THE BELL SYSTEM TECHNICAL JOURNAL
OCTOBER 1974
VOL. 53 NO. 8

LAMP: LOGIC ANALYZER FOR MAINTENANCE PLANNING

THE BELL SYSTEM TECHNICAL JOURNAL

THE BELL SYSTEM TECHNICAL JOURNAL is published ten times a year by the American Telephone and Telegraph Company, J. D. deButts, Chairman and Chief Executive Officer, R. D. Lilley, President, J. J. Scanlon, Executive Vice President and Chief Financial Officer, F. A. Hutson, Jr., Secretary. Checks for subscriptions should be made payable to American Telephone and Telegraph Company and should be addressed to the Treasury Department, Room 1038, 195 Broadway, New York, N. Y. 10007. Subscriptions $15.00 per year; single copies $1.75 each. Foreign postage $1.00 per year; 15 cents per copy. Printed in U.S.A.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 53 October 1974 Number 8

LAMP:

System Description

By H. Y. CHANG, G. W. SMITH, Jr., and R. B. WALFORD

(Manuscript received February 28, 1974)

A general description of the Logic Analyzer for Maintenance Planning (LAMP) system is presented. LAMP is a digital-logic simulation and analysis system used for logic-design verification, for generation and evaluation of fault-detection and diagnostic tests, and for generation of the trouble-location manual (or fault dictionary) data. It is implemented on the IBM 360/370 TSS and OS machines (for both interactive and batch operations), and has been in active use at Bell Laboratories in the development of electronic switching systems, data set facilities, transmission equipment, and advanced integrated circuit technologies.

I. INTRODUCTION

The explosive evolution of digital devices, computers, and systems since the invention of the transistor has necessitated a parallel industry-wide development of tools for the design and test of logic circuits. Whereas the oscilloscope was the mainstay of the digital circuit designer in the early days of discrete-transistor logic circuits, it soon proved to be inadequate for design verification and fault-behavior testing of large systems employing integrated, digital logic. In response to this need for better logic-circuit-development tools, a host of digital-simulator algorithms and simulator systems has been produced.[1–3]

The need for reliable and dependable electronic switching systems (ESS) poses critical design problems. Computer-aided techniques can be used effectively for:

(*i*) Analysis and enhancement of system diagnosability.
(*ii*) Logic-design verification.
(*iii*) Generation of fault-detection tests.
(*iv*) Analysis of faulty-circuit behavior.
(*v*) Verification and evaluation of diagnostic-test designs.
(*vi*) Production of trouble-location manuals (TLMs).

The LAMP system has been designed to attack these problems in a systematic manner.

This paper provides a brief description of the LAMP system organization and features, and is intended to serve as background for the four following papers. These provide details of the logic simulators, the automatic-test-generation system, and the techniques for organizing system design for diagnosability.[4–6] They include a specific example of how LAMP was employed in the development of a large processor for an electronic switching system.[7]

II. EVOLUTION OF THE LAMP SYSTEM

The decision to build a machine-aids system with digital-simulation capability was motivated by the successful use by Bell Laboratories designers of the sequential analyzer.[8] The use of this simulator showed the great advantages of using simulation for logic testing and fault diagnosis. By 1966, Bell Laboratories was incorporating simulation techniques into the design cycle of electronic-switching-system equipment. However, there were several difficulties in the day-to-day use of this simulator. It had a restrictive logic model, long turnaround time due to remote computer location, and no capability for handling large circuits (for example, circuits having as many as 10,000 gates). Because no simulator then available could meet the growing demand for logic-simulation service, a decision was made to develop an advanced logic-simulator system which would grow and adapt to Bell Laboratories current and future needs.

It is instructive that the motivation to develop a design-aids system came from the potential users of that system. Likewise, the initial design objectives and the evolution of the system were influenced to a large extent by the intended users. This has resulted in a very sophisticated, user-oriented system which continues to grow and evolve to meet the changing requirements of the designer.

