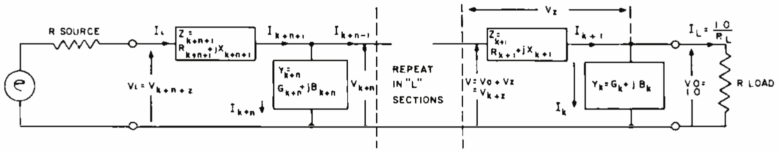
The basic program operation is shown on Fig. 1. A specific ladder structure with a series branch facing the source and a shunt branch facing the load (L-section) is assumed. Resistive source and load terminations are also assumed with unity output volts across the load resistor. Thus, the load current is known, and the current into the first shunt branch can be computed. This current, added to the load current, flows through the next series branch producing a voltage drop which can be summed with the output voltage to given an input voltage to the L-section. These voltages and currents can be complex for the general case where the branches contain *R*, *L*, *C*, *G*. This operation of determining complex branch voltages and currents is repeated, working L-section by L-section, until the source end of the network is reached. Then computations are performed to obtain the magnitude and phase of the source voltage *e*, with and without the network, thus providing the insertion loss relations. From the voltage and current computed for the input branch, the input impedance, reflection coefficient and return loss (using the specified source termination as a reference impedance) can be computed. Those output parameters are then printed, the analysis frequency incremented, and the entire computational process repeated. The analysis frequencies are expressed as a starting frequency (FS), an incrementing frequency (DELF), and a total number of analysis frequencies required (NF) up to 999 Fig. 2 shows the types of branch element configurations presently available.

The present program accommodates ladder networks of up to 100 branches and up to 100 each of *R*, *L*, *C*, *G* elements. No active elements or mutual couplings are allowed. Fig. 3 shows the input data format for the following example filter. Note that this input branch and element data, and source and load resistance data, are printed on the computer output sheets for convenience in identifying networks. Fig. 4 shows a bandpass filter ladder network with lossy coils to be analyzed. The individual L-section types are indicated for the analysis input, and Fig. 5 shows the input data for this example, as printed on the computer output, together with a portion of the output.

Graphs of the various output parameters may be requested by command cards included with the input data, and may be dated and titled from information cards also supplied with the input data. Fig. 6 shows a typical insertion loss graph plot of the output parameters for this bandpass filter (BPF).

---

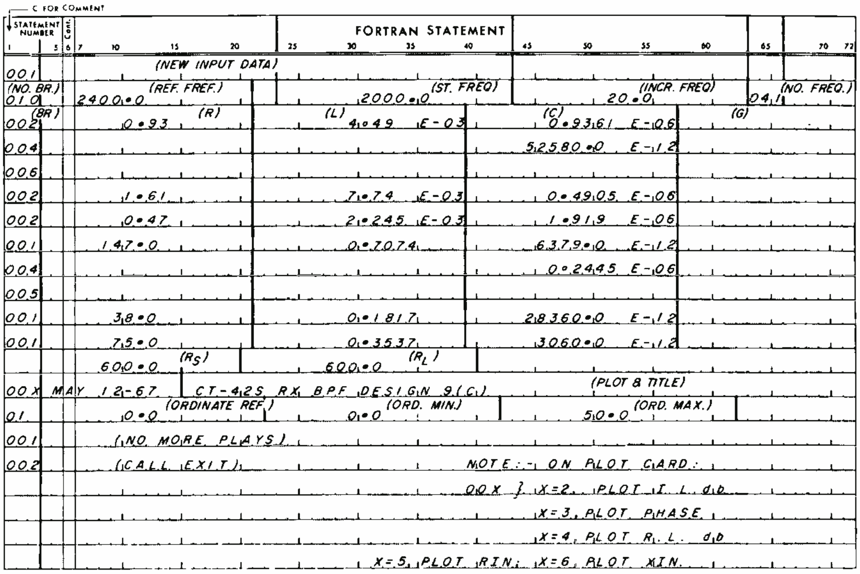
RCA Reprint RE-14-3-8
Final manuscript received June 25, 1968.



INPUT IMPEDANCE $Z_i = \frac{V_i}{I_i}$  TRANSFER CHARACTERISTICS $G = \frac{V_0}{V_i}$  $I = \frac{I_L}{I_i}$

INSERTION LOSS RATIO $L = \frac{I_i R_s + V_i}{(R_s + R_L) I_L}$  $T = \frac{V_0}{e} = \frac{V_0}{V_i + I_i R_S}$

Fig. 1—Ladder network analysis.



```
        (NEW INPUT DATA)
001
(NO. BR.)      (REF. FREF.)              (ST. FREQ)               (INCR. FREQ)    (NO. FREQ.)
010     2400.0                    2000.0                     20.0           041
(BR)           (R)            (L)                  (C)                    (G)
002        0.93            4.49   E-03         0.9361   E-06
004                                            52580.0   E-12
006
002        1.61            7.74   E-03         0.4905   E-06
002        0.47            2.245  E-03         1.919    E-06
001       147.0            0.7074              6379.0   E-12
004                                            0.2445   E-06
005
001       38.0             0.1817              28360.0  E-12
001       75.0             0.3537              3060.0   E-12
          600.0 (RS)       600.0 (RL)
                                                     (PLOT & TITLE)
00X MAY 12-67   CT-42S RX BPF DESIGN 9(C)
              (ORDINATE REF)        (ORD. MIN)                (ORD. MAX.)
01         0.0             0.0                      50.0
001       (NO MORE PLAYS)
002       (CALL EXIT)                       NOTE:- ON PLOT CARD:
                                            00X } X=2, PLOT I.L. db
                                                  X=3, PLOT PHASE
                                                  X=4, PLOT R.L. db
                              X=5, PLOT RIN; X=6, PLOT XIN.
```

Fig. 3—Input data format for analysis of the network shown in Fig. 4.

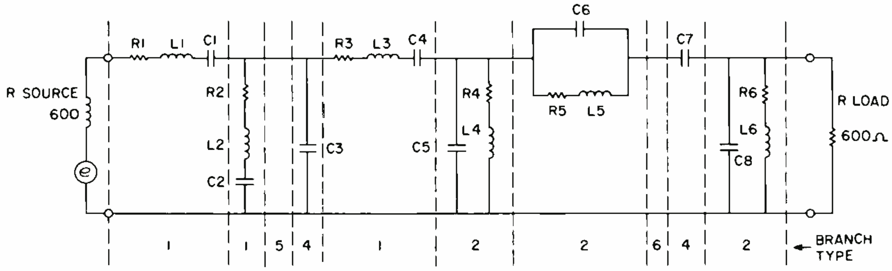


Fig. 4—Example BPF for analysis.

An added feature in the graph plotting routine is the adjustable ordinate scaling, which allows the full ordinate scale to be used for any range of the output parameter to be plotted. For example, the full ordinate range on the insertion loss graph can represent 100 dB, or 10 dB, or 1 dB, etc., thus providing a "magnified" plot of the insertion loss variation with frequency.

The ANALYS program compiles approximately 58 k bytes on a Spectra 70, using the TDOS system compiler. Exclusive of compiler/system control cards and data cards, the source program deck includes 250 cards, and compilation time is about 1 minute.

Execution time is a function of the number of ladder branches, the number of analysis frequencies required, and the number of graphical plots required. For a typical 10 branch, 6 coil, 8 capacitor BPF, analyzed for 41 frequencies with 4 graphical plots, the execution time would be about 2 minutes.

Some significant points in the use of program ANALYS are:

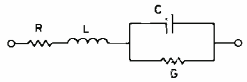
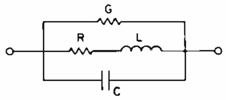
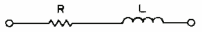
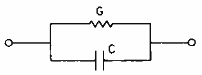
| BRANCH TYPE | BRANCH NETWORK |
|---|---|
| 001 |  |
| 002 |  |
| 003 |  |
| 004 |  |
| 005 | SERIES SHORT o——o |
| 006 | SHUNT OPEN O O |

Fig. 2—Available branch element configuration.

1) Network branches facing load and source must be shunt and series branches respectively (or use branch 006 "shunt open" and/or branch 005 "series short").

2) Branch types 006 and 005 can be used to break a complex branch into several simpler branches to comply with the allowable branch types (Fig. 2).

3) Because of the output print format selected for RIN, XIN (F8.3) on the output parameter print sheet, the maximum values of these values are restricted to ≤ ±99,999.999. Should a value larger than this be computed, the ± sign and leading digits may be lost, since these numerics are right-justified in a 10 column field. The same warning applies to the graphical plot of these values, since the labelling of the ordinate scale is limited to numeric values ≤ ±9,999.99 (F8.2).

4) Element values may range from 0.0 to $\pm 1.0\text{x}10^{74}$ (recommended range is $\pm 1.0\text{x}10^{\pm 20}$).

5) Source and load terminations are pure resistances, $\geq 1.0\text{x}10^{-20} \leq 1.0\text{x}10^{20}$.

6) Up to 999 frequencies may be computed for each run.

7) As many runs as desired may be executed automatically and sequentially by stacking input data sets. Each data card set should be complete. Note that each data set must start with control card 001, to read data. The last data set *must* be terminated by control card 002, which calls exit. Each separate data set must be closed by a control card specifying whether the graphical plots are required. The numerics for this are (Col. 1-3):

001 no plot required
002 plot INSERTION LOSS
003 plot PHASE
004 plot INPUT RETURN LOSS DB
005 plot INPUT RESISTANCE OHMS
006 plot INPUT REACTANCE OHMS

```
     LADDER NETWORK ANALYSIS     PROGRAM ANALYS-1    FMB  5/22/67

              RUN    1

              INPUT DATA

BRANCH TYPE        R                  L                  C                  G

      2       0.92999995E 00     0.44899993E-02     0.93609998E-06     0.00000000E 00
      4       0.00000000E 00     0.00000000E 00     0.52579999E-07     0.00000000E 00
      6       0.00000000E 00     0.00000000E 00     0.00000000E 00     0.00000000E 00
      2       0.16099997E 01     0.77399984E-02     0.49049999E-06     0.00000000E 00
      2       0.46999997E 00     0.22449999E-02     0.19189993E-05     0.00000000E 00
      1       0.14700000E 03     0.70739996E 00     0.63789969E-08     0.00000000E 00
      4       0.00000000E 00     0.00000000E 00     0.24449997E-06     0.00000000E 00
      5       0.00000000E 00     0.00000000E 00     0.00000000E 00     0.00000000E 00
      1       0.38000000E 02     0.18169999E 00     0.28359999E-07     0.00000000E 00
      1       0.75000000E 02     0.35369998E 00     0.13059999E-07     0.00000000E 00
  RSOURCE =  0.60000000E 03   RLOAD =     0.60000000E 03

FREQ, HZ           IL,DB   PHASE DEG     R IN       X IN       REFL.MAGN      REFL.PH.DEG    RL,DB
2.4300000E 03      4.632     5.089     607.102      2.928   6.3645579E-03     22.266      43.925
1.9999988E 03     57.803    45.945      80.446  -1838.402   9.7455460E-01    323.908       0.224
2.0199980E 03     56,000    48.471      81.039  -1724.918   9.7131187E-01    321.710       0.253
2.0399976E 03     54,142    51.262      81.747  -1611.955   9.6744555E-01    319.252       0.287
2.0599971E 03     52,230    54.352      82.602  -1499.336   9.6278375E-01    316.483       0.329
2.0799963E 03     50.270    57.777      83.653  -1386.823   9.5708996E-01    313.337       0.381
2.0999958E 03     48,275    61.565      84.966  -1274.136   9.5002735E-01    309.728       0.445
2.1199954E 03     46.274    65.712      86.646  -1160.888   9.4110978E-01    305.541       0.527
2.1399949E 03     44,324    70.108      88.854  -1046.574   9.2960113E-01    300.616       0.634
2.1599941E 03     42,547    74.311      91.862   -930.429   9.1433364E-01    294.725       0.778
2.1799937E 03     41,198    76.627      96.160   -811.334   8.9334834E-01    287.529       0.980
2.1999932E 03     40.653    69.620     102.735   -687.450   8.6305761E-01    278.490       1.279
2.2199924E 03     37.813    31.469     113.893   -555.581   8.1606865E-01    266.707       1.765
2.2399919E 03     26,722    12.754     136.338   -409.990   7.3438680E-01    250.593       2.682
```

Fig. 5—Computer input data for the example in Fig. 4.

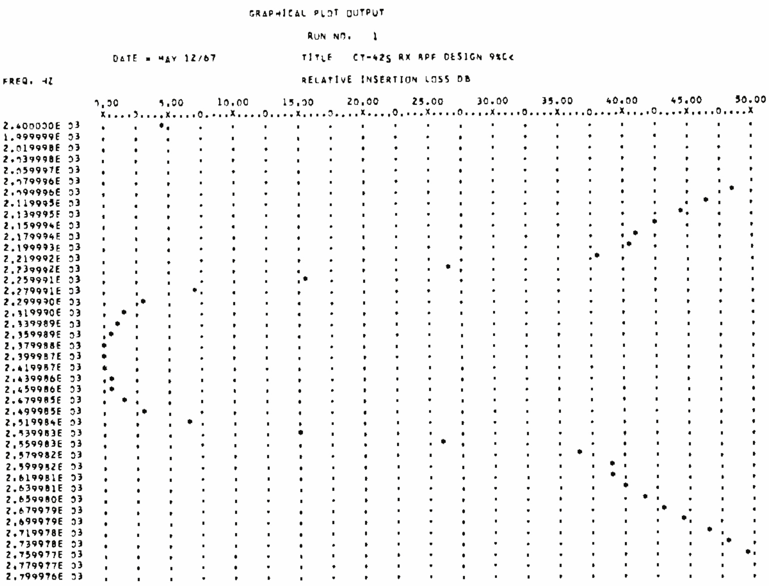


Fig. 6—Typical insertion-loss graph plot.

Given a network transmission function of the form:

$$F(s) = H\left[\frac{\prod_{i=1}^{10}(s+\sigma_{oi})\prod_{i=1}^{20}(s+s_{coi})(s+s_{coi})}{\prod_{i=1}^{10}(s+\sigma_{pi})\prod_{i=1}^{20}(s+s_{cpi})(s+s_{cpi})}\right]$$

$$= H\left[\frac{\prod_{i=1}^{10}(s+\sigma_{oi})\prod_{i=1}^{20}(s^2+2s\sigma_{coi}+\sigma^2_{coi}+\omega^2_{coi})}{\prod_{i=1}^{10}(s+\sigma_{pi})\prod_{i=1}^{20}(s^2+2s\sigma_{cpi}+\sigma^2_{cpi}+\omega^2_{cpi})}\right]$$

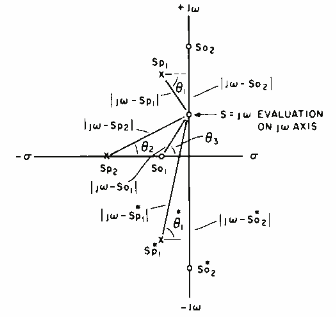
Evaluates: Magnitude db $= 20\log_{10}|F(j\omega)|$

Phase $= \theta = \tan^{-1}\left|\dfrac{I_m F(j\omega)}{R_e F(j\omega)}\right|$ } from pole-zero locations of $F(s)$

Envelope delay $= \dfrac{d\theta}{d\omega}$

where: $s$ = complex frequency = $(\sigma + j\omega)$
$H$ = constant multiplier not equal to zero.
$\sigma_{oi}$ = real zero in r.p.s.
$\sigma_{pi}$ = real pole in r.p.s.
$\sigma_{coi}$ = complex conjugate zero pair real part in r.p.s.
$\omega_{coi}$ = complex conjugate zero pair imaginary part in r.p.s.
$\sigma_{cpi}$ = complex conjugate pole pair real part in r.p.s.
$\omega_{cpi}$ = complex conjugate pole pair imaginary part in r.p.s.

Fig. 8—Mathematical form of the network function analysis.



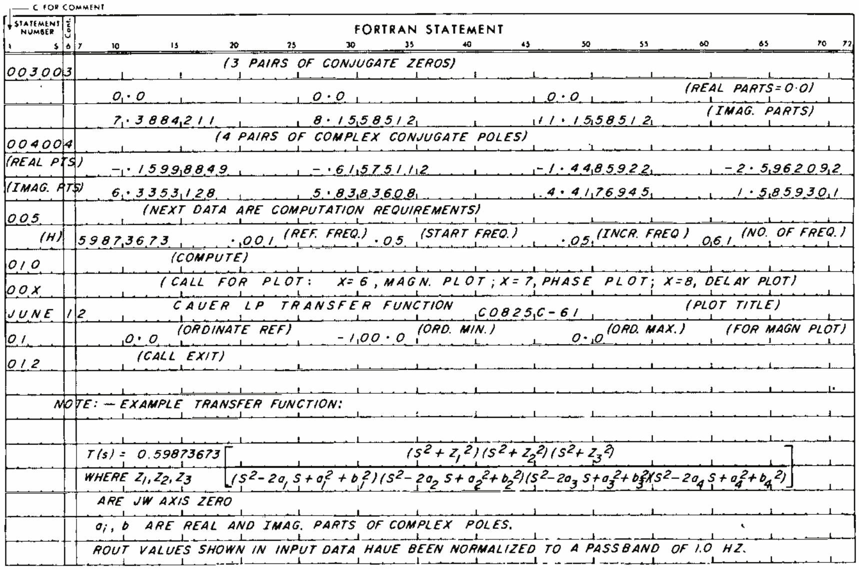
In general:

$$A(\omega) = 20\left[\log H + \sum_{i=1}^{n}\log(j\omega - s_{oi}) - \sum_{i=1}^{n}\log(j\omega - s_{pi})\right]$$

$$\theta(\omega) = (m-n)\pi + \sum_{i=1}^{n}\arg(j\omega - s_{oi}) - \sum_{i=1}^{n}\arg(j\omega - s_{pi})$$

and $A(\omega)$ is in dB.

Fig. 7—Typical network function evaluated in the S-plane.



```
003003          (3 PAIRS OF CONJUGATE ZEROS)
        0.0              0.0                 0.0           (REAL PARTS=0.0)
        7.3884211        8.1558512           11.1558512    (IMAG. PARTS)
004004          (4 PAIRS OF COMPLEX CONJUGATE POLES)
(REAL PTS)  -.15998849   -.61575112   -1.4485922   -2.5962092
(IMAG. PTS)  6.3353128    5.8383608    4.4176945    1.5859301
005             (NEXT DATA ARE COMPUTATION REQUIREMENTS)
     (H) 59873673  .001 (REF. FREQ.) .05 (START FREQ.)  .05 (INCR. FREQ) 061 (NO. OF FREQ.)
010             (COMPUTE)
00X             (CALL FOR PLOT:  X=6, MAGN. PLOT; X=7, PHASE PLOT; X=8, DELAY PLOT)
JUNE 12         CAUER LP TRANSFER FUNCTION   C0825C-61     (PLOT TITLE)
01        0.0 (ORDINATE REF)  -100.0 (ORD. MIN.)  0.0 (ORD. MAX.)  (FOR MAGN PLOT)
012             (CALL EXIT)

     NOTE: — EXAMPLE TRANSFER FUNCTION:
```

$$T(s) = 0.59873673\left[\frac{(S^2+Z_1^2)(S^2+Z_2^2)(S^2+Z_3^2)}{(S^2-2a_1S+a_1^2+b_1^2)(S^2-2a_2S+a_2^2+b_2^2)(S^2-2a_3S+a_3^2+b_3^2)(S^2-2a_4S+a_4^2+b_4^2)}\right]$$

WHERE $Z_1, Z_2, Z_3$
ARE JW AXIS ZERO
$a_i$, $b$ ARE REAL AND IMAG. PARTS OF COMPLEX POLES.
ROUT VALUES SHOWN IN INPUT DATA HAUE BEEN NORMALIZED TO A PASSBAND OF 1.0 HZ.

Fig. 9—Typical input data for cover-type 8th-order elliptic function.

If one or more plots are called for, a control card 001 must follow to indicate that no more plots are required.

8) Note that the output phase parameter values cover 4 quadrants from ≥ 0.0° to <360.0°, with an ambiguity of 360 n°; i.e.: whenever the phase value exceeds 360.0° it reverts to +0.0°.

9) Note that the assumption of a constant $R$ or $G$ associated with each $L$ or $C$ implies a reactive element $Q$ which increases proportionately and linearly with frequency.

## Function Analysis

Another useful program, given the name FUNVAL, has been developed to evaluate filter network functions along the real frequency axis. These network functions can be transfer functions, insertion loss functions, characteristic functions, or driving-point immittance functions.

The network functions must be expressible as the ratio of rational polynomials in complex frequency $S = \sigma \pm j\omega$. The numerator and denominator must be available in factored form as real and complex conjugate root factors; i.e.: as poles and zeros.

The program, in its present form, will handle polynomials up to 100th order —including 20 real roots (including roots at the origin) and 40 complex conjugate root pairs. Thus, the network function to be evaluated may have up to 20 real zeros and 20 real poles, including roots at DC, and may have up to 40 complex conjugate zero pairs and 40 complex conjugate pole pairs.

Evaluation at real discrete frequencies is performed automatically for up to 999 frequencies by setting an initial frequency, an increment frequency, and the number of frequencies required. Also, a reference frequency may be selected, and the output data will then be expressed relative to the output data at the reference frequency. Frequencies must be in Hz, and the function root locations must be in radians/sec. The output data gives: frequency in Hz, magnitude in dB, phase in degrees, and envelope delay in seconds. The envelope delay is defined as: $D = db/dw$ at test frequency.

Fig. 7 shows a typical *s*-plane view of the evaluation process for a network function with two complex conjugate poles, one real pole, and two conjugate zeros on the $j\omega$ axis, as evaluated at the specific $j\omega$ frequency indicated. The required mathematical form of the functions is shown in Fig. 8.

The sign of the phase and delay associated with a LHP function pole has been set as positive. Thus, for conventional transfer functions, the delay will be positive, and the phase will increase with positive slope in the passband. This is in accord with the usual circuit theory concept that a lagging phase angle has positive sign, and increases positively as the frequency is increased.

```
                              RUN NO.  1
                    NETWORK TRANSFER FUNCTION DATA
   NO. OF REAL ZEROS =   0
   NO. OF REAL POLES =   0
   NO. OF COMPLEX CONJ. ZERO PAIRS =   3
   NO. OF COMPLEX CONJ. POLE PAIRS =   4
   H =   5.9873670E-01
                              ROOT  VALUES
               SIGMA (RPS)                           OMEGA (RPS)

COMPLX ZERO     0.0000000E 00                        7.3884211E 00
COMPLX ZERO     0.0000000E 00                        8.1558504E 00
COMPLX ZERO     0.0000000E 00                        1.1155850E 01
COMPLX POLE    -1.5998846E-01                        6.3353119E 00
COMPLX POLE    -6.1575109E-01                        5.8383608E 00
COMPLX POLE    -1.4485922E 00                        4.4176941E 00
COMPLX POLE    -2.5962086E 00                        1.5859299E 00

FREQ. HZ          MAGN. DB          PHASE DEGR.          DELAY SECS.
9.9999993E-04    -2.0115370E-01     2.6594931E-01       7.3874950E-01
9.9999964E-02    -4.0650964E-03     2.6441483E 01       9.0295672E-03
1.1999995E-01    -7.8471303E-03     3.1837723E 01       1.2477577E-02
1.3999993E-01    -1.3668358E-02     3.7259781E 01       1.6180992E-02
1.5999991E-01    -2.2024214E-02     4.2708984E 01       1.9997239E-02
1.7999989E-01    -3.3282518E-02     4.8185638E 01       2.3798048E-02
1.9999987E-01    -4.7742009E-02     5.3689270E 01       2.7477860E-02
2.1999985E-01    -6.5469352E-02     5.9218735E 01       3.0968249E-02
2.3999983E-01    -8.6415589E-02     6.4772614E 01       3.4249485E-02
2.5999981E-01    -1.1020762E-01     7.0349442E 01       3.7357926E-02
2.7999979E-01    -1.3626027E-01     7.5948273E 01       4.0394604E-02
2.9999977E-01    -1.6375172E-01     8.1569214E 01       4.3525815E-02
3.1999975E-01    -1.9163787E-01     8.7213760E 01       4.6982646E-02
3.3999974E-01    -2.1867681E-01     9.2885178E 01       5.1056564E-02
3.5999972E-01    -2.4349785F-01     9.8589157E 01       5.6092501E-02
```

Fig. 10—Partial computer print-out of the network function being analyzed.

Program FUNVAL also includes the graph plotting subroutine, and provides optional plotting of any or all of the output parameters.

Fig. 9 shows typical input data required for a complex filter transfer function example: a Cauer-type 8th-order LP elliptic function providing a 60-dB stop-band attenuation. A partial output print for this example function is shown on Fig. 10.

Graphs of relative magnitude and envelope delay for this example function are shown on Figs. 11 and 12. In addition, Fig. 13 shows a "magnified" magnitude plot for the function pass-band.

The FUNVAL program was prepared in FORTRAN IV for the Spectra 70/45 computer, and compiles in approximately one minute computer time, requiring approximately 49.5 kbytes of memory storage. The source program card deck, exclusive of compiler control cards and data, includes 260 cards.

The computer run is automatic; each problem (i.e.: each set of data cards) is executed automatically and sequentially; output is printed on-line, one frequency at a time. The execute and print time per problem depends on:

1) Order of the polynomials involved;
2) No. of frequency points desired; and
3) The graphs requested to be plotted.

The example network function described above was compiled and executed in about 4 minutes.

Since the basic operation of the program in evaluating a function at a given real frequency requires the repetitive multiplication of factors which include the root values and the given frequency, it is possible to generate a very large numeric during these multiplications if large root values and frequencies are used. If a numeric is generated which is larger than the allowable floating-point numeric value for the Spectra 70 compiler ($10^{74}$), an error halt will occur. To eliminate this hazard, network function root values and evaluation frequencies should be scaled to the smallest numerics possible. The program does contain automatic checks for floating-point numeric "overflow," and when this occurs, will cause a warning to be printed before exiting the run from the Spectra 70 monitor system.

## Conclusion

Programs ANALYS and FUNVAL have proven invaluable in assisting filter and network design, and are illustrative of the speed, flexibility, and low user cost now possible with FORTRAN IV scientific computer programs for use on the Spectra 70 series computers.

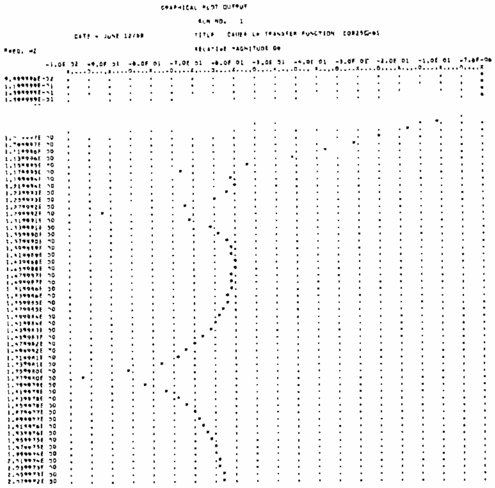


Fig. 11—Relative magnitude plot.

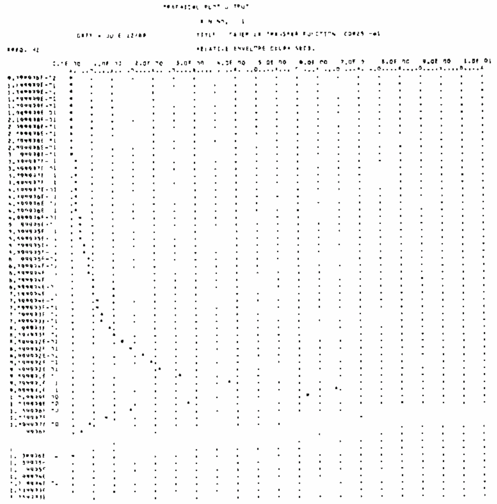


Fig. 12—Envelope delay plot.

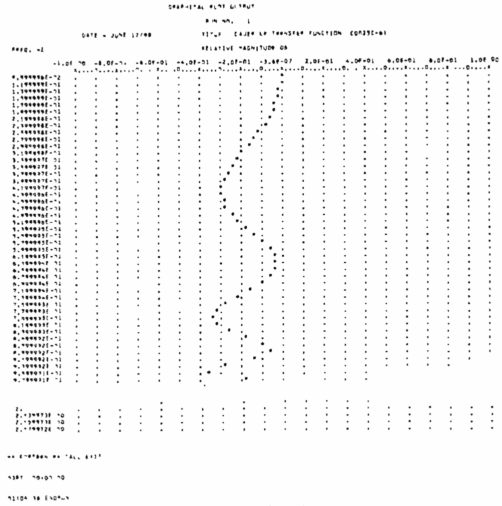
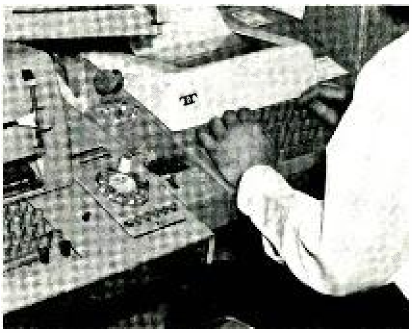


Fig. 13—Expanded magnitude plot.

# Engineering Programs Available



The following is a partial listing of typical engineering and scientific application programs presently being run on the Spectra 70/45. An attempt to update and complete this list is being made; therefore anyone having a program of general interest that is reasonably well documented is encouraged to contact M. G. Pietz, Camden, PC 2853. Space limitations prevented the printing here of engineering application programs written within RCA for other computers, such as the RCA 301 and 501, and the IBM 360 and 7090. Lists of these programs can be obtained at the above number. As reported previously in the **RCA Engineer** (vol. 13, no. 6, p. 83), F. M. Brock (Camden, PC 6053) has a catalog of programs available through the NASA library service.

| Area of Application | Program Name | Program Function | Coding Language | Details Available From |
|---|---|---|---|---|
| Filter Analysis | ANALYS | Analyzes ladder filters for all pertinent steady-state filter characteristics and automatically plots graphs of selected characteristics. | Fortran IV | F. M. Brock, Camden, PC 6053 |
| | FUNVAL | Evaluates complex networks or system functions for amplitude, phase, and delay, and automatically plots graphs of selected characteristics. | Fortran IV | F. M. Brock, Camden, PC 6053 |
| Simulators | DATSIM | Provides a restricted simulation of an FSK data transmission system through given TX/RX BPF defined as points of attenuation and phase vs. normalized frequency. | Fortran IV | F. M. Brock, Camden, PC 6053 |
| | Linear Programming | Provides capability for optimizing complex design problems subject to multiple constraints. Gives optimum solution, sensitivity analysis of variable parameters, and capabilities for producing detail reports. | Fortran IV | C. Robbins, Cherry Hill, PY 6065 |
| | Flow Simulator | General purpose tool for modeling and examining the behavior of systems in management and engineering science areas. | Assembly | C. Robbins, Cherry Hill, PY 6065 |
| Design Automation Aids | Placement-Routing-Folding (PRF) | Places, routes, and folds elements of an LSI MOS chip and generates input data for the ARTWRK program. | Fortran IV | R. Noto, Camden, PC 6755 |
| | ARTWRK | Converts output of PRF program or manually prepared input to control tapes for the Gerber plotter. | Fortran IV | C. Pendred, Camden, PC 2321 |
| | APT | Reduces effort required in programming numerically controlled tools. Input is English-type commands describing movements required by the tool to produce a given part; final system output is a punched paper tape containing all information for producing desired part on the machine tool-control unit. | Fortran IV, Assembly | C. Robbins, Cherry Hill, PY 6065 |
| | HOLES | With information obtained from drawings for hole locations for PC and/or stripline boards, this program provides the necessary input cards for the Gerber plotters and the digtal system drilling machine along with hole schedules for both the plotter and drilling machine. | Fortran IV | M. R. Diezel, Camden, PC 3222 |
| Mechanical Analysis | BEARING | Determines the pressure profile, load, and center of pressure for an arbitrarily shaped thrust (or slider) bearing, that rides on a flat moving surface. | Fortran IV | R. D. Scott, Camden, PC 5442 |
| | MAIN | Calculates the solution of the temperature profile over a rectangular plate with various boundary conditions along the edges of the plate. | Fortran IV | R. D. Scott, Camden, PC 5442 |
| | INERT | The moments of inertia for a complex three-dimensional body are determined by decomposing the body into elemental shapes. Inertia tensor, center of mass, and principal moments of inertia are computed. | Fortran IV | R. D. Scott, Camden, PC 5442 |
| Engineering Management Reports | CHARGE | Summarizes time-shared computer-terminal connect time and CPU time according to activity, shop order, and computer used. Input data comes from log sheets kept at teletype terminal. | Fortran IV | R. S. Miles, Camden, PC 6266 |
| | USORT | Provides a Unit-Bill-of Material from CADRE 2 components parts data. Preliminary version for CADRE 2 system. | Fortran IV, Assembly | R. S. Miles, Camden, PC 6266 |
| | ISORT | Makes an Indented-Parts-List from stored CADRE 2 component parts data. This is a preliminary version for the CADRE 2 system. | Fortran IV, Assembly | R. S. Miles, Camden, PC 6266 |
| | PTSLST | Provides parts-lists for the content of a master file in the CADRE 2 system. General version of a specific routine available on time-sharing version of CADRE 2. | Fortran IV, Assembly | R. S. Miles, Camden, PC 6266 |
| Plotting Subroutines | SICKLE | Transfers tape data to Calcomp plotter. | Assembly | C. Pendred, Camden, PC 2321 |
| | SCPLOT | Graphing routine for S-C 4020. Permits plot of X vs. Y for specified increments on axes, plot of X vs. multiple Y's for specified increments on axes, or X vs. Y for specified increment of X within a range. | Fortran IV | K. Gianapolis, Burlington, 2965 |
| | PLOT | Provides true perspective projection of a set of points specified by three-dimensional coordinates. The viewpoint and projection plane can be placed in any orientation with respect to the point. | Fortran IV | R. D. Scott, Camden, PC 5442 |
| | GERCAL | Converts Gerber outline tape to Calcomp plotter tape. | Fortran IV, Assembly | C. Pendred, Camden, PC 2321 |
| | MTECAP | A program to be used in conjunction with ECAP to produce a paper tape which will in turn produce a graph of the output on a DVST graphical display unit. | Fortran IV, Assembly | R. S. Miles, Camden, PC 6266 |
| Mathematical and Statistical Subroutines | Scientific Subroutine Package (SSP) | Over 200 subroutines in various mathematical and statistical areas. | Fortran IV | J. Peters, Camden, PC 5118 |
| | Statistical System | Series of statistical programs and a monitor system that provide analysis of variance, factor analysis, regression and correlation, nonlinear regression, and function minimizer. | Fortran IV | C. Robbins, Cherry Hill, PY 6065 |
| | Scientific Subroutine System | Contains complex and real matrix subroutines. general matrix routines, complex arithmetic routines, special engineering programs are available to compute binomial coefficients, integrate first-order differential equations, compute Fresnel integral, compute Fourier coefficients, and generate random nos. | Fortran IV | C. Robbins, Cherry Hill, PY 6065 |
| | Square Root and Transcendental Package | Subroutines to evaluate the following functions: square root, exponential ex, natural log, sine and cosine, arctangent, hyperbolic tangent. All subroutines available in single or double precision floating point. | Assembly | C. Robbins, Cherry Hill, PY 6065 |
| | Differential Equation Solver | Gives numerical solutions for differential equations using principles of analytic continuation of a Taylor series. | Fortran IV | R. Noto, Camden, PC 6755 |
| | Hypergeometric | Computes the sum of a given hypergeometric series. Double precision. | Fortran IV | L. Toombs, Camden, PC 2321 |
| Miscellaneous Routines | NEWDEK | Converts program decks punched in IBM 7090 code to Spectra 70/45 code, or vice versa. | Fortran IV | C. Pendred, Camden, PC 2321 |
| | Flow Charting | Symbol-directed program flow charts programs coded in assembly language. | Assembly | J. W. Smiley, Camden, PC 4368 |
| | | Routine also exists to convert CSP-3A source language to flow chart program input. | Cobol | C. J. Moore, Camden, PC 2720 |
| | ALGOL-20 | Carnegie Tech's Algol compiler converted for the Spectra 70. Algol-60 is a subset of Algol-20. | Assembly | J. Peters, Camden, PC 5118 |
| | CSP-3A Assembly Language System | Complete assembly language compilation system for the CSP-3A processor (processor for ICS). | Assembly | C. J. Moore, Camden, PC 2720 |
| | Assembly Language System for R100 Processor | Modification of Spectra 70 assembly language program that provides assembly language system for the R100 processor. | Assembly | L. Hitch, Camden, PC 2864 |
| Logic & Circuit Analysis | LOGSIM IV | Verifies correctness of a logic design at the gate level prior to hardware fabrication. | Assembly | A. Cornish, Camden, PC 6735 |
| | Signal-Trace Program for Digital Logic | Prints out graphs of the elements or gates of a circuit that are electrically interconnected. Capacitance at each node is calculated and underload or overload conditions are printed out. Options include forward-net and reverse-logic trees. | Fortran IV | R. Noto, Camden, PC 6755 |
| | BLPR Input Translation | Converts logic connectivity input data developed for an IC layout design program into input for Logsim. | Assembly | J. W. Smiley, Camden, PC 4368 |
| | LBPR Input Translation | Converts input data developed for Logsim into input data for an IC layout design program. Logsim input is redescribed as more complex logic circuit types by grouping the circuits according to a specific program library of circuit types. | Assembly | J. W. Smiley, Camden, PC 4368 |
| | ECAP | Electronic Circuit Analysis Program produces DC, AC, and/or transient analysis of electrical networks. Various versions accommodate 30 nodes, 50 nodes, and plotted outputs. | Fortran IV | R. Crosby, Camden, PC 4864 |
| | CMOSTR | Performs dc and transient analysis for C-MOS, P-MOS, and N-MOS digital circuits with linear capacitors. | Fortran IV | A. Feller, Camden, PC 3257 |
| | CASMOS II | Performs DC and transient analysis for P-MOS circuits containing nonlinear capacitors. | Fortran IV | A. Feller, Camden, PC 3257 |

# Computer Centers at RCA

| Location | Contact | Computer Complement | Memory Capacity | Turn-Around Time (hour) | Cost $/hr | Remarks |
|---|---|---|---|---|---|---|
| Advanced Tech.<br>Camden, N.J. | M. Pietz | Spectra 70/45G | 262k bytes | 2 | 140 | Turn-around time is based on prime-shift operation; second and third-shift turn around is faster. |
| DCSD<br>Camden, N.J. | P. J. Helmer | Spectra 70/45<br>RCA-301<br>RCA-3301 | 131k bytes<br>10k char.<br>80k char. | — | — | This computer is for administrative-applications, scientific and engineering work is referred to the Advanced Technology computer center. |
| M&SR<br>Moorestown, N.J. | J. B. Vail | IBM-7090<br>IBM-1401<br>IBM-1401<br>Spectra 70/55 | 32,768 words<br>4000 char.<br>8000 char.<br>131k byte | 2<br>2<br>2 | 246<br>50<br>50 | The turn-around times and rates for the newly installed Spectra 70/55 have not yet been established. |
| | P. J. Dwyer | Spectra 70/45<br>RCA-301 | 131k bytes<br>10k char. | — | — | This computer is for management information and administrative applications. |
| ASD<br>Burlington, Mass. | F. Congdon | RCA-301<br>Spectra 70/45G | 40k char.<br>262k byte | 2 | 140 | The Spectra equipment is being installed in January; the 301 system will remain until April 69. Contact H. Brodie for all engineering applications. |
| AED<br>Hightstown, N.J. | O. J. Wenzel | Spectra 70/45G | 262k byte | less than 2 | 140 | Typical small jobs are handled immediately. Contact R. H. Goerss for all engineering applications. |
| Record Club<br>Indianapolis, Ind. | F. Giegerich | RCA-301<br>RCA-301<br>RCA-3301<br>RCA-3301 | 10k char.<br>20k char.<br>120k char.<br>120k char. | less than 2 | 75<br>180 | One RCA-3301 computer is shared with Record Operations, and one RCA-301 (20k) is shared with Record Operations. |
| ISD<br>Cherry Hill, N.J. | M. Longo | Five Spectra 70/45's<br>Spectra 70/35<br>Spectra 70/15<br>Spectra 70/25 | <br>65k byte<br>8k byte<br>65k byte | 24 | 100 | The memory capacities of the five Spectra 70/45's range from 65k bytes to 262k bytes. |
| EC<br>Harrison, N.J. | J. Lynch | Spectra 70/45<br>Spectra 70/45 | 131k byte<br>262k byte | per schedule<br>per schedule | 100<br>100 | There is considerable emphasis placed on time sharing in the EC engineering groups. The rates given are estimated. |
| Record Operations<br>Indianapolis, Ind. | R. Murphy | RCA-301 | 20k char. | 24 | 75 | |
| EC<br>Lancaster, Pa. | E. J. Brabits | RCA-301<br>Spectra 70/45 | 40k char. | 24 | 81 | The Spectra equipment is presently being installed; therefore cost and turn-around times have not yet been established. |
| EC<br>Marion, Ind. | J. Schrock | SDS-920 | | | | This computer is used to control the color picture tube automatic test equipment Test data generated by this real-time system are used by Marion engineers. |
| RCA Ltd.<br>Montreal, Canada | W. Kievit<br>G. Payette | Spectra 70/45 | 131k byte | 24 | | The billing rate has not yet been established since the computer is newly installed. Although the standard scheduling provides 24-hr. service, priority jobs are handled immediately. |
| GSD<br>Dayton, N.J. | E. Fagan | Spectra 70/45<br>Spectra 70/35<br>Graphic 70/822<br>Graphic 70/832 | 131k bytes<br>65k bytes<br>—<br>— | | 175<br>150<br>155<br>155 | Each department has computer time scheduled on a daily basis; during the scheduled time, the jobs for that department are run immediately. Overnight runs are also handled. |
| Inst. Systems<br>Palo Alto, Calif. | R. Hazeltine | Spectra 70/45 | 131k bytes | | | This is a real-time system which is interconnected with the Palo Alto educational system. |
| ISC<br>Los Angles, Calif. | M. Freeman | Spectra 70/45<br>RCA-301 | 262k bytes<br>20k char. | 24<br>24 | 125<br>125 | This center has the capability for handling inputs from remote card and paper-tape terminals. Since this is a new computer center, final billing rates have not yet been established. |
| CISC<br>New York, N.Y. | J. Branco | Three Spectra 70/45<br>RCA-301<br>RCA-3301 | 262k bytes<br>30k char.<br>160k char. | immediate<br>4<br>overnight | 125<br>100<br>75 | The three billing rates are based on priority assigned to the specific job being processed. |
| Laboratories<br>Princeton, N.J. | L. Berton | Two Spectra 70/45<br>RCA-601/604<br>RCA-301 (slave) | 262k bytes<br>16k words<br>10k char. | 2<br><br>2 | 180<br><br>325 | This center has the capacity for handling batch-run inputs from remote teletype terminals; the time sharing terminals provide this remote capability. |

# MIS and its relationship to the engineer

**B. G. Curry | J. R. Gates**

**The objective of RCA's internal MIS program is to provide timely information and analyses for all levels of RCA management as required for effective operation of the corporation's many businesses. In RCA, with its broad technical base, the engineer is an important part of the management to be serviced by the MIS program. This paper briefly discusses several of the engineer's functions—e.g., information storage and reference, data reduction and analysis, problem solving, project management, and process definition and design—and shows how these functions can be well served by modern information systems.**

The RCA MIS program is best described by formal MIS long range plans for each division and for Corporate staff. Each year the MIS activity from each division, and Corporate staff, updates and submits their plans for major developments in their management information systems for the ensuing five years. These are reviewed with the Staff Vice-President, MIS, the applicable division General Manager, and other responsible management to insure that the plans and objectives are responsive to operating needs.

These plans have several purposes; e.g., to insure consistency with available manpower and with Corporate plans, and to act as a communications device through all levels of management. The plans are budgeted and controlled as part of each business plan at Division and Corporate Staff levels.

Guidance, counseling and (when appropriate) direct aid is supplied throughout RCA by the office of the RCA Staff Vice-President, Management Information Systems. Projects affected can be for Corporate Staff use only, single Divisional or multi-Divisional effort. This year, computer-based management and operating systems at RCA represent an expense of $50 million.

### Information Storage and Reference

The engineer or scientist must know what has been done and what is being done by others in technical fields related to his current tasks. This information must be extracted from large files of information, representing a variety of sources, ranging from technical reports through formally published material. To be effective, the systems to handle such files must condense and index the material so that an engineer can efficiently identify whether needed information is likely to be in the system, and if so, then be directed to highly relevant sources where he can go into greater depth.

RCA's Technical Information Systems (TIS) activity is presently designing and implementing systems to serve RCA engineers in these areas. This activity—part of Corporate Engineering Services, Research and Engineering—plans to provide a set of interrelated, computer-based information services both for engineering groups and for our technical libraries (which are, of course, important traditional links in the information system).

Present effort of TIS is focusing on services of the "current awareness" variety. This involves means for periodically alerting an individual or group having well-defined technical interest "profiles" as to the existence of newly issued information of relevance. These services also involve bulletins summarizing new information resources; such bulletins are prepared by computer screening and cross-referencing of source material so that they can be readily "browsed".

These are being supplemented with systems work on the problems of computerizing of technical library records and files—so that needed documents can be most effectively located.

Future systems work will be directed toward the searching of retrospective files (i.e., older as well as current information) and toward bringing into the system specialized information resources such as descriptions of work in progress (i.e., "who is doing what").

In other areas of engineering information, a major effort is being conducted jointly by the corporate MIS group and the Patents and Licensing activity. Its objective is to develop an integrated computer-based system for classification, dissemination, and retrieval of domestic and foreign patent information.

Other MIS systems concepts which will ultimately be of assistance to the engineer include configuration control, engineering drawing systems, and even preparation of documents, such as technical proposals.

### Data Reduction and Analysis

Tapping major manufacturing product systems for information feedback on product performance after leaving the drawing board provides another area of fruitful endeavor. Left to his own devices, an engineer can often get his needed data; however, a good system would present him with *information*, rather than bury him in *data*. This work calls into play instrumentation and data collection devices to pick up the data, computers to analyze and reduce the output to useful information and output devices such as plotters to bring the answers back to the engineer in a better form for decision making. Many times the system can help him ignore data which is irrelevant. Again, the *system* for managing the data and bringing out information is the important ingredient.

### Problem Solving

In this area, the engineer has done some of his own best work. Knowingly or not, he has set up information systems as an extension of his slide rule and desk calculator. He has arranged for the collection of measurement data and written computer programs to solve very complex mathematical problems. He has built mathematical models of his devices and allowed the computer to be a part of his management system in inspecting the inner workings of his devices. He has set up computer systems to simulate a circuit or even a group of circuits so that he could "build" without ever cutting metal or soldering a wire. The advent

---

Reprint RE-14-3-14
Final manuscript received September 13, 1968.

of time-sharing computers has caused a rapid acceleration of this type of computer use in engineering.

### Engineering Operations

Engineering systems in Operations break down two ways: engineering management and technical direction. To the engineer directing a project or group, project control is essential, and a good management information system in this area will allow the engineering manager to do more engineering and still know more of where he stands in his projects.

The engineer is often the key to a technical manufacturing operation. He may be the creator of a paper tape to drive a machine tool or the supplier of a formula to be used in a process. If he is involved in "cookbook" design, he may well wish to turn his formulas over to a computer management system to be handled so that he can get onto more important and creative activities.

One example of the role that MIS can play in technical operations is in the generation of a formal bill of materials. In the past, engineering and manufacturing prepared separate bills of materials for a particular product—neither list being fully satisfactory. As process controls were introduced, however, the process engineer generating the controlling flow chart was able to specify a formal bill of materials that was satisfactory in all respects. Thus, in addition to increasing efficiency through process control, the project documentation was improved significantly. Also, with the better controls better information was available on labor, cost, shrinkage, amount of materials used and in stock, and schedule estimates.

### Availability of Spectra Systems

As mentioned earlier, the introduction of time-sharing was an important factor in extending the computers to the engineer. For the larger engineering systems, however, the engineer will continue to use internal business computers. By the end of 1968, there will be some 27 third-generation Spectra 35, 45 and 55's internally available for RCA Engineers. The chart in the centerspread of this issue shows the location, cost, turn-around time and contact for each of these systems.

The key to effective engineering use of these systems lies in the formalized MIS plans for each division described earlier in this paper. The engineer must be a part of these plans and this requires a close communication between engineering and MIS management.

Engineering must then apply basic engineering principles to its computer needs by specifying performance for its key systems and then by working with MIS to develop a program for creating these systems. The nature of most systems will require participation with other functional areas.

How many times have you heard, or said yourself, "I can't get those computer people (the business computer) to process my work; I left input two weeks ago and they didn't get to it yet; that is the accountants computer we need our own"? These are complaints of long standing which today must be classed only as excuses. This feeling from the past is a source of great dissatisfaction among the engineering community and yet with RCA's present computer power and service-oriented MIS organizations *it is also being solved.* All RCA MIS organizations have published five-year plans. Engineering must insist on being an important part of these plans as to projects and computer usage. Division MIS management is committed to providing this service to insure the engineer's role as a major contributor to, and benefiter of, the MIS program.

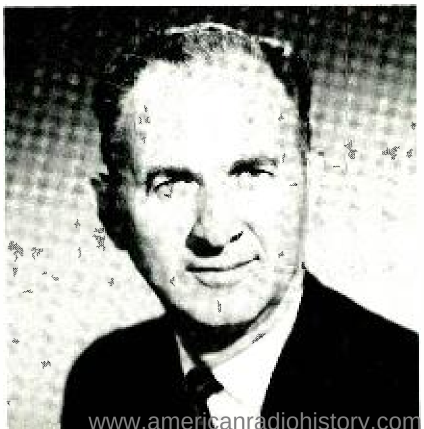
### Concluding Remarks

The relationship between MIS and the engineer will not in any way slacken off. Indeed, the tie will become stronger, especially because our products are becoming more technically complex. The engineer's information systems will become an integral part of the business systems. The process control computer will tend to merge with other computers. The automatic machine tool will get its direction as a by-product of the larger system. Much problem solving will be done on-line using data flowing from the management information system so that quick decisions can be made and fed back for control. Much of this work will be handled via time shared computers. The engineer is an important key to good MIS design. He must be prepared to understand, and be a party to, the systems surrounding him in business, if these systems are to serve his needs.

---

**Bruce G. Curry, Dir.**
Management Information Systems Programs
New York, N.Y.
received the BS degree in 1952 at the Massachusetts Institute of Technology and the MBA in 1955 from the University of Michigan. He joined the Radio Corporation of America in 1959 as Methods Manager, Cleveland District, in the EDP Division In the following year, he was made Methods Manager for the Eastern Region. In 1963, he assumed his present position on the RCA Corporate Staff where he is responsible for the Corporation's internal M.I.S. Program. Prior to his service with RCA, he was at International Business Machines Corporation in 1955 as a Technical Service Representative. He went to Dow Chemical in 1956, was engaged in distribution analysis and transportation studies for Dow Products, and subsequently served as Project Leader on an IBM 709 Project. He was with the Chrysler Corporation in a similar capacity during 1958-1959. Mr. Curry is a member of the ACM and was on the IFIP Congress 65 Data Processing Committee.

**John R. Gates, Mgr.**
Information Systems Planning
Management Information Systems
New York, N.Y.
received the BSEE from the University of Nebraska in 1942 and went to work for RCA. In his first 13 years with the Electron Tube Division, he held positions of Equipment Engineer, Manager, Mechanical Equipment Design, and Manager, Advanced Equipment Development. In 1947, he received the MS from Stevens Institute of Technology. He was awarded a Sloan Fellowship in 1955 to study at MIT, and received an SM in Industrial Management in 1956. In 1956, he assisted in forming the Automation Systems Development group for the Electron Tube Division, and was Manager, Automation Projects Administration; and Manager, Technical Systems. In 1963, he moved to the RCA Corporate Staff with a newly formed Mangement Information Systems group. Mr. Gates is a Member of the IEEE and the Institute of Management Sciences.



# A dialog about computers with a materials analyst

**J. R. Woolston**

**Characters**

MATERIALS ANALYST
COMPUTER EXPERT
COMPUTER

**Scene 1: The computer expert visiting a materials laboratory has just suggested that the materials analyst should use a computer to aid him in his calculations.**

ANALYST: Me—use a computer? But I'm an x-ray specialist, not a mathematician! I analyze materials, not functions or payrolls. Besides, if I really need to use a computer in my work I can get someone to take care of it for me. Can't I?”

EXPERT: That statement, or something like it, has been the traditional retort of you materials analysts whenever the subject of computers has been broached. But it is being heard less and less and someday, not too far off, it will sound as ridiculous as: “Me—drive a car?”.

The reason is simple. The statement represents several misconceptions that are rapidly being cleared up by a growing band of venturesome analysts who have taken the plunge and have found out that the water's not too deep, and is, in fact, rather comfortable! Let's look at some of these misconceptions. The first is that it takes some kind of mathematical genius or specialist to write computer programs, and that this task is somehow beyond the range of the analyst. Another is that computers are intended for use in complex mathematical work and for handling vast bookkeeping jobs. And the most persistent misconception is that an analyst can easily have someone else—a computer specialist, or programmer—prepare any computer program he might need for him. This last misconception is more than just an erroneous assumption—it represents naiveté.

ANALYST: All right, maybe so, but just why should I think about using computers in my work? Just for the sake of doing it? To be up-to-date?—modern?

EXPERT: By no means, but rather to save money, manpower, time, and to improve your analytical results.

ANALYST: Fine. I'm all for that, along with apple-pie and motherhood, but that's a pretty generalized answer. Besides, I thought computer time was expensive.

EXPERT: Let's talk about the expense first, then we'll get to some examples. OK?

ANALYST: OK.

EXPERT: Compared to your hourly pay, $200 to $500 per hour probably sounds pretty costly, but analytical problems are handled in a matter of seconds on a big computer, not the hours it takes to process large bookkeeping problems, and you pay only for the time you use. Then there's time-sharing, which we'll talk about in more detail, where the cost is only $10 to $15 per hour. How's that sound to you?

ANALYST: Groovy.

EXPERT: Admittedly, you'll spend some money getting your programs written and debugged, but once done you'll realize tangible savings. Of course, you'll find yourself handling a lot more analytical work with the time the computer saves you. And you may find it desirable to obtain some special equipment, some of it rather costly, to facilitate collecting your data for the computer to process.

ANALYST: I knew there was a catch somewhere!

EXPERT: Not really. With most analytical problems, often the greatest time-saving is obtained not by the computer itself, but via automatic data collection devices. Besides, these are capital expenses; your time isn't. Now let's get down to some concrete examples.

**J. Rogers Woolston**
Process and Materials Applied
Research Laboratory
Laboratories
Princeton, N.J.

received the BSE from Princeton University in 1955, in a combined program of electrical engineering and physics. He spent the summers of 1952 and 1953 at RCA Laboratories, and the summer of 1954 at Bell Telephone Laboratories, working primarily on semiconductor devices. Since 1955, he has been with RCA Laboratories, working initially on materials research in both thermoelectrics and intermetallic compound semiconductors. He received a U.S. Patent on a power transistor design, and designed and developed a crystal growth apparatus for GaAs. In 1958 he became a member of the Materials Research Laboratory where he has specialized on various aspects of solids mass spectrography and on computer processing of analytical data. This experience has included the design, development, and improvement of equipment and techniques for thermal vaporization analysis and for RF spark-source analysis. In addition, he has written many programs for the RCA 601 and Spectra 70 computers as aids to both analysis and research with the MS7 mass spectrograph, and for x-ray diffractometry. He is the author or coauthor of numerous papers and publications bearing on these subjects. Mr. Woolston is a member of ASTM Committee E-14.

---

Reprint RE-14-3-19
Final manuscript received July 25, 1968.

ANALYST: Fine.

EXPERT: First, let me explain that there are several ways in which a computer can be used to aid you. For instance, there's batch-mode usage, time-sharing, time-sharing with automatic data collection, even real-time!

ANALYST: Whoa! You've lost me already.

EXPERT: Oh, sorry about that. Well, batch mode is the most straightforward and by far the most common, historically. It simply means that when you've completed your analysis, you punch all your data on cards and submit them together with a program to a computer. Sometime later, usually the next day, you collect your output—your analytical results. You look it over (at least you should) and if it looks OK you send it on to your customer. See, here is a typical batch-mode output for a solids mass spectrographic analysis.

ANALYST: Neat.

EXPERT: In this case, the computer did all the mathematical calculations to get the individual results, which would have taken a good man most of a day to do by hand, then it geometrically averaged related results and printed them out in a nice, legible form.

ANALYST: It's better than my handwriting!

EXPERT: And it saves having a secretary type them. And after all, she's overhead.

ANALYST: Be careful how you talk about my secretary!

EXPERT: Programs of this sort can and have been written for almost every analytical discipline—x-ray diffractometry, emission spectrography, neutron activation, you name it.

ANALYST: Even head-shrinking?

EXPERT: Sorry, that's not my kind of analysis.

ANALYST: Well, if such programs have already been written, why can't I just get a copy of one of them and use it myself?

EXPERT: Perhaps you can. In rare cases this expedient has worked. But in most cases such programs have to be modified to suit your individual requirements, and who's going to do that for you? Modifying programs is usually very difficult, if not impossible for all but the person who originally wrote them. Don't get me wrong. Existing programs can be very useful to you as models around which to build your own, and there are some very good ones in the literature. Now, if you're interested in high resolution organic analysis by mass spectrometry, you can buy a complete package that includes the computer and the programs to do the job, all set to go. But that's a special case.

ANALYST: Sounds expensive.

EXPERT: It is, but if you have no computer at your location, it can be worth it. Actually, high resolution mass spectrometry is a very good example of a case where computers have not only aided the analyst, they have led to the creation of a whole new analytical discipline. Here you are working with literally thousands of six- or seven-digit numbers, the exact molecular masses, for every analysis—a task that would be impossible to do manually. I'm glad you mentioned it.

ANALYST: Mentioned what?

EXPERT: But let's get on to time-sharing.

ANALYST: Can I choose who I'm going to share it with?

EXPERT: Time-sharing is a fairly recent development of the computer art, and its potentialities for analytical applications have only begun to be explored. The big difference between it and batch mode operation is this: You are in the system.

ANALYST: Wow!

EXPERT: Consider this: Most computer programs are built like a tree. They start at the trunk, and as they go upward they encounter branches—decisions that have to be made. Each branching is followed by further branching, until the end is reached. In batch mode, you—the analyst and programmer—have to anticipate all the possible branchings, and since you may not know which way to go *a priori* at any particular branch, you have to allow the computer to follow all feasible branches, often producing the whole tree! This can result in a great deal of output.

ANALYST: Yes, I've seen people carrying great stacks of output away from our computer center.

EXPERT: With time-sharing, you can interact directly with the computer as it's processing your data. In other words, when a decision is needed, you can bring your vast analytical experience to bear on it and make it yourself.

ANALYST: Right or wrong!

EXPERT: Right. And if you do make the wrong decision, you'll know it right away and can correct it. A good example of this is lattice constant determination by x-ray powder pattern analysis.

ANALYST: Hey, that's my field!

EXPERT: OK then, let's look at how this time-sharing program is actually used. As you know, after running the sample in your x-ray powder camera you have a long strip of film with a lot of arc-shaped lines on it.

ANALYST: Yes, I know. I look at several of them every day. Too many!

EXPERT: Then you have to measure the exact positions of every line, go through a lot of work with a desk calculator to figure out the diffraction angles for each line, and then try to index the pattern to determine the crystallographic class and the lattice constant.

ANALYST: You don't have to tell me.

EXPERT: And perhaps your sample had more than one phase. What do you do then?

ANALYST: I usually give up.

EXPERT: You might, but a computer won't! Look, instead of my telling you about it, let's go and do it.

ANALYST: Here? But we haven't got a computer here!

EXPERT: Have you got a teletype in the building someplace?

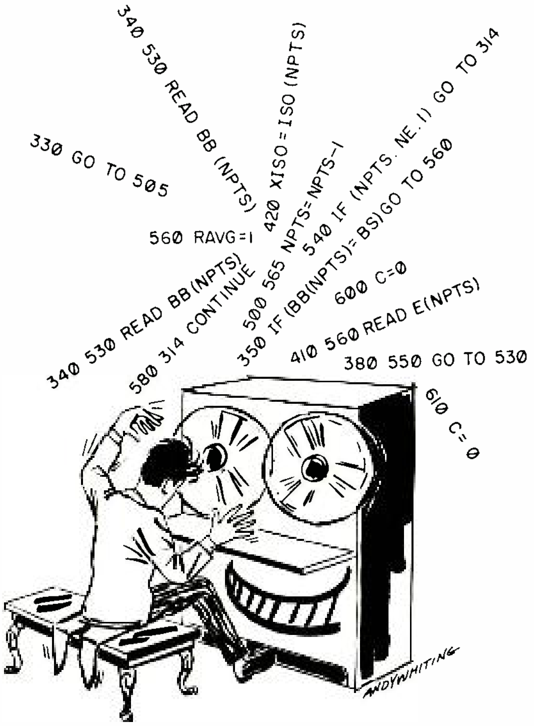
ANALYST: Yes, there's one down the hall.

EXPERT: Fine, we'll use the computer at the Labs in Princeton.

ANALYST: No kidding?

EXPERT: You can run the computer from Timbuktu or from your own house, if you install a teletype.

ANALYST: Man, that'd be great for my kids' homework.



**Scene 2: At the teletype console**

EXPERT: Now, all we do is dial the computer's number, give it a valid usercode, and type in the name of the program we want. Now, do you happen to have some x-ray powder camera data handy?

ANALYST: You bet I do! Here's some data from a sample that must have not two, but three very close phases in it. I've been having a terrible job trying to figure it out!

EXPERT: OK, let's have at it. First you see, the computer—the program, really—asks us a question. It's going to ask a lot more.

COMPUTER:

```
DO YOU WISH TO ENTER DATA FROM THE KEYBOARD?
A=YES
```

[*Note: the statements covered by tone have been typed in by the user.*]

EXPERT: You see, we answer each question yes or no or we provide the data it asks us for. Now it's asking for your data, so we type it all in. Now look what it says! You measured one of the lines incorrectly, and it has spotted the inconsistency in the data and it's telling you what it thinks the data ought to be.

ANALYST: No wonder I was having so much trouble with this sample! I did all that work for nothing.

COMPUTER:

```
THE NUMBER 6 FRONT REFLECTION LINE WAS READ IMPROPERLY:
PLEASE CORRECT.
MY CORRECTED VALUES ARE:
                    FRL(6)=144.016
                    FRR(6)=197.666
```

ANALYST: Man, that's cool! It even tells you what they should have been.

EXPERT: Right. So now we can use its values, or you can go and remeasure the film.

ANALYST: Look's like it knows what it's doing, so let's use its values.

EXPERT: Now it's listing your lines and their calculated diffraction angles and their *d*-values. And now we've got more questions to answer.

COMPUTER:

```
WOULD YOU LIKE ME TO ATTEMPT TO INDEX A POSSIBLE
CUBIC PHASE(S)?
A=YES

WHICH LINE SHALL I START WITH?
LINE=1

DO YOU KNOW THE N-VALUE OF THE LINE?
A=NO

WHAT TOLERANCE SHALL I USE?
TOLERANCE=0.3
```

EXPERT: Of course we want to try to index your lines, and we may as well start with the first line. I recommend a trial starting tolerance of 0.3, and let's pretend that we don't know the n-value of the first line, though it's probably a 111.

ANALYST: I'll buy that.

EXPERT: Look, it's doing a pretty good job on the first phase.

COMPUTER:

```
    #        N       HKL        A
    1        3       111        5.6579
    3        8       220        5.6511
    6       11       311        5.6554
    9       16       400        5.6567
   12       19       331        5.6554
   15       24       422        5.6501
   17       25       500        5.6523
   18       27       333,511    5.6457
   21       32       440        5.6468
   24       35       531        5.6491
   27       40       620        5.6545
   30       43       533        5.6521
   35       51       551,711    5.6534

WOULD YOU LIKE TO CHANGE THE TOLERANCE AND TRY AGAIN?
A=NO
```

ANALYST: Hey, it goofed! Line 17 doesn't belong in there. I know that this material is a spinel, and there shouldn't be a 500 reflection in the pattern.

EXPERT: You're right. That line belongs to another phase, but we'll take care of that in a moment. Now, would you like to use Cohen's least-squares methods to get the lattice constant?

ANALYST: Sure.

COMPUTER:

```
WOULD YOU LIKE TO APPLY COHEN'S METHOD?
A=YES
       LATTICE CONSTANT = 5.6540  +OR- 0.0001

SHOULD I OMIT ANY LINES?
A=YES

HOW MANY LINES SHOULD I OMIT?
NUMBER=1
LINE=17
       LATTICE CONSTANT = 5.6540  +OR- 0.0001
```

EXPERT: That's how we get rid of unwanted lines, and as you can see, it didn't change our answer. Now it's listing all the lines that didn't fit in the first phase, so we'll answer the questions again for the second phase—and here's our lattice constant for the second phase.

COMPUTER:

```
LATTICE CONSTANT = 5.5826  +OR- 0.0004
```

EXPERT: And for the third phase.

COMPUTER:

```
LATTICE CONSTANT = 5.5443  +OR- 0.0003
```

ANALYST: Man, that is cool! And it only took us about ten minutes to do it.

EXPERT: For which my usercode will be charged about two dollars.

ANALYST: That would have taken me all day, even with the right data to start with.

EXPERT: So you're beginning to like computers. We'll, we can do better than this. If you can justify it, talk your boss into getting an automatic densitometer with digital output and you won't even have to spend the hour or so it took you to measure the line-positions on the film. The data can be fed to the computer automatically. Or, for a bit more money, dispense with the film altogether and digitize your x-ray equipment.

ANALYST: You're not going to dispense with me next, are you?

EXPERT: No, you'll always be a vital link in the process. But you may have time to do some research, and you'll find yourself writing more and more sophisticated programs for the computer.

ANALYST: Me?

EXPERT: Yes, you!

## Acknowledgement

I wish to thank Mr. R. T. Smith, author of the lattice constant time-sharing program, for his permission to use the example quoted.

# Use of the computer in speech-processing analyses

**E. S. Rogers** | **Dr. P. W. Ross**

**The evaluation of speech recognition systems, which abstract spectral and envelope features from the speech signal in binary form, presents an interesting problem in data analysis. In general, the techniques of pattern recognition are useful for the evaluation of a wide range of feature-recognition systems. By use of a digital computer, it is possible to simulate various recognition algorithms and predict the behavior of a complete speech recognition system. This paper discusses the results of the application of various of these techniques to an experimental speech recognition system. The data logging system and the structure of the recognition algorithms used are described.**

RESEARCH on machine processing of speech is aimed toward discovering the useful, invariant, information-bearing portions of the acoustic signal. A sound wave may be completely described in terms of the amplitude and frequency of the wave as a function of time; therefore, the fundamentals of any speech recognition system must be based in some manner upon amplitude, frequency, and time. The real goal is to employ these three fundamental parameters in such a manner that the speech elements can be classified from a minimum number of features, characteristics, and procedures which lead to the ultimate objective—identification.

To analyze the different elements of speech, there must be some means of segmenting the nearly continuous flow of speech. The segmentation involves sentences, words, syllables, syblets,[1] and phonemes. The choices of an optimum size speech element for analysis required a balance between ease of segmentation and the size of the vocabulary created by the various basic elements. At present, the speech element that appears to offer the best compromise for analysis, transmission and synthesis is the syllable-syblet combination.[2]

## Speech Analyzer

A complete system embodying the idea of segmenting speech into syblets, analyzing by syblets and recognizing by syllables is depicted in Fig. 1. The analyzer segments spoken syllables into smaller elements for processing in stages and produces a digital output or display for each syllable. Both envelope and spectral information are utilized to develop the display. The features used include intrasyllabic pauses, voice-to-unvoiced sound transitions, relative rate of growth and decay, duration and relative spectral information. A brief description of these features is outlined as follows:

1) The system uses the syllable as the basic speech unit for recognition, but it makes use of acoustic syllables, syblets, and phonemes when they can be separated by machine.

2) Segmentation of syllables is effected by intrasyllabic pauses, and voiced-unvoiced sound transitions.

3) The coarse envelope features used include rate of growth, duration, rate of decay, weak sound, and loud sound.

4) Three types of amplitude normalization have been used. Relative channel energies are measured by amplitude comparison between channels. The channel outputs are logarithmic to give the ratio of channel energies rather than the difference. A volume compressor is used to normalize the signal level in the unvoiced sound input channel.

5) Time normalization has been accomplished by having the processor sample only on significant spectral changes.

6) Frequency normalization is included in the form of a selection switch. Four voice ranges are available.

**Edward S. Rogers**
Communications Research Laboratory
Laboratories
Princeton, N.J.
received the BA in Mathematics and Physics from Susquehanna University in 1942 and the MS in Physics from Case Institute of Technology in 1943. From 1943 to 1945 he served as a Member of the Technical Staff with Columbia University Division of War Research, National Defense Research Council and the USN Underwater Sound Reference Laboratory engaged in Underwater Acoustics. In 1945 he became a Member of the Technical Staff of the Laboratories in Princeton. His work has been primarily in the field of acoustics specialing in underwater sound, ultrasonics, speech analysis, noise reduction, and solid state electromechanical transducers. He has also engaged in circuit work for application to color television synchronization methods. His publications include papers on noise reduction systems, solid state electromechanical transducers, and automatic speech recognition systems. He has been granted a number of patents in these fields. He has received three RCA Achievement Awards for outstanding work in research. He is a member of Sigma Xi and the Acoustical Society of America.

**Dr. Paul W. Ross**
Communications Research Laboratory
Laboratories
Princeton, N.J.
received BE in 1960 from Yale University. In 1961 and 1963, he received the ME and DE, respectively, from the same institution. His thesis was concerned with an analysis of traveling-wave parametric amplifiers, which was supported by a National Science Foundation research contract. In the summers of 1959 and 1960, he was employed by Edgerton, Germeshausen and Grier, Inc., with a group developing test instrumentation for high-speed pulse measurement systems. In 1963, he joined the Laboratories as a Member of Technical Staff, assigned to a group working on speech recognition by machine. His work in this group has centered about digital systems and the use of computer technology in speech processing. He is an associate member of the Acoustical Society of America, and a member of the Society of Sigma Xi.

---

Reprint RE-14-3-6
Final manuscript received June 25, 1968.

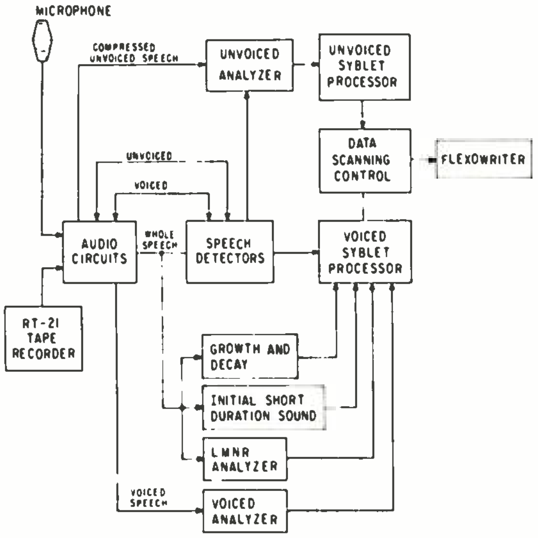


Fig. 1—Block diagram of speech analyzer.

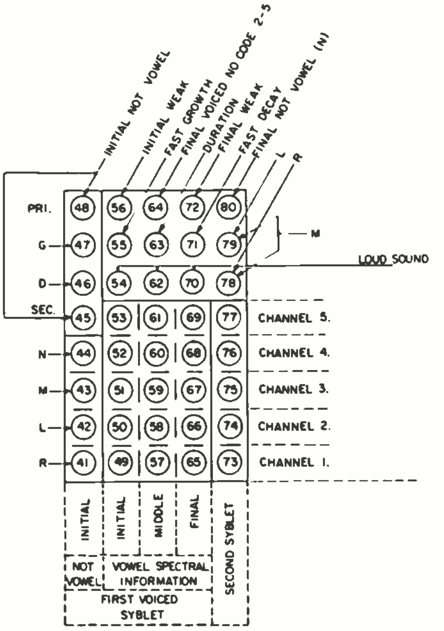


Fig. 2—40-bit digital display, voiced syblet processor.

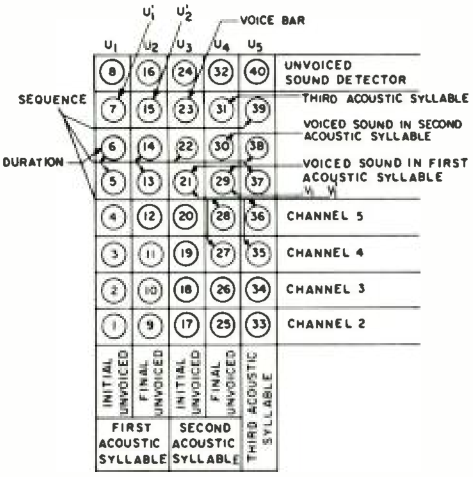


Fig. 3—40-bit digital display, unvoiced syblet processor.

7) The system uses separate voiced and unvoiced spectral analyzers. This permits greater flexibility of the two systems and allows an optimum choice of frequency channels, dynamic range, and integrating time for each of these types of sounds.
8) The digital displays derived for each type of sound are displayed in two 40-light matrices, which gives immediate knowledge of the digital output obtained for each syllable.

The assignment of features in the voiced and unvoiced matrices are shown in Figs. 2 and 3

The acoustic spectrum of repeated voicings of a syllable is subject to substantial variations even though the voicings are made by the same person. Although the analyzer normalizes and extracts pertinent speech features, usually a large number of digital displays are generated for a given syllable. To establish the boundaries of the 80-bit displays pertaining to a particular syllable, a large amount of data is needed. This is obtained from many voicings of each syllable by several speakers. The effectiveness of a particular analyzer is measured by its ability to produce sets of digital displays that are non-intersecting for each speech element of the system vocabulary.

## Data Analysis

The problem of evaluating the performance of a speech analyzer can be separated into two parts:

1) The development of a suitable data logging system for the efficient and rapid acquisition of the processed speech data generated by the feature abstraction system;
2) The reduction of this large quantity of data to determine the degree of recognition or separation of the test vocabulary that can be obtained.

The data analysis problem includes the evaluation of the strength and consistency of a particular feature, such as the repeatability of a voiced-unvoiced sequence or the presence of an intrasyllabic pause. This data can be used to provide a statistical evaluation of various feature occurrences and it can be used to develop a model of the partitioning effected by an analyzer to give a measure of the separation of various speech elements from one another.

### Data Logging System

The digital output from the speech analyzer is displayed in two eight-row, five-column matrices as shown in Figs. 2 and 3. Each of the 80 points may be present or absent for a given voicing. The eighty matrix points, which represent various speech features, were encoded four at a time into sixteen possible combinations. Each combination of four features was assigned a letter of the alphabet from *A* through *P*. For the 80-feature system, this results in a sequence of 20 alphabetic characters for each voicing. The encoding process was done with 20 relay trees. A stepper switch was used to scan the 20 relay trees. The resulting signals from the tree outputs activate a diode matrix that generates the correct bit patterns to operate a Friden Flexowriter and its tape punch. Each data logging cycle was initiated from the control circuitry of the speech processor. After each key lever or punch operation, the stepper switch advances one position to interrogate the next relay tree. At the end of each data logging cycle a gap-feed signal is produced, the stepper is returned to the initial position, and a reset signal is sent to the speech processor to clear the display for the next speech sample.

A simple data-taking procedure has been devised that requires minimum effort on the part of the operator. The system requires operator intervention only to insert identifying information. Various items of identifying information, such as the name of the speaker, the syllable being spoken, the run number, or test conditions may be punched on the paper tape through the Flexowriter keyboard. Identifying data may also be entered through the Flexowriter paper tape reader. In fact, this was used to insert word identification. The syllable numbers and speaker names were stored on paper tape and read into the data tape at the beginning of each new syllable. With the described data logging system, it has been possible to process speech at the rate of 3000 to 4000 speech samples per day.

### Computer Programs

The next phase in the data acquisition procedure is the transcription of the punched paper tape to a computer magnetic tape file. A standard software routine is used. An associated program then updates a master tape file with the transcribed data labeled as to speaker element for each voicing.

With this data logging and transcription system, it is possible to acquire data over a period of time, and then retrieve and batch-process whatever data runs are desired in one group. The analysis programs recall all desired data that has been collected, sort it by syllable number, and compute the frequency of occurrence of each feature for a given syllable. Programs are available that give the conflicting displays between syllables, the number of different displays for a given syllable, and summaries of the 80-bit displays for all voicings of each syllable.

#### Evaluation

From a series of experimental observations, it is possible to divide the speech features, or measurements made upon the speech signal, into two broad classes. The first class contains what are termed "strong" features. These are features determined by the gross spectral and envelope attributes of the speech element. The strong features are used to detect transitions such as voiced to unvoiced conditions, or speech to silence, and vice versa. For a given class of speech elements, these basic sequences may be detected with great reliability.

The second class of features consist of what are termed "weak" features. These are such measurements as local spectral balance, and the classification of vowels and consonants into various categories. These features are more prone to variations due to enunciation.

The data reduction and analysis problem may be directed in two ways—first, the statistical evaluation of various feature occurrences; second, a representation and analysis of the partitioning of the discrete-feature signal space, so as to determine what sets of speech elements may be differentiated from other speech element sets. It is also useful to determine the general configuration of the local regions associated with each set of speech elements, so as to arrive at a concept of an eventual operational speech recognition system and algorithm.

The statistical summary of the feature occurrence is done by a computer program that retrieves all of the desired runs of data from the master data file, as previously described. Each sample pattern from the analyzer has been identified with the syllable number that it represents. The desired patterns are transferred onto a work tape and are then sorted in order.

The sorted patterns are now read back into a buffer area. The patterns, which had been encoded into alphabetic characters, are decoded into the original bit pattern. A summary of the frequency of occurrence for each bit, or feature, is tabulated for each syllable number. This process continues until the selected and sorted sample list is exhausted.

An elementary sub-code partitioning algorithm is used to process each of the feature frequency summaries. This is a procedure where a boundary is defined to enclose all of the patterns produced by the voicing of a particular speech element. A threshold, or decision level, may be set to decide if a feature is significantly present or absent. If the feature frequency falls within an intermediate range of frequency of occurrence, then this feature is not considered to be significant, as the feature frequency indicates a considerable degree of statistical variation if it is not near the threshold.

Each of the arrays of feature-frequency summaries for the different syllables are tested in this manner. Thus, an eighty-tuple pattern is generated for each syllable consisting of ones, zeros, and blanks, indicating respectively the fact that a feature is "significantly present," "significantly absent," or is "not significant" at a particular decision level. In this manner, it is seen that a set of sub-cubes in the discrete feature space is defined, one for each syllable. These sub-cubes are then compared one with the other, in all possible combinations. If any pair of sub-cubes are identical, then an entry of this pair of syllables is also made on the output list. If a particular sub-cube is unique, this is indicated and tabulated.

From the foregoing procedure, it is possible to evaluate, in a somewhat pessimistic manner, the potential behavior of the experimental speech processing system. The sub-cubes represent a first approximation of a "tree" configuration for decoding the patterns. This result is pessimistic, in the sense that it is often possible to separate some syllables by defining particular points or smaller regions in the hyperspace, instead of determining only the larger sub-cubes that enclose all of the voicings of a syllable. An example would be the case of a "dumbbell" shaped region, which should be considered as two distinct "volumes", instead of a large one.

The limiting case for the foregoing procedure occurs when the threshold is set for a 50-50 split, that is, a "dimension" of the sub-cube is assigned a one or zero value upon the basis of the most probable value of that particular feature. As a result of the application of this limiting case, a set of points is defined, which approximately corresponds to the "center of gravity" of all of the displays for a particular syllable. If any of the centers of gravity for any pair of syllables are the same, or "close," it is a strong indication that these syllables might not be separable.

A more difficult situation occurs when the samples for two or more syllables are intermingled. This is liable to occur when phonetically similar speech elements are found to map into the discrete signal space with a similar configuration to that which they have in ordinary speech signal space, such as the words "an" and "and." To evaluate this situation, it is useful to implement a "search" program that compares all pattern samples with all other pattern samples in the test set. This procedure should only be used as a last resort, because of the large number of comparisons required. For N samples, the number of comparisons would be $C^N$. A more reasonable approach would be to examine only those sets of patterns whose centers of gravity are in the "neighborhood" of the set of interest.

### Typical Results

A number of experimental speech processing systems have been developed. The objective of the latest in this series of systems has been to develop a system for the separating and recognizing 550 different speech elements. This system was tested with pre-recorded speech samples for 960 different syllables, with four different individuals speaking each syllable five times, for a total of 19,200 speech samples. This set of speech samples was played back through the experimental speech processing system, and the resulting data logged by the system described in this paper. The analysis program was then run for various decision levels, for the assignment of the sub-cube boundaries. A typical analysis, for the case of 20 samples per syllable, was made at decision levels of 5 and 15, where 5 sets the level at which the system assigns a "zero" to that feature and 15, where a "one" is assigned. The results indicated that 208 syllables were completely separable from any other syllable. This data can be represented by a confusion matrix, where the input syllables are presented on one coordinate, and the same list is used as the other coordinate. If all syllables are separable, then all terms would lie on the major diagonal of the matrix. Those falling off the diagonal represent the cases of confusion of one syllable with another. The number of terms falling off the diagonal was tabulated by the program. The printout from the analysis program was in the form of a chart listing all the various confusions. By eliminating syllables that are often confused, the number of completely separable syllables is increased to 560.

As an additional part of the analysis program, the number of times that a particular feature was assigned a value of "one," "zero," or "not significant" was tabulated. It is possible to obtain a measure of how "strong" a feature is by observing the number of times that it is assigned to "one," "zero," or "not significant." A strong feature can be construed as one that is assigned, approximately, an equal number of times to the "one" and "zero" categories for a particular set of speech element samples and very few times to the "not significant" category. The analysis program also has a speech element set selection feature that allows a certain selected list of syllables to be analyzed. This is useful for a final check on the performance of the system once a preliminary sorting of the initial large number of syllables has been completed. An example of this tabulation is shown in Table I for the first 10 features for all 960 syllables.

Table I—Sample Listing of Syllable Confusions for 960 Syllables.

| *Feature Number* | *Significantly Present* | *Significantly Absent* | *Not Significant* |
|---|---|---|---|
| 1 | 149 | 734 | 77 |
| 2 | 163 | 732 | 65 |
| 3 | 108 | 796 | 56 |
| 4 | 96 | 850 | 14 |
| 5 | 201 | 672 | 87 |
| 6 | 357 | 472 | 131 |
| 7 | 142 | 762 | 56 |
| 8 | 309 | 591 | 60 |
| 9 | 36 | 833 | 91 |

## Model for Analysis of Binary Feature Matrices

The complementary problem to that of determining the valid features of the speech signal is the evaluation of the data generated by testing such systems. It is useful to develop a model of the speech processing system, in respect to the output data that is generated during the testing procedure. The data is generated in the form of binary information. The variables may have only one of two values: "one" or "zero," corresponding respectively to the presence or absence of a particular speech feature. A sequence of $n$ different binary features represents an $n$-dimensional discrete space. The distinctive nature of this data allows the use of many of the aspects of Boolean algebra, and the application of the techniques that have been applied to problems in coding theory and pattern recognition.

The patterns, or "points" in the space, are considered to define separable pattern classes if all the patterns of any one class of speech elements are never the same as any of the patterns generated by members of another class of speech elements. A useful measure of the system performance is that as more speech samples are presented to the system for analysis, the number of different patterns that are generated have an upper bound. This is, in effect, requiring patterns generated for a given syllable to define a bounded region in the discrete signal space.

The number and variation of the patterns that are generated for a given class also gives some indication of the requirements that would be placed upon a final functional system, and influences the nature of an optimum recognition algorithm.

A metric for such a discrete space is the familiar Hamming distance of coding theory, where the distance from one "point" to the other is the number of bits in a pattern that differ from the bit configuration of another pattern. The application of the concept of Hamming distance, to express the "nearness" of one set of patterns to another set, must be used with some care when applied to the problem of evaluating speech processing systems. This is necessary for the following reasons. In the usual coding theory problem, it is assumed that all bits are equally likely to change, or be in error. This is usually not the case for the inter-related features measured by the speech processing system under consideration. This results in a predisposition of the speech analysis system to partition the discrete signal feature space into unequal "volumes" of irregular "shape."