The initial system was made available to users in late 1969 on IBM System/360 TSS at Bell Laboratories, Naperville, Illinois. It had only a modest set of features. However, the user reactions were generally favorable. Since then, substantial improvements in system performance and capabilities have been incorporated. The TSS version of LAMP was converted to run on IBM System/360 OS in mid-1970 and was made available to Bell Laboratories users at Holmdel, New Jersey, and Columbus, Ohio. Automatic-test-generation capability was incorporated in early 1972; and the facilities to analyze system structural diagnosability were implemented in late 1972. The LAMP system is in active use in the development of many ESS projects as well as other non-ESS work such as the development of data-set facilities, transmission equipment, and advanced integrated-circuit technologies. The current user group includes twenty organizations from nine Bell Laboratories locations (Murray Hill, Whippany, Holmdel, Allentown, Columbus, Merrimack Valley, Indianapolis, Denver, and Naperville).

III. SYSTEM ORGANIZATION

LAMP is a system of programs designed to be used for logic-design verification, evaluation of fault-detection tests and diagnostic programs, and generation of the trouble-location manual (or fault dictionary) data. It is implemented on the IBM 360/370 TSS and OS machines (for both interactive and batch operations). The current version can handle circuits containing up to 65,000 gates. The system is composed of four basic parts:

(*i*) A circuit-description-language compiler.
(*ii*) A command-language interpreter.
(*iii*) A collection of design tools composed of an automatic-test-generation (ATG) system;[4] a controllability, observability, and maintenance engineering techniques (COMET[5]) system; and a family of simulators.[6]
(*iv*) An output system.

A block diagram showing the functional relationship of the various parts of the LAMP system is presented in Fig. 1. A logic circuit can be described to the LAMP system through a user-oriented language called *LSL-LOCAL*. The circuit description is then translated by the language compiler into *simulation tables*. The *command-language interpreter* directs all the actions of simulation, test generation, and diagnosability analysis in accordance with user-specified commands and information stored in the simulation table.

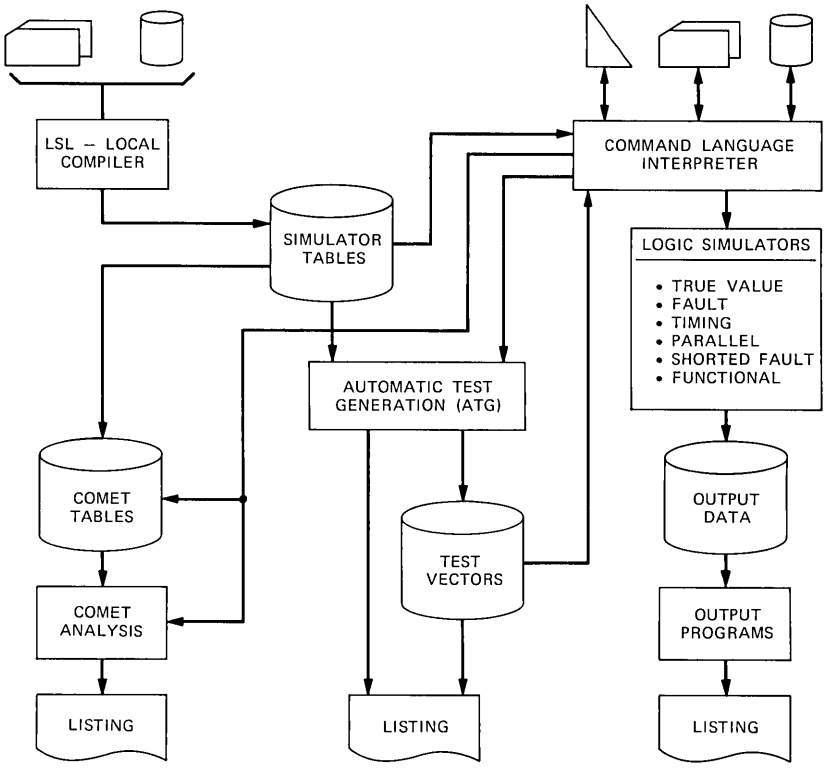


Fig. 1—Block diagram of LAMP system.

For a given logic-circuit description, the ATG system can automatically produce the test-vector information. To verify logic design and to study faulty-circuit behavior, a family of *simulators* can be used. The inputs applied to the simulators can be manually generated and/or generated by the ATG programs. The simulators are capable of simulating circuit behavior in either fault-free or faulty mode, with facilities to handle race and oscillation conditions and to perform detailed timing analysis.