For the case of eighty dimensions, there will be $2^{80}$ (approximately $10^{24}$) possible combinations, or patterns, of ones and zeros. This is such a large number that it is not, as a practical matter, possible to form a list of all the possible combinations, or produce any reasonable graphical representation of the entire 80-dimensional space. However, it appears that one feasible way to consider the problem is to treat the small regions of this 80-dimension space immediately in the vicinity of a particular point of interest, i.e., patterns that differ only a few bits from the pattern of interest, and in a certain sense, represent near neighbors to that point.

The result of this line of thought was the development of a technique to present the regions in the 80-dimensional space in the vicinity of the patterns generated by a particular set of enunciations of a syllable. It was desired to determine the degree of variation of the patterns about some center of gravity, or "most likely point," in the 80-dimensional space.

Once the center of gravity of the patterns for a particular set of voicings has been determined, then it is possible to compare each pattern for this speech element with the center of gravity. The member that is the farthest from the center of gravity then defines the maximum radius hypersphere which will enclose all sample vectors of this set. As a result of these calculations for each syllable, a chart is printed, which gives the syllable number, the number of different vectors or patterns for this set of voicings, the diameter of the set (an optional operation), the radius of the hypersphere, and the center of gravity. This operation is done for all of the input data.

The second phase of the computation then produces a presentation of the immediate vicinity in the 80-dimensional space of each set of voicings. If all pairwise measurements were made, it would require a great number of computations, and probably not be very informative. Consequently, only those sets in the immediate neighborhood of the set of interest are examined. A threshold was set to examine only nearby sets whose centers of gravity were less than or equal to ten bits from the center of interest; those that are found were listed, together with the center-to-center distance. The occurrence of all members of these selected neighboring sets which have members that lie within two bits or less (an optimal threshold) of any members of the set of interest was also noted and printed out with this minimum distance. Included are any conflicts, i.e., vectors identical for two different syllables, where the distance is zero.

At the completion of the analysis for each set, the total number of nearby centers of gravity is printed. The total number of sets which have members within two bits of the syllable of interest is also printed. In this manner, it is possible to evaluate the "nearness" of the various sets of voicings, and to obtain an excellent idea of the possibility of separating two or more syllables.

It is possible to draw a number of conclusions from the use of this analysis procedure in this case. The first conclusion: syllables or speech elements that are similar in a phonetic sense are also found to be similar in the 80-dimensional space generated by the speech processing equipment. Further conclusions may be drawn. The centers of gravity of some sets are often quite close together, which indicates that the different sets are intermingled, but not necessarily intersecting, at least on the basis of the number of samples of each set presented for analysis. In a large number of cases, it was found that the number of different vectors often exceeded 75% of the number of samples for that set. This is a strong indication, in most cases, that the region occupied by samples obtained for a particular syllable is not yet saturated; if more samples are introduced, it is possible that the boundaries will expand, and that there will be a greater degree of intersection between the different sets of data.

Two examples from the analysis program printout are shown in Table II. The first represents what might be termed a "good" syllable. The three nearby centers are associated with syllables that bear only a slight similarity to the example set. The vowel value is somewhat similar, and a possible similarity between the initial consonant can be seen. The fact that the closest center is at distance 6 suggests that any reasonable decision-making algorithm will be able to distinguish this set from all others.

Table II—Two Examples from the Speech Data Analysis Program.

| | |
|---|---|
| *Example 1* | |
| Syllable 302 (CORE) | 20 Different Vectors<br>Radius 17 |
| Syllable 283 (CALL) | Centers at 7 |
| Syllable 343 (HAR) | Centers at 6 |
| Syllable 354 (HOW) | Centers at 9 |
| 3 Nearby Centers | 0 Nearly Syllables |
| *Example 2* | |
| Syllable 356 (KER) | 20 Different Vectors<br>Radius 11 |
| Syllable 295 (CLEAR) | Centers at 7 |
| Syllable 298 (CO) | Centers at 6 |
| Syllable 300 (COM) | Centers at 6 Min. Dist. 1 |
| Syllable 309 (CULL) | Centers at 7 |
| Syllable 340 (FUR) | Centers at 9 |
| Syllable 343 (HAR) | Centers at 9 Min. Dist. 2 |
| Syllable 347 (HER) | Centers at 3 Min. Dist. 1 |
| Syllable 348 (HEV) | Centers at 9 |
| Syllable 353 (HON) | Centers at 8 |
| Syllable 354 (HOW) | Centers at 8 |
| Syllable 392 (SIRE) | Centers at 6 Min. Dist. 2 |
| Syllable 410 (TER) | Centers at 9 Min. Dist. 1 |
| 12 Nearby Centers | 5 Nearby Syllables |

The second example is typical of what can be termed a "difficult" syllable. There are twelve nearby centers in this case, and five sets have nearby members. Attention can be called to the two cases where there is a minimum distance of one between a member of the example set and a member of the two other sets. The first case—*ker-com*—indicates that the vowel value and *m* versus *r* discrimination may be marginal. This might be safely ignored, as the centers of the two sets are 6 bits apart, but the variation, or maximum hypersphere radius, is found to be 11 and 17 bits respectively, for the two sets, which suggests a probable difficulty in discrimination. The remaining cases may be treated in a similar manner. The two examples presented show the utility of this programming technique in conducting a "post mortem" on the speech processing system behavior.

## Summary

The method of taking data, and the computer programs for processing it, provide a powerful method of simultaneously evaluating many feature extraction schemes. The partitioning of the discrete sample space by sub-cubes provides a means of attaining a first approximation toward implementing a final system. The modeling technique, which considers the 80 binary features as defining an 80-dimensional discrete space, is valuable for determining the syllables that are likely to be confused. This technique, which points up potential trouble areas, shows that speech elements that are similar in a phonetic sense are also similar in the 80-dimensional space generated by the speech processing equipment. This indicates that a forcing communication system could be implemented in which an output would be produced for every input utterance, and the system could be encoded so that the output would be phonetically similar to the input.

Analysis in terms of hyperspheres in 80-dimensional space points up potential troubles, but a more specific analysis is required to determine whether these difficulties are in fact real or not. In particular, information might be obtained on the fine structure of two clusters of points representing a pair of hard-to-discriminate classes. For example: To what extent are the clusters "ellipsoidal" instead of "spherical"? To what extent do clusters fill a region "solidly", or spread evenly? These questions require substantial computation to resolve, but are assessible now that the problem-pairs of syllables have been recognized (and can be determined again for any subsequent feature-extraction system).

## Acknowledgments

Acknowledgement is gratefully made to Dr. R. O. Winder of the Laboratories for his aid in the development of the concepts and computer programs associated with the section on the "nearby neighbor" technique for analysis of the 80-dimensional space. The work described in this paper was sponsored by the Defense Communications Agency under Contract No. DCA100-66-C-0092.

## References

1. Olson, H. F., Belar, H., and Rogers, E. S., "Research Towards a High Efficiency Voice Communication System", *J. of the Audio Engineering Soc.*, Vol. 14, No. 3 (July 1966).
2. Olson, H. F., Belar, H., and Rogers, E. S., "Speech Processing Techniques and Applica-tion", *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-15 (Sept. 1967) pp. 120-126.

# Computer programming for electronic circuit analysis

**W. J. Pratt**

**Design automation has recently become one of the foremost endeavors of many design engineering activities. One area of special interest to the design engineer is that of circuit analysis by computer. This allows the circuit designer to economically examine the performance of a circuit during the various stages of design by using the computer instead of a "breadboard". Many circuit analysis programs have been written with specific analysis goals in mind, while others are intended for the general usage of the engineering community. This article relates to some basic features of circuit analysis programming; these basics serve as a foundation for many special types of analysis, that can be obtained by program repetition and effective manipulation of the solution.**

**William J. Pratt**
Signal Processing Engineering
Missile and Surface Radar Division
Moorestown, New Jersey
received the BSEE from Villanova University in 1964. Upon graduation, he joined Bionic Instruments, Inc., where he was responsible for the design of electronic equipment for the medical field. In June of 1965 he joined the Signal Processing Engineering group in the Missile & Surface Radar Division of RCA. Since that time, he has participated in the design of the AN/FPS-49 AN/FPS-92 radar systems. He is presently engaged in the design of signal processing equipment for the AN/FPS-95 radar system. Mr. Pratt is a member of the IEEE.

THE ABILITY of the digital computer to analyze electronic networks is fundamentally the capability first to generate (from the input data) a descriptive matrix that specifies both the network and all excitation sources, then to solve a set of linear, simultaneous, complex equations, and finally to present the solutions of these equations in a manner that yields the desired analysis.

The computer input data must contain all circuit parameters, the network topology, a description of the excitation sources, and other parameters required for the analysis—e.g., frequency. The essential requirements of the input-data format are that it must allow the user to input the required information directly from inspection of the circuit diagram without lengthy conversions, and that it must be sufficiently standardized that the computer can correctly determine the network matrix.

The most commonly used method of defining an electronic network is to locate and designate all the elements and voltage nodes in the circuit and to define each of the elements in terms of its type (resistor, capacitor, etc.), value and nodes of connection. Fig. 1 shows a typical input data format for the simple network of Fig. 2. Each voltage in the network is defined with respect to a ground or datum node whose voltage is set to zero volts. Thus, the number of nodes in the electrical network defines the total number of discreet voltages to be found in the system, any or all of which may be solved for by application of Kirchoff's current law to each node in the network and then solving the set of simultaneous equations.

The input data in the format shown is normally the only information required by the computer for analysis. This input may take the form of a deck of punched cards, a paper tape, or may be typed directly into the system from either a local or remote teletype depending on the type of system available to the user. The example shown was typed on a teletype for analysis by a remote time-shared computer.

## Inputs for Circuit-Analysis Program

Inputs to the computer for a circuit analysis program can be divided into the following seven categories.

### System Commands

These commands direct the computer to call up and compile the applicable source program, and direct the operational mode of the system. The command language varies with the computer in use. In batch-processing systems, the computer operator generally handles all system commands; the user only has to know the system command that specifies the desired program. For on-line systems, such as time-sharing computers, the user must learn a few simple commands. In the example of Fig. 1, the first three lines are system commands. The command FORTRAN directs the Fortran Language Compiler on line. The "COMPILE, 1:100, /ECA/" directs the computer to compile the program /ECA/, which will yield the desired frequency response analysis. The EXECUTE directs execution of the program.

### Comments

Most systems allow the user to insert arbitrary text into the printout so that records of the analysis can be kept and separated from other analyses. These statements have no effect on the operation of the program.

### Program Commands

These commands allow the user to activate optional portions of the program. For example, some circuit-analysis programs have built-in

---

Reprint RE-14-3-10
Final manuscript received July 25, 1968.

subroutines for graphically plotting the results of the analysis; a PLOT command will activate such a subroutine. Some programs also require an EXECUTE command to start the actual analysis after the input data has been entered.

### Network Data

The network or circuit inputs describe the electrical network in its entirety, i.e., component type, value, and connection points (nodes) for each electrical component. These entries can be made directly from a circuit diagram in which each voltage node has an arbitrary number. The only restriction on selecting node labels is that they be a series of consecutive numbers starting with node zero, which is the ground or datum node. The number of component types is generally limited to the types shown in Fig. 3. Using these elements, equivalent circuits can be developed for virtually all linear components, active or passive. The parameter values can generally take on any value between $10^{99}$ and $10^{-99}$ for most computers and can be specified to seven or more significant digits.

### Analysis Parameters

These inputs describe the conditions under which the analysis is to be made. The statement FREQUENCY 2E2, 2E3, 2E2 directs the computer to begin the analysis at 200 Hz and continue to 2000 Hz in steps of 200 Hz (E denotes powers of 10). Other examples of analysis parameters would be time durations and output intervals in a time or transient analysis program.

### Output Control

These statements describe the desired output data. The output data is generally a node voltage, but may also be a component (or branch) current or power dissipation.

### Modification Instructions

If an analysis yields unsatisfactory results, one or more of the component parameters can be modified, or more components can be added until the desired results are achieved. In general, the program accepts modification instruction after each analysis is completed, allowing a change in any component or analysis parameter or a change in circuit excitation. This is exemplified in the second analysis in

```
B1  ICS
    NODES 1,0
    VALUE= 2E-3
B2  RES
    NODES 1,0
    VALUE= 1E3
B3  CAP
    NODES 1,0
    VALUE= 0.25E-6
B4  CAP
    NODES 1,2
    VALUE= 0.1E-6
B5  IND
    NODES 1,2
    VALUE= 0.1
B6  CAP
    NODES 2,0
    VALUE= 0.25E-6
B7  RES
    NODES 2,0
    VALUE= 1E3
B8  END
FREQ= 2E2,2E3,2E2
 SOLVE FOR NODE 2


         F                 MAG              PHASE          TD
    200.000 HZ        0.9714944959       -21.1679      .29400E-03
    400.000 HZ        0.9149008393       -40.4754      .26816E-03
    600.000 HZ        0.8844560385       -58.7922      .25440E-03
    800.000 HZ        0.9318810105       -81.7061      .31825E-03
      1.000 KHZ       0.9747501612      -127.9395      .64213E-03
      1.200 KHZ       0.4311316013       171.4327     -.41579E-02
      1.400 KHZ       0.1086728573       145.1441      .36512E-03
      1.600 KHZ       0.0028741581       -46.7707      .26655E-02
      1.800 KHZ       0.0474502370       -53.7646      .97138E-04
      2.000 KHZ       0.0668104291       -58.5172      .66008E-04


MODIFY B4
NEW VALUE= 0.15E-6


         F                 MAG              PHASE          TD
    200.000 HZ        0.9716212749       -21.1986     -.51831E-03
    400.000 HZ        0.9170979857       -40.7892      .27209E-03
    600.000 HZ        0.8979507685       -60.4975      .27373E-03
    800.000 HZ        0.9780919552       -90.9638      .42310E-03
      1.000 KHZ       0.6725900769      -162.7694      .99734E-03
      1.200 KHZ       0.1018166542       151.7371     -.43681E-02
      1.400 KHZ       0.0549358912       -44.2439      .27220E-02
      1.600 KHZ       0.1031702161       -52.5278      .11505E-03
      1.800 KHZ       0.1187140942       -57.8627      .74097E-04
      2.000 KHZ       0.1220604777       -61.6974      .53260E-04
```

Fig. 1—Computer frequency response analysis of Fig. 2 circuits. (Underlined characters are typed by the program user, all others are printed by the computer.)

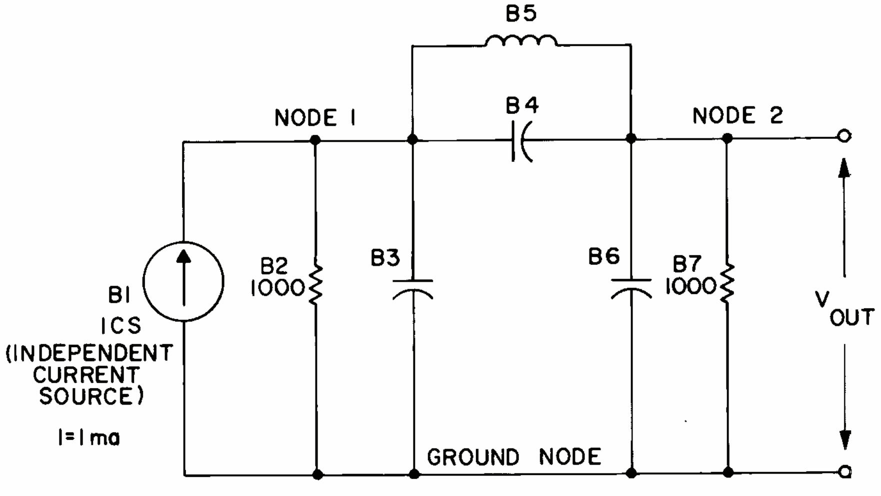


Fig. 2—Elliptic low-pass filter ($F_c$ = 1kHz).

Fig. 1 which shows the effect on the bandwidth of the filter caused by lowering the value of the capacitor in branch 4.

## Steady-State Analysis Programming

Sinusoidal steady state analyses (including DC analysis) are the most widely used circuit analysis programs. These programs are simple to use, yield extremely accurate results, and are relatively simple to write. A general usage circuit analysis program of this type is short enough, usually under 150 statements in length when written in FORTRAN IV, to be used on small computers and time-sharing systems. The disadvantages are that it does not account for transient responses in the network and will not accommodate non-linear elements and devices operated as such. Time and transient analysis programs are available on larger computers, but are much more complex and will not be discussed here. Fig. 4 shows the basic structure of a steady state analysis program. The individual blocks are described below.

### Data Input

The input data is usually formated in a manner similar to that shown in the example of Fig. 1. This portion of the program accepts all of the network data, program commands, analysis parameters, and output control statements and permanently stores the information for later use. The primary purpose of this part of the program is to expedite the inputting of the data.

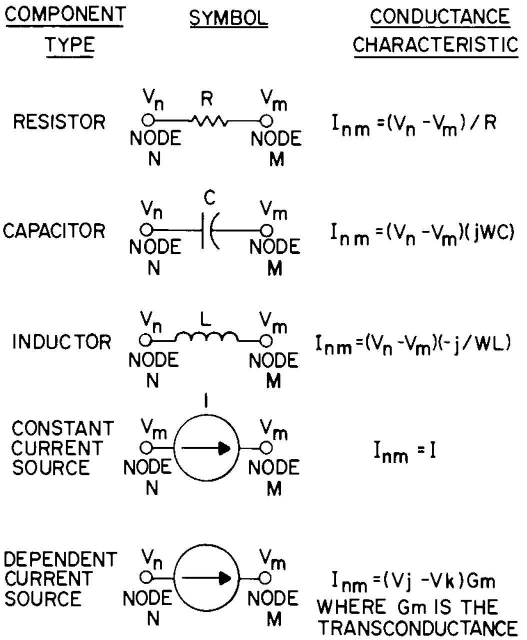


Fig. 3—Component types recognized by steady-state circuit analysis program.

Besides storing the analysis frequency data and output-control data, the program makes entries into arrays whose indexes correspond to the number of a particular component in the network. The arrays store the data for (1) component type (a Hollerith or alphanumeric array); (2) connection nodes (two integer arrays); (3) parameter value; and (4) nodes or branch of dependence (controlled sources only). For example, the entries for branch 4 of the circuit of figure one are

B(4) = 3HRES,
N(4) = 2,
M(4) = 0, and
VALUE(4) = 1000,

where 3HRES is the Hollerith representation of the component type (resistor). N and M are the connection nodes, and VALUE is the parameter value. The information stored in these arrays will be used repeatedly each time the admittance matrix is formed and, therefore, must be saved until termination of the analysis or until modified by the parameter modification portions of the program.

A circuit-analysis program intended for general usage recognizes the types of circuit components listed in Fig. 3. This listing contains the minimum number of component types required to enable the program to analyze an arbitrary electrical network containing both passive and active, lumped-constant parameters. Some programs also recognize components such as transistors, transformers, and other devices for which a linear equivalent circuit can be generated. All of these devices, however, can be represented by the component types listed in Fig. 3 through the use of equivalent circuits.

One commonly used circuit component not shown in Fig. 3 is the voltage source. When doing a nodal analysis, which is the case for virtually all circuit analysis programs, it is necessary to have $n-1$ independent equations to solve a network of $n$ nodes. One exception is when a voltage source is connected between two nodes in the network. In that case, the nodes are defined with respect to one another by the voltage source. Therefore, it is necessary to have some finite resistance in series with all voltage sources in the network, and to ensure that the point of connection of the resistor and voltage source is not defined as a node. When these conditions are met, the nodal analysis can be performed by converting the series combination into an equivalent network consisting of a current source in parallel with the same resistor by application of Norton's Theorem. Many existing programs accept the series combination of the voltage source and resistor as a single circuit element or branch and perform the necessary arithmetic to convert the branch to a current source in parallel with the resistor. If this is not the case, however, the progam user can easily make the substitution of networks before beginning the analysis.

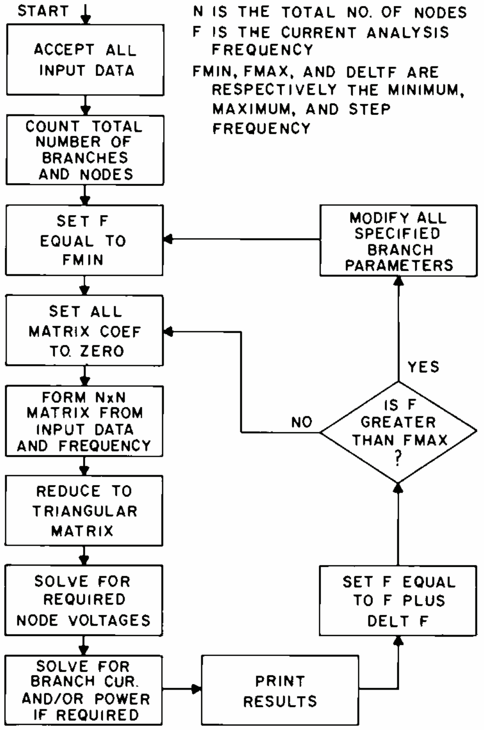


Fig. 4—Block diagram of steady-state circuit analysis program.

### Admittance Matrix Generation

The system admittance matrix used by the computer in the nodal analysis is generated from the input data. This matrix contains all of the coefficients of the current-law equations for each voltage node in the network. Each coefficient is complex, in general, the real part being evaluated from the parameter values of the resistors and sources, and the imaginary part being evaluated from the parameter values of the capacitors and inductors in the network.

The independent current sources form a constant matrix which contains only one column. The system matrix equation takes the following general form.

$$\begin{bmatrix} \mathrm{G}(1,1) & \cdots \mathrm{G}(1,\mathrm{J}) & \cdots & \mathrm{G}(1,\mathrm{N}) \\ \mathrm{G}(2,1) & \cdots \mathrm{G}(2,\mathrm{J}) & \cdots & \mathrm{G}(2,\mathrm{N}) \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{G}(\mathrm{I},1) & \cdots \mathrm{G}(\mathrm{I},\mathrm{J}) & \cdots & \mathrm{G}(\mathrm{I},\mathrm{N}) \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{G}(\mathrm{N},1) & \cdots \mathrm{G}(\mathrm{N},\mathrm{J}) & \cdots & \mathrm{G}(\mathrm{N},\mathrm{N}) \end{bmatrix} \begin{bmatrix} \mathrm{V1} \\ \mathrm{V2} \\ \vdots \\ \mathrm{VI} \\ \vdots \\ \mathrm{VN} \end{bmatrix} = \begin{bmatrix} \mathrm{C}(1) \\ \mathrm{C}(2) \\ \vdots \\ \mathrm{C}(\mathrm{I}) \\ \vdots \\ \mathrm{C}(\mathrm{N}) \end{bmatrix}$$

where

$\mathrm{G}(\mathrm{I},\mathrm{J}) = \mathrm{GR}(\mathrm{I},\mathrm{J}) + j\mathrm{GI}(\mathrm{I},\mathrm{J})$, and
$\mathrm{C}(\mathrm{I}) = \mathrm{CR}(\mathrm{I}) + j\mathrm{CI}(\mathrm{I})$.

GR and GI represent the real and imaginary parts, respectively, of the complex coefficients; CR and CI represent the real and imaginary parts, respectively, of the constant current source terms; and N is the total number of nodes in the network, not counting the ground or reference mode. Each row of the matrix contains all of the coefficients of the current-law equation for one of the nodes. If the node voltages are represented by V1 through VN, the simultaneous equations can be written as follows:

$$\mathrm{G}(1,1)\mathrm{V1} + \cdots + \mathrm{G}(1,\mathrm{J})\mathrm{VJ} + \cdots + \mathrm{G}(1,\mathrm{N})\mathrm{VN} = \mathrm{C}(1)$$

$$\mathrm{G}(2,1)\,\mathrm{V1} + \cdots + \mathrm{G}(2,\mathrm{J}) + \cdots + \mathrm{G}(2,\mathrm{N})\mathrm{VN} = \mathrm{C}(2)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\mathrm{G}(\mathrm{I},1)\mathrm{V1} + \cdots + \mathrm{G}(\mathrm{I},\mathrm{J})\mathrm{VJ} + \cdots + \mathrm{G}(\mathrm{I},\mathrm{N})\mathrm{VN} = \mathrm{C}(\mathrm{I})$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\mathrm{G}(\mathrm{N},1)\mathrm{V1} + \cdots + \mathrm{G}(\mathrm{N},\mathrm{J})\mathrm{VJ} + \cdots + \mathrm{G}(\mathrm{N},\mathrm{N})\mathrm{VN} = \mathrm{C}(\mathrm{N})$$

To evaluate each of the admittance terms from the input data, each input element must be taken into account and its contribution to the overall matrix must be entered at the proper locations. The first task performed by the program is to sort the input element according to type. Once this breakdown is accomplished, the data associated with each element may be used to make the appropriate matrix entry for each element. The element value and topology data is used by one of five subroutines, one for each element type, to determine the admittance of the element in question and to insert the computed value into the appropriate matrix location. Prior to starting this process, all coefficients are set to zero, then, as each admittance term is generated, it is added algebraically to any previous entry into the matrix. Thus, if no previous element has entered into the computation of a given coefficient, the new term is added to zero. This process is continued until all input elements are accounted for in the admittance matrix. (All unmodified coefficients remain equal to zero.)

The resistance subroutine receives all resistive elements from the input data and performs the following operations: First, the admittance (conductance) is found by simply inverting the resistor value. Then, four entries are made into the matrix. Positive entries are made to the real main diagonal coefficients and two negative entries are made on off-diagonal real coefficients corresponding to the two connection nodes of the resistors. For example, if the resistor in question is connected between nodes I and J. The following operations are performed.

GR(I,I) is replaced by GR(I,I) + 1/(resistor value)
GR(J,J) is replaced by GR(J,J) + 1/(resistor value)
GR(I,J) is replaced by GR(I,J) — 1/(resistor value)
GR(J,I) is replaced by GR(J,I) — 1/(resistor value)

These statements simply mean that the old value of each GR-coefficient is replaced by that same value with the admittance of the resistor either added or subtracted to it. In the case where the resistor is connected to ground (node 0), only one entry is made into the matrix, namely, the real part of the main diagonal coefficient corresponding to the other connection node.

The same procedure is valid for capacitive and inductive components except that the entries are made to the imaginary part of the coefficient instead of the real part. Also, since the radian frequency enters into the calculation of susceptance for capacitors and inductors, the matrix must be reconstructed each time that the excitation frequency is changed.

The remaining entries into the matrix are from the dependent current sources. For each source, four entries are made, two in each of two rows of the matrix. The two rows correspond to the nodes to which the source is connected; the columns correspond to the nodes that affect the current source. Because the current is dependent upon a node voltage, the term to be entered is transconductance, *Gm*. If the dependent source is connected between nodes K and M, and the control node is I with respect to J, and an increasing voltage at node I with respect to J causes an increase of current into node K, the matrix entries would be as follows:

$$\mathrm{GR}(\mathrm{K},\mathrm{I}) = \mathrm{GR}(\mathrm{K},\mathrm{I}) - Gm \qquad (1)$$

$$\mathrm{GR}(\mathrm{K},\mathrm{J}) = \mathrm{GR}(\mathrm{K},\mathrm{J}) + Gm \qquad (2)$$

$$\mathrm{GR}(\mathrm{M},\mathrm{I}) = \mathrm{GR}(\mathrm{M},\mathrm{I}) + Gm \qquad (3)$$

$$\mathrm{GR}(\mathrm{M},\mathrm{J}) = \mathrm{GR}(\mathrm{M},\mathrm{J}) - Gm \qquad (4)$$

For purposes of illustration, refer to the transistor circuit in Fig. 5 and the equivalent circuit in Fig. 6. By previous definition, the node which receives increasing current with increasing voltage at node I is node K (the node to which the arrow points).

The collector current, $I_c$, for a transistor is $\beta I_b$. Since transconductance, rather than current gain, is required to make the matrix entry, the program is required to make the following conversion.

$I_c = Gm\ (V_b - V_e)$, and
$I_c = \beta\, I_b = \beta (V_b - V_e)\ /\ h_{ie}$,
$Gm = \beta\ /\ h_{ie}$.

As noted under input data, a branch of dependence is required for all controlled sources. In this case, the branch of dependence is the resistor corresponding to the short-circuit input resistance of the transistor, branch 5 in the equivalent circuit. Thus the method of defining *Gm* is to divide the parameter value of the controlled source ($\beta$) by the parameter value of the branch of dependence (*R4*). In the case of a field effect transistor, where *Gm* is commonly used rather than $\beta$, the transconductance term is used directly. Therefore, the input-data portion of the program is generally set up to handle either a current gain or a transconductance term for the parameter value of a controlled source.

The node equations can be written from inspection of Fig. 6 and are:

$$[(1/R1) + (1/R2) + (1/R3) + (1/R4)]V1 - (1/R4)V2 = 1\mathrm{x}10^{-3},$$

$$-(1/R4)V1 + [(1/R4) + (1/R5)]V2 - I_c = 0, \text{ and}$$

$$I_c + [(1/R6) + (1/R7)]V3 = 0$$

Because $I_c = \beta/R4\ (V_{be}) = Gm\ (V1 - 2)$, the node equations can be written as follows:

$$[(1/R1) + (1/R2) + (1/R3) + (1/R4)]V1 - (1/R4)V2 = 1\times10^{-3},$$

$$(-(1/R4) - Gm)V1 + [(1/R4) + (1/R5) + (Gm)]V2 = 0, \text{ and}$$

$$(Gm)\,V1 + (-Gm)\,V2 + [(1/R6) + (1/R7)]V3 = 0.$$

An examination of these equations reveals the placement of *Gm*, an suggests the operations necessary to account for the controlled current sources in the system matrix, as outlined by Eq. 1 through 4

The remaining input elements to be accounted for are the independent current sources. These sources are used to construct the single column, constant matrix which is found on the right side of the equal sign of the system matrix equation. Since there may be more than one source in the network and these sources can take on arbitrary phase angles with respect to one another, the terms in the constant matrix are, in general, complex numbers. The real and imaginary parts must be evaluated from the magnitude and phase information on the source. The procedure is as follows:

$$\text{CR} = \text{MAG}\ (\cos \emptyset)$$
$$\text{CI} = \text{MAG}\ (\sin \emptyset)$$

where MAG and Ø are the magnitude and phase angle, respectively, of the controlled source.

#### Solution of the Martix Equation for Node Voltages

The net result of the procedures outlined above is a two dimensional array of stored complex numbers that represent the admittance matrix of the electrical network at a specified frequency, ω. A one-dimensional constant array representing the excitation sources is also stored in memory. These arrays represent a system of linear, complex, simultaneous equations, the solution of which will yield all of the node voltages in the electrical network.

The most prevalent method of solving a group of simultaneous equations by computer is a procedure commonly referred to as "Gauss Elimination". This procedure involves transforming an arbitrary matrix into a "triangular" matrix, i.e., a matrix in which all of the coefficients to the left (or right) of the main diagonal are equal to zero. Thus the last equation in the set has only one unknown variable with a non-zero coefficient, and can be solved for the unknown involved. Then by a process of sequential back substitution, the remainder of the unknowns are found. The basic process can be extended to equations involving complex coefficients and is the basis for nearly all electronic circuit analysis programs. To illustrate this method of solution for the network node voltages, consider the network shown in Fig. 7. The general form of the simultaneous equations for a two node electrical network is

$$\text{G}(1,1)\text{v}1 + \text{G}(1,2)\text{v}2 = \text{c}(1), \text{ and}$$
$$\text{G}(2,1)\text{v}1 + \text{G}(2,2)\text{v}2 = \text{c}(2).$$

For the network in question, these equations become

$$1.2\text{v}1 - 0.2\text{v}2 = 1, \text{ and}$$
$$-0.2\text{v}1 + 0.3\text{v}2 = 0.$$

All of the imaginary parts of the coefficients are zero in this example because there are no reactive elements in the network and no phase angle associated with the input source. To find the node voltages, *V1* and *V2*, the following procedure is executed.

1) The first equation is multiplied by the factor G(2,1) / G(1,1), and becomes — 0.2v1 + 0.0333v2 = — 0.1666.
2) This equation is then subtracted from the second equation, yielding 0.2666 V2 = 0.1666.

The 0 coefficient of v1 in the last equation resulted from the choice of G(2,1)/G(1,1) as the multiplier. If there were more equations in the group, the same procedure could be used to eliminate the coefficient of v1 from all of the remaining equations. For the present example, of course, it is unnecessary to procede further since the voltages v1 and v2 can now be easily determined.

This procedure may be expanded to yield solutions for any number of simultaneous equations. In a system of *n* equations. v1 is eliminated from all except the first equation by setting the coefficients of v1 to zero by the method outline above. Then v2 is eliminated from all except the first two equations, and so on, until in the $n^{th}$ equation, only the coefficient of VN remains non-zero. Solution for VN is then obtained and sequential back substitution will yield the remainder of the node voltages

The generalized procedure is to first eliminate v1 from equations 2 through *n* by defining *n-1* multipliers as follows:

$$\text{U}(\text{I}) = \text{G}(\text{I},1)/\text{G}(1,1), \text{ where I} = 2,3\ldots,n.$$

U(I) is then multiplied by the first equation *n*-1 times, once for each value of I, each time subtracting the product from the $\text{I}^{th}$ equation, thereby generating *n-1* new equations with v1 eliminated. The new general coefficient for equations 2 through *n*, G(I,J)′, is

$$\text{G}(\text{I},\text{J})' = \text{G}(\text{I},\text{J}) - \text{U}(\text{I})\text{G}(\text{L},\text{J}).$$

The process is then repeated using the newly formed second equation as the "pivot", defining *n*-2 multipliers, which are

$$\text{U}(\text{I}) = \text{G}(\text{I},2)/\text{G}(2,2), \text{ where I} = 3,4,\ldots,n.$$

U(I) is multiplied by the second equation *n-2* times, once for each value of I, each time subtracting the product from the $\text{I}^{th}$ equation, generating *n-2* equations with the variable v2 eliminated.

The process is continued in this manner until V(N-1) is eliminated from the last equation. The resultant triangular matrix takes the following form:

$$\begin{bmatrix} \text{G}(1,1) & \cdots & \text{G}(\text{L},\text{J}) & \cdots & \text{G}(\text{L},\text{N}) \\ & \cdots & \text{G}(2,\text{J}) & \cdots & \text{G}(2,\text{N}) \\ & & \cdot & & \cdot \\ & & \cdot & & \cdot \\ & & \cdot & & \cdot \\ & & \text{G}(\text{I},\text{J}) & \cdots & \text{G}(\text{I},\text{N}) \\ & & & & \cdot \\ & & & & \cdot \\ & & & & \cdot \\ & & & & \text{G}(\text{N},\text{N}) \end{bmatrix} \begin{bmatrix} \text{V}1 \\ \text{V}1 \\ \cdot \\ \cdot \\ \cdot \\ \text{VI} \\ \cdot \\ \cdot \\ \cdot \\ \text{VN} \end{bmatrix} = \begin{bmatrix} \text{C}(1) \\ \text{C}(2) \\ \cdot \\ \cdot \\ \cdot \\ \text{C}(\text{I}) \\ \cdot \\ \cdot \\ \cdot \\ \text{C}(\text{N}) \end{bmatrix}$$

The node voltage, VN. can be easily found and is

$$\text{VN} = \text{C}(\text{N})/\text{G}(\text{N},\text{N})$$

Back substitution yields the rest of the node voltages. i.e.,

$$\text{VI} = \frac{\text{C}(\text{I}) - \sum_{\text{J}=\text{I}+1}^{\text{N}} \text{G}(\text{I},\text{J})\ \text{VJ}}{\text{G}(\text{I},\text{I})}.$$

Since the coefficients are complex numbers, all of the arithmetic involved in the "Gauss Elimination" routine and the back substitution is complex. Most FORTRAN compilers are able to perform complex arithmetic, which results in a simple routine for the solution of the node voltages. The results obtained for the node voltages are also complex

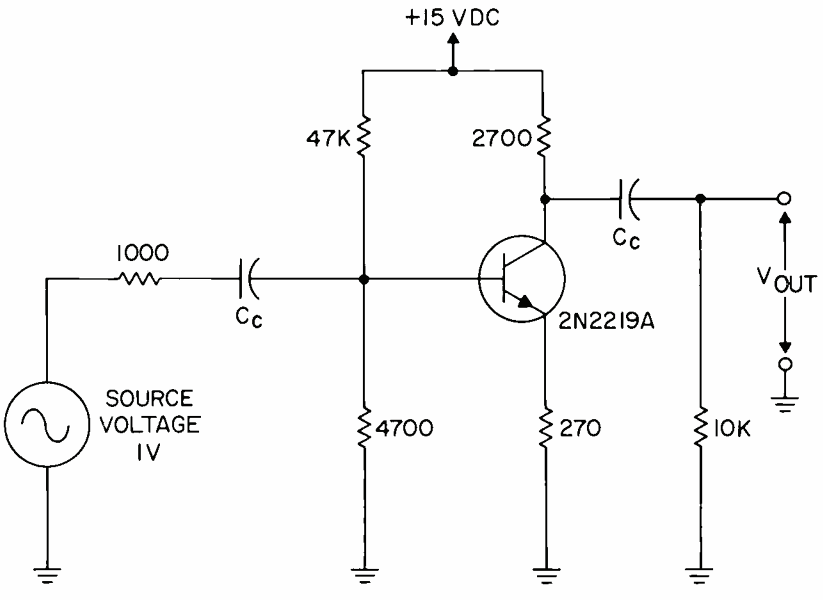


Fig. 5—Single-stage transistor amplifier.

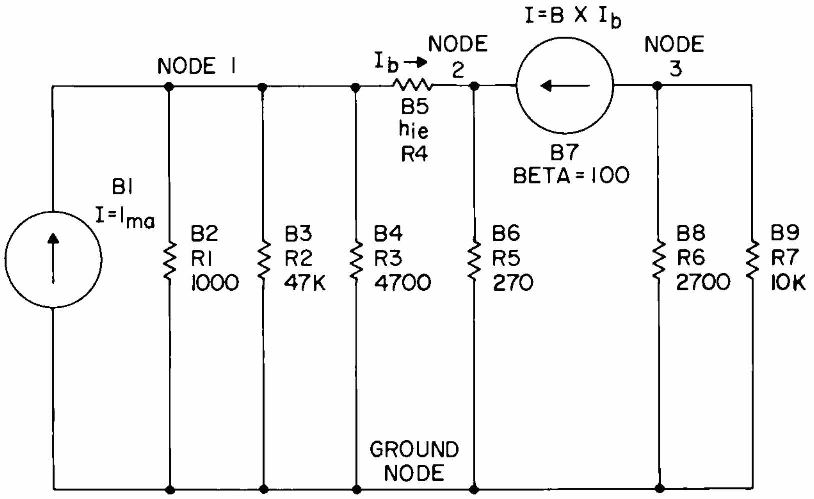


Fig. 6—Mid-band equivalent circuit for transistor amplifier circuit of Fig. 5.

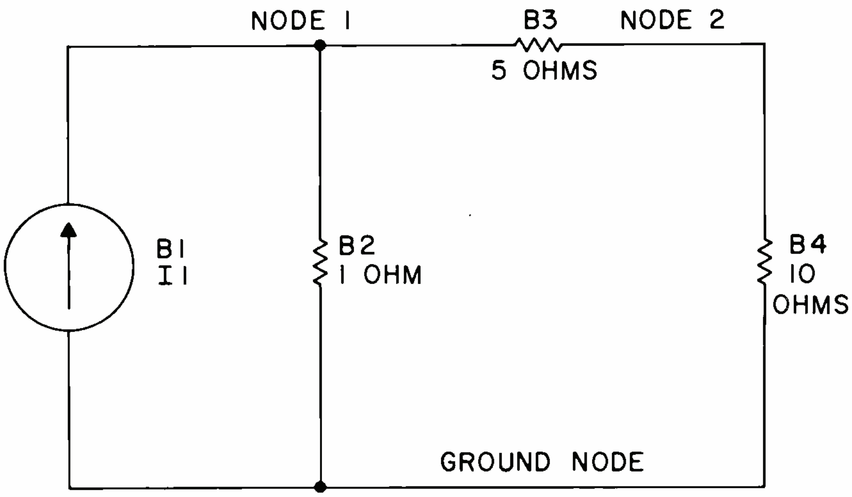


Fig. 7—Simple resistor network used to illustrate Gauss elimination procedure.

numbers. The magnitude and phase response at a node is easily found from the complex number as follows:

$$\text{mag of } V_i = (VR_i^2 + VI_i^2)^{1/2}$$
$$\text{phase of } V_i = \tan^{-1}(VI_i/VR_i),$$

where $VR$ and $VI$, respectively, are the real and imaginary parts of $V_i$. It should be noted that the arctangent function is periodic, repeating every $\pi$ radians. If the quadrant information, that is, the signs of the real and imaginary part of $V$, are taken into account, the phase angle can be defined unambiguously over the range from $\pi$ to $-\pi$ radians. Outside of this range, the phase angle will be in error by some multiple of $2\pi$ radians, always giving a result lying in the range between $-\pi$ and $+\pi$. This phenomenon is shown in the frequency response analysis of Fig. 1. Notice that as the phase angle approaches —180°, there is an abrupt change in phase to a positive number. The actual phase is the number listed minus 360°.

## Conclusions

Electronic circuit analysis programs are fast and efficient tools which help significantly in the design of electronic circuitry. The analysis of Fig. 1, for example, required less than 15 minutes on the console and delivered the results for less than $5. The programs can be used by any engineering group within RCA through use of the RCA Corporate or other time-sharing systems. When large quantities of analytical data must be obtained, in-house computers operating in the batch-processing mode using commercially available programs, such as ECAP,[1] are more efficient.

The accuracy of results, in general, is excellent. One rule-of-thumb that should be followed to ensure highly accurate results is that the differential magnitude of the impedences connected to any node in the network be less than ½ of the precision of the computer in orders of magnitude. For example, when using a computer which operates with 12 places of significant digits, the maximum differential impedences connected to any node should be less than $10^6$.

## Bibliography

1. IBM, "1620 Electronic Circuit Analysis Program User's Manual."

# NATS—an easy-to-use computer program for analyzing linear circuits

**D. G. Ressler**

**Persons having no experience with computers may easily learn to use a network analysis program developed for a time-shared computer. The user describes a circuit by answering questions at a Teletype console, and the program responds with a linear steady-state analysis of the circuit. The circuit may consist of passive elements, transistors, and generators of various types. The analysis consists of calculating quantities such as gain, node voltages, and input and ouput impedances at desired frequencies, and either printing these quantities on the Teletype, or plotting them on a Model T storage-tube display or a hard-copy plotter.**

COMPUTERS don't have to be hard to talk to. But because they speak a language quite different from that of humans, a programmer often spends as much or more time smoothing the flow of communication between program user and computer as he does specifying the necessary computations.

The program NATS—Network Analysis Time Shared—includes sophisticated computer-user communication facilities, which permit a user with no computer experience to easily manipulate a large and powerful computer program for performing steady-state analysis on linear electrical circuits. This is important in the computer-aided design field for two reasons:

1) Many engineers engaged in circuit design have little or no experience with computers; and
2) Circuit design problems are complicated by themselves; the added complexity of a hard-to-use analysis program often discourages the engineer from performing any analysis at all.

The NATS program is easy to use because it has the following characteristics:

1) The program runs on a time-shared computer, so the user can interact with the full power of the computer in real time while sitting at a Teletype console.
2) Only one arbitrary command need be learned by the user; all other interaction is in English text and Arabic numbers.
3) The user may interrogate the program for an explanation of any question he does not understand.
4) As the user gains experience, he can use shortcuts to improve speed and efficiency.

Experience has shown that the best way to learn to use NATS is to sit down and try it. There is no need for a laboriously detailed instruction manual. Most new users get results in just a few minutes; an hour or more of use qualifies one as an expert.

## What NATS Can (and Cannot) Do

The NATS program can perform a steady-state linear analysis of electrical networks described to the computer by the user. A network to be analyzed may consist of passive elements (resistors, capacitors, inductors), transistors and generators of various types. The nodes are numbered arbitrarily and the circuit values read in. The results of the analysis consist of the gain, input and output impedances, admittances, node voltages, and equivalent y-matrix of the circuit. Each of these quantities is calculated at one or more frequencies specified by the user. The output may be printed on the Teletype, or plotted against frequency on the Teletype, a Model T storage tube display,[1] or a precision plotter.

Nonlinear analysis or transient analysis cannot be done using NATS (although the latter can be simulated by hand using Fourier analysis). Thus, bias point problems involving nonlinear elements cannot be solved; nor can sensitivity analysis be performed using NATS.

The program operates in sections, usually in sequence shown in Fig. 1. However, the user may jump from one section to any other section at will and make as many changes as he desires, analyzing each change at his pleasure.

**David G. Ressler**
Consumer Electronics Research Laboratory
Laboratories
Princeton, N.J.
received the BS in chemistry from Rutgers University in 1960 and the BSEE, also at Rutgers, in 1963. He joined the Laboratories in June 1963 on the Research Training Program, and in 1964 joined what is now the Consumer Electronics Research Laboratory on permanent assignment in the area of electro-optic displays. He received the MSEE from the University of Pennsylvania in 1965 and is currently working toward the PhD at the Polytechnic Institute of Brooklyn. Mr. Ressler is currently investigating active device models used in computer-aided design.

The program will detect most improper replies, so mistakes are easily corrected. The sections shown in Fig. 1 are explained as follows:

1) START: The user enters the program and has a set of instructions printed out if he wants them.
2) INPUT/OUTPUT: The user specifies the node numbers of the input and output of the circuit.
3) PASSIVE ELEMENTS: A resistor, capacitor, and inductor may be placed in series or in parallel with each other between any pair of nodes. A node pair may also be shorted. Any values entered here or elsewhere in the program can be easily changed at any time.

---

Reprint RE-14-3-7
Final manuscript received August 8, 1968.

4) SOURCE/LOAD: An impedance including a voltage source may be added to the input terminals of the circuit. This source voltage will be reflected in the circuit voltages calculated by the program. The source impedance will not be included in the calculated input impedance. An output impedance may similarly be added.

5) TRANSISTORS: The program accepts *y*-, *s*-, *h*- or hybrid-*pi* small-signal transistor parameters and enters them into a permanent library which is consulted by the program when needed.

6) GENERATORS: The user can specify generators with the following characteristics: a) Voltage or current generator; b) Generator dependent on, or independent of, circuit qualities; c) If dependent, then dependent either on voltages at nodes, or currents through branches.

7) FREQUENCY OPTION: There are several ways the frequencies of analysis can be specified, and the user does so at this point in the program.

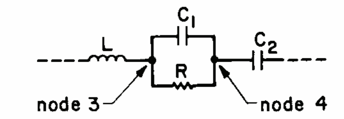
8) OUTPUT CHOICE: Since printout on the Teletype is at the rate of only 10 characters/second, print-out of all possible results would be time-consuming. Therefore, the user types in only the quantities he wants printed. He makes a similar choice of what he wants plotted (vs. frequency only) and by what medium (Teletype, storage tube, or Calcomp plotter).

9) ANALYZE: This is the section in which the network admittance matrix is calculated from the circuit description and is inverted. Details about this are given here. Following analysis, the results are returned to the user.

10) WHAT NEXT?: This is the most powerful question in the program, for the user has the choice of returning anywhere in the program to make changes, or to perform any number of related tasks, such as listing out the circuit components. The user can get to this section from anywhere in the program with the special command "/" followed by a letter.

One of the features of the program referred to earlier is the ability to store circuit descriptions on the computer's magnetic disc for later re-analysis. This allows the user to stop using the program, then come back to his problem conveniently some other time. The frequency and output specifications are stored along with the circuit description.

A paper-tape description of a circuit can be produced to be used with NAB, an analysis program similar to NATS, but operating in the batch-process mode on another computer; NATS can also read such a tape.



$$Y_{33} = \frac{1}{R} + j\left(\omega C_1 - \frac{1}{\omega L}\right) \qquad Y_{34} = -\frac{1}{R} - j\omega C_1$$

$$Y_{43} = -\frac{1}{R} - j\omega C_1 \qquad Y_{44} = \frac{1}{R} + j\omega(C_1 + C_2)$$

Fig. 2—This section of a circuit is shown with the elements of the corresponding Y-matrix.

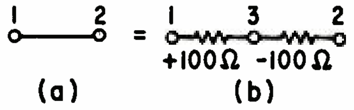


Fig. 3—Adding a node and connecting resistances as in b) has the same effect as shorting nodes 1 and 2.

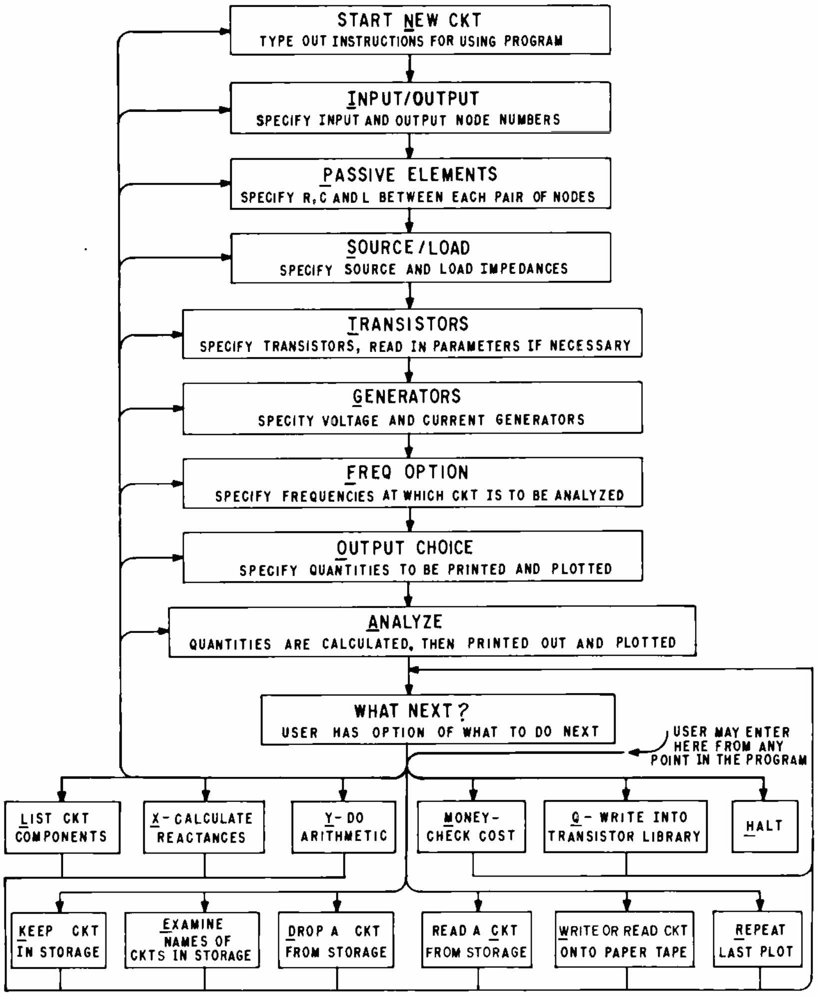


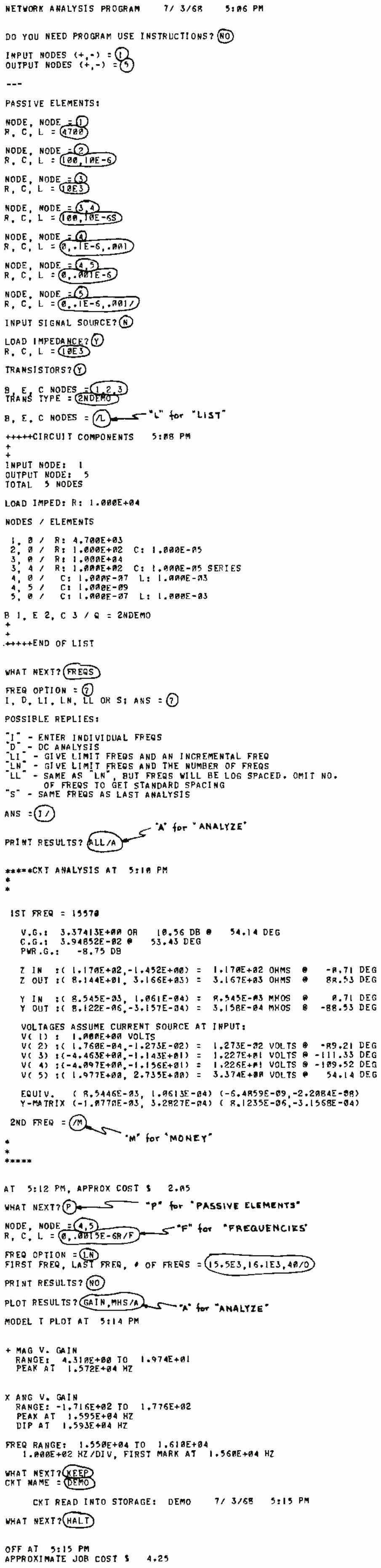
Fig. 1—The usual "flow" of NATS is as shown; however, it is possible to move from one point to any other with a simple command.

## How NATS Works

The description of the circuit is typed in by the user to build an admittance matrix $[Y]$ at each frequency of analysis. Each element of $[Y]$ consists of the self (or transfer) admittances at the corresponding node of the circuit. Fig. 2 shows a portion of a circuit and the appropriate admittances added to the $Y$-matrix describing the circuit. Transistors and dependent sources are added in a similar manner.

The $Y$-matrix appears in the circuit equation $[Y]\,\mathbf{V} = \mathbf{I}$ which, when solved by inverting $[Y]$, gives the desired quantities, including gain, input and output impedances, and node voltages $\mathbf{V}$. The vector $\mathbf{I}$ represents the currents injected into the nodes of the circuit from external independent sources.

A difficulty with using this kind of analysis is that a short circuit between two nodes (for example, a voltage source with zero internal impedance) would result in an infinite term added to the admittance matrix. NATS overcomes this difficulty by adding a third node between two shorted nodes and using a negative resistance, as shown in Fig. 3. The voltages at the two nodes must be the same but there is no in-



```
NETWORK ANALYSIS PROGRAM    7/ 3/68    5:06 PM

DO YOU NEED PROGRAM USE INSTRUCTIONS? NO

INPUT NODES (+,-) = 1
OUTPUT NODES (+,-) = 5

---

PASSIVE ELEMENTS:

NODE, NODE = 1
R, C, L = 4700

NODE, NODE = 2
R, C, L = 100,10E-6

NODE, NODE = 3
R, C, L = 10E3

NODE, NODE = 3,4
R, C, L = 100,10E-6S

NODE, NODE = 4
R, C, L = 0,.1E-6,.001

NODE, NODE = 4,5
R, C, L = 0,.001E-6

NODE, NODE = 5
R, C, L = 0,.1E-6,.001/

INPUT SIGNAL SOURCE? N

LOAD IMPEDANCE? Y
R, C, L = 10E3

TRANSISTORS? Y

B, E, C NODES = 1,2,3
TRANS TYPE = 2NDEMO

B, E, C NODES = /L  <--- "L" for "LIST"

+++++CIRCUIT COMPONENTS   5:08 PM
+
+
INPUT NODE:  1
OUTPUT NODE:  5
TOTAL  5 NODES

LOAD IMPED: R: 1.000E+04

NODES / ELEMENTS

 1, 0 /  R: 4.700E+03
 2, 0 /  R: 1.000E+02  C: 1.000E-05
 3, 0 /  R: 1.000E+04
 3, 4 /  R: 1.000E+02  C: 1.000E-05 SERIES
 4, 0 /   C: 1.000E-07  L: 1.000E-03
 4, 5 /   C: 1.000E-09
 5, 0 /   C: 1.000E-07  L: 1.000E-03

B 1, E 2, C 3 / Q = 2NDEMO
+
+
.+++++END OF LIST


WHAT NEXT? FREQS

FREQ OPTION = ?
I, D, LI, LN, LL OR S; ANS = ?

POSSIBLE REPLIES:

"I" - ENTER INDIVIDUAL FREQS
"D" - DC ANALYSIS
"LI" - GIVE LIMIT FREQS AND AN INCREMENTAL FREQ
"LN" - GIVE LIMIT FREQS AND THE NUMBER OF FREQS
"LL" - SAME AS "LN", BUT FREQS WILL BE LOG SPACED. OMIT NO.
       OF FREQS TO GET STANDARD SPACING
"S" - SAME FREQS AS LAST ANALYSIS

ANS = I/
                                   'A' for 'ANALYZE'
PRINT RESULTS? ALL/A  <---


*****CKT ANALYSIS AT  5:10 PM
*
*

 1ST FREQ = 15570

   V.G.:  3.37413E+00 OR   10.56 DB @   54.14 DEG
   C.G.:  3.94852E-02 @   53.43 DEG
   PWR.G.:   -8.75 DB

   Z IN  :( 1.170E+02,-1.452E+00) =  1.170E+02 OHMS  @   -0.71 DEG
   Z OUT :( 8.144E+01, 3.166E+03) =  3.167E+03 OHMS  @   88.53 DEG

   Y IN  :( 8.545E-03, 1.061E-04) =  8.545E-03 MHOS  @    0.71 DEG
   Y OUT :( 8.122E-06,-3.157E-04) =  3.158E-04 MHOS  @  -88.53 DEG

   VOLTAGES ASSUME CURRENT SOURCE AT INPUT:
   V( 1) :  1.000E+00 VOLTS
   V( 2) :( 1.760E-04,-1.273E-02) =  1.273E-02 VOLTS @  -89.21 DEG
   V( 3) :(-4.463E+00,-1.143E+01) =  1.227E+01 VOLTS @ -111.33 DEG
   V( 4) :(-4.097E+00,-1.156E+01) =  1.226E+01 VOLTS @ -109.52 DEG
   V( 5) :( 1.977E+00, 2.735E+00) =  3.374E+00 VOLTS @   54.14 DEG

   EQUIV.   ( 8.5446E-03, 1.0613E-04) (-6.4859E-09,-2.2084E-08)
   Y-MATRIX (-1.0770E-03, 3.2827E-04) ( 8.1235E-06,-3.1568E-04)

 2ND FREQ = /M  <---
                          'M' for 'MONEY'
*
*
*****


AT  5:12 PM, APPROX COST $   2.05

WHAT NEXT? P  <---   "P" for "PASSIVE ELEMENTS"

NODE, NODE = 4,5
R, C, L = 0,.0015E-6R/F  <--- "F" for "FREQUENCIES"

FREQ OPTION = LN
FIRST FREQ, LAST FREQ, # OF FREQS = 15.5E3,16.1E3,40/D

PRINT RESULTS? NO

PLOT RESULTS? GAIN,MHS/A  <--- "A" for "ANALYZE"

MODEL T PLOT AT  5:14 PM


+ MAG V. GAIN
  RANGE:  4.310E+00 TO  1.974E+01
  PEAK AT  1.572E+04 HZ


X ANG V. GAIN
  RANGE: -1.716E+02 TO  1.776E+02
  PEAK AT  1.595E+04 HZ
  DIP AT  1.593E+04 HZ

FREQ RANGE:  1.550E+04 TO  1.610E+04
   1.000E+02 HZ/DIV, FIRST MARK AT  1.560E+04 HZ

WHAT NEXT? KEEP
CKT NAME = DEMO

     CKT READ INTO STORAGE:  DEMO    7/ 3/68   5:15 PM

WHAT NEXT? HALT


OFF AT  5:15 PM
APPROXIMATE JOB COST $   4.25
```

Fig. 4b—Teletype printout of analysis. User's replies have been circled.

finite admittance to be added. The user of NATS is not aware of the existence of the extra node.

## Example

A sample analysis will demonstrate some of the features of NATS, using the circuit containing passive elements and a transistor shown in Fig. 4a; Fig. 4b shows the Teletype printout. The circuit is entered, the circuit components are listed, the results are printed out at one frequency, a capacitor is changed, and a plot of circuit gain versus frequency made on a storage tube display as shown in Fig. 4c. For this second analysis, 40 frequencies are chosen between 15.5 kHz and 16.1 kHz. Note that the plot shown in Fig. 4b shows the time of day as given in the printout. Finally the circuit is put into storage in the computer for reanalysis, at some later time. In this example, the user is somewhat experienced and uses some short cuts a beginner would probably not use. But at one point he forgets what options were open to him and asks the computer by typing a question mark. Note that the response is not satisfactory the first time, so a second question mark elicits a more detailed response.

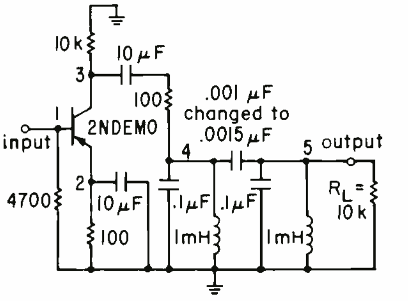


Fig. 4a—Circuit diagram. Parameters of transistor 2nDEMO had been placed previously in a library in the computer.

The transistor specified by the circuit has been entered earlier into the permanent library of the computer. If it had not been entered, the user would be so informed and given an opportunity to enter it.

Note how easily the user moved from one section of the program to another with the command "/" followed sometimes by a letter. If no letter is used, the program proceeds to the next section shown in Fig. 1. The user could get a list of the appropriate letters to use in this command by typing "/?".

## Conclusion

The NATS program is a powerful tool for designing and analyzing circuits that is easily mastered. The engineer can save so much time using NATS that he will probably obtain more information about a circuit from the program than he would otherwise calculate by hand. The cost of using NATS is low: typical computer charges for a one-hour session might be from $10 to $20, and quite a lot can be accomplished in an hour. The program is currently available to run on the PDP-10 computer at Applied Logic Corporation in Princeton, New Jersey.

## Reference

1. J. C. Miller and C. M. Wine, "A Simple Display for Characters and Graphics," *RCA reprint PE-387.*

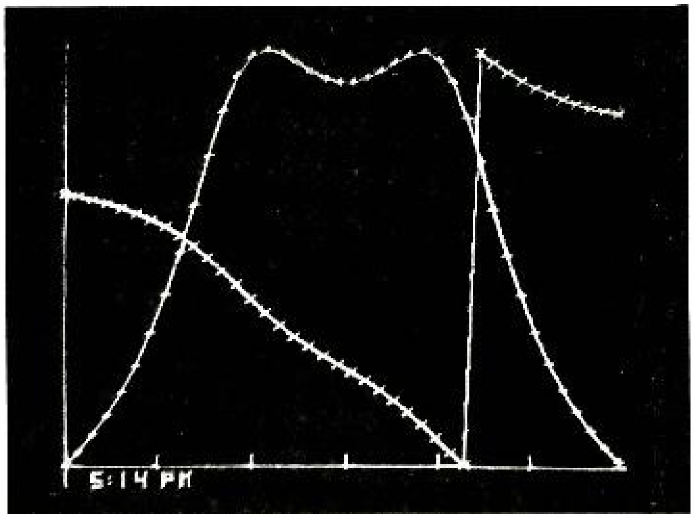


Fig. 4c—Model-T storage tube display of voltage gain magnitude and angle versus frequency. Note the scaling information given in the printout.

# Task scheduling for the Spectra 70/46 time-sharing operating system

**G. Oppenheimer**
**N. Weizer**

**This article discusses task scheduling and resource balancing for a medium-size virtual memory paging machine in relation to a combined batch-processing and time-sharing environment. It includes a synopsis of the task scheduling and paging algorithms that were implemented for the machine. Throughout the discussion, particular emphasis is placed on balancing the system performance relative to the characteristics of all the system resources.**

THE Spectra 70/46 Time Sharing Operating System (TSOS) is designed to control a medium-size, time sharing and multiprogramming batch system that will support up to 48 conversational users and process, simultaneously, a combined total of 64 batch and interactive tasks. Included in the paper are descriptions of some of the general considerations that influenced the design of the scheduling and paging algorithms adopted to balance the available resources and provide adequate response times.

## Hardware Environment

To provide a background for the later discussion of the scheduling and paging algorithms a short description of the Spectra 70/46 features important to the development of the algorithms is presented here.

The Spectra 70/46 is basically a Spectra 70/45 with memory-address translation hardware added. The virtual memory and paging facilities of the 70/46 are achieved using this translation memory. Special hardware functions implemented in the read-only memory of the 70/46 supplement the translation memory. They are used as additional privileged instructions which provide the ability to load or unload all or selected parts of translation memory and to scan translation memory such that only the entries of those pages that have been accessed are stored into main memory.

The control bits contained in each translation memory entry include indicators that show whether a page has been written into and/or accessed. These bits are set automatically by hardware. The usable bit, indicating that a page is in memory, and the space for the physical page number that corresponds to the virtual page are also included in the translation memory entry. These are set by software when a page is brought into memory.

Each of the 512 entries in the translation memory (TM) represent one 4-kbyte page of the 2-million-byte virtual memory. These entries are grouped in 64 page units which are called segments. The length of each segment is equal to the largest number accommodated by the hardware address arithmetic adders.

The first four segments (256 pages) of virtual memory are available for user tasks. This portion of virtual memory, called user virtual memory, is allocated uniquely for each user. That is, each program in the system gets its own private 256 pages of virtual memory.

When a task is to be given control of the processor, the necessary portion of the 256 entries belonging to it are loaded into the TM after the previous program's TM entries have been stored in a predesignated portion of the system virtual memory.

**G. Oppenheimer, Mgr**
New Product Line Operating Systems
Systems Programming
Information Systems Division
Camden, N.J.

received the BS in Industrial Engineering from N.Y.U. and the MS from the Graduate School of Industrial Administration at Carnegie Tech. Mr. Oppenheimer joined RCA in June of 1961 and was responsible for the 301 PERT/Time program and the 3301 Flow Simulator language, in addition to participating in information retrieval and sales forecasting efforts. In April of 1966, he joined the Control Systems area of Systems Programming where he has been participating in the development of the Time Sharing Operating System (TSOS) for the Spectra 70/46. Prior to joining RCA, Mr. Oppenheimer worked with the Operations Research Center at M.I.T. developing an Inventory Management System for the U.S. Army Ordnance Corps. He is a member of the ACM and TIMS, and the Tau Beta Pi and Alpha Pi Mu honor societies.

**N. Weizer, Ldr**
TSOS Central Control and
Memory Mgmt. Subsystems
Systems Programming
Information Systems Division
Camden, N.J.

received the BA in Physics in 1960 and the MSE in Computer and Information Sciences in 1965 both from the University of Pennsylvania. Mr. Weizer joined RCA in November of 1966 as a member of the Control Systems area of Systems Programming. In this activity he has been participating in the design and development of the Time Sharing Operating Systems (TSOS) for the Spectra 70/46. Prior to joining RCA, Mr. Weizer was employed by Burroughs Corporation as a systems designer working on the design of new hardware and software systems. Mr. Weizer is a member of the ACM and the IEEE.

Reprint RE-14-3-8
Final manuscript received May 17, 1968.

The four segments of virtual memory that are not available to user programs are used by the Control Program and shared routines. This system virtual memory is always allocated in this manner. It is independent of which program is loaded in the other 4 segments; it is resident in the virtual memory; and it need not be unloaded from TM as control is passed from one task to another.

### System Configuration

Physical memory is 262 kbytes (64 pages). The backing store is a drum with a 333 kbyte/second transfer rate. The drum capacity may be 800 or 1600 pages, assuming a single page per track. There is a one-to-one correspondence between read/write heads and tracks.

Data transfer from peripherals is either across selector channels for high-speed devices or across a multiplexer for lower speed units. There is a maximum of four selector channels on the system. Transfers to and from memory steal memory cycles at the rate of 1.44 μs/2 bytes for the selectors and 14.4 μs/byte for the multiplexer.

A typical configuration is assumed to be a 262-kbyte processor, an 800-page drum on selector channel 1, 4 discs (156-kbyte/second transfer rate with approximately 7.5 million bytes of storage each) on selector channel 2, 5-6 tapes (9 channel, 120 kbyte/second) on selector channel 3, 20-40 terminals connected through the multiplexer, a card reader, and two printers also connected through the multiplexer.

## Algorithm Development

System simulations were performed to evaluate alternate scheduling and paging algorithms. The various algorithms that were exercised were built into the model structures along with the hardware characteristics. Parameters for task characteristics, which were used to effect different load conditions, were established. Consequently, both the models and the loads were modified regularly as the studies progressed.

The primary measures used to evaluate the relative performances of the algorithms were the response times and the effective compute times achieved by the hypothetical problem programs. Response time is defined as the period between the time the last character of a message is received from a terminal and the time the first character of the responding message is sent to the terminal. The effective compute time represents raw instruction time for user tasks. All I/O interference and overhead functions are netted out of this figure.

### Model Loading

A load of twenty-six user tasks was used to evaluate the relative performance of the six algorithms simulated. Of these tasks, nineteen were interactive and seven were batch. Two of the seven batch tasks were taken to be completely compute bound with no I/O. The average time range, before a new page was required, was 11 to 19 ms for the interactive tasks and 2 to 6 ms for the batch tasks. A total of 400 pages were required by the twenty-six tasks.

The I/O interference rate used in these studies was 1.68 μs/byte, rather than the present 1.44 μs/2 bytes. This interference rate implies that nearly 8 μs of processor time was used to accommodate data transfer for each page transferred. Subsequent studies incorporated the present interference rate, but they are not presented here as they do not pertain specifically to the algorithm development.

Table I is a condensation of the results obtained for the six algorithms.

Table I—Simulation Results for the 6 Models.

| *Model Number* | *Effective Compute Time (%)* | *Processor Idle (%)* | *Average Response Time (seconds)* |
|---|---|---|---|
| 1 | 24.8 | 68.9 | 5.48 |
| 2 | 25.1 | 69.3 | 0.81 |
| 3 | 28.9 | 51.7 | 0.81 |
| 4 | 30.3 | 50.0 | 1.59 |
| 5 | 30.2 | 49.5 | 1.69 |
| 6 | 41.6 | 10.9 | 2.14 |

### Model 1

The first model provides a basis for comparison for the subsequent, more sophisticated models. It has only one scheduling queue. The task at the head of the queue is given control of the processor when its predecessor is waited for an I/O or runs out its time slice. When a task runs out its time slice, it is placed at the end of the queue. When it is waited for an I/O operation, it is placed in a wait state. At the termination of the I/O, it is placed at the end of the queue. Each time a task gains control of the processor it receives a full time slice. At each paging interrupt, indicating the task requires a page not in memory, the processor is permitted to idle while the page is transferred from the drum to memory. The time required to make this transfer is charged to the task's time slice.

The paging algorithm always attempts to remove a page from memory belonging to a task other than the one in control of the processor. Pages that have not been modified are always selected before pages that have been written into.

### Model 2

In an attempt to improve the response times for the interactive users, a change was made in the queueing structure of model 1. Rather than placing a task at the end of the queue for the processor when an I/O wait condition is removed, it is placed in one of the 2 other queues. One queue is for the tasks for which responses have just been received from a terminal; that is, when a task has its I/O wait condition removed by receiving a message from a terminal, that task is placed in the queue of *tasks-with-I/O-from-the-terminal-completed.*

The second queue is for the tasks for which any other type of I/O operations (excluding paging operations) have been completed. The third queue now represents the queue of tasks that have run out their time slices.

When a task is waited for I/O or runs out its time slice, it is removed from control of the processor as in the first model. Now, however, if there is a task in the *terminal-I/O-completed* queue it is given control of the processor before any other task. Upon gaining control of the processor from this queue, a task always receives a new, full, time slice. If the *terminal-I/O-completed* queue is empty an attempt is made to service the *non-terminal-I/O-completed* queue. If there is a task in this queue, it is given control of the processor, but with the time slice set to that portion of its full time slice, which was not used. This procedure guards against the possibility of having tasks that repeatedly gain control of the processor from this queue, but never run out their time slice, thereby preventing a task that has run out its time slice from ever gaining control of the processor.

If both *I/O-completed* queues are empty, a task in the *time-slice-run-out* queue is given control of the processor. Such tasks always get a new full time slice. The paging algorithm for this approach is the same as for model 1.

#### Model 3

To improve processor utilization (and drum utilization) the next logical step was to permit two tasks to compete for the processor and the paging routine simultaneously; i.e., rather than always permitting the processor to idle when paging is in progress, one additional task is allowed to compete with the original for memory space and processor time. To effect this change, it was necessary to create an additional single task queue for the processor and a similar queue for the paging routine.

The paging rules in this environment become a bit more complex. The pages in memory associated with both active tasks must be identified as currently in use, yet a distinction between the tasks with which each page is associated must also be maintained.

When one of the two active tasks requires a page, the page-out decision attempts to pick a page that does not belong to either of the two active tasks. In the event that all the pages in memory belong to the active tasks, processing is delayed for the task requiring additional paging, and the other proceeds as though the system were operating with in-line paging only. The decision to delay the task requiring the page was based on ease of implementation.

Time-slice charges also became somewhat more complex in this environment. The difficulty was how to account for the processor idle time resulting from paging waits. The procedure established was to charge the time slice of the task for which paging was being performed.

The rationale for charging a time slice for processor idle time resulting from paging was based primarily on the concern that a single task might require more pages in memory than the available memory capacity would allow. If this condition arose and the task's time slice was not charged for in-line paging, it would be possible for the task to maintain control of the processor for many seconds, thus excessively lengthening response times. This scheme prevents such an occurrence although it does so at the expense of efficiency in the processing of a very large task.

#### Model 4

The next step in the algorithm development was to attempt to improve upon the use of the processor, memory, and paging mechanism by making the number of active tasks—the tasks permitted to compete simultaneously for these resources—more adaptive to the requirements of the tasks themselves. To do this, as many tasks as possible were made active, up to the point that the estimated number of pages for the active tasks, in the period they were active, did not exceed a prespecified maximum.

The estimated number of pages required by a task during an active period is obtained by using the count of the number of pages brought in for the task in its previous active period.

The prespecified maxium number of active pages is set equal to the actual number of pageable pages in the system. The actual number in the simulation run was set at 54. This number was chosen based upon an assumption that 10 of the 64 pages in the system would be devoted to resident control program functions. Variations in the prespecified maximum number of active pages over the range of 100 to 200 percent of the actual number of pageable pages in the system were tried also. No definitive gain was found, however, by altering the choice from the 100-percent value.

#### Model 5

As an offshoot of the adaptive approach, a combination of the adaptive and two-at-a-time techniques was tried. In this approach, whenever the adaptive technique would normally permit only one task to be active, a second task was also made active.

#### Model 6

Analysis of the results for the first five models showed that to overcome the high level of processor idle time, it would be necessary to be able to use the processor when it was otherwise idling for paging. This was achieved by making one of the non-interactive tasks resident. Model 6 extends model 5 to incorporate this feature.

As indicated in Table 1, making a single task resident and changing no other parameters resulted in a 38% improvement in processor utilization over our previous best results, and maintained the average response time within tolerable limits.

### Task Scheduling

The previously described simulation effort led to the development of the two central control algorithms: *the task scheduling algorithm and the paging algorithm.*

The task scheduling algorithm is based on the adaptive approach of model 4. This model has been extended, however, to incorporate additional queues, multiple resident tasks, interrupt-driven tasks, a modification in the procedure to make tasks inactive, and a cycle-checking function. In addition, the paging algorithm has been extended to accommodate the modifications in the task scheduling algorithm.

#### The Queue Structure

The technique used in the structuring of queues is to give precedence to interactive tasks by dividing each major queue into subqueues such that tasks that have terminals attached to them are placed in higher priority subqueues within the main queues. The three main queues in the system are 1) the queue for the processor, 2) the queue for the paging routine, and 3) the queue for ready, non-resident inactive tasks. Fig. 1. shows the queue structure and the flow through the system.

#### The Path of an Active Task

The normal path of a task, once it is made active is 1) to enter the queue for the paging routing, 2) to have its requests processed by the paging routine, 3) to enter the queue for the processor, and 4) to gain control of the processor and to process until a) it requires a page not in memory, b) it runs out its time slice, or c) it is waited for an I/O operation.

#### Resident Tasks

If the paging demand rate of the active pageable tasks is greater than the drum

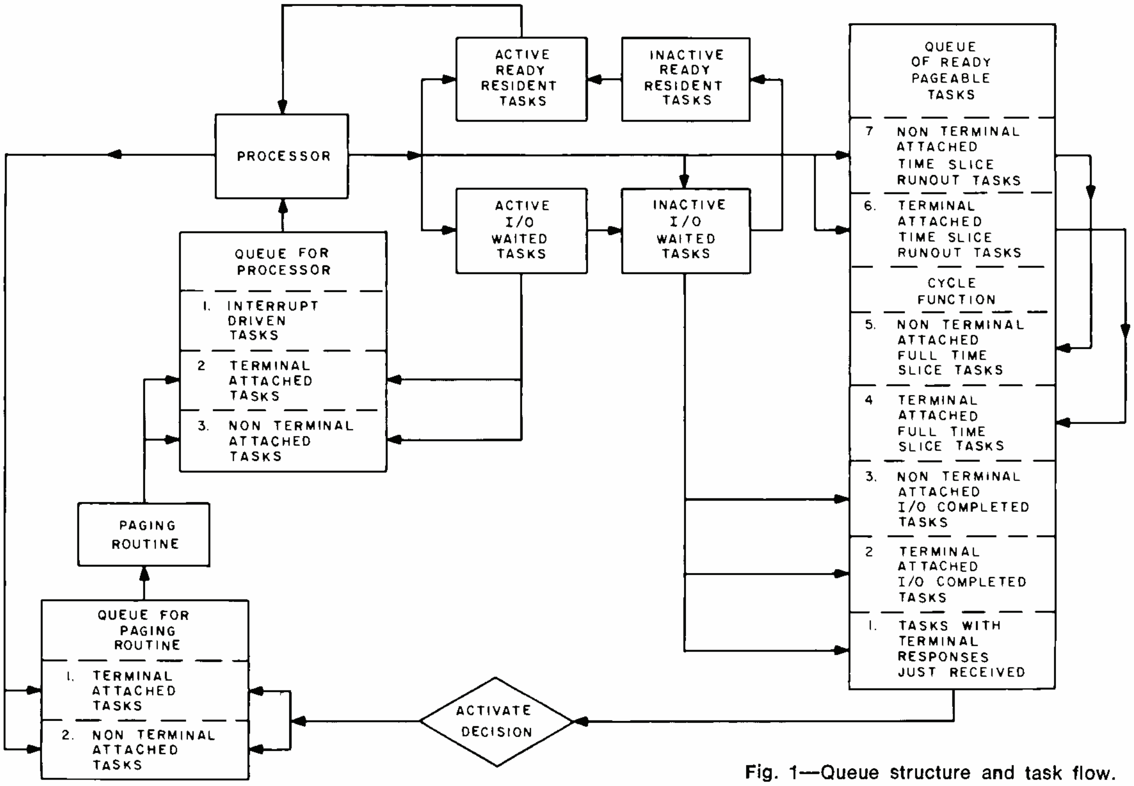


Fig. 1—Queue structure and task flow.

transfer rate, the queue for the processor will tend to be depleted. If this queue is empty when a task completes processing and there is a task in the *active-ready-resident* queue the resident task is given control of the processor. However, as soon as paging is completed for one of the pageable tasks the resident task is put back on the *active-ready-resident* queue and the pageable task is given control of the processor.

This procedure gives the pageable tasks almost as much processor time as when there is no resident task. The results of the simulations have shown that the processing achieved for the resident task is virtually free when contrasted to a purely paging environment.

### Interrupt Driven Tasks

While a task is in control of the processor it may be interrupted as a consequence of a number of different asynchronous operations, which do not require any loading or unloading of the translation memory. One such interrupt, that of an I/O termination for an interrupt driven task, is of particular interest.

When there is an interrupt for an interrupt-driven task, an attempt is made to service the task immediately. The code for interrupt-driven tasks is kept resident in system virtual memory and can be made resident in physical memory when the task is initiated. Therefore, if there is not another interrupt-driven task in control of the processor, the task for which the interrupt occurred may be given control immediately without any TM change or paging required. If there is another interrupt-driven task processing, the task for which the interrupt occurred is placed in the highest priority of the subqueues for the processor.

A task removed from control of the processor by an interrupt-driven task is placed at the head of the processor subqueue for terminal-attached tasks. This procedure guarantees that this task will be the next non-interrupt-driven task to be given control of the processor. No changes in either TM or physical memory are necessary to accommodate this task-switching process.

### Active, I/O Waited Tasks

One special feature has been incorporated into the algorithm for active tasks in an attempt to reduce the amount of paging in the system. This feature is an attempt to keep active a pageable task that has been waited for I/O. When a pageable task is waited for a nonterminal I/O, it is not made inactive unless, or until, the paging routine is not busy.

The normal effect of making a task inactive is to initiate the activate routine to attempt to make additional tasks active. As tasks are made active, they are placed in the queue for the paging routine and, when paging is performed, the newly paged task will tend to occupy the memory space previously occupied by the task just made inactive. As a consequence, when that task is again made active most of its pages will have been removed from memory and it is necessary to initiate paging action to bring them in again. However, if the paging routine is kept busy without making additional tasks active, there will be no gain in terms of processor utilization by making these tasks active. It is for this reason that we made the decision to delay making I/O waited pageable tasks inactive.

### A Cycle-Checking Function

The processing of the inactive ready queue is in priority order. But to guard against a condition that would allow a terminal-attached task (which continually runs out its time slice) from monopolizing the processor, a cycle-checking function has been added within this queue.

When the cycle checking function is initiated, no tasks for which I/O waits were removed (or which are entitled to new time slices) are left in the inactive ready queue. The only tasks that might be in the queue would be in the subqueue for tasks that have run out their time slices. The cycle checking function transfers these tasks to the new time slice subqueues, permitting them to be made active.

## The Paging Algorithm

The paging algorithm may be divided into two parts; the analysis routine and the paging routine. The analysis routine gains control of the processor whenever a paging queue interrupt occurs. This routine, utilizing a hardware special analysis function, determines all of the pages that are involved in the execution of the instruction causing the interrupt. The analysis routine then determines which pages, required by the instruction, are not resident in memory and builds a stack containing all the information needed to bring these pages into memory. Finally, the analysis routine places the tasks that caused the interrupt in the queue for the paging routine.

The paging routine gains control of the processor whenever a paging drum I/O termination is detected, or when the paging drum is idle, upon the termination of the processing of the analysis routine.

To facilitate the selection of the pages to be paged out, link lists are maintained of *accessed-only* and *written-to* pages that are in memory and associated with inactive tasks. These lists are extended each time a task is made inactive, and they are purged when pages belonging to a task just made active are still located on the lists.

Selection of the page to be paged out is made from the accessed-only list unless it is empty, in which case a written-into page is chosen. If both lists are empty, the pageable control program and shared pages in memory are checked for recency of access. Those that were not accessed during the last 500-ms period are then linked to the appropriate list and the normal scan is resumed. If there are not more control program or shared pages eligible to be linked, an I/O-waited active task (in LIFO order) is made inactive and its pages are linked to the appropriate lists. If there are no I/O-waited tasks to make inactive, tasks are removed from the queue for the paging routine (in LIFO order). These tasks are made inactive, their pages are linked appropriately, and the tasks themselves are requeued to the head of the subqueue of tasks with terminal responses completed. (This queue placement ensures that tasks made inactive in this manner will be made active again before any other tasks in the system.) Finally, the condition may arise that all the pageable pages in memory are associated with the tasks for which the paging is being performed. When this condition occurs, it becomes necessary to cannibalize from the task itself. In this instance, care is taken not to remove from memory any page required for the current instruction.

### Prepaging

When a task is made active, the P-counter page is brought into memory before the task is given control of the processor. The need for this page is clear and its identity is readily obtained because this data was stored when the task was made inactive.

A similar procedure to permit prepaging additional pages has been established. However, concern over the 3-ms-per-page loss incurred for every incorrect page brought into memory has led to a postponement in the implementation of more extensive prepaging until a more comprehensive evaluation can be made using live data.

### Time-Slice Alternatives

Analysis of various simulation runs have shown that very few tasks tend to run out of their time slice during a single period of being made active.

The significance of the time slice, therefore, tends to be that it prevents a single task from monopolizing control of the processor for an extended period of time. As another effect, it also tends to get I/O orders issued from I/O-oriented tasks that would otherwise tend to be postponed until a compute-bound task completed processing. This procedure, in turn, improves the overall use of all system resources and improves system throughput. Thus, provided the time slice is not made as small as to cause an undue amount of system overhead, the size of the time slice is not too significant. Currently, a time slice of 2 seconds is provided for every task in the system.