If the purpose at hand is to determine the diagnosability of the design, the *COMET system* can be used to assist the users to organize systems design for diagnosability by systematically determining the optimum placement of control-access and monitor points. Simulation and analysis results are then collected under the control of an *output system*. Numerous output options can be specified that allow users to obtain information concerning logic verification, timing analysis,

and other data-processing information at the time of simulation or afterwards.

In the following sections, the salient features of the various major functional blocks in the LAMP system will be described.

3.1 Circuit description input language

A logic circuit is described to the LAMP system through a user-oriented language called LSL-LOCAL. This language permits the entry of all information concerning the particular circuit either at the gate level or at the functional level. At the gate level, circuits are described in terms of logic elements such as NANDS, NORS, ANDS, ORS, and NOTS, whereas the functional level the circuits are expressed as memories, registers, clocks, etc. LSL-LOCAL was designed as an easily extendible language, to be used by circuit designers and diagnosticians who may not be trained as programmers.

Once the circuit description is entered, the LSL-LOCAL language processor compiles the description into data tables to be used by the simulator(s), the ATG system, and the COMET analysis programs. The language processor has a substantial number of checks built into it to detect and intercept most errors before they can get into the system. These checks include syntax checks (for missing parameters, illegal characters, etc.) and circuit connectivity and consistency checks such as fan-in/fan-out limits. These features enable the users to check the coding of a circuit efficiently in terms of cost and time.

The original version of the language processor was developed in late 1969. Since then, three major revisions have been implemented to enhance its capability and performance. Many of the improvements were incorporated to support a wider range of applications, and the language has become a standard logic design input language in Bell Laboratories.

As an example of the LSL-LOCAL circuit description, an exclusive-OR circuit as shown in Fig. 2a can be encoded as:

```
CKTNAME:  XOR;
  INPUTS:  A, B;
 OUTPUTS:  X;
     NOT:  A′, A;
           B′, B;
    NAND:    AB′,        (A, B′);
             BA′,        (B, A′);
             A × B,      (AB′, BA′),    (X);
           (gate name)   (input list)   (output)
```

The description generally consists of three parts: (*i*) the CKTNAME statement, which introduces the circuit description and declares the name of the circuit; (*ii*) connection declarations, which specify the names and the types of all the input/output signals of the circuit; and (*iii*) interconnection blocks, which specify elements and networks used in the circuit and how these are interconnected. The hierarchical structure of the language allows the specification of circuits in a modular fashion. Thus, the exclusive-OR circuit can be used as an element in describing a single-bit adder [see Fig. 2(b)]:

```
CKTNAME:  ADDER1;
  INPUTS:  A, B, K;
 OUTPUTS:  C, K_;
     XOR:  A × B, (A, B), (X);
           D, (X, K), (C);
     NOT:  A′, A;
           B′, B;
    NAND:  ANB, (A, B);
           AORB, (A′, B′);
           AORBNK, (AORB, K);
           K_, (ANB, AORBNK);
```

These single-bit circuits can then be used to describe an n-bit adder or other more complex logic element(s). There is no explicit limit on the number of levels of nesting in describing circuits using LSL-LOCAL. A user can very conveniently construct the data base of a large circuit or system by combining the various data bases from its component circuit modules.

3.2 Command system

The control of LAMP system action for test generation, simulation, and COMET analysis is accomplished by means of a command-language structure. A large selection of interactive commands is available which enables the users to compile and edit a circuit description, specify simulation-test vectors, make simulation runs, observe circuit behavior, gather circuit statistics, determine optimal placement of maintenance-access and observation points to enhance diagnosability, and specify types of simulation and analysis output. At present, there are approximately 80 commands in the system, many of which were implemented at the request of users. The commands are highly user-oriented so that one can easily learn the use of the system after a relatively minor amount of instruction.