## Conclusions

The Spectra 70/46 TSOS is a medium-scale system oriented to a mixed batch processing and time-sharing environment. The hardware characteristics, which (in conjunction with environment) largely determined the structure of the algorithms, should be fairly representative of the price and performance class in which this system is located. In particular, the memory size, the processor speed and the existence of I/O interference reasonably typify systems in the medium-scale range. The results of studies indicate that a medium-scale system capable of supporting a reasonable amount of interactive activity could reasonably incorporate the following guiding principals.

Where I/O interference significantly reduces processor speed, one or more batch tasks should be made resident in memory. Demand paging should then be considered for interactive tasks to avoid the high cost of mistakes in bringing the wrong pages into memory.

A mechanism should be provided for monitoring and adjusting the load on critical system resources such as I/O channels and memory to prevent oversaturation. A method of dynamically and adaptively controlling the load competing for these resources avoids significant degradation of system performance.

A queueing structure more sophisticated than the simple round-robin approach should be implemented. Distinction among the factors causing tasks to give up control of the processor provide a reasonable basis for the queue organization. Servicing of queues to put interactive tasks back in control of the processor as soon as possible distinctly improves response times at no cost to processor utilization.

### BIBLIOGRAPHY

Dennis, J. B., *private communication.*

# Microwave solid-state power sources

**Dr. Fred Sterzer**

**There are today five important classes of solid-state devices for generating microwave power: microwave transistors, transistor-driven varactor harmonic-generator chains, tunnel diodes, avalanche transit-time devices, and devices whose operation depends on the negative differential mobility produced in certain semiconductors by the transfer of conduction electrons from high-velocity states to low-velocity states.**

TRANSISTORS, varactor diodes, and tunnel diodes are pure junction devices. Avalanche transit-time devices contain junctions, but some of the processes essential to their operation occur in the bulk of the semiconductor. Transferred-electron devices are pure bulk devices devoid of any junctions. The pure junction devices are the oldest and most mature members of this group of devices, and not surprisingly their rate of improvement has begun to slow down perceptibly. Bulk devices, on the other hand, are at an early stage in their development, and important improvements in performance are forthcoming at a rapid rate. This paper summarizes the state-of-the-art of all five classes of solid-state microwave power sources, with emphasis on bulk devices because the reader is most likely already familiar with the operation of the junction devices.

## MICROWAVE TRANSISTORS

Most modern microwave transistors are silicon planar epitaxial diffused N-P-N structures with emitter geometries that combine long emitter edges (for high current capabilities) with large ratios of emitter edge to emitter area (for short emitter transit times). [Note—at high current levels, the emitter current, $I_e$, is concentrated at the edge of the emitter, so that the maximum value of $I_e$ is proportional to the length of the emitter edge. The emitter transit time, $T_e$, which is one of the important factors limiting the frequency response of a transistor, is proportional to the ratio $C_e/I_e$ where $C_e$ is the emitter capacitance. $C_e$ is proportional to the emitter area, so that $T_e$ is inversely proportional to the ratio of emitter edge-to-area.] The two most widely used emitter geometries are the RCA-developed "overlay" geometry[1], in which as many as 408 individual emitter sites are tied together by an aluminum overlay (Fig. 1), and the interdigital geometry in which the emitters are built like a pair of interlocking combs. The power outputs obtainable from single microwave transistors are at present approximately as follows: 40 W at 400 MHz, 15 W at 1 GHz, 5 W at 2 GHz, 200 mW at 3 GHz, and a few milliwatts at 6 GHz.

Microwave transistors are now generally mounted in low-inductance packages that are specifically designed for operation at microwave frequencies. Two such packages have been developed by D. R. Carley's group at Electronic Components in Somerville (Fig. 2). One package is for operation in coaxial circuits, and the other is for operation in stripline circuits. Transistor chips mounted in these microwave packages perform significantly better at microwave frequencies than similar chips mounted in conventional low-frequency transistor packages.

High-power microwave transistor amplifiers and oscillators are usually built in either coaxial line or stripline. Fig. 3 is a photograph of a compact coaxial amplifier that uses three transistors mounted in parallel on the outer shell of a coaxial line. This amplifier, designed by H. C. Johnson of the Microwave Applied Research Laboratory in Princeton, has a power output of 21 W at a gain of 7 dB, a collector efficiency of 65%, and a bandwidth of 25 MHz centered at a frequency of 700 MHz. At lower power levels, hybrid integrated circuits on alumina or sapphire substrates are increasingly being used. A photograph of a thin-film, lumped-element hybrid integrated 2-GHz amplifier built on a polished sapphire substrate is shown in Fig. 4. This amplifier has a power output of 1 W at a gain of 4.7 dB, a collector efficiency of 30%, and a 1-dB bandwidth in excess of 7%. It was built by S. P. Knight, D. A. Daly, and M. Caulton of the Microwave Research Laboratory in Princeton.

Transistors can also be operated as amplifier-multipliers and oscillator-

**Dr. Fred Sterzer, Director**
Microwave Applied Research Laboratory
Electronic Components, Princeton, N.J.
received the BS in physics in 1951 from the College of the City of New York, and the MS and PhD degrees in 1952 and 1955, respectively, from New York University. From 1952 to 1953 he was employed by the Allied Control Corporation in New York. During 1953 and 1954 he was an instructor in physics at the Newark College of Engineering in New Jersey, and a research assistant at New York University. He joined the RCA Electron Tube Division in Harrison, N.J., in October, 1954, and transferred to Princeton, N. J., in 1956 where he is now Director of the Microwave Applied Research Laboratory. Dr. Sterzer's work has been in the field of microwave spectroscopy, microwave tubes, light modulators and demodulators, microwave solid state devices (including parametric amplifiers, tunnel-diode microwave amplifiers and frequency converters, microwave computing circuits, and bulk-effect devices. Dr. Sterzer is a member of Phi Beta Kappa, Sigma Xi, the American Physical Society, and the IEEE. He holds 18 patents in the microwave field and is the author of approximately 50 technical papers.

---

Reprint RE-14-3-1 (ST-3613)
Final manuscript received May 2, 1968.

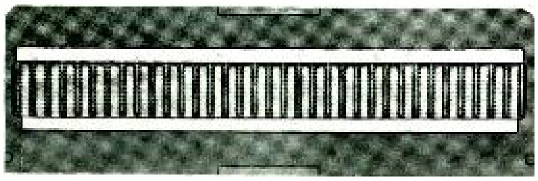


Fig. 1—Photograph showing the 408 emitters of the RCA Dev. No. TA2675 overlay transistor. The dimensions of each emitter are 0.1 mil x 1.4 mils.

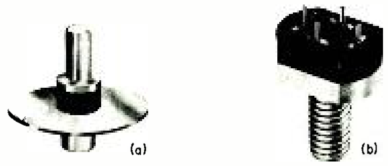


Fig. 2—Photograph of two transistor packages specifically designed for operation at uhf and microwave frequencies: a) coaxial package, b) stripline package.

multipliers to produce power output at frequencies well above their maximum fundamental frequency of oscillation. The nonlinear element primarily responsible for harmonic generation in transistors is the depletion-layer-capacitance of the base-collector junction. This junction acts much as the junction in the varactor diode, and most of the design considerations for varactor multipliers can also be applied to the design of transistor multipliers.

Caulton, et al[2] of the Microwave Research Laboratory have built amplifier-multipliers using a single 2N3375 overlay transistor. With an input of 1 W at 500 MHz, they obtained the following results:

| *Frequency (GHz)* | *Power Output (watts)* | *Collector efficiency (RF output/ DC Input—%)* |
|---|---|---|
| 1.0 | 2.6 | 41 |
| 1.0 | 3.6 | 30 |
| 1.5 | 1.8 | 20 |

### TRANSISTOR-DRIVEN VARACTOR HARMONIC-GENERATOR CHAINS

During the past two years, substantial improvements in the power-handling capabilities of varactor multipliers have been achieved by the following techniques: 1) use of multiplier circuits in which the RF power is divided between two or more varactor diodes,[3-5] 2) use of multiple-junction varactor diodes[6], and 3) use of series stacked varactor diodes[7]. The current state-of-the-art of varactor multipliers is summarized in the following table:

| *Input freq. (GHz)* | *Mult. factor* | *Output power (W)* | *Eff. (%)* | *No. of varactor diodes* | *Ref.* |
|---|---|---|---|---|---|
| 0.37 | 4 | 17.7 | 45 | 2 | 5 |
| 0.9 | 2 | 1.8 | 88 | 1 | 4 |
| 0.9 | 2 | 16 | 80 | 2 | 4 |
| 0.9 | 3 | 8 | 55 | 4 | 4 |
| 1.5 | 4 | 3 | 55 | 4 | 3 |
| 4 | 3 | 1.2 | 62 | 1[a] | 7 |

[a] One stacked varactor using four diodes in series.

The state-of-the-art of the CW power output of transistor-driven varactor multiplier chains is approximately as follows: 20 W at 2 GHz, 10 W at 4 GHz, 5 W at 8 GHz, and 1 W at 13 GHz.

For multiplication ratios greater than four, step-recovery diodes (i.e., varactor diodes that are specifically designed to take advantage of charge-storage effects) are usually used. The following data are representative of the performance of multipliers using a single step-recovery diode:[8]

| *Input Freq. (GHz)* | *Input Power (W)* | *Output Freq. (GHz)* | *Power Output (W)* |
|---|---|---|---|
| 0.2 | 18 | 2 | 2.4 |
| 0.4 | 15 | 2 | 5 |
| 2 | 5 | 10 | 0.35 |

### TUNNEL DIODES

Microwave tunnel-diode oscillators are only useful in applications that require microwatts or at most a few milliwatts of power output.[9] The state-of-the-art of tunnel-diode oscillators is summarized in the following table. The table shows that the power output of tunnel-diode oscillators is limited to the milliwatt range at the lower microwave frequencies, and to the microwatt range at the higher microwave frequencies.

| *Fixed-freq. oscillators* | *Freq. (GHz)* | *Output Power (mW)* | *Method of Tuning* | *Ref.* |
|---|---|---|---|---|
| | 1.6 | 26 | —— | 10 |
| | 6 | 4 | —— | 11 |
| | 50 | 0.2 | —— | 12 |
| *Electronically tunable oscillators* | | | | |
| | 0.9-1.8 | 0.4 | Bias | 13 |
| | 1.8-4.0 | 0.5 | Varactor | 14 |
| | 5.1-10.2 | 0.5 | Garnet | 15 |

*Note:* In bias tuning, the oscillation frequency is varied by changing the bias voltage across the tunnel diode; in varactor tuning, a varactor diode is incorporated in the resonator of the oscillator and tuning is accomplished by varying the voltage across the varactor; in garnet tuning, the resonant circuit is a garnet, and the frequency is varied by varying the magnetic field in the garnet.

### AVALANCHE TRANSIT-TIME DIODES

In 1958, W. T. Read[16] of Bell Telephone Laboratories proposed a multilayer N$^{++}$-P$^{+}$-I-P$^{++}$ diode for generating microwave power. The operation of this diode was based on a combination of impact avalanche breakdown and electron transit-time effects, and diodes of this general type are now known as avalanche transit-time diodes, IMPATT (impact avalanche and transit time) diodes, or simply avalanche diodes. The *avalanche transit time* principle was first experimentally verified in 1965 by Johnston, DeLoach, and Cohen[17], who achieved pulsed power outputs of 80 mW at 12 GHz from a silicon P-N junction diode driven into avalanche. Advances since 1965 have been so rapid that today avalanche-diode oscillators are established as one of the most important new microwave solid-state power sources.

In the diode proposed by Read, holes are generated by avalanche multiplication in a narrow *avalanche* region and are injected into an adjacent drift region. The diode is biased so that the field in the avalanche region is high enough to cause breakdown during the positive half of the RF voltage cycle, but is below the critical field required for breakdown during the negative part of the voltage cycle. As a result, the hole current generated in the avalanche region grows during the positive half of the RF voltage cycle and dies down during the negative half. The hole current therefore reaches its maximum value one quarter of a cycle after the voltage reaches its maximum, i.e., the hole current lags the RF voltage by 90°. During the entire RF cycle, the electric field in the drift region is kept high enough to cause the injected holes to travel with the (constant) limiting velocity. If the AC hole current injected from the avalanche region into the drift region is given by $I_A = I_o \cos \omega\tau$, then the current through the drift region due to this injected current (assuming one-dimensional flow) will be given by*

$$I_T = I_o \frac{\sin(\omega\tau/2)}{(\omega\tau/2)} \cos(\omega\tau - \omega\tau/2) \tag{1}$$

where $\tau$ is the transit time of the injected holes through the drift region. [Note—Eq. 1 can be derived as follows: the total current (i.e., the sum of the conduction current $I_C$ and the

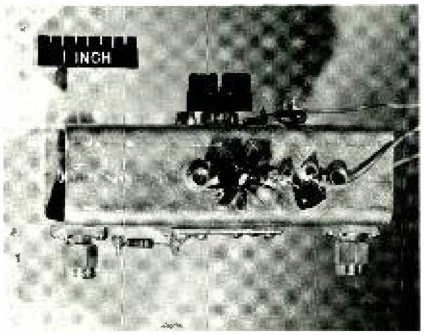


Fig. 3—Photograph of a 20-watt coaxial 700-MHz amplifier using three transistors in parallel.

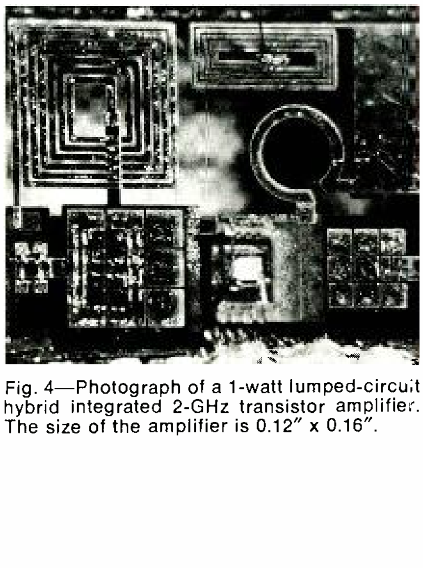


Fig. 4—Photograph of a 1-watt lumped-circuit hybrid integrated 2-GHz transistor amplifier. The size of the amplifier is 0.12" x 0.16".

displacement current $A\,\partial D/\partial t$) must be continuous, i.e., $\partial\,(I_c + A\,\partial D/\partial X)/\partial x = 0$. Therefore

$$\frac{1}{w}\int_0^w \left( I_c + A\,\frac{\partial D}{\partial t}\,dx \right) = I_c$$

$$+ A\,\frac{\partial D}{\partial t} = \frac{1}{w}\int_0^w \left( I_c + C\,\frac{dV}{dt} \right) dx$$

where $w$ is the width of the depletion region and $C = \varepsilon A/w$ is the capacitance. Now,

$$\frac{1}{w}\int_0^w I_c dx = \frac{1}{w}\int_0^w I_o \cos\omega\,(t - x/v)\; dx$$

$$= I_o\,\frac{\sin(\omega\tau/2)}{(\omega\tau/2)}\,\cos(\omega\tau - \omega\tau/2),$$

where $I_o \cos\omega\tau$ is the current injected at $x = 0$, and $v$ is the drift velocity of the charge carriers.]

Eq. 1 shows that $I_T$ lags $I_A$ by a phase angle of $\omega\tau/2$. Therefore, if the length of the drift region is chosen to make $\omega\tau = \pi$, then $I_T$ will lag $I_A$ by 90°, and the phase difference between the applied voltage and $I_T$ will be 180°; in other words, the diode will act as a negative resistance.

Gilden and Hines[18] have carried out a detailed small-signal analysis of Read's avalanche diode and have derived the following expression for its impedance:

$$Z = r_s + \frac{w^2}{v\varepsilon A\,(1 - \omega^2/\omega_a^2)} + \frac{1}{j\omega C\,(1 - \omega_a^2/\omega^2)} \qquad (2)$$

where $r_s$ is the (positive) series resistance of the diode structure, $C$ is the depletion layer capacitance when the diode is biased just below the avalanche voltage, $\omega_a$ is the "avalanche" frequency given by $(2\alpha_a v I_o/\varepsilon A)^{1/2}$, $\alpha_a$ is the derivative of the average ionization coefficient with respect to the electric field, and $I_o$ is the average current through the diode. The equivalent circuit corresponding to Eq. 2 is shown in Fig. 5. The second term in Eq. 2, which corresponds to $R_d$ in the equivalent circuit, becomes negative for $\omega > \omega_a$, and the diode becomes an active device whenever $-R_d > r_s$. The third term of Eq. 2 corresponds to the parallel resonant circuit of Fig. 5. Note that its resonant frequency is proportional to the square root of the current. [Note—Eq. 2 was derived with the assumption that the ionization rate and drift velocity of electrons and holes are equal. An expression for the impedance of Read-type avalanche diodes for the case of unequal electron and hole ionization rates and drift velocities is given in reference 19.]

Most practical microwave avalanche diodes differ from Read avalanche diodes in that avalanching is not confined to a narrow well-defined region but rather occurs over a significant portion or over the entire depletion region. These types of diodes behave similarly to Read diodes and also exhibit negative resistance when the transit time of the charge carriers becomes a significant fraction of an rf period.[20]

Microwave avalanche diode devices have been fabricated in a variety of semiconductor materials (Ge, Si, GaAs), shapes (mesa and planar), and impurity profiles (abrupt and graded P-N junctions, P-I-N junctions, $N^{++}$-$P^{+}$-I-$P^{++}$ junctions, etc.). At this early date in the development of avalanche diodes, it is not known which of these many types is optimum for a given application.

The state-of-the-art of microwave avalanche-diode oscillators is summarized in Table I. The table shows that power-(frequency)$^2$ products of nearly 4000 watts-(GHz)$^2$ have already been achieved. This Pf$^2$ product is more than two orders of magnitude greater than the largest Pf$^2$ product that can be obtained from transistors.

Both the AM and the FM noise of avalanche-diode oscillators are quite high.[28,29] Thus, for example, if an avalanche-diode oscillator is to be used as the local oscillator in a superheterodyne receiver, it is usually mandatory to use a balanced mixer to suppress at least some of the oscillator noise.

Avalanche diodes have also been used successfully as the active element in reflection-type circulator-coupled microwave amplifiers. Such amplifiers are much noisier than, for example, tunnel-diode amplifiers, but their power output can be at least two orders of magnitude greater. The lowest reported noise figures of amplifiers are about 40 dB with Si diodes, and about 30 dB with Ge diodes. With GaAs diodes, J. R. Collard and J. Kuno of the Microwave Applied Research Laboratory have obtained noise figures of about 20 dB.[30,31] An amplifier with saturated cw power output as high as 40 mW has been reported by Scherer.[32] This amplifier, which uses a Si diode, is tunable from 8 to 12 GHz, and has a gain of 10 to 15 dB and an instantaneous bandwidth of 30 to 100 MHz.

### TRANSFERRED-ELECTRON DEVICES

In 1961, Ridley and Watkins[33] suggested the possibility of obtaining bulk negative resistance in certain semiconductors by transferring electrons heated by high electric fields from high-mobility to low-mobility conduction sub-bands. Detailed calculations by C. Hilsum[34] in 1962 showed that GaAs and GaAs-GaP alloys were good candidates for exhibiting bulk negative resistance through transferred-electron effects, and the first experimental verification of this bulk negative resistance was achieved in GaAs by Gunn in 1963.[35] Bulk negative resistance has also been observed in compounds like InP, CdTe, ZnSe, and InAs, but so far only GaAs and GaAs-GaP alloys have produced useful amounts of microwave power.

Fig. 6 shows the drift velocity of conduction electrons as a function of electric field for N-type GaAs. For electric fields below about 3kV/cm, the drift velocity is proportional to the electric field; for drift velocities above about 3kV/cm, the drift velocity decreases as the electric field is increased. In other words, the differential mobility of GaAs is positive for electric fields below about 3kV/cm, but is negative for electric fields above about 3kV/cm. [The mobility becomes positive again when the electric field in the crystals becomes high enough to cause breakdown (~150 kV/cm).] The transfer of electrons from high-velocity states to low-velocity states takes place in a time short compared to the period of a microwave signal; Fig. 6 should therefore be valid for electric fields varying at microwave rates.

Fig. 7 shows a slab of high-resistivity N-type GaAs of thickness $l$ with ohmic contacts on opposite faces. The sample is initially in thermal equilibrium (no applied voltage), and then a voltage step function of amplitude $V = E_a l$ is applied across its ohmic contacts, where $E_a$ is greater than the threshold field $E_m$ defined in Fig. 6, and where the risetime of the step function is a small fraction of the dielectric relaxation time. [The dielectric relaxation time is the time constant of the growth or decay of space charge in a semiconductor.] When the voltage first reaches its full value $V$ the charge distribution in the sample has not had time to change from its initial thermal equilibrium distribution, and the electric field throughout the sample is therefore at first uniform and equal to $V/l$ any edge effects are neglected. At this point in time, the signal AC equivalent circuit of the sample consists of a negative resistance $R$ in parallel with a capacitance $C$, where $R$ and $C$ are given by:

$$R = -\frac{l}{en|\mu|A} \text{ and } C = \frac{A\varepsilon}{l} \tag{3}$$

In Eq. 3, $e$ is the electronic charge; $n$ is the donor density; $|\mu|$ is the mobility corresponding to a field $V/l$; $A$ is the cross-sectional area of the sample; and $\varepsilon$ is its dielectric constant.

Once the voltage across the slab of GaAs is established and current starts to flow, the charge distribution and the electric field in the slab rapidly become non-uniform, and Eq. 3 ceases to be applicable. The steady-state distribution of the charges and of the electric field in the sample will be stable (i.e., non-oscillatory) if[37]

$$nl < 2.7\text{x}10^{11}\text{cm}^{-2} \tag{4}$$

The impedance $Z$ of a stable sample is approximately*

$$Z \sim \frac{l^2}{\varepsilon v_0 A} \; \frac{e^{-\alpha} + \alpha^2 - 1}{\alpha^2} \tag{5}$$

where $v_o$ is the average drift velocity of the electrons in the crystal; $\alpha = [(\xi ne/\varepsilon) - j\omega]\, l/v_o$, and $\xi$ is the average value of the derivative of the drift velocity of the electrons with respect to the electric field. [Note—Eq. 5 can be derived starting from Poisson's equation in one dimension: $\partial^2V/\partial x^2 = -\partial E/\partial x = -p/E = (p_d - J_c/v)/E$ where $p$ is the charge density; $p_d$ is the donor density (assumed to be uniform throughout the crystal); $J_c$ is the conduction current density; and $v$ is the drift velocity of the electrons. Now $J_c = J - \varepsilon\, \partial E/\partial t$, where $J$ is the total current density, so that Poisson's equation can be rewritten as $\varepsilon\, \partial E/\partial x = p - J/v + (\varepsilon/v)\, \partial E/\partial t$. Assume $E = E_o + E_1\, e^{j\omega t}$; $J = J_o + J_1 e^{j\omega t}$; and $v = v_o - \xi E_1 e^{j\omega t}$ $(\xi E_1 << v_o)$. Putting these assumptions into Poisson's equation yields

$$E_1 = \frac{J_1}{j\omega\varepsilon - \varepsilon J_o/v_o}\left[1 - e^{(j\omega\varepsilon - J_0/V_0)}\, x/v_o\varepsilon\right]$$

The AC impedance of the sample is $Z =$

$$\frac{1}{A}\frac{\partial V_1}{\partial J_1} = \frac{1}{A}\frac{\partial}{\partial J_1}\left[-\int_0^l E_1\, dx\right],$$

which becomes Eq. 5 if one assumes that $J_o/v_o \sim ne$.] The real part of Eq. 5, i.e., the AC resistance of the sample, is positive at all frequencies if $\xi > 0$. However, if $\xi < 0$, then the AC resistance is negative at frequencies in the neighborhood of the transit-time frequency $f_t = v_o/l$ and its harmonics, and the sample can be used as the active element in negative resistance amplifiers.

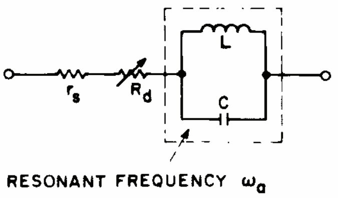


Fig. 5—Small-signal equivalent circuit of a Read avalanche diode.

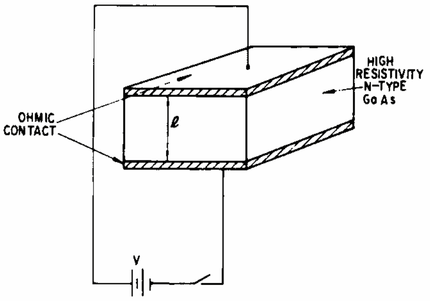


Fig. 7—Sketch of a transferred-electron device.

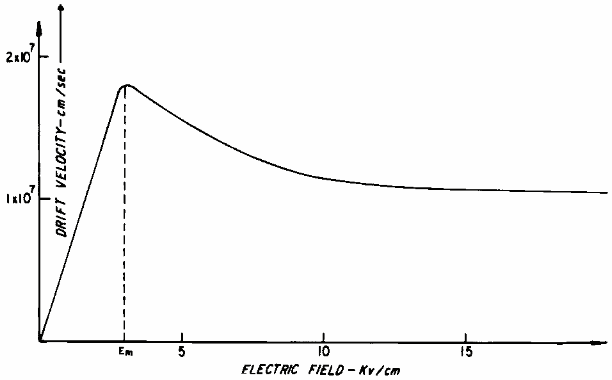


Fig. 6—Drift velocity of conduction electrons in GaAs as a function of electric field.

If inequality 4 is not satisfied, travelling space charge dipole layers will generally form in the sample, and its steady state will be oscillatory. The electric field inside such dipole layers is very high—typically of the order of tenths-of-thousands of volts/cm, while the electric field outside the dipole layers is less than the threshold field $E_m$. The dipole layer and the accompanying high-field region or high-field 'domain' usually form at or near the cathode, move through the crystal with the drift velocity of the electron stream, and disappear at the anode. A new domain is then formed and the process is repeated. The fundamental frequency of oscillation will be approximately equal to the transit-time frequency $f_t = v_o/l$.

Several important modes of operating bulk GaAs diodes depend on superimposing an RF voltage on the DC bias voltage. In one of these modes the frequency of oscillation is made higher than the transit-time frequency $f_t$ by extinguishing each domain before it reaches the anode. This is accomplished by adjusting the RF circuitry surrounding the diode in such a way that the sum of the DC voltage (which is greater than the threshold voltage $E_m/l$) and the RF voltage becomes appreciably smaller than $E_m/l$ whenever a domain has traversed a given distance across the crystal. The frequency of oscillation will in this case be approximately $v_o/l_d$, where $l_d$ is the distance from the cathode to the point in the sample where the domains are extinguished.

A second important mode of operation that depends on the existence of an RF voltage across the sample is the limited space-charge accumulation (LSA) mode first described by Copeland.[37] In LSA operation, the sample is DC-biased with a voltage $V_{DC}$ several times the threshold voltage $V_m$, and the RF circuitry is adjusted to produce an RF voltage $V_{RF}$ large enough so that the sum of $V_{DC}$ and $V_{RF}$ is less than $V_m$ during part of each cycle. The doping of the sample and the frequency of oscillation are selected so that the dielectric relaxation time $\varepsilon/\mu_n e$ is a small fraction of an RF period when $E$ is less than $E_m$, but is greater than about 3 RF periods when $E$ is greater than $E_m$; i.e., for N-type GaAs:

$$3\varepsilon/|\mu_n|e \approx 2\text{x}10^5 > n/f >>$$

$$\varepsilon/\mu_0 e \approx 1.4\text{x}10^3 \text{ sec/cm}^3 \quad (6)$$

where $\mu_n$ is the average (differential) mobility for $E > E_m$, and $\mu_0$ is the average mobility for $E < E_m$. As a result of inequalities 6, no high-field domains are formed in LSA operation. During the portion of the RF cycle when the average electric field $E_{AVE}$ is greater than $E_m$, there is not enough time to accumulate all the space charge necessary for the formation of a domain, and whatever space charge is accumulated is dissipated during the portion of the cycle when $E_{AVE}$ is less than $E_m$. The frequency of oscillation in the LSA mode is therefore independent of the carrier transit time and is solely determined by the circuit around the sample. As a result, at a given frequency of operation an LSA sample can be made much thicker than a sample operating in the transit-time mode (LSA samples have been successfully operated with thicknesses of up to 100 $V_0/f\varepsilon$) and the impedance of LSA samples operating at a given power level and frequency can be made much higher than the impedance of transit-time devices. [Note—The maximum thickness of the active region of transistors, avalanche transit-time diodes, and transit-time transferred-electron devices is inversely proportional to the operating frequency. As a result, all of these devices have fundamental limitations on their maximum (power-impedance level) (frequency)$^2$ products, as has been shown by E. O. Johnson[38] of RCA Electronic Components and by B. C. DeLoach.[39] Since the maximum thickness of an LSA sample is — in principle, at least — independent of frequency, Johnson-DeLoach-type theories are not applicable to LSA devices; it is not known at present which factors will ultimately limit the power output of LSA devices.]

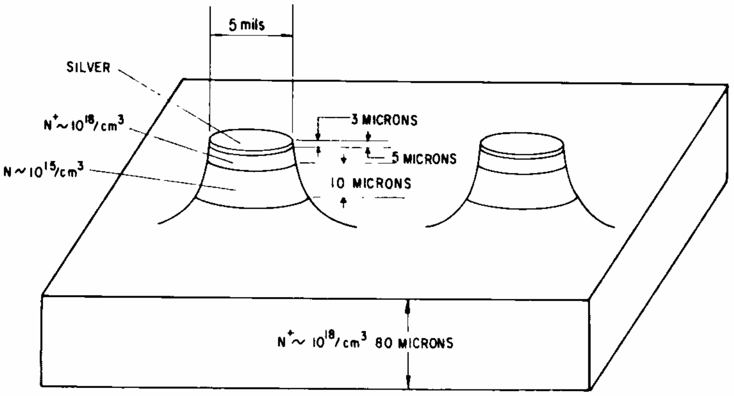


Fig. 8—Sketch of a two-mesa X-band cw transferred-electron oscillator. The mesas are chemically etched from an epitaxial wafer.

Most modern transferred-electron (TE) devices are fabricated from epitaxial GaAs. Epitaxial rather than bulk-grown material is used for the following reasons:

1) Epitaxial material is generally purer, more homogeneous, and more reproducible than bulk grown material.
2) High-resistivity epitaxial N-type material can be capped with N+ layers to avoid the difficult problem of having to make ohmic contacts directly to high-resistivity material.
3) Epitaxial material, unlike most bulk material, usually has a positive temperature coefficient of resistance, and thus offers a built-in protection against thermal runaway.

Fig. 8 shows a sketch of a two-mesa X-band TE devices fabricated from GaAs $N^+$-N-$N^+$ sandwiches epitaxially grown by A. Gobat of the Microwave Applied Research Laboratory using a vapor-phase deposition technique developed by J. J. Tietjen and J. A. Amick of the Materials Research Laboratory in Princeton. S. Y. Narayan of the Microwave Applied Research Laboratory has obtained as much as 60 mW of cw power from such a single-mesa TE device and 220 mW from a device with six mesas in parallel. A transferred-electron oscillator (TEO) based on the work by Gobat and Narayan is now being marketed by RCA Solid-State Microwave Operations (RCA S-229). A photograph of a TEO built by B. E. Berson and J. F. Reynolds of the Microwave Applied Research Laboratory is shown in Fig. 9. This oscillator has a pulsed power output of 100 W at 1090 GHz.

The state-of-the-art of TEO's using a sandwich geometry is summarized in Table II. The table shows that power-(frequency)$^2$ products in excess of

**Table I. State-of-the-art of avalanche transit-time diode oscillators.**

| *Frequency (GHz)* | *Power Output (watts)* | *Efficiency (%)* | *Mode of Operation* | *Diode Material* | *Type of Junction* | *Ref.* |
|---|---|---|---|---|---|---|
| 0.425 | 435 | 22 | Pulsed | Si | diffused epitaxial abrupt mesa | 21 |
| 1.05 | 420 | >30 | Pulsed | Si | diffused epitaxial abrupt mesa | 22 |
| 6 | 0.5 | 8 | CW | Ge | diffused mesa | 23 |
| 8 | 30 | ~4-5 | Pulsed | Si | diffused planar abrupt | 24 |
| 13.3 | 4.7 | ~8 | CW | Si | abrupt | 25 |
| 21 | 8.8 | 4-5 | Pulsed | Si | diffused planar abrupt | 24 |
| 39 | 0.32 | 7 | CW | Si | abrupt | 26 |
| 115 | 0.075 | 0.1 | Pulsed | Si | linearly graded | 27 |

**Table II. Power output and efficiency of transferred electron oscillators ($B$ = bulk-grown material $E$, = epitaxially grown material).**

| *Frequency (GHz)* | *Power Output (watts)* | *Efficiency (%)* | *Type of Crystal* | *Mode of Operation* | *Ref.* |
|---|---|---|---|---|---|
| 1.1 | 360 | 8 | B | Pulsed Gunn | 40 |
| 1.5 | 3000 | 10 | B | Pulsed LSA | 40 |
| 1.9 | 112 | 24.7 | E | Pulsed Gunn | 41 |
| 2.2 | 143 | 18.6 | E | Pulsed Gunn | 42 |
| 7.7 | 0.34 | 5.5 | E | CW Gunn | 43 |
| 8 | 350 | 3 | B | Pulsed LSA | 44 |
| 88 | 0.02 | 2 | E | CW LSA | 45 |

$2x10^4$ watts-$(GHz)^2$ have already been obtained. This $Pf^2$ product is three orders of magnitude greater than the largest $Pf^2$ product that can be obtained from transistor oscillators.

For frequencies in excess of 100 KHz from the carrier, the AM and FM noise of TEO's are significantly lower than those of avalanche oscillators.[29] At 30 MHz from the carrier, the noise output of TEO's is so low that they can be used as local oscillators for low-noise 30-MHz single-ended IF mixers.

Thim and Barber[46] have built circulator-coupled reflection-type negative-resistance amplifiers in the 2-to-10-GHz range using stable wafers, i.e., wafers satisfying inequality 4. They report maximum gains of about 10 dB, minimum noise figures of about 23 dB, and saturated power outputs of about −3 dBm. Similar amplifiers operating in the 20-to-60-GHz range are described in reference 47.

Linear amplification has also been obtained from oscillating wafers, i.e., from wafers that violate inequality 4. This type of amplifier takes advantage of the negative resistance of propagating domains. Thim[45] describes such an amplifier using a TE wafer oscillating at 8 GHz. Using circulator coupling, he obtained at 6 GHz a linear gain of 9 dB with a power output of 60 mw at 1 dB gain compression, and a minimum noise figure of about 18 dB.

## REFERENCES

1. Carley, D. R., et al, "The Overlay Transistor, Part I: New Geometry Boosts Power" *Electronics* (Aug 23, 1965) pp. 71-77.
2. Caulton, M., et al, "Generation of Microwave Power by Parametric Frequency Multiplication in a Single Transistor" *RCA Review* Vol. 26 (Jun 1965) pp. 286-311.

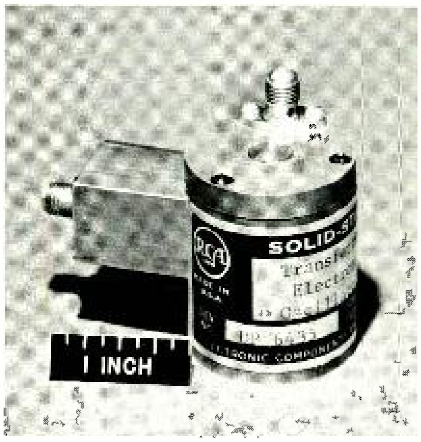


Fig. 9—Photograph of a pulsed 100-watt L-band transferred-electron oscillator.

3. Brooks, R. D. and Gewartowski, J. W., "A Unilateral 6-GHz 2½-Watt Varactor Quadrupler" *Digest of Technical Papers* (1967) pp. 18-19; International Solid-State Circuits Conference, Philadelphia, Pa. (Feb 1967).
4. Parker, D. and Grayzel, A. I., "Varactor Arrays" *1967 NEREM Record* (Nov. 1967) pp. 144-145.
5. Sun, C., "High-Power Varactor Multipliers" *1967 NEREM Record* (Nov 1967) pp. 148-149.
6. Collard, J. R., et al, "High-Power Microwave Varactors and Varactor Multipliers", WESCON/66 (Aug 1966).
7. Irvin, J. C. and Swan, C. B., "A Composite Varactor for Simultaneous High-Power and High-Efficiency Harmonic Generation" *IEEE Trans. on Electron Devices*, Vol. ED-131 (May 1966) pp. 466-471.
8. Application Note 913, Hewlett-Packard, 1501 Page Mill Road, Palo Alto, California.
9. Sterzer, F., "Tunnel-Diode Devices", Chapter in Vol. 2 *Advances in Microwaves* (Academic Press, New York, 1967).
10. Presser, A. and Roswell, A. E., "L-Band GaAs Tunnel-Diode Oscillators with Power Outputs from 10-100mW" *Proc. IRE* Vol. 51 (Jan 1963) pp. 224-225.
11. Hauer, W. B., "Definition and Determination of the Series Inductance of Tunnel Diodes", *IRE Trans. on Electron Devices*, Vol. ED-8 (Nov 1961) pp. 470-475.
12. Young, D. T., et al, "High-Efficiency Millimeter-Wave Tunnel-Diode Oscillators" *Proc. IEEE* Vol. 52 (Oct 1964) pp. 1260-1261.
13. Wright, M. L., "Octave-Range Voltage-Tuned Tunnel-Diode Oscillator" *Proc. IEEE* Vol. 52 (Apr 1964) p. 442.
14. Fortgang, M. M. and Hindin, H. J., "Wide-Band Varactor-Tuned Tunnel-Diode Oscillators" *Proc. IEEE* Vol. 52 (Apr 1964) pp. 419-420.
15. Hornbostel, D. H., "5-10 GHz Yig-tuned tunnel-diode oscillator" *Proc. IEEE* Vol. 53 (Nov 1965) p. 1751.
16. Read, W. T., "A Proposed High-Frequency Negative-Resistance Diode" *Bell System Technical Journal* Vol. 37 (1958) pp. 401-446.
17. Johnston, R. L., et al, "A Silicon-Diode Microwave Oscillator" *Bell System Technical Journal* Vol. 44 (Feb 1965) pp. 369-372.
18. Gilden, M. and Hines, M. E., "Electronic Tuning Effects in the Read Microwave Avalanche Diode" *IEEE Trans. on Electron Devices* Vol. ED-13, (Jan 1966) pp. 169-175.
19. Fisher, S. T., "Small-Signal Impedance of Avalanching Junctions with Unequal Electron and Hole Ionization Rates and Drift Velocities" *IEEE Trans. on Electron Devices* Vol. ED-14 (Jun 1967) pp. 313-322.
20. Misawa, T., "Negative Resistance in p-n Junctions Under Avalanche Breakdown Conditions, Part I and Part II" *IEEE Trans. on Electron Devices* Vol. ED-13 (Jan 1966) pp. 137-151.
21. Prager, H. J., et al, "High-Power High-Efficiency Silicon Avalanche Diodes at Ultra-High Frequencies" *Proc. IEEE* Vol. 55 (Apr 1967) pp. 586-587.
22. Prager, H. J., et al, "Anomalous Avalanche Diodes", 25th Conference on Electron Device Research, McGill University, Montreal, Canada (Jun 21-23, 1967).
23. Dunn, C. N. and Olson, H. M., "Germanium Epitaxial Mesa Diodes for a 6-GHz ½-watt CW Avalanche Oscillator", 1967 IEEE International Electron Devices Meeting, Washington, D. C. (Oct 1967).
24. Gilden, M. and Moroney, W., "High-Power Pulsed Avalanche-Diode Oscillators for Microwave Frequencies" *Proc. IEEE* Vol. 55 (Jul 1967) pp. 1227-1228.
25. Swan, C. B., "Improved Performance of Silicon Avalanche Oscillators Mounted on Diamond Heat Sinks" *Proc. IEEE* Vol. 55 (Sep 1967) pp. 1617-1618.
26. Swan, C. B., et al, "Continuous Oscillation at Millimeter Wavelength with Silicon Avalanche Diodes" *Proc. IEEE* Vol. 55 (Oct 1967) pp. 1747-1748
27. Bowman, L. S. and Burrus, C. A., Jr., "Pulse-Driven Silicon p-n Junction Avalanche Oscillators for the 0.9 to 20 mm Band" *IEEE Trans. on Electron Devices* Vol. ED-14 (Aug 1967) pp. 411-418.
28. Josenhans, J., "Noise Spectra of Read Diode and Gunn Oscillators" *Proc. IEEE* Vol. 54 (Oct 1966) pp. 1478-1479.
29. Rulison, R. L., et al, "Improved Performance of IMPATT Diodes Fabricated from Ge" *Proc. IEEE* Vol. 55 (Feb 1967) pp. 223-224.
30. McIntyre, R. J., "Multiplication Noise in Uniform Avalanche Diodes" *IEEE Trans. on Electron Devices* Vol. ED-13 (Jan 1966) pp. 164-168.
31. Hines, M. E., "Noise Theory for the Read-Type Avalanche Diode" *IEEE Trans. on Electron Devices* Vol. ED-13 (Jan 1966) pp. 158-163.
32. Scherer, E. F., "Large-Signal Amplification with Avalanche Devices" *Proc. IEEE* Vol. 55 (Mar 1967) pp. 464-465.
33. Ridley, B. K. and Watkins, T. B., "The Possibility of Negative-Resistance Effects in Semiconductors" *Proc. Phys. Soc.* Vol. 78 (Aug 1961) pp. 293-304.
34. Hilsum, C., "Transferred-Electron Amplifiers and Oscillators" *Proc. IRE* Vol. 50 (Feb 1962) pp. 185-189.
35. Gunn, J. B., "Microwave Oscillations of Current in III-V Semiconductors" *Solid-State Communications* Vol. 1 (Sep 1963) pp. 88-91.
36. McCumber, D. E. and Chynoweth, A. G., "Theory of Negative-Conductance Amplification and of Gunn Instabilities in 'Two-Valley' Semiconductors" IEEE Trans. on *Electron Devices* Vol. ED-13 (Jan 1966) pp. 4-21.
37. Copeland, J. A., "A New Mode of Operation for Bulk Negative-Resistance Oscillators" *Proc. IEEE* Vol. 54 (Oct 1966) pp. 1479-1480.
38. Johnson, E. O., "Physical Limitations on Frequency and Power Parameters of Transistors" RCA Review Vol. 26 (Jun 1965) pp. 163-177.
39. DeLoach, B. C., Jr., "Recent Advances in Solid State Microwave Generators" Chapter in Vol. 2 *Advances in Microwaves* Academic Press, New York 1967.
40. Dow, D. G., "Two Gunn Holdups Slow Microwave Oscillators" *Electronics* (Nov 13, 1967) pp. 129-130.
41. Narayan, S. Y., Berson, B. E. and Reynolds, J., "Pulsed Epitaxial GaAs Gunn Oscillators" 1967 *NEREM Record* (Nov 1967) pp. 24-25.
42. Berson, B E. and Narayan, S. Y., "High Peak Power Epitaxial GaAs Oscillators" *IEEE Trans. on Electron Devices* Vol. ED-14 (Sep 1967) pp. 610-611.
43. Murakami, H., et al, "Electrical Performance of GaAs Epitaxial Gunn-Effect Oscillators" *IEEE Trans. on Electron Devices* Vol. ED-14 (Sep 1967) pp. 611-612.
44. Kennedy, W. K., Jr., et al, "LSA Operation of Large-Volume Bulk GaAs Samples" *IEEE Trans. on Electron Devices* Vol. ED-14 (Sep 1967) pp. 500-504.
45. Copeland, J. A., "CW Operation of LSA Oscillator Diodes" *Bell System Technical Journal* (Jan 1967) pp. 284-287.
46. Thim, H. W. and Barber, M. R., "Microwave Amplification in a GaAs Bulk Semiconductor" *IEEE Trans. on Electron Devices* Vol. ED-13 (Jan 1966) pp. 110-114.
47. Thim, H. W. and Lehner, H. H., "Linear Millimeter Wave Amplification with GaAs Wafers" *Proc. IEEE* Vol. 55 (May 1967) pp. 718-719.
48. Thim, H. W., "Linear Microwave Amplification with Gunn Oscillators", *IEEE Trans. on Electron Devices* Vol. ED-14 (Sep 1967) pp. 517-522.

# The TIROS decade

## A. Schnapf

**The Space Age . . . started by the launch of Sputnik I . . . is now 10 years old. Men have orbited earth, and spacecraft have landed on the moon and have travelled to distant planets. Communications satellites make possible live television among continents and, with the aid of meteorological satellites, giant strides are being made in our understanding of weather. RCA has played a key role in many of these areas, but none would appear more important than our role in the Meteorological Satellite Program, TIROS (Television and Infra-Red Observation Satellite). Throughout the space decade, RCA has been involved in this program, which provided the first important peaceful and beneficial use of outer space by nations of the world with the implementation of the TIROS Operational System (TOS) in 1966.**

**This paper reviews the performance of the ten TIROS and seven ESSA satellites (part of the TIROS Operational System) orbited during this decade and gives some insight into the improved TOS satellites that will be launched in the near future.**

RCA's PARTICIPATION in the Tiros Meteorological satellite program started in 1958, with the first Tiros satellite being launched on April 1, 1960. Since then, a total of 10 Tiros and 7 Essa satellites have been successfully orbited (Fig. 1), providing nearly continuous space observations of this planet's weather phenomena for more than eight years.

Originally, Tiros was an experimental system. However, in February 1966, the Tiros Operational System was implemented with the successful orbiting of the Essa 1 and 2, expanding the basis Tiros system and providing the world's first operational satellite system capable of observing the earth's cloud cover on a daily routine basis. Now, second-generation Tos satellites are under devolopment; these will further enhance the potential of the Tiros system for global operational weather observation and forecasting.

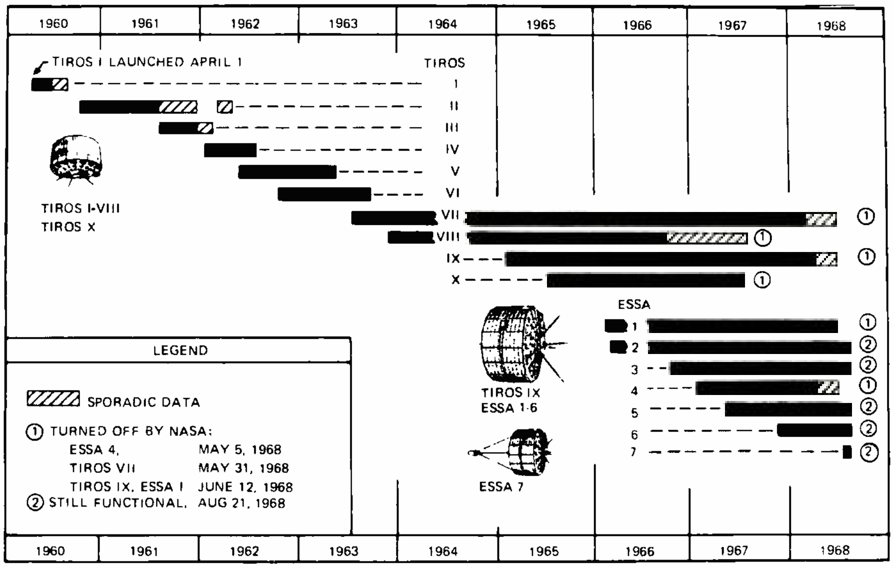


Fig. 1—TIROS/TOS Performance Summary.

### Meteorological Benefits

Through the Tiros research and development programs, a reliable and useful meteorological tool has been developed, namely, the Tiros Operational System. This system has been in operation for more than two years, providing routinely and without interruption, daily global weather observation to the National Environmental Satellite Center of the United States and local APT weather photographs to more than 300 stations located throughout the world.

The important product of Tos is the observation of weather conditions in all parts of the world and the provision of this weather data rapidly and in useful form. Major weather systems. such as fronts, storm centers, tropical and extratropical flows, hurricanes, typhoons, and distinctive cloud patterns are viewed by the TV cameras in the Essa satellites. This data is then relayed to earth, where it can be applied to other, previously obtained, data for analysis and forecasting. As more satellite-gathered data has been accumulated and analyzed, the use of the photographic data has been enhanced through user experience. Now, meteorologists can determine the direction of circulation, deduce wind speeds in large vortices, trace the paths of jet streams, detect sea ice and snow cover, and readily identify squall lines and cloud bands associated with frontal systems.

Several of the more than 1,030,000 television pictures returned by the Tiros and Essa satellites are shown in Figs. 2, 3 and 4. Although reproduction processes result in the loss of detail, the high quality of the Tiros and Tos photographs is evident in these illustrations. The photographs were taken with TV cameras equipped with wide-angle lenses.

**Abraham Schnapf**
TIROS/TOS Program Manager
Astro-Electronics Division
Princeton, N.J.

received the BSME degree from the City College of New York in 1948 and his MSME degree from Drexel Institute of Technology in 1953. Mr. Schnapf, has been Manager of the TIROS and TOS programs at RCA Astro-Electronics Division since the first TIROS program was started by the National Aeronautics and Space Administration in 1960. He has had the responsibilty for the management of design and fabrication of ten TIROS and seven ESSA weather satellites; all were successful. In 1967, his reponsibilities were increased to include the TIROS M/ITOS program. He is a professional engineer in the state of New Jersey and an Associate Fellow of the AIAA. He is listed in American Men of Science and he is a member of the New York Academy of Science. He holds a number of patents and has presented and published numerous papers.

---

Reprint RE-14-3-23
Final manuscript received July 8, 1968.

Fig. 2—First Complete view of the World's weather. This photomosaic is composed of 450 photographs taken by TIROS IX during its 12 orbits on February 13, 1965. (Courtesy of U.S. Environmental Science Services Administration.)

## Evolution of the Global Weather Satellite System

TIROS I, the world's first meteorological satellite, was launched April 1, 1960, with the primary objective of demonstrating the feasibility of observing the earth's cloud cover by means of slow-scan television cameras in an earth-orbiting, spin-stabilized satellite. This satellite included both a wide-angle and a narrow-angle TV camera, and was placed in a circular orbit at an altitude of 400 miles, with the orbit inclined 48 degrees to the equator.

The first, historic television pictures from space were received on the very first orbit of TIROS I, immediately and clearly demonstrating the feasibility of the system. Fig. 5 shows the first picture taken on the first orbit, a wide-angle picture showing for the first time the earth's cloud cover from space, the northeastern part of the United States and part of Canada. With the reception of pictures from TIROS I, a new and powerful tool for the meteorological community became a reality.

TIROS II was orbited on November 23, 1960, to demonstrate, in addition to the wide- and narrow-angle television cameras, an experimental, 5-channel, scanning IR radiometer and a 2-channel non-scanning IR device. Both of these devices were developed by the NASA Goddard Space Flight Center. They measured the thermal energy of both the earth's surface and atmosphere in order to provide data on the planet's heat balance and add a new dimension for the understanding of weather. A magnetic torquing coil was added to TIROS II (and all TIROS satellites thereafter) so that a controlled magnetic field about the satellite would interact with the earth's field in space and, hence, provide control of the satellite's attitude. In this way, camera pointing, thermal control, and the use of available solar power were enhanced.

TIROS III, IV, V, VI and VII were launched between July 1961 and June 1963 to provide continuous observation of the earth's cloud cover for limited operational use. With each of these satellites, particular emphasis was given to providing early warning of severe tropical storms, hurricanes, and typhoons. In addition to cloud-cover observation, the satellites were employed experimentally to detect sea ice and snow cover, and to support the Indian Ocean expedition, Ice Reconnaissance experiments, and other research programs. These spacecraft each contained two, slow-scan, ½-inch vidicon television camera systems as the primary sensors.

TIROS VIII, launched in December 1963, included both a ½-inch TIROS camera and a 1-inch Automatic Picture Transmission (APT) camera. This marked the first in-space use of the APT system that had been developed for Nimbus I. The APT camera utilizes a very slow-scan vidicon, as compared to the ½-inch TV camera. The latter requires 2 seconds to scan its 500-TV-line image; the APT camera requires 200 seconds for readout of its 800-TV-line image. By virtue of the 2-kHz bandwidth of the APT system, TIROS VIII was able to transmit direct, real-time, television pictures to a series of 45, relatively inexpensive APT ground stations located around the world.

TIROS IX, the first "wheel-mode" satellite, was launched in January 1965 with the objective of expanding the capability of the TIROS satellites to provide complete global weather observation on a daily basis. This represented an increase of four times the daily observation provided by the predecessor TIROS satellites. With its new design, TIROS IX differed from its predecessors in many aspects, and was the forerunner of the satellites now used in the TIROS Operational System.

A primary difference was that in TIROS I through VIII, the two TV cameras were mounted on the baseplate of the satellite with the optical axes parallel to the inertially stabilized spin axis. Hence, the camera axes were parallel to the orbit plane and viewed the earth for about 25 percent of each orbit. In the TIROS IX configuration, the TV cameras were mounted diametrically opposite one another and looked out through the sides rather than through the baseplate of the satellite. The satellite was injected into orbit with the spin axis in the orbital plane; however, the spin axis was then maneuvered by an improved magnetic-torquing system to an attitude normal to the orbit plane. Thus, the spinning satellite "rolled" along its orbital path and the field of view of each camera passed through the local vertical once during each spin or "roll." At the proper interval in the picture-taking sequence, the camera shutter was triggered to take a photo of the local scene when the camera was looking down at the earth. Hence, throughout the sunlit portion of the orbit, the earth below the satellite could be observed by means of a sequence of overlapping photos. By placing the wheel satellite in a near-polar, sun-synchronous orbit, the entire earth could be observed on a daily basis.

TIROS X, the last of the research and development series of standard TIROS satellites, was launched in July 1965 to provide hurricane and tropical storm observations.

ESSA 1 was launched on February 3, 1966 into a 400-n. mile, near-polar, sun-synchronous orbit to become the first operational satellite providing global observations on a daily basis. This satellite (like its predecessor, Tiros IX) utilized two ½-inch vidicon camera systems, wherein a pair of pictures (one from each camera) produced a picture swath 2200 miles wide and 800 miles long along the orbit track. Contiguous coverage was provided by programming the cameras to take "picture pairs" every two minutes along the sun-illuminated portion of the orbital track. With the 14.5 orbits completed each day, a total of 450 TV photos were available for transmission to the Tiros ground stations at Fairbanks, Alaska, or the station at Wallops Island, Virginia. From these stations, the satellite television and telemetry data was retransmitted to the Environmental Science Services Administration's (Essa's) National Environmental Satellite Center (Nesc), for processing, analysis, and retransmission to major weather centers in the United States, as well as abroad.

#### The TOS Satellites

With Essa 1 on station and providing operational global observation for readout in the United States, the second Essa satellite, Essa 2, was successfully placed in orbit on February 28, 1966. Essa 2 was actually the first of the Tos-design spacecraft. It was launched into a 750-nautical-mile, sun-synchronous orbit to complement Essa 1 in the Tiros Operational System by providing direct, real-time readout of APT pictures to the APT ground stations located throughout the world.

This pair of operating satellites fulfilled the commitment made by the United States to provide an operational meteorological satellite system in the first quarter of 1966.

Later in 1966, on October 2, the Essa 3 satellite was launched. It replaced Essa 1, in which a TV sensor had ceased operating. Essa 3 was launched into a 750-nautical-mile, sun-synchronous orbit. In this satellite, the original Tiros wide-angle camera system was replaced with a modified Nimbus camera, the Advanced Vidicon Camera System (AVCS), which provided higher resolution and a larger picture area than the ½-inch Tiros cameras. To ensure uninterrupted daily global photocoverage, Essa 4 (the second of the Tos APT satellites) was placed in orbit on January 26, 1967. This satellite replaced Essa 2, which was providing limited global coverage due to a slow orbital drift that had taken place since launch. Essa 5, the second Tos AVCS satellite, was placed in orbit on April 20, 1967 to provide back-up for Essa 3. Essa 6, carrying APT cameras, was successfully launched on November 10, 1967, to replace Essa 4, whose one remaining operational camera did not provide satisfactory operational data. Essa 7 was launched on August 16, 1968 to ensure an uninterrupted supply of AVCS photographs, and extended the perfect record of the Tiros/Tos programs to 17 successful spacecraft in 17 launches.

### Description of the TIROS Operational System

The Tiros Operational System is sponsored by the U.S. Department of Commerce, and is managed and operated by the Environmental Science Services Administration's National Environmental Satellite Center, under the technical direction of the National Aeronautics and Space Administration's Goddard Space Flight Center (Nasa/Gsfc). To meet the full operational objectives of the system, it is required that two Tos meteorological satellites be in orbit at all times; one carrying the APT subsystem for direct local readout to APT stations throughout the world, and the second carrying the AVCS, which is capable of storing global video data for readout to associated ground stations that immediately relay the data to the Nesc for processing and analysis.

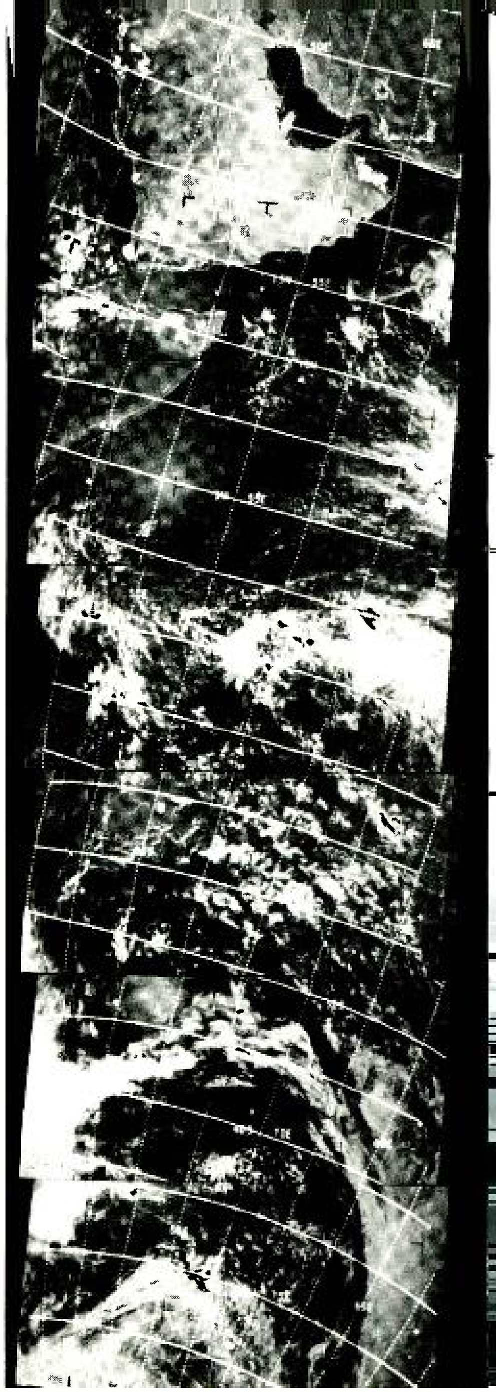


Fig. 3—Portion of an orbital sequence of ESSA 3 television photographs showing Saudi Arabia, Somaliland, and Indian Ocean. (Gridding added by digital computer at the U.S. National Environmental Satellite Center).

#### The TOS Ground Station Network

The two primary Tos ground stations are the Command and Data Acquisition (CDA) stations located near Fairbanks, Alaska and Wallops Island, Virginia. The locations of these stations permit direct communications with a Tos satellite on every orbit except one, on a daily basis. The two stations are similar, and each is equipped for either manual or automatic transmission of commands to the satellite by means of a low-gain antenna. A separate, 85-foot, steerable, parabolic antenna is used to receive the satellite-transmitted telemetry data and the AVCS video transmissions; APT video need not be received by the CDA stations.

All telemetry data transmitted from the satellites is recorded on 7-channel tape recorders at the CDA station; when AVCS video data is received, a second tape recorder is employed. The telemetry data from the satellite's 137.77-MHz beacon also is recorded on paper-chart recorders and, in addition, is transmitted in real time to the Tos Operations Center (TOC) at the National Environmental Satellite Center, Suitland, Maryland, and to the TOS Evaluation Center at the Goddard Space Flight Center, Greenbelt, Maryland. The video data is played back at an 8-to-1 reduced rate over the 48-kHz broadband transmission line to TOC.

The TOS satellite ephemeris data is provided by NASA's Space Tracking and Data Acquisition Network (STADAN). This data is used for providing the CDA stations and APT stations with orbit tracking data; it is also used by TOC to permit picture gridding by computer and to facilitate programming for future events.

Under ESSA control, programming instructions and pertinent ephemeris and tracking data (derived from NASA STADAN station data) are forwarded from TOC to the primary CDA stations in advance of that station's contact with the satellite. TOC also monitors the performance of the satellite throughout its operational life, in order to budget the power supply and maintain the spin-axis attitude, the spin period, and the operating temperatures of the satellite within the optimum design limits.

### Physical Description of the Satellite

The TOS APT and TOS AVCS satellites are similar in their general external physical characteristics. The satellite structure is similar to that of previous TIROS satellites, consisting of an 18-sided right polyhedron, 22.5 inches high and 42 inches in diameter. A reinforced baseplate carries most of the subsystem components, and the cover assembly ("hat") provides mounting area for the solar cells on its outer top and side surfaces. The dynamics-control devices provided on the satellite consist of attitude- and spin-control magnetic coils, and mechanical and liquid precession dampers. These devices are mounted inside the hat structure. Openings in the hat provide viewing ports for various sensors mounted on the baseplate. A crossed-dipole antenna projects from the underside of the baseplate and a monopole, or whip, antenna extends vertically from the center of the hat. On ESSA 7, this antenna was modified to combine, in one structure, the receiving antenna and a biconical S-band transmitting antenna, which is being given an "in-space" test prior to its use on the improved TOS system described later. In addition, the AVCS satellite is equipped with terrestrial-radiation sensors developed by the University of Wisconsin for measurement of the earth's heat balance.

The APT satellite, with redundant APT cameras, weighs 285 pounds; the AVCS satellite, with redundant AVCS cameras and video recorders, and the University of Wisconsin radiometers and associated equipment, weighs 325 pounds (345 pounds with the new S-Band transmitting system). Except for data recorders and an IR subsystem, the functional diagram for the TOS AVCS spacecraft is essentially the same as that for the APT spacecraft. Fig. 6 permits a side-by-side comparison of the two types of TOS spacecraft.

### Satellite Dynamics Control

The identical dynamics-control subsystems for the two types of TOS satellites include attitude- and spin-control coils, and nutation dampers. The primary technique used for controlling the spinning satellite's attitude is magnetic torquing. Since the mission requires a sun-synchronous orbit, in which the orbit plane precesses in synchronism with the earth's motion around the sun, the satellite's inertially stabilized spin axis must be precessed at the same rate in order to maintain the "wheel" attitude (in which the spin axis is normal to the orbit plane). To achieve this continuous slow drift in the proper direction, a Magnetic Bias Coil (MBC) is used in conjunction with the Quarter Orbit Magnetic Attitude Control (QOMAC) coil. The MBC is similar to the magnetic attitude-control devices employed on TIROS II through VIII; the QOMAC coil is similar to that used first on TIROS IX.

The current-carrying MBC generates an electromagnetic field of selectable strength, whose dipole moment is colinear with the satellite's spin axis. The device consists of a coil of wire and a stepping switch to regulate the amount and direction of the current in the coil.

The QOMAC coil, operating in a low-torque mode, provides the fine control over spin axis motion required to keep the satellite in mission attitude. In its high-torque mode of operation, this coil provides the rapid attitude change required to initially achieve the wheel attitude. This attitude change, of approximately 90 degrees, is required because upon injection into orbit the satellite's spin axis lies in or near the orbit plane, rather than normal to it. In the high-torque mode, the spin axis processes approximately 10 degrees per orbit, and in the low-torque mode, approximately 2 degrees per orbit.

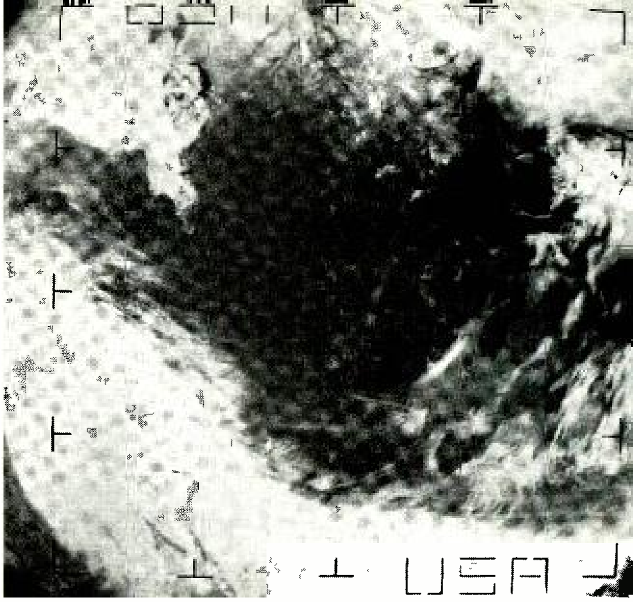


Fig. 4—The one-millionth picture taken by the TIROS series; picture shows the Hudson Bay area and Northern New England on May 27, 1968.

The Magnetic Spin Control (MASC) coil is used to maintain the satellite spin rate at the optimum level for operation of the TV cameras. Current through the MASC coil is computed at ½-spin intervals to provide a motor effect that can be used to increase or decrease the spin rate. Normally the MASC coil is activated only near the earth's poles, where its operation is most efficient.

## The Second-Generation TIROS Operational System

In 1966, design studies were initiated by NASA/GSFC, in consultation with ESSA, and are being implemented by RCA under the TIROS M and Improved

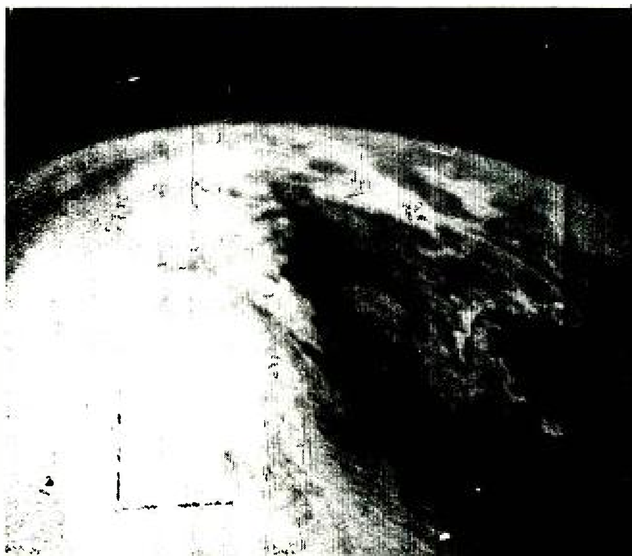


Fig. 5—First television picture of earth's weather received from a satellite; weather over Eastern U.S. and Canada on April 1, 1960.

TOS (ITOS) program. This program utilizes, whenever possible, the proven technology and satellite hardware developed by the TIROS and TOS satellites, and sensors developed for other NASA programs.

Since the TOS system is an operational system, the changes planned for TIROS M/ITOS were chosen to reflect an orderly transition. The phase-in process between TOS and TIROS M/ITOS reflects the requirements of common usage of existing ground-station and data-processing facilities to accommodate the simultaneous use of TOS and TIROS M/ITOS during the initial, developmental flights of the new series of satellites.

An underlying requirement for the development of operational systems is cost effectiveness. In keeping with this requirement, the second-generation satellite has been configured to the capabilities of the Delta launch vehicle, a reliable, low-cost launch vehicle. In addition, the system design calls for placing all sensors on a single satellite, rather than having separate APT and AVCS satellites. Thus, only one satellite and one launch, as opposed to two satellites and two launches in the present operational system, will be required to meet the mission requirements for local and global data readout.

Like TOS, TIROS M/ITOS offers the capability for providing daytime, direct, real-time APT readout to stations throughout the world, and readout of stored AVCS daytime observations to NESC. However, TIROS M/ITOS will also be configured with a scanning radiometer (SR) subsystem capable of daytime and nighttime cloud-cover observations, for direct readout of local data and stored readout of global data. Use of the SR subsystem in conjunction with the AVCS will enable photocoverage of the entire earth every 12 hours.

All of the primary sensors will be supplied in redundant sets on TIROS M/ITOS to provide the necessary backup in the event of failure or degradation of a device.

In addition to the primary sensors, the TIROS M/ITOS satellite will be configured with the following secondary sensors:

- A Flat Plate Radiometer, developed by the University of Wisconsin, to provide global measurements of the earth's thermal energy.
- A Solar Proton Monitor, developed by the Applied Physics Laboratory of Johns Hopkins University, to measure the solar proton energy at the satellite's orbital altitude.

The capability for using all of these sensors on a single spacecraft results from the use of an advanced stabilization technique on the TIROS M/ITOS satellite. This type of stabilization allows the main body of the spacecraft to rotate at only one revolution per orbit. Thus, the sensor-side of the satellite always faces toward earth. This is in contrast to the basic TOS satellites in which the entire spacecraft spins at either 10.5 or 9.2 r/min, depending on the basic timing requirements of the sensors (APT or AVCS) employed.

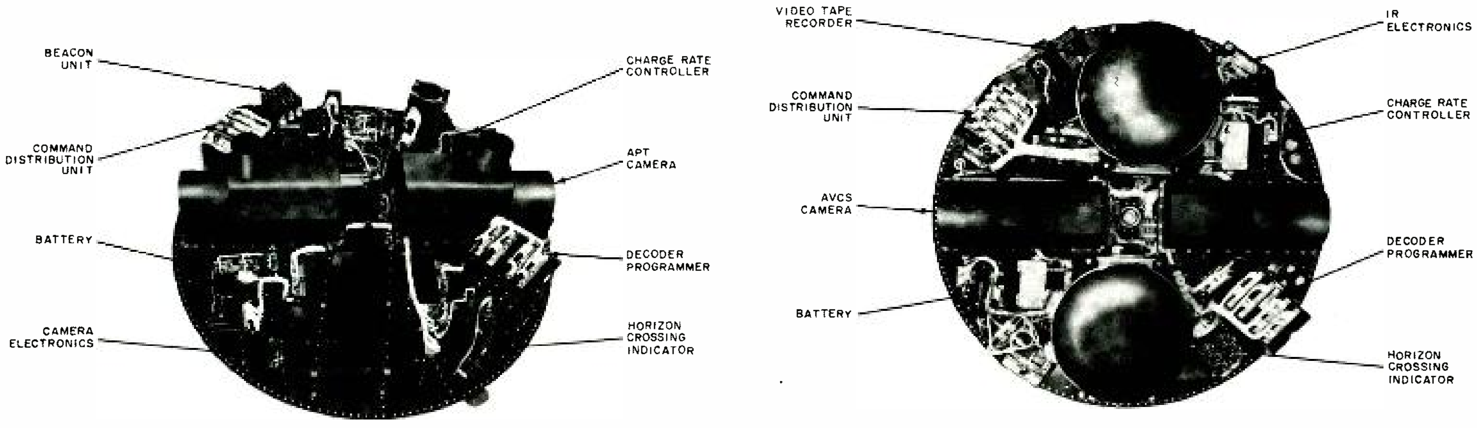
### System Design

The general physical configuration of the TIROS M/ITOS satellite is shown in Fig. 7. The satellite is a rectangular, box-shaped structure, with each of the sides measuring approximately 48 inches in length and 40 inches in width. On the bottom of the structure, a cylindrical transition section attaches to the 37-inch-diameter ring section of the second stage of the launch vehicle.

The key to the control of the TIROS M/ITOS satellite is the momentum flywheel system. The spinning flywheel, coupled through bearings to a despun platform, is used to maintain effective stabilization. The flywheel axis is colinear with the pitch axis and contains a scanning mirror that will enable fixed IR bolometers to detect sky-earth and earth-sky transitions in their field of view. The satellite's pitch-control system will regulate the speed of the motor-driven flywheel based upon position and rate signals derived from these IR bolometers.

TOS-type QOMAC coils will be used to correct roll and yaw errors, as well as to perform the initial orientation maneuver. The MBC coils will be utilized to correct for the residual magnetic dipole and provide the basic one degree-per-day precession rate to track the orbit regression of a sun-synchronous orbit. A magnetic spin-control coil will provide momentum control about the pitch axis, and liquid dampers will reduce satellite nutation.

As shown in Fig. 7, the three solar panels will be mounted along the main body of the satellite with their hinge lines at the top of the structure; during launch these panels will be held flat against the sides of the spacecraft in a stowed position. Once the satellite achieves mission mode, the panels will be deployed by actuators. In the deployed position, they will be normal to the sides and parallel to the top of the structure and will be approximately in the orbit plane.



TOS APT SPACECRAFT

TOS AVCS SPACECRAFT

Fig. 6—Comparison of the TOS APT and TOS AVCS baseplate layouts.

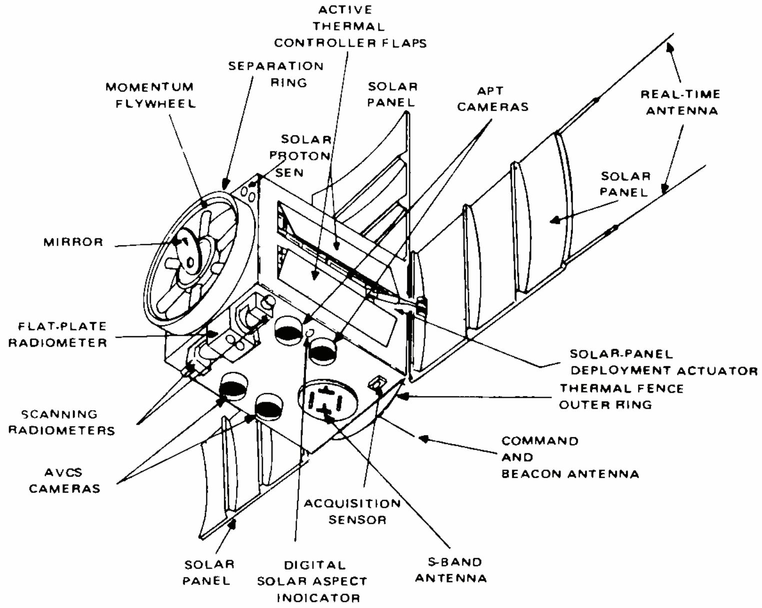


Fig. 7—The TIROS M/ITOS satellite in mission mode.

Thermal control will be achieved by the application of passive and active thermal-control techniques. Most of the satellite is covered by multilayer insulation blankets, except for the primary sensor openings and the areas used for the active control device. Passive thermal control will be provided by a variable absorptivity device, designated the "thermal fence," mounted on the top of the satellite. As the sun angle varies with respect to the two vertical walls of the fence, the amount of solar absorption will also vary. The heights of the fence walls were selected to provide maximum heat input at a sun angle of approximately 60 degrees. This passive-control device will, by itself, maintain the satellite temperatures within design limits. However, an active-thermal-control system will also be utilized to augment the passive design, providing a narrower range of temperature variations throughout the satellite's mission life. This active device consists of thermal flaps that can be opened or closed to vary the effective emissivity of the spacecraft.

All of the satellite's electronic equipment will be mounted on three load carrying sections within the structure or main body of the satellite. The equipment will be arranged on two side wall members and on the base section. Sufficient volume will be provided for additional equipment, and the simple structural design will permit a variety of possible layouts. The two side walls will support the basic camera and recording subsystems, whereas the base structure will contain command, control, power supply, and communications equipment. The scanning radiometers will be mounted on the base section, at the lower edge of the earth-oriented side.

### System Operation

The TIROS M orbit-injection sequence will be initiated immediately after separation from the upper stage of the launch vehicle. The events from injection to the achievement of the mission-mode attitude, under nominal conditions, are expected to take up to 24 hours. Upon separation from the upper stage, the satellite will be spinning at approximately 3.5 r/min, with the spin axis approximately normal to the orbit plane. This spin rate will provide the required momentum value for controlling satellite attitude. The momentum wheel in the pitch control system (stabilite) will then be spun-up to 75 r/min. At this point, the satellite will be completely stabilized about the spin axis of the flywheel. Then, the magnetic torquing coils will be employed to adjust the momentum vector (spin axis) from the initial injection attitude to mission mode, in which the spin axis will be normal to the orbit plane.

The pitch-axis control system will be commanded to achieve local orientation of the sensor platform by transferring most of the total momentum of the satellite to the flywheel. The spin rate of the flywheel will increase to 150 r/min, while the main body of the satellite will decrease to one revolution per orbit in order to keep the sensor side of the satellite facing earth throughout the orbit.

When the satellite reaches mission mode attitude and desired spin rate, the solar panels will be deployed. In the deployed position, the panels will be in the orbit plane and fully illuminated by the sun. Under nominal orbital conditions, the sun vector will be at 45 degrees to the spin axis; however, complete mission operations can be realized with a sun angle within the range of 30 to 60 degrees with respect to the spin axis.

Each APT and AVCS camera on TIROS M/ITOS satellites will scan an area similar to that covered by the TOS satellite cameras, and each camera will be programmed for an 11-picture sequence on each orbit. The SR sensor will provide continuous coverage (i.e., will scan continuously) whenever it is turned on. Although this sensor will normally be used for nighttime cloud cover observations, it will also be used for daytime observation when desired.

The secondary sensor data (from the flat plate radiometer and the solar proton monitor) will be stored in serial, digital form on two tracks of an incremental recorder. A third track in the recorder will record timing data.

The TIROS M/ITOS communications link will utilize the same beacon telemetry, APT transmission, and command links employed on TOS; however, the stored video data for the AVCS cameras, the SR sensors, and the secondary sensors will be transmitted at an S-band frequency, nominally 1.7 GHz.

The capacity of the TIROS M/ITOS command and control subsystem has been increased over that of the TOS subsystem because of the greater number of commands required for the multiple-sensor configuration used on TIROS M/ITOS and to provide for future growth.

## Summary

The TIROS Operational System and the second-generation ITOS satellites represent the culmination of an orderly and progressive research and development program initiated over 10 years ago. The routine practical application of meteorological satellites has been one of the most important products of this first decade in space, and today many countries of this planet are deriving direct, beneficial use from the TIROS Operational System.

# An integral-cavity L-band TRF amplifier—a new concept

**K. W. Uhler**

**Existing IFF transponders use transistorized superheterodyne receivers with all the attendant problems of such receivers: spurious radiation, image rejection, crystal burn-out, limited dynamic range, and difficulty in maintaining the required passband over the temperature range. As a solution to these problems the Naval Research Laboratories used a new TRF approach to develop an advanced transponder design which is both compact and lightweight.**

THERE are many anecdotes told of instances when the sudden failure of IFF (identification friend or foe) equipment at a crucial moment led to an unorthodox and often humorous, method of identification. At WW-II speeds, a faulty IFF system was easily circumvented because there was time. However, the supersonic attack velocities of today's sophisticated weapons systems have reduced the decision time from minutes to milliseconds, and high kill ratios have thrown a vital emphasis on instantaneous identification.

There has been considerable effort, particularly over the last several years, to improve the existing equipment in an attempt to meet current and anticipated requirements. The most successful approach to date has been a complete departure from the approach used in present IFF transponders.

The success of this approach depended on the development of the FD2201, (Fig. 1) an integral-cavity 1030-MHz TRF amplifier. The basic prototype FD2200 and the subsequent variant, FD2200V1 were developed by the Special Products Engineering Group of Receiving Tube Engineering.

Final manuscript received May 16, 1968.

**K. W. Uhler, Ldr.**

Special Product Developments
Receiving Tube Engineering
Electronic Components, Somerville, N.J.

received the BS in Science from Pennsylvania State University in 1947, continuing with graduate work through 1948. In 1948 he joined the RCA Receiving Tube Engineering activity at Harrison. From 1948 to 1951 he worked as an engineer on receiving tube specifications for computer applications, environmental testing and the development of new test methods. In 1952 he was promoted to Engineering Leader. His work involved the early development of test methods and equipment for semiconductors, low-crossmodulation circuits, beam deflection tube SSB detection, and industrial and military nuvistor applications. He had the responsibility for the basic concept, product design and application of integral-cavity TRF amplifiers for transponder use. Mr. Uhler is a Senior Member of the IEEE and has served on numerous industry and service committees including the IRE/AIEE Subcommittee on Definitions of Semiconductor Devices (1952-1956) and committees on methods of testing (1954-1956).

## AMPLIFIER DESIGN

The basic electrical characteristics for a universal transponder satisfying both military and commercial needs were set forth by the AIMS committee:

| | |
|---|---|
| Center frequency | 1030 MHz |
| —6-dB bandwidth | 7 MHz (min) |
| —40-dB bandwidth | 25 MHz (max) |
| Attenuation at 1005 MHz | 60 dB (min) |
| Attenuation at 1055 MHz | 60 dB (min) |

[AIMS is an acronym that was contrived as follows: **A** for Air Traffic Control Radar Beacon System; **I** for IFF; **M** for Mark XII Secure Identification System; and **S** for System.]

The environmental requirements included operation over the temperature range of —54°C to +95°C as well as a 15-g vibration rating, and a 50-g shock rating. The system requires a fairly flat response curve to allow for transmitter drift. Because the frequency-response requirements are essentially Gaussian, the —3-dB bandwidth is about 5.5 MHz when the —6 dB bandwidth is between 7.5 and 8.0 MHz. The gain should be in the order of 45dB to override the tangential noise from the video detector. Because crystal burnout was one of the more serious field problems with existing equipment, the amplifier should provide limiting well below the video-detector rating.

During the early stages of development it became apparent that a two-minute warmup from the cold storage temperature of —62°C did not bring the amplifier up to —54°C. Later equipment measurements indicated that long periods of operation in the RF compartment raised the amplifier shell temperature to +125°C. The practical operating range is then —62°C to +125°C, a difference of 187°C. Invar, with a coefficient of expansion of 1.1x$10^{-6}$/in/in/°C, was the only available material that would provide dimensional stability over this wide temperature range. In strip form, Invar lent itself to forming and embossing for mechanically formed cavities, but was somewhat difficult to plate. A technique was developed to plate high-density silver which yielded unloaded cavity Q's in the order of 1800 to 2200. This technique was particularly useful in designing a dimensionally stable low-loss preselector.

The 8058 nuvistor has a coefficient of expansion of 9.5x$10^{-6}$ in/°C. In mating the 8058 to the cavity center line, a sliding contact on the top cap was used to provide for linear differential expansion. Radial expansion was accommodated by means of a flanged seam with a "C"-shaped stainless steel slide to maintain approximately constant compression on the internal parts. The cutaway view of the 8058 (Fig. 2) shows how the grid is internally connected to the nuvistor shell. A flange attached to the nuvistor shell is used to integrate the 8058 into the cavity.

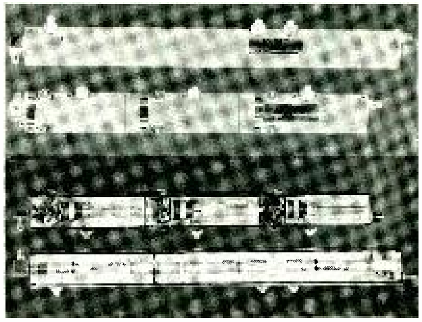


Fig. 1—The components of the FD2201 transponder.

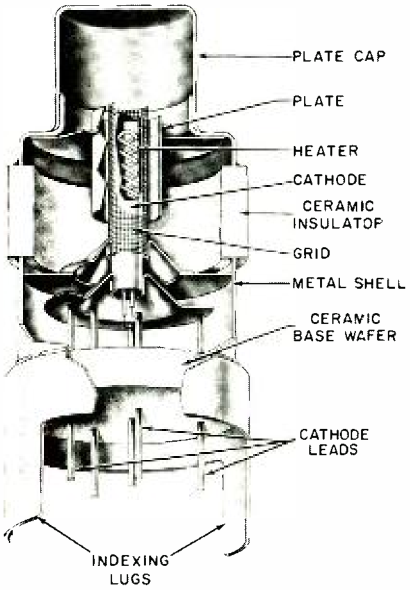


Fig. 2—Cutaway view of the 8058.

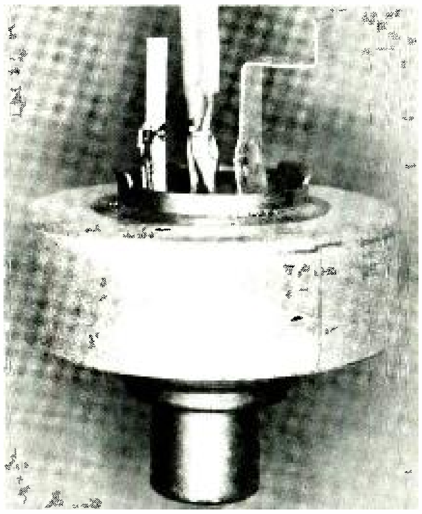


Fig. 3—Subassembly of the 8058 showing impedance-matching arrangement.

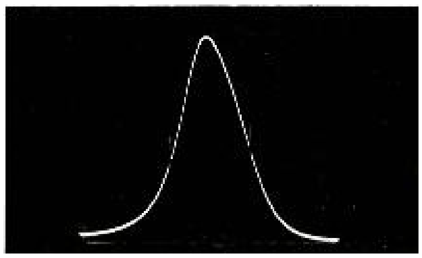


Fig. 4—Response curve of FD3101 amplifier.

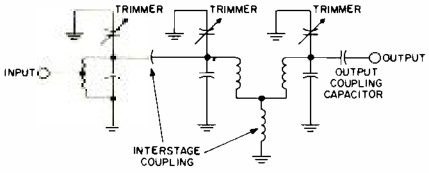


Fig. 5—Circuit diagram of the FD3100 preselector.

The flange is isolated from DC ground by a 3-mil-thick mica sheet clamped between the flange and shell. Grid-to-ground reactance is reduced by silvering the mica on both sides; as a result, the capacitance is raised from about 300 pF to approximately 1000 pF.

In a grounded-grid configuration, the intrinsic gain depends on the cavity loading. To avoid changes in coupling due to temperature variations and to keep the design coaxial, an inductive loop is used for output coupling. The measurements were simplified by the use of a 50-ohm reference load, but this then required matching the input impedance of the nuvistor to 50 ohms. Nominally in the order of 150-250 ohms, the input impedance of the nuvistor was matched in a unique manner by utilizing the three cathode leads. The cut-away view of the 8058 (Fig. 2) shows the cathode leads and their connection to the internal flange supporting the cathode itself. By use of one lead as a part of a shunt-matching inductance, the matching is accomplished close to the cathode and on the active side of the input lead inductance and the shunt capacitance across the base dielectric (ceramic base wafer). A disc feed-through capacitor provides 470 pF to ground; the shape and length of the third lead connection to the feed-through capacitor controls the total shunt inductance and thereby provides a controlled input match. To minimize series inductance, the RF input is connected to the two cathode leads by a flat ribbon (Fig. 3).

In the final design of the amplifier, the internal feedback capacitance of the 8058 affected the bandpass characteristics and caused an unbalanced response. This problem was solved by staggering the gain; that is, by designing the first stage for 17.3 dB, the middle stage for 12.5 dB, and the output stage for 16.8 dB. The middle stage has a larger effect on the skirt selectivity with relatively little effect on overall gain. When the first and third stages are synchronously tuned, the skirt selectivity can then be controlled by tuning the center stage. In reality, for a balanced response curve, the center stage is tuned somewhat below the center frequency.

The completed FD3101 amplifier is 7.7 inches long with an outside diameter of 0.87 inch. The over-all gain is 46.5 dB with a noise figure of 11 dB. Fig. 4 shows a typical response curve taken with a sweep generator and a square-law detector. Some asymmetry due to feedback is noticeable at the base of the curve.

## PRESELECTOR DESIGN

The FD3100 preselector presented some unique problems. Three stages of preselection were necessary to obtain control of the deep skirts. Because the insertion loss is directly additive to the amplifier noise figure, the attenuation has to be held to a minimum. Spurious responses outside of the passband would lead to problems with any high-powered transmitters in the vicinity. The dimensional stability could be readily obtained by use of Invar, but the physical layout would have marked effect on the uniformity of manufacture and hence on the ultimate cost. Choosing a quasi-Butterworth response to maintain a fairly flat-top bandpass curve, an in-line coaxial design was used. This approach also permitted the use of jigged assembly techniques where close control on the position of the internal parts can be realized.

The FD3100 preselector uses an 8.4-inch length of seamless Invar tubing for the 0.87-inch-O.D. outer shell and 0.25-inch-O.D. Invar center lines. The RF input is coupled to the quarter-wave line by an input loop. As indicated in Fig. 5, the center line is capacitively end-coupled to the next quarter-wave line. This line is iris-coupled to the output section. The preselector output is coupled to the RF connector by a concentric capacitive probe. The probe was designed to control the coupling capacitance by the probe O.D. rather than its axial position to take up variations in the RF connector. This method also provides DC isolation and eliminates a DC blocking capacitor on the amplifier input.

The quarter-wave lines, cantilevered from the shorted ends, had a mechanical resonance near 500 Hz. To support the cantilever and add rigidity, 0.06-inch-thick fosterite discs were added at the center of each line; the discs are edge-metallized for soldering. Because the fosterite has a coefficient of expansion of $9.5 \times 10^{-6}$ in/°C, the outer rim was designed with three equally spaced flats so that the disc contacts the shell for about half of its circumference in three separated segments. This arrangement allows flexure of the shell between the soldered segments and reduces the stress on the ceramic well below the fracture point. Because the center hole-to-line joint is soldered at 230°C, the joint remains in compression over the operating temperature range and no stress relief is required.

The completed FD3100 preselector has a typical insertion loss of 0.9dB with a —3-db bandpass of 9.5 MHz and a —40-dB bandpass of 36 MHz.

### THE FD2201

The FD2201 consists of the FD3100 preselector coupled through a half-wave length of coaxial line to the FD3101 amplifier. The response curves shown in Fig. 6 are more meaningful in light of the following cw measurements:

| | |
|---|---|
| Gain at 1030 MHz | 45 ± 2 dB |
| Center Frequency | 1030 ± 0.5 MHz |
| — 3-dB bandwidth | 5 MHz (min) |
| —40-dB bandwidth | 25 MHz (max) |
| —60-dB bandwidth | 50 MHz (max) |
| Linear dynamic range | 60 dB |
| Noise Figure | 13 dB (max) |
| $\Delta f_o$,—62°C to +125°C* | ±1.0 MHz (max) |
| Δ gain, —62°C to +125°C | 1.5 dB (max') |

*Shell temperature

The features which make the FD2201 particularly useful from the system point of view are its light weight (7 oz.) and its low power drain (6.3 W).

The reliability may be surprising to those unfamiliar with the nuvistor. The original sample of nuvistors placed on life test four years ago are still operating well. From statistical data taken on the 8058 on 5000-hour life tests, the composite MTTF for the FD2201 is 570,000 hours. In addition, the FD2201 meets all of the requirements for class-3 equipment of MIL-E-5400G.

The flexibility of the design concept is of practical significance. Amplifiers have been built, with and without preselectors, in the frequency range of 600 MHz through 1200 MHz with any required gain from 15 dB to 50 dB.

The success of the design concept is self evident. The FD2201 is now being manufactured by Electronic Components and is being used in the APX-72 transponder. The FD2200V1 is also being used in the APX-71 radar interrogator. The excellent thermal stability of the nuvistor, its long life and unusually uniform electrical characteristics with relatively small size, have led to additional integral cavity possibilities.

A two-stage mechanically tunable 600-to-1000 MHz version, the A15528A, is being used in several specialized military applications. A 300-W 1000-MHz grid-pulsed transmitter, the A15569, has been developed using the 1-kV-rated A15526 nuvistor for possible application in phased-array radar. Oscillators using the 8627 nuvistor will supply the drive power to the A15569 for transmitter applications. In addition, a 1090-MHz version of the FD-2201, the A15515, was developed for airborne interrogators, and has been provided on a sample basis to several customers both here and abroad.

### LOOKING AHEAD

There are no direct competitors to the FD2201 today, but the size of the market will make competition inevitable. The advent of 1-GHz silicon transistors has led to the design of a solid-state preamplifier. Nearing design completion, the preamplifier (Fig. 7) was located within the amplifier input circuit without increasing the over-all length. This improved version can give either 45 or 50 dB gain with a system noise figure of 8 dB maximum. The natural evolution of this design is towards a hybrid IC stage with a discrete inductance to allow adjustment during manufacture. Hybrid FD2201 amplifiers will lead to all-solid-state designs having reduced size and lower power drain. Wherever the future trends may lead, the FD2201 has set a performance mark that will be hard to beat.

### ACKNOWLEDGMENT

The author wishes to acknowledge the contributions of the FD2201 design team: W. A. Harris, for his theoretical analysis and computer programs; C. Gonzalez, who developed the preselector from a laboratory model to a practical working unit; and L. J. Striednig, whose inventiveness and practical application of theory made the amplifier possible.

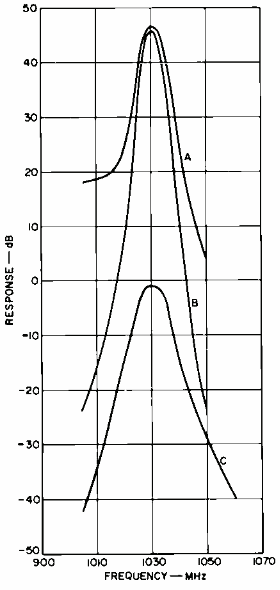
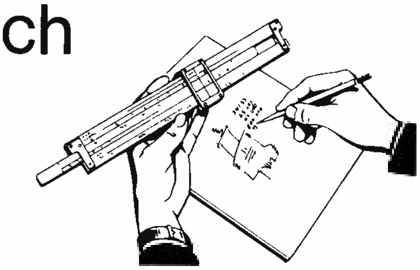


Fig. 6—Bandpass curves for a) FD3101 amplifier alone, c) FD3100 preselector alone, and b) FD2201.

Fig. 7—Recently designed solid-state preamplifier for FD2201.

# Engineering and Research Notes



Brief Technical Papers
of Current Interest

## Use of Polaroid Negatives

M. Aguilera

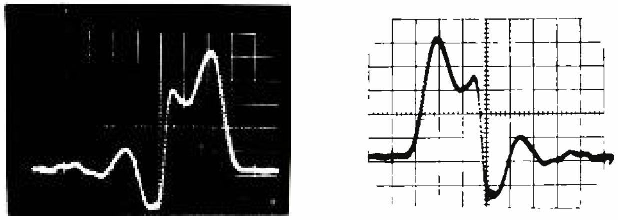
Defense Communications Systems Division
Camden, New Jersey

The engineer or technician associated with design, evaluation, or testing is frequently required to produce a report quickly. One of the obstacles to producing a quick report is the inability to present waveforms of various signals within the system.

Presently, the oscilloscope trace must be reduced to reproducible form by 1) hand drawing, 2) tracing a photograph or 3) photolithography. All of the methods are time consuming and may prohibit the inclusion of many desirable waveforms in a report.

The author has discovered that an acceptable method for reproducing waveforms exists to anyone having access to an electrostatic copier and a Polaroid* camera.

The procedure for this reproduction is to first take a Polaroid picture of the oscilloscope trace. The photograph is then mounted in an Engineering Notebook for safeguarding of the information. The negative is kept. This is trimmed, coated with Polaroid print coater, and mounted within the body of a typewritten report. The electrostatic copier can then be used to make one or two copies of the report or to produce an offset master to be used in making several hundred copies.



Positive

Negative

Fig. 1—Electrofax reproduction of Polaroid positive and negative.

The advantage of using the Polaroid negative is shown in Fig. 1. The electrostatic copier does not reproduce the solid black areas as shown in Fig. 1a; note the washout of detail. However, the copy of the Polaroid negative (Fig. 1b) shows the improvement in detail and contrast when using the negative.

The disadvantage of using the negative is also illustrated in Fig. 1. The time base of the negative is reversed when compared with the positive. To correct the time base, the sweep of the oscilloscope can be reversed. The oscilloscope vendor should be contacted for the correct method for doing this. The author has found that labeling the negative with the time-base direction is usually sufficient.

Negatives using Polaroid Type-47 film have been exposed to room lighting conditions for at least four weeks without showing severe fading. This is not believed to be a problem.

This method provides a quick, convenient, and adequate method of presenting important waveforms in technical reports, both small volume and large volume.

---

*Trademark of the Polaroid Corporation

Reprint RE-14-3-24
Final manuscript received September 22, 1968.

## Adjustable Tape-Guide Roller

B. Siryj

Advanced Mechanical Technology
Advanced Technology
Defense Electronic Products
Camden, New Jersey

A tape-guide roller capable of maintaining accurate perpendicularity bewteen a tape and a reference plane within a magnetic recorder has been designed for the Defense Communications Systems Division. Tape in these recorders moves rapidly; this alignment prevents damage due to excessive stress in the tape. Although designed for a magnetic recorder using a helical scan, the device can be adapted for use in most tape or film recorders.

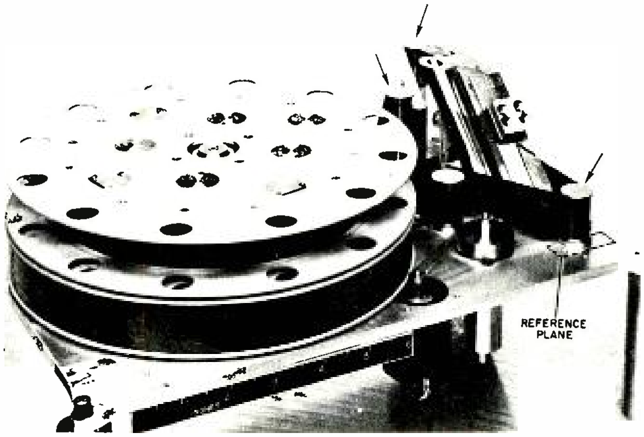
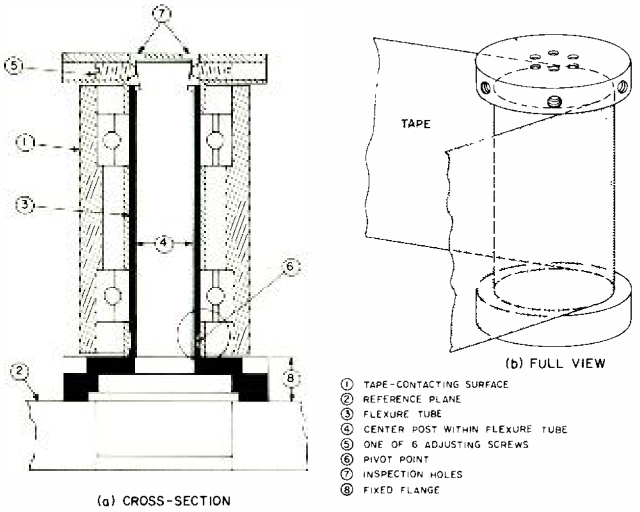


Fig. 1—Magnetic recorder showing tape-guide rollers (arrows).

Fig. 1 shows an arrangement of several fixed-flange rollers assembled in an operating recorder. Fig. 2 depicts the roller configuration, showing the tape-contacting surface (1) aligned perpendicular to the reference plane (2). The flexure tube (3) in conjunction with the inserted center post (4) provides the mechanism for aligning the tape-contacting surface with the reference plane.



(a) CROSS-SECTION

(b) FULL VIEW

Fig. 2—Adjustable tape-guide roller.

The modulus of elasticity and the moment of inertia of the stainless-steel center post are considerably larger than those of the magnesium flexure tube; therefore, when the two are forced against each other, principal deflection occurs in the flexure tube rather than the post. Alignment is accomplished by adjusting the six screws (5) mounted in the roller cap (actually part of the flexure tube). When a screw in the cap is advanced (and the opposing screw backed off), a cantilever-beam action at pivot point (6) forces the flexure tube and the tape-contacting surface to tilt. This alignment procedure thus adjusts the perpendicularity of the guided tape with respect to the reference plane. Visual inspection holes (7) aid in observing the position of the flexure tube relative to the center post.

Reprint RE-14-3-24
Final manuscript received August 22, 1968.

---

## Woven-Mesh Printed Circuits

William J. Stotz
and
Richard J. Araskewitz

Speech Processing
Advanced Technology
Defense Electronic Products
Camden, New Jersey

The authors have devised a technique for improving the reliability of printed circuits.

The fabrication of printed circuit boards, in accordance with state of the art techniques, begins with a dielectric substrate having a cladding of conductive foil mounted thereon; the cladding consisting of a layer of electrolytically deposited copper having a thickness in the order of from one to five mils. The board is generally coated with a layer of photosensitive material, exposed through artwork, developed and etched, to produce the desired circuitry. Fig. 1 is a partial top view of one layer of a multilayer circuit board, including a plated-thru hole, fabricated in accordance with this technique.

Under environmental conditions of temperature and vibration, stresss are often induced at the junction (not shown) of the plated-thru hole and the etched circuitry of the encapsulated layers of a multilayer circuit board. These stresses have the tendency to cause any imperfection which may exist at the junction, or be formed thereby, to propagate circumferentially, resulting in a complete separation between the plated-thru hole and the etched circuitry of the encapsulated layers. Any flaws in the form of notches or nicks, as shown in Fig. 1 tend to propagate, resulting in partially broken or open circuits.

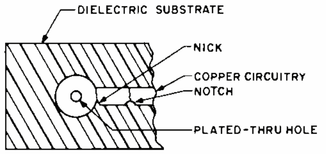


Fig. 1—Partial top view of one layer of a printed-circuit board.

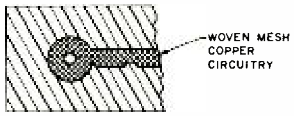


Fig. 2—Woven-mesh copper cladding.

By utilizing a woven mesh conductive cladding, i.e., copper, as shown in Fig. 2, there are formed many independent points of contact which have a greater degree of flexibility and strength than the single rigid joint shown in Fig. 1. The propagation of flaws is generally alleviated, thus making the etched circuitry less sensitive to notches and nicks. To provide required electrical characteristics, the intersection points of the mesh cladding should make positive electrical contact as distinguished from the ordinary type of woven mesh used in residential screens or the like.

Reprint RE-14-3-24
Final manuscript received April 15, 1968.

---

## Digital Multiplier

Lawrence N. Merson

Digital Communication Equipment
Defense Communications Systems Division
Camden, New Jersey

The circuit shown in Fig. 1 operates as a frequency doubling circuit using conventional digital logic elements. The delay circuits are arranged to produce a reset of the flip-flops C and D to produce respective output signals having pulse widths which are compatible to produce the output E. Waveshapes found in the circuit are shown in the timing diagram of Fig. 2. The circuit shown in Fig. 1 may be used in cascade to produce a succession of frequency doubling operations wherein the final frequency would be the initial frequency multiplied by $2^n$ where $n$ equals the number of cascaded circuits of Fig. 1.

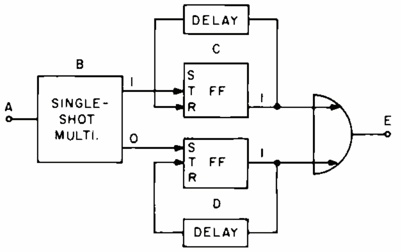


Fig. 1—Digital multiplier circuit.

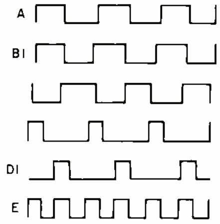
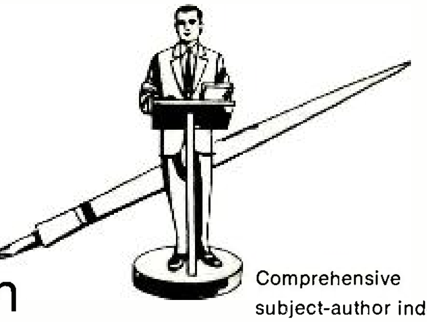


Fig. 2—Circuit waveshapes.

Reprint RE-14-3-24
Final manuscript received January 15, 1968



# Pen and Podium

Comprehensive
subject-author index
to Recent RCA
technical papers

Both published papers and verbal presentations are indexed. To obtain a published paper, borrow the journal in which it appears from your library, or write or call the author for a reprint. For information on unpublished verbal presentations, write or call the author. (The author's RCA Division appears parenthetically after his name in the subject-index entry.) For additional assistance in locating RCA technical literature, contact: **RCA Staff Technical Publications, Bldg. 2-8, RCA, Camden, N.J. (Ext. PC-4018).**

This index is prepared from listings provided bimonthly by RCA Division Technical Publcations Administrators and Editorial Representatives—who should be contacted concerning errors or omissons (see inside back cover).

**Subject index categories are based upon the Thesaurus of Engineering Terms, Engineers Joint Council, N.Y., 1st Ed., May 1964.**

## Subject Index

**Titles of papers are permuted where necessary to bring significant keyword(s) to the left for easier scanning. Authors' division appears parenthetically after his name.**

### AMPLIFICATION

**IC RF and IF AMPLIFIERS**—H. M. Kleinman (EC, Som) *Electronics World;* 7/68

**PARAMETRIC AMPLIFIER, Microelectronic**—Dr. W. Y. Pan (DCSD, W. Windsor) National Electronics Conf., Washington, D.C.; 5/10/68

### ANTENNAS

**CIRCULAR APERATURE ANTENNA, Difference Pattern Characteristics of**—C. E. Profera, L. H. Yorinks (MSR, Mrstn) *Microwaves,* Vol. 7, No. 6; 6/68

### CHECKOUT

**MAINTAINABLE ELECTRONICS on Long Duration Missions, Requirements for**—M. Johnson (ASD, Burl) 2nd National Conf. on Space Maintenance, Las Vegas, Nevada; 8/6-8/68

### CIRCUITS, INTEGRATED

**CMOS INTEGRATED CIRCUITS**—J. Hilibrand (EC, Som) Summer Course on Field-Effect Transistors, Massachusetts Inst. of Technology; 7/29-30/68

**HETERODYNE EXCITER Using FET's, A Novel**—G. D. Hanchett (EC, Som) ARRL National Convention, San Antonio, Texas; 6/7-9/68

**IC RF and IF AMPLIFIERS**—H. M. Kleinman (EC, Som) *Electronics World;* 7/68

**LARGE-SCALE INTEGRATION of Linear IC's**—S. Katz (EC, Som) *Electronics World;* 7/68

**L-BAND TRANSMITTER, Hybrid Integrated**—E. Belohoubek, H. Johnson, A. Presser (EC, Pr) *IEEE Trans. on Electron Devices;* 7/68

**LSI TECHNOLOGY TRENDS—Bipolar and MOS**—E. E. Moore (EC, Som) IEEE LSI Conf., Los Angeles, Cal.; 6/25/68

**MICROWAVE INTEGRATED CIRCUITS, Notes on**—H. Sobol (EC, Som) Integrated-Circuits Course, U. of Michigan; 6/10-14/68

### CIRCUITS, PACKAGED

**MICROMIN for the 70's**—H. Sobol (EC, Som) Microwave Exposition, San Francisco, Calif.; 6/4-6/68

**WOVEN MESH Printed Circuits**—W. J. Stotz, R. J. Araskewitz (DCSD, Cam) *Technical Notes #21;* 6/3/68

### COMMUNICATIONS COMPONENTS

**AIRBORNE MILITARY TRANSCEIVER Finds Room in Crowded Spectrum**—L. P. Magasiny (DCSD, Cam) *Electronics;* 4/15/68

**50 MHz TRANSVERTER for Transceiver Operation, Design Concepts for a**—H. W. Brown (DCSD, Cam) Mt. Airey VHF Club, Phila., Pa.; 7/19/68

**FM RECEIVERS Using High-Gain Integrated-Circuit IF Ampliers, Design of High-Performance**—L. Kaplan, T. J. Robe (EC, Som) IEEE Conf. on Broadcast and TV Receivers, Chicago, Ill.; 6/17-18/68

**HETERODYNE EXCITER Using FET's, A Novel**—G. D. Hanchett (EC, Som) ARRL National Convention, San Antonio, Texas; 6/7-9/68

**HIGH-POWER FREQUENCY DOUBLERS Using Coupled TME Lines**—C. Sun (EC, Pr) *RCA Review;* 6/68

**L-BAND TRANSMITTER, Hybrid Integrated**—E. Belohoubek, A. Presser, H. Johnson (EC, Pr) IEEE Trans. on Electron Devices; 7/68

**LOW-DELAY DISTORTION FDM TERMINAL**—A. Acampora, G. Winram (DCSD, W. Windsor) International Communications Conf., Philadelphia, Pa.; 6/12/68

### COMMUNICATIONS SYSTEMS

**SINGLE SIDEBAND STORY—Developing the STR-150 and SBA-1K**—E. W. Mahland (CESD, Mdw. Lds) APCO Spring Conf.; 4/24/68 and IMSA Conf.; 5/24/68

### COMMUNICATION, VOICE

**VOICE CONTROLLER for Astronaut Maneuvering Unit**—M. B. Herscher, T. P. Kelly (AT, Cam) 2nd National Conf. on Space Maintenance & Extravehicular Activities, Los Angeles, Calif.; 8/8/68

### COMPUTER APPLICATIONS

**EMI DATA REDUCTION, Prediction, and Analysis Using Time-Shared Computers**—A. Dimarzio (ASD, Burl) IEEE 1968 Symposium on Electromagnetic Compatibility, Seattle, Washington; 7/23-25/68; *Symposium Record;* 7/68

### COMPUTERS, PROGRAMMING

**LOGIC SIMULATION PROGRAM, Effective Utilization of a**—A. P. Moll (MSR, Mrstn) Master's Thesis, U. of Penna.; 5/68

### COMPUTER STORAGE

**TAKING CRYOELECTRIC MEMORIES out of Cold Storage**—R. A. Gange, J. J. Carrona (EC, Pr) *Electronics;* 4/17/68

### COMPUTER SYSTEMS

**COMPUTER SYSTEMS ANALYSIS AND DESIGN, Fundamentals of**—F. Congdon (ASD, Burl) AMA Seminar; 5/68 and 8/68

### CONTROL SYSTEMS

**MESSAGE SWITCHING**—I. N. Suskind (DCSD, Cam) Seminar Pennsylvania Military College; 6/68

### DISPLAYS

**THIN-FILM CIRCUITS for Scanning Image-Sensor Arrays**—G. Sadasiv, P. K. Weimer, W. S. Pike (Labs., Pr) *IEEE Trans. on Electron Devices;* 4/68

### EDUCATION

**TECHNOLOGICAL ADVANCES in Electronics**—J. M. Forman (EC, Lanc) *RCA Family News;* 4-5/68

### ELECTROMAGNETIC WAVES

**MULTIPLE-LOOP FREQUENCY-COMPRESSIVE Feedback for Angle-Modulation Detection**—H. Heinemann, A. Newton, J. Frankle (DCSD, Cam) *RCA Review;* 6/68

**NUCLEAR MAGNETIC RESONANCE Determination of the $^{125}$Te Hyperfine Field on Ferromagnetic $CuCr_2Te_4$**—S. B. Berger, J. I. Budnick, T. J. Burch (Labs., Pr) *Physics Letters,* Vol. 26A, No. 10; 4/8/68

**PHASE TRANSITIONS of an Isotropic Ferromagnet in an External Magnetic Field**—M. Rayl, P. J. Wojtowicz (Labs., Pr) *Physical Review Letters,* Vol. 20, No. 26; 6/24/68

### ENERGY CONVERSION

**50,000-WATT HEAT-PIPE SPACE RADIATOR, Design of a**—R. C. Turner, W. E. Harbaugh (EC, Lanc) ASME Meeting, Beverly Hills, Calif.; 6/16-19/68

**HEAT PIPE—A New Tool in Heat Transfer**—G. Y. Eastman (EC, Lanc) *Scientific American;* 5/68

### ENVIRONMENTAL ENGINEERING

**50,000-WATT HEAT-PIPE SPACE RADIATOR, Design of a**—R. C. Turner, W. E. Harbaugh (EC, Lanc) ASME Meeting, Beverly Hills, Calif.; 6/16-19/68

**HEAT PIPE—A New Tool in Heat Transfer**—G. Y. Eastman (EC, Lanc) *Scientific American;* 5/68

### GRAPHIC ARTS

**EXPOSING SOURCES for Kodak Resists**—G. F. Damon (EC, Som) Kodak Photoresist Seminar, Los Angeles, Calif.; 5/20/68

**GRAPHIC COMMUNICATIONS IN THE 1970s, The Impact of the Computer on**—A. H. Coleman (GSD, Dayton) Oregon Center for Continuing Education in Conf. on Industrial and Technical Communications; 7/26/68; and the Society of Technical Writers and Publishers, Management Group, Lexington, Mass.; 5/23/68

### INTERFERENCE

**EMI DATA REDUCTION, Prediction, and Analysis Using Time-Shared Computers**—A. Dimarzio (ASD, Burl) IEEE 1968 Symposium on Electromagnetic Compatibility, Seattle, Washington; 7/23-25/68; *Symposium Record;* 7/68

**NOISE IN TELEVISION BROADCAST EQUIPMENT, Study of**—K. Sadashige (CESD, Cam) 103rd Annual SMPTE Conf., Los Angeles, Calif.; 5/8/68

### LABORATORY TECHNIQUES

**CHEMICAL ANALYSIS by X-Ray Secondary-Emission (Fluorescence) Spectrometry, Practical Aspects of**—E. P. Bertin (EC, Hr) Summer Workshop in X-Ray Spectrometry, State University of New York, Albany, N.Y.; 6/10-14/68

**CURVE TRACER that Frees Diode Characteristics of Internal Resistance**—J. I. Pankove, J. E. Berkeyheiser (Labs., Pr) *Review of Scientific Instruments,* Vol. 39, No. 6; 6/68

**$GaAs_{1-x}P_x$ INJECTION LASERS, Time Delay and Memory Effects in**—J. I. Pankove (Labs., Pr) 9th International Conf. on the Physics of Semiconductors, Moscow, USSR; 7/23-29/68

### LASERS

**INJECTION LASER of Semi-Planar Geometry, Low-Threshold, High-Efficiency**—H. W. Becke (EC, Som) IEEE Quantum Electronics Conf., Miami, Fla.; 5/14-17/68

**LASER ACTION in Field-Ionized Bulk GaAs**—P. D. Southgate (Labs., Pr) *Applied Physics Letters,* Vol. 12, No. 3; 2/1/68

### LOGIC THEORY

**ERROR CORRECTIONS IN MEMORY SYSTEMS, Coding Schemes for**—C. V. Srinivasan (Labs., Pr) Northwestern University, Evanston, Illinois; 7/9/68

**THRESHOLD LOGIC will cut costs, especially with boost from LSI**—R. O. Winder (Labs., Pr) *Electronics;* 5/27/68

### MANAGEMENT

**DECISION PROCESS, Understanding the**—S. M. Perone (DCSD, Cam) *Administrative Management;* 5/68

**LEADERSHIP—A Theory on Scientific Management**—H. J. Wood (DCSD, Cam) American Management Visual Training Library; 7/68

**THRESHOLD LOGIC will cut costs, especially with boost from LSI**—R. O. Winder (Labs., Pr) *Electronics;* 5/27/68

### MEDICAL ELECTRONICS

**BIOMEDICAL ENGINEERING, Another Look at**—L. E. Flory (ME, Trenton) The Philadelphia Chapter of the G-EMB of the IEEE; 5/14/68

### PARTICLE BEAMS

**PARTICLE SIZE AND MORPHOLOGY of Zinc Sulfide: I—Influence of Precipitating Conditions**—R. A. Brown (EC, Lanc) *Electrochemical Technology;* 7-8/68

### PROPERTIES, ATOMIC

**SECONDARY-ELECTRON EMISSION**—R. E. Simon, B. F. Williams (EC, Pr) *IEEE Trans. on Nuclear Science;* 6/68

### PROPERTIES, MOLECULAR

**ANISTROPIC CRYSTALS, On the Band Structure of**—G. Harbeke (Labs., Pr) *Physica Status Solidi*, Vol. 27; 1968

**CLOSE-SPACED GROWTH of Degenerate P-Type GaAs, GaP, and Ga(As,P) by $ZnCl_2$ Transport**—P. A. Hoss, L. A. Murray, J. J. Rivera (EC, Som) *J. of the Electrochemical Society*; 5/68

**FERROMAGNETIC HALOCHALCOGENIDE SPINELS ($CuCr_2X_3Y$) and Some Properties of the Systems $CuCr_2Se_3Br$-$CuCr_2Se_4$ and $CuCr_2Te_3I$-$CuCr_2Te_4$**—M. Robbins, P. K. Baltzer, E. Lopatin (Labs., Pr) *J. of Applied Physics*, Vol. 39, No. 2, Part I; 2/1/68

**LARGE STOICHIOMETRIC MAGNESIUM ALUMINATE SPINEL SINGLE CRYSTALS, Growth and Characterization of**—C. C. Wang, S. H. McFarlane III (Labs., Pr) International Conf. on Crystal Growth, Birmingham, England; 7/15-19/68

**LATTICE EXPANSION AND SUPERCONDUCTIVITY of $Nb_3Sn$ as a Function of Hydrogen Content**—P. R. Sahm (Labs., Pr) *Physics Letters*, Vol. 26 A, No. 10; 4/8/68

**SINGLE CRYSTALS of $CaNb_2O_6$ by Chemical Transport, Preparation of**—F. P. Emmenegger (Labs., Pr) *J. of Crystal Growth*, Vol. 2; 1968

**$ZnRh_2O_4$ SINGLE CRYSTALS, Growth of**—R. H. Arlett (Labs., Pr) *J. of the American Ceramic Society*, Vol. 51, No. 5; 5/68

### PROPERTIES, SURFACE

**CESIUM ANTIMONIDE FILMS Deposited on Various Substrates, An Electron and X-Ray Diffraction Study of**—W. H. McCarroll (EC, Pr) *J. of Applied Physics*; 5/68

**PIEZOELECTRIC SEMICONDUCTORS, High-Electric-Field Galvanomagnetic Effects in**—R. S. Crandall (Labs., Pr) *Physical Review*, Vol. 169, No. 3; 5/15/68

**SILICON-ON-SAPPHIRE FILMS, Electrically and Optically Active Defects in**—D. J. Dumin, P. H. Robinson (Labs., Pr) International Conf. on Crystal Growth, Birmingham, England; 7/15-19/68

**THIN-FILM CIRCUITS for Scanning Image-Sensor Arrays**—G. Sadasiv, P. K. Weimer, W. S. Pike (Labs., Pr) *IEEE Trans. on Electron Devices*; 4/68

### PROPERTIES, CHEMICAL

**TERNARY METAL OXIDES by Chemical Transport, Crystal Growth of**—F. P. Emmenegger (Labs., Pr) International Conf. on Crystal Growth, Birmingham, England; 7/15-19/68

### PROPERTIES, ELECTRICAL

**AVALANCHE QUENCHED DOMAINS in Transferred-Electron Oscillators**—J. R. Collard (EC, Pr) Conf. on Electron-Device Research, Boulder, Colo.; 6/19-21/68

**ELECTRICAL CONDUCTION in n-Type Cadmium Sulfide at Low Temperature**—R. S. Crandall (Labs., Pr) *Physical Review*, Vol. 169, No. 3; 5/15/68

**FERROELECTRIC PROPERTIES of Stable and Metastable Phase III $KNO_3$**—G. W. Taylor, B. J. Lechner (Labs., Pr) *J. of Applied Physics*, Vol. 39, No. 5; 4/68

**RELAXATION PHENOMENA in Planar Ge(Li) Detectors**—P. P. Webb (RCA Ltd., Montreal) 11th Scintillation & Semiconductor Counter Symp. of IEEE, Washington, D.C.; *IEEE Trans. on Nuclear Science*, Vol. NS-15, No. 3; 6/68

**STATIC NEGATIVE RESISTANCE in Transferred-Electron Devices**—F. Sterzer (EC, Pr) Conf. on Electron-Device Research, Boulder, Colo.; 6/19-21/68

### PROPERTIES, OPTICAL

**RARE-EARTH OXYSULFIDES—A New Family of Phosphor Hosts for Rare-Earth Activators**—M. R. Royce, A. L. Smith (EC, Lanc) Electrochemical Society Meeting, Boston, Mass.; 5/5-9/68

### PROPERTIES, THERMAL

**ABSORPTION EDGE OF GaAs, Temperature Dependence of the**—M. A. Afromowitz, D. Redfield (Labs., Pr) 9th International Conf. on the Physics of Semiconductors, Moscow, USSR; 7/23-29/68

**REFRACTORY MATERIALS AS DIFFUSION BARRIARS, Application of**—P. J. Chao (EC, Hr) Plansee Seminar on High-Temperature Materials, Tyrol, Austria; 6/24/68

### RADAR

**PULSE DOPPLER SYSTEM, A Monopulse Instrumentation Tracking Radar**—J. F. O'Brien (MSR, Mrstn) Master's Thesis, U. of Penna.; 5/69

### RECORDING

**EQUALIZATION TECHNIQUES for Extended Bandwidth FM Recording**—G. T. Rogers (AT, Cam) University of Penna., Moore School of Electrical Engineering, *Thesis*, Phila., Pa.; 7/26/68

**QUADRUPLEX RECORDING, Velocity Errors in**—R. N. Hurst (CESD, Cam) 103rd Annual SMPTE Conf., Los Angeles, Calif.; 5/10/68

### RELIABILITY

**EXTENDING THE LIFE OF CHROMIUM—Silver Metallization of Silicon Devices**—E. C. Ross, J. T. Wallmark (Labs., Pr) 1967 Annual Symposium on Reliability Physics

**LITHIUM MOBILITY WITH RADIATION SENSITIVITY in Lithium Containing Solar Cells, Correlation of**—A. G. Holmes-Siedle, G. Brucker, T. Faith (AED, Pr) IEEE Nuclear Radiation Effects Conf., Missoula, Mont.; 7/15/68

**POWER-TRANSISTOR FAILURES in Inverters Driving Resistive or Capactive Loads, Cut Down on**—D. M. Baugher (EC, Som) *Electronic Design*; 5/9/68

**UNIFORM RADIATION SENSITIVITY in Planar Transistors, A Selection System for**—A. G. Holmes-Siedle (AED, Pr) IEEE Nuclear Radiation Effects Conf., Missoula, Mont.; 7/15/68

### SOLID-STATE DEVICES

**$Al_2O_3$-SILICON Insulated Gate Field Effect Transistors**—A. Waxman, K. H. Zaininger (Labs., Pr) *Applied Physics Letters*, Vol. 12, No. 3; 2/1/68

**HIGH-FREQUENCY TRANSISTOR Selection Checklist**—R. L. Wilson (EC, Som) *Electronic Products*; 5/68

**LITHIUM MOBILITY WITH RADIATION SENSITIVITY in Lithium Containing Solar Cells, Correlation of**—A. G. Holmes-Siedle, G. Brucker, T. Faith (AED, Pr) IEEE Nuclear Radiation Effects Conf., Missoula, Mont.; 7/15/68

**POWER-TRANSISTOR FAILURES in Inverters Driving Resistive or Capacitive Loads, Cut Down on**—D. M. Baugher (EC, Som) *Electronic Design*; 5/9/68

**UNIFORM RADIATION SENSITIVITY in Planar Transistors, A Selection System for**—A. G. Holmes-Siedle (AED, Pr) IEEE Nuclear Radiation Effects Conf., Missoula, Mont.; 7/15/68

### SPACE COMMUNICATION

**DATA RELAY SATELLITES**—J. D. Kiesling (AED, Pr) Annual Meeting American Society of Engineering Education at UCLA, Los Angeles, Calif.; 7/17/68

**SEALING FRIT in Color Picture Tubes, Use of**—R. K. Schneider (EC, Marion) American Ceramic Society Indiana Branch, Carmel, Indiana; 6/3/68

**ORBITING DATA RELAY NETWORK**—C. R. Whelan, J. Kiesling, S. H. Durrani, H. Berkowitz (AED, Pr) Joint seminar with Rutgers and Princeton Univ. EE Departments; 4/18/68

**VOICE CONTROLLER for Astronaut Maneuvering Unit**—M. B. Herscher, T. P. Kelly (AT, Cam) 2nd National Conf. on Space Maintenance & Extravehicular Activities, Los Angeles, Calif.; 8/8/68

### SPACECRAFT

**CANADIAN DOMESTIC SATELLITES—defining the Systems Considerations**—D. Jung (RCA Ltd., Montreal) *Electronics & Communications*, Vol. 16, No. 6; 6/68

**METEOROLOGICAL SATELLITES**—E. A. Goldberg, L. Krawitz, W. J. Haneman, R. Molloy, M. Shepetin (AED, Pr) Seminar with Princeton Univ.—Mechanical and Aerospace Science Dept.; 4/30/68

### SUPERCONDUCTIVITY

**LATTICE EXPANSION AND SUPERCONDUCTIVITY of $Nb_3Sn$ as a Function of Hydrogen Content**—P. R. Sahm (Labs., Pr) *Physics Letters*, Vol. 26 A, No. 10; 4/8/68

**RCA SUPERCONDUCTIVE MATERIALS**—Presented as part of Panel Discussion in Summer Study Session, Brookhaven National Laboratory, L.I., N.Y.; 6/27/68

**TAKING CRYOELECTRIC MEMORIES out of Cold Storage**—R. A. Gange, J. J. Carrona (EC, Pr) *Electronics*; 4/17/68

### TRANSMISSION LINES

**HIGH-POWER FREQUENCY DOUBLERS Using Coupled TME Lines**—C. Sun (Labs., Pr) *RCA Review*; 6/68

**OPTIMUM MULTILAYER PRINTED CIRCUIT TRANSMISSION LINE Design for High Density Picosecond Digital Applications**—J. J. Surina (AT, Cam) International Electronic Circuit Packaging Symposium, Los Angeles, Calif.; 8/19-20/68

### TUBES, ELECTRON

**CERAMIC-METAL PHOTOMULTIPLIERS, High-Temperature Performance of**—R. M. Matheson, F. A. Helvy (EC, Lanc) *IEEE Trans. on Nuclear Science*; 6/68

**DISTRIBUTED KLYSTRONS, A Theoretical and Experimental Analysis of the Large-Signal Behavior of**—C. Sun (EC, Pr) *IEEE Trans. on Electron Devices*; 2/68

**EXTERNAL-ANODE TUBES for the Radio Amateur**—L. W. Aurick (EC, Lanc) CQ; 6/68

**NEW IMAGE ISOCON**—E. M. Musselman (EC, Lanc) Institute on Photo-Electronic Devices, University of Rhode Island; 7/15/68

**RCA PERMA-CHROME COLOR PICTURE TUBE, Development of the**—R. H. Godfrey, T. M. Shrader, R. C. Demmy (EC, Lanc) IEEE Conf. on Broadcast and TV Receivers, Chicago, Ill.; 6/17-18/68

**TELEVISION CAMERA TUBES, Transfer Characteristics & Spectral Response of**—L. D. Miller (EC, Lanc) Institute on Photoelectronic Devices, University of Rhode Island; 7/18/68

**TELEVISION PICTURE TUBE, Transfer Characteristics and Spectral Response of**—L. D. Miller (EC, Lanc) Course on Photo-Electronic Imaging Devices, University of Rhode Island; 7/15/68

### TUBE COMPONENTS

**EUOPIUM-ACTIVATED RED PHOSPHORS, Properties of Some Selected**—S. S. Tron, J. S. Martin, J. P. Stanavage, A. L. Smith (EC, Lanc) Electrochemical Society Meeting, Boston, Mass.; 5/5-9/68

---

## Author Index

**Subject listed opposite each author's name indicates where complete citation to his paper may be found in the subject index.**

### ASTRO ELECTRONICS DIVISION

**Berkowitz, H.** space communication
**Brucker, G.** reliability
**Brucker, G.** solid-state devices
**Durrani, S. H.** space communication
**Faith, T.** reliability
**Goldberg, E. A.** spacecraft
**Haneman, W. J.** spacecraft
**Holmes-Siedle, A. G.** reliability
**Holmes-Siedle, A. G.** solid-state devices
**Holmes-Siedle, A. G.** reliability
**Holmes-Siedle, A. G.** solid-state devices
**Kiesling, J. D.** space communication
**Kiesling, J. D.** space communication
**Krawitz, L.** spacecraft
**Molloy, R.** spacecraft
**Shepetin, M.** spacecraft
**Whelan, C. R.** space communication

### AEROSPACE SYSTEMS DIVISION

**Congdon, F.** computer systems
**Dimarzio, A.** computer applications
**Dimarzio, A.** interference
**Johnson, M.** checkout

### ADVANCED TECHNOLOGY

**Herscher, M. B.** communication, voice
**Herscher, M. B.** space communication
**Kelly, T. P.** communication, voice
**Kelly, T. P.** space communication
**Rogers, G. T.** recording
**Surina, J. J.** transmission lines

### COMMERCIAL ELECTRONIC SYSTEMS DIVISION

**Hurst, R. N.** recording
**Mahland, E. W.** communications systems
**Sadashige, K.** interference

### DEFENSE COMMUNICATIONS SYSTEMS DIVISION

**Acampora, A.** communications components
**Araskewitz, R. J.** circuits, packaged
**Brown, H. W.** communications components
**Frankle, J.** electromagnetic waves
**Heinemann, H.** electromagnetic waves
**Magasiny, L. P.** communications components
**Newton, A.** electromagnetic waves
**Pan, W. Y.** amplification
**Perone, S. M.** management
**Stotz, W. J.** circuits, packaged
**Suskind, I. N.** control systems
**Winram, G.** communications components
**Wood, H. J.** management

### ELECTRONIC COMPONENTS

**Aurick, L. W.** tubes, electron
**Baugher, D. M.** reliability
**Baugher, D. M.** solid-state devices
**Bertin, E. P.** laboratory techniques
**Becke, H. W.** lasers
**Belohoubek, E.** circuits, integrated
**Belohoubek, E.** communications components
**Brown, R. A.** particle beams
**Carrona, J. J.** computer storage
**Carrona, J. J.** superconductivity
**Chao, P. J.** properties, thermal
**Collard, J. R.** properties, electrical
**Damon, G. F.** filters, electrical

**Demmy, R. C.** tubes, electron
**Eastman, G. Y.** energy conversion
**Eastman, G. Y.** environmental engineering
**Forman, J. M.** education
**Gange, R. A.** computer storage
**Gange, R. A.** superconductivity
**Godfrey, R. H.** tubes, electron
**Hanchett, G. D.** circuits, integrated
**Hanchett, G. D.** communications components
**Harbaugh, W. E.** energy conversion
**Harbaugh, W. E.** environmental engineering
**Helvy, F. A.** tubes, electron
**Hilibrand, J.** circuits, integrated
**Hoss, P. A.** properties, molecular
**Johnson, H.** circuits, integrated
**Johnson, H.** communications components
**Kaplan, L.** communications components
**Katz, S.** circuits, integrated
**Kleinman, H. M.** amplification
**Kleinman, H. M.** circuits, integrated
**Martin, J. S.** tube components
**Matheson, R. M.** tubes, electron
**McCarroll, W. H.** properties, surface
**Miller, L. D.** tubes, electron
**Miller, L. D.** tubes, electron
**Moore, E. E.** circuits, integrated
**Murray, L. A.** properties, molecular
**Musselman, E. M.** tubes, electron
**Musselman, E. M.** tubes, electron
**Presser, A.** circuits, integrated
**Presser, A.** communications components
**Rivera, J. J.** properties, molecular
**Robe, T. J.** communications components
**Royce, M. R.** properties, optical
**Schneider, R. K.** tube components
**Shrader, T. M.** tubes, electron
**Simon, R. E.** properties, atomic
**Smith, A. L.** properties, optical
**Smith, A. L.** tube components
**Sobol, H.** circuits, packaged
**Sobol, H.** circuits, integrated
**Stanavage, J. P.** tube components
**Sterzer, F.** properties, electrical
**Sun, C.** tubes, electron
**Sun, C.** communications components
**Sun, C.** transmission lines
**Trond, S. S.** tube components
**Turner, R. C.** energy conversion
**Turner, R. C.** environmental engineering
**Williams, B. F.** properties, atomic
**Wilson, R. L.** solid-state devices

## GRAPHIC SYSTEMS DIVISION

**Coleman, A. H.** graphic arts

## LABORATORIES

**Afromowitz, M. A.** properties, thermal
**Arlett, R. H.** properties, molecular
**Berkeyheiser, J. E.** laboratory techniques
**Berger, S. B.** electromagnetic waves
**Baltzer, P. K.** properties, molecular
**Budnick, J. I.** electromagnetic waves
**Burch, T. J.** electromagnetic waves
**Crandall, R. S.** properties, electrical
**Crandall, R. S.** properties, surface
**Dumin, D. J.** properties, surface
**Emmenegger, F. P.** properties, molecular
**Emmenegger, F. P.** properties, chemical
**Harbeke, G.** properties, molecular
**Lechner, B. J.** properties, electrical
**Lopatin, E.** properties, molecular
**McFarlane, S. H.** properties, molecular
**Pankove, J. I.** laboratory techniques
**Pankove, J. I.** laboratory techniques
**Pike, W. S.** displays
**Pike, W. S.** properties, thermal
**Rayl, M.** electromagnetic waves
**Redfield, D.** properties, thermal
**Robbines, M.** properties, surface
**Robinson, P. H.** properties, surface
**Ross, E. C.** reliability
**Sadasiv, G.** displays
**Sadasiv, G.** properties, surface
**Sahm, P. R.** properties, molecular
**Sahm, P. R.** superconductivity
**Southgate, P. D.** lasers
**Srinivasan, C. V.** logic theory
**Taylor, G. W.** properties, electrical
**Wallmark, J. T.** reliability
**Wang, C. C.** properties, molecular
**Waxman, A.** solid-state devices
**Weimer, P. K.** displays
**Weimer, P. K.** properties, surface
**Winder, R. O.** logic theory
**Winder, R. O.** management
**Wojtowicz, P. J.** electromagnetic waves
**Zaininger, K. H.** solid-state devices

## MEDICAL ELECTRONICS

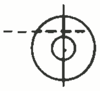
**Flory, L. E.** medical electronics

## MISSILE AND SURFACE RADAR DIVISION

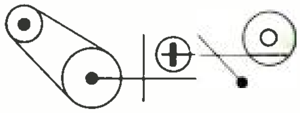
**Jung, D.** spacecraft
**Moll, A. P.** computers, programming
**O'Brien, J. F.** radar
**Profera, C. E.** antennas
**Webb, P. P.** properties, electrical
**Yorinks, L. H.** antennas

---

# Patents Granted



## to RCA Engineers



As reported by RCA Domestic Patents, Princeton

## ASTRO-ELECTRONICS DIVISION

**Thermocouple Assembly**—R. C. Turner (AED, Pr) U.S. Pat. 3,377,208, April 9, 1968 (Patent assigned to U.S. Government)

**Tape Cushioning Apparatus for Tape Transports**—R. Herman, R. W. Raynor (AED, Pr) U.S. Pat. 3,398,910, August 27, 1968

**Tape Transports**—S. P. Clurman (AED, Pr) U.S. Pat. 3,392,927, July 16, 1968

**Directive Antennas**—J. D. Kiesling (AED, Pr) U.S. Pat. 3,396,394, August 6, 1968

## LABORATORIES

**Cryogenic Heat Pump Including Magnetic Means for Moving a Normal Zone along a Superconductive Rod**—J. Pearl (Labs., Pr) U.S. Pat. 3,393, 526, July 23, 1968

**Light Frequency Shifter**—F. Sterzer (Labs., Pr) U.S. Pat. 3,393,955, July 23, 1968

**Signaling Loop Including Unbroken Railroad Track**—F. L. Hatke, G. W. Gray (Labs., Pr) U.S. Pat. 3,398,275, August 20, 1968

**Composite Metal Articles**—J. J. Hanak, F. D. Rosi (Labs., Pr) U.S. Pat. 3,395,000, July 30, 1968

**Sensor Array Coupling Circuits**—P. K. Weimer (Labs., Pr) U.S. Pat. 3,397,325, August 13, 1968

**Method of Making Divalent Rare Earth Laser Crystals in an Electric Field**—F. K. Fong (Labs., Pr) U.S. Pat. 3,393,140, July 16, 1968 (Assigned to U.S. Government)

**Semiconductor Devices and Circuits Using the Pinch Effect**—J. Glicksman, M. C. Steele (Labs., Pr) U.S. Pat. 3,396,283, August 6, 1968

**Ferroelectric Control Circuits**—B. J .Lechner, G. W. Taylor (Labs., Pr) U.S. Pat. 3,393,345, July 16, 1968

**Excitation Circuits for an Array of Electrical Elements**—B. J. Lechner, J. Tults (Labs., Pr) U.S. Pat. 3,393, 346, July 16, 1968

**Frequency Control Circuit Utilizing Switching Means**—R. F. Sanford (Labs., Pr) U.S. Pat. 3,393,379, July 16, 1968

**Integrated Semiconductor Diode Matrix**—R. A. Shahbender (Labs., Pr) U.S. Pat. 3,399,390, August 27, 1968

## CONSUMER ELECTRONIC DIVISION

**Size Stabilization**—T. J. Christopher. J. A. McDonald (CED, Indpls) U.S. Pat. 3,388,285, June 11, 1968

**Frequency Modulation Detector Circuit Suitable for Integration in a Monolithic Semiconductor Body**—J. Avins (CED, Som) U.S. Pat. 3,383,607, May 14, 1968

**Integrated Circuit Biasing Arrangements**—L. A. Harwood (CED, Som) U.S. Pat. 3,383,612, May 14, 1968

**Blanking Circuits for Television Receivers**—G. E. Anderson (CED. Indpls) U.S. Pat. 3,392,306, July 9, 1968

**Television Deflection Circuits**—L. R. Kirkwood, C. S. Liu (CED, Indpls) U.S. Pat. 3,379,924, April 23, 1968

**Signal Translating Circuit**—W. E. Davis, G. P. Lee (CED, Indpls) U.S. Pat. 3,399,-277, August 27, 1968

**FM Counter-Type Detector Especially Suited for Integrated Circuit Fabrication**—J. Avins (CED, Som) U.S. Pat. 3,399,353, August 27, 1968

**Electron Beam Convergence Apparatus**—G. K. Sendelweck (CED, Indpls) U.S. Pat. 3,393,343, July 16, 1968

**Regulated Power Supply**—N. W. Hursh (CED, Indpls) U.S. Pat. 3,395,311, July 30, 1968

**Television Deflection Power Recovery Circuit**—G. F. Rogers (CED, Indpls) U.S. Pat. 3,395,313, July 30, 1968

**Energizing System for Color Purity Apparatus**—E. Lemke, N. W. Hursh (CED, Indpls) U.S. Pat. 3,398,319, August 20, 1968

**Dynamic Color Purity Apparatus**—N. W. Hursh (CED, Indpls) U.S. Pat. 3,398,320, August 20, 1968

## COMMERCIAL ELECTRONIC SYSTEMS DIVISION

**Multivibrator with Its Timing Cycle Determined and Initiated by first two Pulses of Input Clock but then Isolated Therefrom for Remainder of Count**—R. A. Dischert (CESD, Cam) U.S. Pat. 3,382,375, May 7, 1968

**Indoor Television Antenna**—J. D. Callaghan (CESD, P&A, Deptford) U.S. Pat. D211,025, May 14, 1968

**Pulse Rate Limiting Circuit**—G. R. Kamerer, L. F. Crowley (CESD, MdwLds) U.S. Pat. 3,390,339, June 25, 1968

**Film Threading Arrangement**—J. L. Young, R. Lichalk (CESD, Mdw. Lds.) U.S. Pat. 3,393,847, July 23, 1968

**Transistorized Sync Stripper**—L. J. Baun (CESD, Cam) U.S. Pat. 3,398,298, August 20, 1968

**Horizontal Deflection Linearity Control Circuit**—L. J. Bazin (CESD, Cam) U.S. Pat. 3,398,318, August 20, 1968

## MISSILE AND SURFACE RADAR DIVISION

**Tape Transport Apparatus**—M. Yamamoto, T. F. Carlin (MSR, Mrstn) U.S. Pat. 3,380,683, April 30, 1968 (Assigned to U.S. Government)

**Multichannel Time Delay System Employing Less Delay Lines than the Number of Channels**—A. A. Gorski (MSR, Mrstn) U.S. Pat. 3,392,444, May 7, 1968 (Assigned to U.S. Government)

**Digital Frequency and Phase Discriminator**—R. J. McCurdy (MSR, Mrstn) U.S. Pat. 3,391,343, July 2, 1968 (Assigned to U.S. Government)

## ELECTRONIC COMPONENTS

**Superconductive Magnet Construction**—H. C. Schindler (EC, Hr.) U.S. Pat. 3,394,-330, July 23, 1968

## DEFENSE COMMUNICATIONS SYSTEMS DIVISION

**Vehicle Identifier System**—J. W. Daniel (DCSD, Cam) U.S. Pat. 3,399,405, August 27, 1968

**Coil Winding Apparatus**—H. E. Haslau (DCSD, Cam) U.S. Pat. 3,392,760, July 16, 1968

**Means for Servicing a Plurality of Data Buffers**—R. S. Klein (DCSD, Cam) U.S. Pat. 3,395,398, July 30, 1968

## DEFENSE MICROELECTRONICS DIVISION

**Gated Flip-Flop Employing Plural Transistors and Plural Capacitors Cooperating to Minimize Flip-Flop Recovery Time**—E. K. C. Yu (DME, Som) U.S. Pat. 3,398,-300, August 20, 1968

## INFORMATION SYSTEMS DIVISION

**Coupling Circuit**—C. M. Wright (ISD, Cam) U.S. Pat. 3,388,265, June 11, 1968

**Tape Reel Latch**—J. M. Uritis (ISD, Cam) U.S. Pat. 3,383,067, May 14, 1968

**Multistage Amplifier Circuitry Used in Conjunction with High Speed Digital Computer Memories**—T. R. Mayhew (ISD, Cam) U.S. Pat. 3,383,666, May 24, 1968

**Mechanical Movement**—W. Deighton (ISD, Cam) U.S. Pat. 3,386,742, June 4, 1968

**Filter Circuit**—J. M. Bailey (ISD, Pennsauken) U.S. Pat. 3,394,346, July 23, 1968

**Micro-magnetic Grooved Memory Matrix**—L. Wu (ISD, Cam) U.S. Pat. 3,395,403. July 30, 1968

**Time Delay Circuit Employing Logic Gate**—A. Sheng, E. Hebert (ISD, Cam) U.S. Pat. 3,396,282, August 6, 1968

**Circuit for Generating Two Consecutive Same-Duration Pulses, each on Separate Output Terminals, regardless of Triggering Pulse Duration**—J. A. Vallee (ISD, W. Palm) U.S. Pat. 3,393,367, July 16, 1968

**Printed Circuit Connector**—S. M. Shelley (ISD, Cam) U.S. Pat. 3,393,392, July 16, 1968

## ADVANCED TECHNOLOGY

**Tape Transport Drive Means**—L. H. Fulton (AT, Cam) U.S. Pat. 3,396,890, August 13, 1968

**Direct Current Electrical Neuron Circuit**—T. B. Martin, E. P. McGrogan (AT, Cam) U.S. Pat. 3,394,266, July 23, 1968

**Transient Signal Analyzer Circuit**—G. J. Dusheck (AT, Cam) U.S. Pat. 3,394,309, July 23, 1968

**Logic Circuits**—T. B. Martin (AT, Cam) U.S. Pat. 3,394,351, July 23, 1968

## ELECTROMAGNETIC AND AVIATION SYSTEMS DIVISION

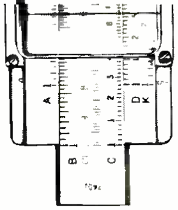
**Card System**—A. Gattuso (EASD, Van Nuys) U.S. Pat. 3,392,734, July 16, 1968

**Overdrive Circuit for Inductive Loads**—C. R. Corson, P. Hoffman (EASD, Van Nuys) U.S. Pat. 3,396,314, August 6, 1968

**Information Storage and Retrieval**—E. H. Irasek (EASD, Van Nuys) U.S. Pat. 3,394,-247, July 23, 1968

## RCA VICTOR CO. LTD.

**Digital Storage and Generation of Video Signals**—R. J. Clark (Ltd, Montreal) U.S. Pat. 3,388,391, June 11, 1968

# Engineering  News and Highlights

## Dr. Brown and Dr. Hillier Named to New Executive Positions

**Dr. George H. Brown** has been named Executive Vice President, Patents and Licensing, with responsibility for RCA's patent operations and for the company's worldwide licensing and technical aid activities.

**Dr. James Hillier** has been appointed Vice President, Research and Engineering with responsibility for research and development throughout the corporation.

In announcing these two appointments, RCA President **Robert W. Sarnoff** said: "We anticipate continuing rapid change in all aspects of electronic technology, affecting every major area of RCA's business at home and overseas. This new technical executive alignment should provide increased management attention and direction for our internal research and engineering activities and our external programs involving technical relations with the industry here and abroad, by separating the two areas and placing each under an outstanding scientist and technical executive.

"We expect this change to enhance RCA's ability to generate, use, and market new technology, and we believe it will give the company far greater flexibility in responding to future opportunities for growth and diversification."

Dr. Brown, whose contributions to communications technology have won international recognition, has headed RCA's research and engineering programs since 1961 and was elected an Executive Vice President in 1965. Dr. Hillier has won recognition both for his research and engineering activities and his role as a manager of research. He is a pioneer in electron microscopy and has headed RCA's central research organization as Vice President, RCA Laboratories, since 1958.

## Dr. William M. Webster to Head RCA Laboratories

**Dr. William M. Webster**, research executive and a leader in the field of solid-state electronics, has been appointed to head RCA's central research organization as Staff Vice President, RCA Laboratories.

Dr. Webster, who has been Staff Vice President, Materials and Device Research, will be responsible for all of the research activities of RCA Laboratories, which has its principal facilities at the David Sarnoff Research Center, Princeton, N.J. He will report to Dr. Hillier, Vice President, Research and Engineering.

Dr. Webster is widely known for his work in semiconductor and gaseous electronics. He was graduated from Union College in 1945 and received the PhD in physics from Princeton University in 1954. He joined RCA Laboratories in 1946 as a specialist in vacuum and solid-state electronics, and subsequently made a number of significant contributions to tube and transistor development. From 1954 to 1959, he was Manager, Advanced Development for the RCA Semiconductor and Materials Division. In 1959, he returned to RCA Laboratories as Director, Electronic Research Laboratory, and was promoted to Staff Vice President, Materials and Device Research, in 1966. He holds a number of patents relating to television, vacuum tubes, gas tubes, circuitry, and semiconductor devices, and his studies and writings have made notable contributions to the understanding of semiconductor device operation. Dr. Webster is a Fellow of the Institute of Electrical and Electronics Engineers and a member of Sigma Xi.

Contents: September 1968 *RCA Review*
Volume 29 Number 3

The *RCA Review* is published quarterly. Copies are available in all RCA libraries. Subscription rates are as follows (rates are discounted 20% for RCA employees):

| | DOMESTIC | FOREIGN |
|---|---|---|
| 1-year | $4.00 | $4.40 |
| 2-year | 7.00 | 7.80 |
| 3-year | 9.00 | 10.20 |

## Parker Named Chief Engineer of DCSD

**S. N. Lev,** Division Vice President and General Manager, Defense Communications Systems Division, has named **D. J. Parker** Chief Engineer to replace **C. K. Law** who was appointed Manager of Technical Planning, Defense Engineering. Mr. Parker was formerly Manager, Advanced Technology, Defense Engineering.

Mr. Parker received the BS degree in Optics from the Institute of Optics, University of Rochester, N.Y. in 1950. He joined the Applied Research Department of RCA after graduation, becoming supervisor of the Optics Group in 1954 and Manager of Applied Research in 1963. He was promoted to Manager of Advanced Technology in 1967. Mr. Parker is a member of the Optical Society of America, the Society of Photographic Engineers, the American Physical Society, the Society of Motion Picture and Television Engineers, and is a senior member of the IEEE.

## Vollmer Fills Post Vacated by Parker

**D. Shore,** Chief Defense Engineer has appointed **Dr. James Vollmer,** Manager, Advanced Technology to replace **D. J. Parker** who became Chief Engineer of DCSD. As Manager of Advanced Technology, Dr. Vollmer has charge of a group of 120 engineers and physicists charged with translating the recent advances of basic research into useful techniques and devices.

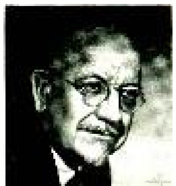
Dr. Vollmer received the BS in General Science at Union College in 1945, an MA and PhD in Physics at Temple University in 1951 and 1956, respectively. His research interests, publications, and patents have covered a wide variety of fields, ranging from infrared properties of materials to plasma physics to quantum electronics. His professional experience includes, in order, five years of teaching at Temple University, eight years of supervising a research group at Honeywell, Inc., and nine years of research supervision at the Radio Corporation of America. Dr. Vollmer is a Fellow of the AAAS, a senior member of the IEEE, and a member of the American Physical Society. His honors include membership in Phi Beta Kappa, Sigma Xi, and Sigma Pi Sigma. He is currently listed in American Men of Science, Who's Who in the East, and Leaders in American Science.

## Schneider Named Chief Engineer of RCA Communications, Inc.

**H. R. Hawkins,** President of RCA Communications, Inc. has named **Philip Schneider,** Chief Engineer. Mr. Schneider was formerly Manager of Advanced Telecommunications Systems for DCSD.

Mr. Schneider received the AB in electrical engineering in 1949 from Columbia College, the BSEE in 1950, and the MSEE in 1952 from Columbia University. He also attended Harvard University's Graduate School of Business Administration, completing a program for management development. From 1952 to 1959, Mr. Schneider worked for Bell Telephone Laboratories where he advanced to Supervisor of Systems Engineering. After serving as Director, Systems Development for the Teleregister Corporation from 1959 to 1961, Mr. Schneider joined RCA as Manager, Advanced Switching Systems and Techniques. Mr. Schneider is a member of the IEEE Switching Committee, Tau Beta Pi, Sigma Xi, and RESA.

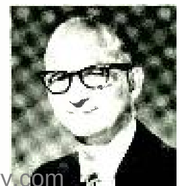
Dr. G. H. Brown



Dr. J. Hillier

Dr. W. M. Webster

D. J. Parker



Dr. J. Vollmer

P. Schneider

## Staff Announcements

### Patents and Licensing

**Dr. G. H. Brown,** Executive Vice President, Patents and Licensing announced the organization of Patents and Licensing as follows: **M. E. Karns,** Vice President, Patents and Licensing; **H. W. Leverenz,** Staff Vice President; **J. Epstein,** Administrator, Staff Services; **H. R. L. Lamont,** Director, European Technical Relations.

### Research and Engineering

**Dr. J. Hillier,** Vice President, Research and Engineering, has appointed **W. M. Webster,** Staff Vice President, RCA Laboratories; **W. C. Morrison,** Staff Vice President, Corporate Engineering Services; **A. N. Curtiss,** Staff Vice President, Administration, Research and Engineering; **J. Hillier,** Acting Director, Advanced Technical Planning; **R. H. Edmondson,** Staff Engineer, Product Engineering; **E. W. Herold,** Staff Engineer, Product Engineering; **H. Kihn,** Staff Engineer, Product Engineering; **E. M. Leyton,** Staff Engineering, Product Engineering; **H. Rosenthal,** Administrator, Staff Services.

**W. M. Webster,** Staff Vice President, RCA Laboratories, announced the organization of the Laboratories as follows: **J. A. Rajchman,** Staff Vice President, Data Processing Research; **F. D. Rosi,** Staff Vice President, Materials and Device Research; **T. O. Stanley,** Director, Systems Research; **A. A. Barco,** Staff Advisor; **R. E. Quinn,** Manager, Technical Administration.

### Information Systems Division

**J. R. Bradburn,** Executive Vice President, Information Systems, has appointed **A. W. Carroll,** Division Vice President, Systems Programming; and **F. M. Hoar,** Division Vice President, Advertising and Public Affairs.

### Consumer Products and Components

**D. L. Mills,** Senior Executive Vice President, Consumer Products and Components, announced that the Magnetic Products Division will become associated with Electronic Components. **J. Stefan** will continue as Division Vice President and General Manager of the Magnetic Products Division and will report to **J. B. Farese,** Executive Vice President, Electronic Components.

### Electronic Components

**J. B. Farese,** Executive Vice President, has announced the organization of Electronic Components as follows: **G. C. Brewster,** Manager, Operations Planning and Support; **C. E. Burnett,** Division Vice President and General Manager, Solid State and Receiving Tube Division; **M. J. Carroll,** Manager, Equipment Marketing Relations; **J. T. Cimorelli,** Division Vice President and General Manager, Memory Products Division; **W. C. Dove,** Purchasing Agent; **G. W. Duckworth,** Division Vice President, Equipment Sales; **A. M. Durham,** Manager, News and Information; **A. M. Glover,** Division Vice President, Technical Programs; **J. A. Haimes,** Division Vice President, Distributor Products; **L. A. Kameen,** Manager, Personnel; **J. Koppelman,** Controller, Finance; **C. H. Lane,** Division Vice President and General Manager, Industrial Tube Division; **W. H. Painter,** Division Vice President, Electronic Components International Operations; **H. R. Seelen,** Division Vice President and General Manager, Television Picture Tube Division.

**A. M. Glover,** Division Vice President, Technical Programs has appointed **J. F. Wilhelm** as Manager Commercial Engineering.

**J. F. Wilhelm,** Manager, Commercial Engineer has appointed **A. P. Sweet,** Manager, Industrial Products and Picture Tubes Commercial Engineering.

### Consumer Products and Components

**A. Mason,** Chief Engineer has appointed **R. Guenther** as Staff Technical Advisor.

### RCA Service Company

**Robert W. Sarnoff,** President has announced that Random House Inc., will report to **A. L. Conrad** as Vice President, Education Systems. The Education Systems organization will be responsible for overall planning and coordination of all RCA education activities. In addition, this organization will contain the following operating units: RCA Institutes, Inc.; Instructional Systems (Palo Alto); the education activities of the Instructional and Professional Electronic Systems Dept. now in the Commercial Electronic Systems Division. **C. V. Newsom,** Vice President, Education, will continue as RCA's principal corporate officer and **T. A. Smith,** Executive Vice President, will continue to devote his full attention to the field of education.

**C. M. Odorizzi,** Senior Executive Vice President, Services has appointed **E. H. Griffiths,** President and **S. D. Heller,** Division Vice President, Operations.

### Staff

**T. G. Paterson,** Manager, Systems Development, has appointed **R. H. Baker,** Manager, Walt Disney World Project.

**G. A. Fadler,** Vice President, Manufacturing Services and Materials has announced the General Product Assurance function is transferred from Product Engineering to Manufacturing Services and Materials. **H. E. Schock** will continue as Administrator, General Product Assurance.

**B. V. Dale,** Manager, Automatic Test and Measurement Systems has appointed **M. H. Lazar** to the newly created position of Manager, Systems Programming.

**Chase Morsey, Jr.,** Vice President Marketing announced that the Special Development Projects activity is transferred from the Laboratories to the Marketing organization. **L. R. Day** will continue as Director, Special Development Projects, and will report to the Vice President, Marketing. **W. H. Enders** is appointed Director, Advanced Product Planning.

---

**Errata:** In Vol. 13, No. 4 (Dec. 1967, Jan. 1968), p. 72, Mr. R. J. Gildea (formerly with RCA Aerospace Systems Division and presently with the Mitre Corporation) should have been included as a co-author of "Computer-Aided Selection of Test Points and Fault Isolation Procedures" by R. S. Fisher and F. R. Hawke.

In Vol. 14, No. 2 (Aug./Sept. 1968) pp. 10 and 13, the diagrams in Figs. 1 and 2 of Mr. Turkington's paper should be interchanged.

---

## Degrees Granted

| | |
|---|---|
| **Walter G. Gibson,** Labs., Pr. | MSEE, Newark College of Engineering, 6/68 |
| **I. Cherkas,** ISD, Camden | MSEE, University of Pennsylvania, 6/68 |
| **Robert L. Bailey,** Lancaster | MS, Physics, Franklin and Marshall, 6/68 |
| **Irving E. Martin,** Lancaster | MS, Physics, Franklin and Marshall, 6/68 |
| **Don R. Carter,** Lancaster | MS, Physics, Franklin and Marshall, 6/68 |
| **Augutus D. Saxton,** Lancaster | PhD, Chemistry, Clarkson College, 6/68 |
| **Donald R. Tshudy,** Lancaster | MS, Chemistry, Franklin and Marshall, 6/68 |
| **Francis E. Geiger,** Lancaster | BS, Physics, Franklin and Marshall, 6/68 |

---

**DCSD Best paper award:** The first winner of Defense Communications Systems Division's Best Paper Award is John D. Rittenhouse of Magnetic Recording Engineering in Camden. Rittenhouse (right) accepts award from DCSD Chief Engineer, C. K. Law. (Mr. Law has since joined DEP Staff, Moorestown.) Award of certificate and savings bond was for the technical paper "Rotary Head Instrumentation Recorders for Telemetry Recordings." DCSD will give the award quarterly and annually. Winners will also be honored at DCSD annual authors banquet.

### Dr. Max to Retire

**Dr. A. M. Max,** Manager of the Chemical and Physical group of Record Engineering Laboratory is taking early retirement and will assume the position of Professor of Mechanical Engineering at Purdue University, Indianapolis Regional Campus. He will continue his association with Record Engineering in an advisory capacity.

Dr. Max was born in Sheboygan, Wis., attended the University of Wisconsin receiving his BS in 1934, MS in 1935 and PhD in 1937 for work in corrosion and inhibitors under the late Prof. O. P. Watts. From 1937 to 1941, Dr. Max was a chemist with the Ternstedt Div. of General Motors Corp. in Detroit. In 1941-44 he was assistant Professor of Chemical Engineering at the University of North Dakota. Since 1944, Dr. Max has been with the Engineering and Development section of the Record Division in Indianapolis. In 1948, he was awarded the RCA Award of Merit for the development of matrix processing methods which contributed materially to improved record quality and helped to make high fidelity records possible. His continued efforts in the field of high speed electroforming of copper, iron and nickel have paced the technology and made possible nickel electroforming at 600 ASF with excellent mechanical properties. Dr. Max has been associated with the RCA ENGINEER since its inception and has also served as Technical Publications Administrator for the Record Division. Dr. Max was instrumental in helping to plan and expedite two issues devoted to audio, tape, and record engineering.

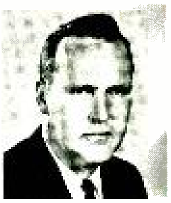

Dr. A. M. Max


C. Hoyt

### C. Hoyt is TPA for CED

**Clyde Hoyt** recently was named Chairman of the Editorial Board and TPA for the Consumer Electronics Division to replace **Ken Chittick.** Mr. Hoyt will be responsible for the review and approval of technical papers, and for promoting the preparation of papers for the RCA ENGINEER and other journals, both internal and external.

Mr. Hoyt is a graduate of Morningside College and Iowa State University in Arts and Electrical Engineering. He has been actively engaged in many phases of Home Instruments engineering; his work in television dates back to the introduction of the 630TS in 1946. In recent years he has been Staff Engineer for Consumer Electronics. He is presently the Manager, General Engineering Administration. Mr. Hoyt holds approximately 24 patents, primarily in the TV area.


R. J. McLaughlin


S. B. Ponder

### New Ed Reps for Information Systems Division—McLaughlin, Moffa, and Ponder

To assist **Murray Kaminsky** in his duties as Technical Publications Administrator for the Information Systems Division, **R. J. McLaughlin** will represent the Marlboro, Mass. activity (presently located at Framingham, Mass.); **S. B. Ponder** will represent the Palm Beach, Fla., activity; and **M. Moffa** will represent the Camden, N.J. area. Messrs. **McLaughlin, Moffa,** and **Ponder** will be responsible for planning and processing articles for the RCA ENGINEER.

Mr. McLaughlin received the BS in Business Administration from Northeastern University in 1960 and will shortly receive his MBA. He joined RCA in 1961 as a Technical Recruiter with Aerospace Systems Division. In 1963, he was appointed Engineering Administrator for the Systems Support Engineering product lines, and in 1966 he became Leader, Engineering Administration in support of Automatic Test Equipment Engineering, Systems Support Engineer, and Engineering Controls and Support. In April of this year, he joined the Information Systems Division's new Peripheral Equipment facility in Marlboro, Massachusetts as Manager, Engineering Services.

Mr. Ponder received the BS in Education from Indiana University and did graduate work at the University of Minnesota. After working for the Allison Division of General Motors Corp. and for Minneapolis Honeywell, Mr. Ponder joined RCA in 1961. He is presently Manager of Special and International accounts.

## Professional Activities

#### Information Systems Division

Recognition for contributions advancing the state of the art has been accorded to engineers on the staff at Palm Beach Gardens: **R. Taynton, B. Glass, T. Floyd, J. Watson, C. Miller, E. Nelson, T. Sariti, T. Prieto, B. Salzer, C. James, D. Hall, B. Peyton, A. Turecki, J. Schell.**

**B. W. Pollard,** Camden, has been elected to the Mid-Eastern Area Committee of the IEEE Computer Group; he was also a speaker at the Second Annual IEEE Computer Group Conf. held in June at Los Angeles, Calif. **N. Garaffa** served as Secretary of the Fifth Annual Design Automation Workshop sponsored by the ACM-IEEE. Mr. Garaffa was also elected Chairman of next year's conference. At the same conference, **Sam Heiss** of Palm Beach presented a paper.

#### Astro-Electronics Division

**Robert Feuchtbaum** has been appointed to the Steering Committee for the Electrical Insulation Conference — Materials and Application.

#### Laboratories

**Dr. A. G. Revesz** was elected to the membership in the New York Academy of Science and was invited to accept Fellowship in the American Institute of Chemists at the coming meeting in Atlantic City "in recognition of his contributions in inorganic chemistry."

#### Aerospace Systems Division

**F. Gardiner** has accepted a position on the Program Committee of Boston Chapter of IEEE Group on Engineering Management. **Richard J. Geehan,** has been selected as June Engineer of the Month, in recognition of his accomplishments on the main Memory for the Airborne Data Automation (ADA) System.

#### RCA Missile Test Project

The Seventh Annual IEEE Region III Convention will be held in Cocoa Beach, Florida. **A. L. Conrad,** Vice President, is a member of the Executive Advisory Committee and **Denton Clark,** serves as General Chairman. Serving as chairman of the session on computers is **H. N. Morris,** ISD and **Dr. L. E. Mertens,** MTP is co-chairman for technical papers. Delivering technical papers will be **R. W. Avery,** ISD, Palo Alto; **E. T. Johnson,** MTP, Florida; and **A. E. Smith,** MSR, Mrstn.

#### Defense Communications Systems Division

The IEEE Group on Reliability has chosen: **M. Tall,** Mrstn., Chairman; **J. H. Goodman,** Camden, secretary; **J. F. Chalupa,** Camden, Publicity Committee chairman; **R. E. Killion,** Mrstn., Symposium Committee, chairman; **G. Ashendorf,** Camden, Education and Training Committee, chairman; **N. Salatino,** Camden, Membership Committee, chairman; and **G. Hunt,** Camden, Arrangements Committee, chairman.

#### Defense Communications Systems Division

**C. W. Fields,** has been elected to the IEEE Engineering Writing and Speech Group National Administrative Committee. **J. Neubauer,** Chaired IEEE Vehicular Communications Group Standards Committee 16-2, which issued the new "Standard for Testing FM Mobile Communications Receivers." The standard is also being processed for issuance as a USA Standards Institute standard.

#### Electromagnetic and Aviation Systems Division

**Mr. George F. Fairhurst** was elected a Fellow of the Society of Logistics Engineers—in recognition of his contribution to the enhancement of logistics management in the field of project engineering and his service to the Society of Logistics Engineers.

# Editorial Representatives

The Editorial Representative in your group is the one you should contact in scheduling technical papers and announcements of your professional activities.

| | |
|---|---|
| **Defense and Commercial Systems** | |
| **Defense Electronic Products** | |
| **Aerospace Systems Division** | Engineering, Burlington, Mass. |
| **Electromagnetic and Aviation Systems Division** | Engineering, Van Nuys, Calif.<br>Engineering, West Los Angeles, Calif. |
| **Astro-Electronics Division** | Engineering, Princeton, N.J.<br>Advanced Development and Research, Princeton, N.J. |
| **Missile & Surface Radar Division** | Engineering, Moorestown, N.J. |
| **Defense Communications Systems Division** | Engineering, Camden, N.J.<br>Technical Communications, Camden, N.J.<br>Engineering, Camden, N.J.<br>Advanced Communications Laboratory, West Windsor, N.J. |
| **Defense Engineering** | Advanced Technology, Camden, N.J.<br>Defense Microelectronics, Somerville, N.J.<br>Systems Engineering, Evaluation, and Research, Moorestown, N.J.<br>Advanced Technology, Camden, N.J.<br>(Acting) Central Engineering, Camden, N.J. |
| **Commercial Electronics Systems Division** | Chairman, Editorial Board, Camden, N.J.<br>Mobile Communications Engineering, Meadow Lands, Pa.<br>Professional Electronic Systems, Burbank, Calif.<br>Studio, Recording, & Scientic Equip. Engineering, Camden, N.J.<br>Microwave Engineering, Camden, N.J.<br>Broadcast Transmitter & Antenna Eng., Gibbsboro, N.J. |
| **Industrial and Automation Systems** | Engineering, Plymouth, Mich. |
| **Information Systems** | |
| **Information Systems Division** | Engineering, Camden, N.J.<br>Engineering, Camden, N.J.<br>Palm Beach Engineering, West Palm Beach, Fla.<br>Engineering, Marlboro, Mass. |
| **Graphic Systems Division** | Engineering, Dayton, N.J. |
| **Research and Engineering** | |
| **Laboratories** | Research, Princeton, N.J. |
| **Consumer Products and Components** | |
| **Electronic Components** | Chairman, Editorial Board, Harrison, N.J. |
| **Solid State and Receiving Tube Division** | Solid State Power Device Engrg., Somerville, N.J.<br>Commercial Receiving Tube and Semiconductor Engineering, Somerville, N.J.<br>Solid State Signal Device Engrg., Somerville, N.J.<br>Receiving Tube Operations, Woodbridge, N.J.<br>Semiconductor and Conversion Tube Operations, Mountaintop, Pa.<br>Receiving Tube Operations, Cincinnati, Ohio<br>Semiconductor Operations, Findlay, Ohio |
| **Television Picture Tube Division** | Television Picture Tube Operations, Marion, Ind.<br>Television Picture Tube Operations, Lancaster, Pa. |
| **Industrial Tube Division** | Power Tube Operations and Operations Svcs., Lancaster, Pa.<br>Conversion Tube Operations, Lancaster, Pa.<br>Microwave Tube Operations Harrison, N.J. |
| **Memory Products Division** | Memory Products Dept., Needham, Mass. |
| **Technical Programs** | Engineering, Harrison, N.J. |
| **Consumer Electronics Division** | Chairman, Editorial Board, Indianapolis, Ind.<br>Advanced Devel., Indianapolis, Ind.<br>Radio "Victrola" Product Eng., Indianapolis, Ind.<br>TV Product Eng., Indianapolis, Ind.<br>Electromech. Product Eng., Indianapolis, Ind.<br>TV Product Eng., Indianapolis, Ind.<br>Resident Eng., Bloomington, Ind. |
| **Record Division** | Record Eng., Indianapolis, Ind. |
| **Services** | |
| **RCA Service Company** | EDP Service Dept., Cherry Hill, N.J.<br>Consumer Products Service Dept., Cherry Hill, N.J.<br>Govt. Service Dept., Cherry Hill, N.J.<br>Consumer Product Administration, Cherry Hill, N.J.<br>Tech. Products, Adm. & Tech. Support, Cherry Hill, N.J.<br>Missile Test Project, Cape Kennedy, Fla. |
| **RCA Communications, Inc.** | RCA Communications, Inc., New York, N.Y. |
| **New Business Programs** | Engineering, New Business Programs, Princeton, N.J. |
| **National Broadcasting Company, Inc.** | Staff Eng., New York, N.Y. |
| **RCA International Division** | Clark, N.J |
| **RCA Victor Company, Ltd.** | Research & Eng., Montreal, Canada |
| **Education Systems** | |
| **Instructional Systems** | Instructional Systems Engineering, Palo Alto, Cal. |

* **Technical Publication Administrators**