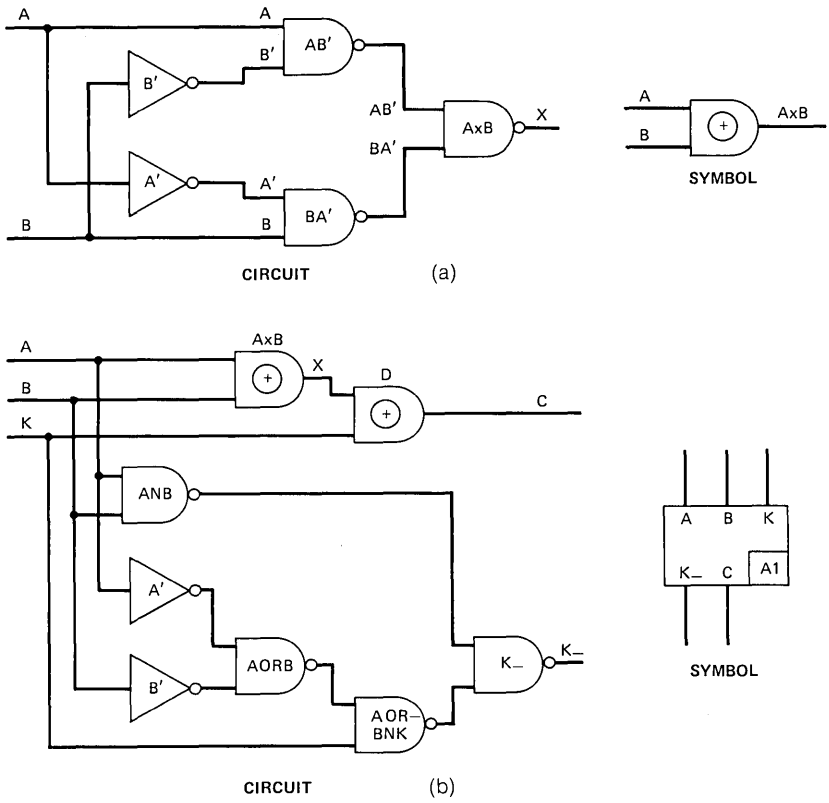


Fig. 2—(a) Exclusive-OR circuit. (b) One-bit adder circuit.

The system structure is implemented with four levels of hierarchy. On the base level is the executive routine which reads commands entered by the system user and interprets them as to type. It then calls the appropriate routine to handle the command. On the next level are the command handlers whose functions are to process the command line and call the appropriate functional processors and service routines. On the third level are the functional processors; they are designed to perform specific functions such as simulation, circuit-description and test-vector compilation, circuit modification, processing control, and interactive control. On the fourth level are the various service routines that perform such tasks as gate-name retrieval, print control, vector translation, preliminary processing of data lines, file accessing, and printing.

To illustrate the richness of the command language, a few of the most commonly used commands for logic simulation are described. Referring to Fig. 3, to enter circuit descriptions into LAMP, the LSL-LOCAL encoding of the circuit will be first compiled (using SOURCE) and the resultant simulator tables loaded (using LOAD). A circuit can also be formed by combining several circuits into one using LINK. Should it become necessary to modify the circuit logic without recompiling the entire circuit, then CKTCHANGE can be used to connect/disconnect gates, add gates, and rename, change, or remove gates.

The input test vectors for simulation can be described in either trinary (0, 1, and "don't know"), octal, or hexadecimal form (using INVEC), or in terms of vector names defined by PATTERN. In certain applications, the STATE command is used to set the circuit-state variables to initialize a circuit before a simulation run. Internal gates of the circuit can be treated as additional circuit outputs or test points by issuing the MONITOR command. Conversely, normal circuit-output leads can be MASKED out for a particular run.

The what, when, and how much of the simulation statistics that are to be processed after a run are defined through RESULTS. A simulation is initiated by the RUN command and can be temporarily halted by a STOP command. At a STOP, the user may interrogate the state of the simulation and obtain simulation statistics accumulated up to that point (by using the DISPLAY command), or he may overwrite the next input vector in the INVEC data set by issuing an ALTER command. The simulation can be resumed by issuing a GO command. If the user wishes to change the course of simulation during a STOP, he can use the JUMP feature to skip unwanted test vectors.

To facilitate the use of the LAMP system in the production mode, many commands have been developed for analyzing circuit topology, gathering circuit statistics, and performing audits. Some examples are the CKTCHECK command to check the consistency of simulation tables and to provide statistical information such as counts of gate and functional types, average fan-in and fan-out for each type, percentages of types to total, etc., and the CKTSTAT command which prints a brief summary of circuit statistics including number of gates, number of circuit inputs, number of circuit outputs, number of clocks, and number of nonfaulted gates. For topological analysis, the LOOPS command allows one to identify all loops within a circuit or contained by a specified gate, the FEEDBACK command identifies the minimum number of feedback loops within the circuit, the PATH command finds the shortest path between a specified gate and any input, and the MSC

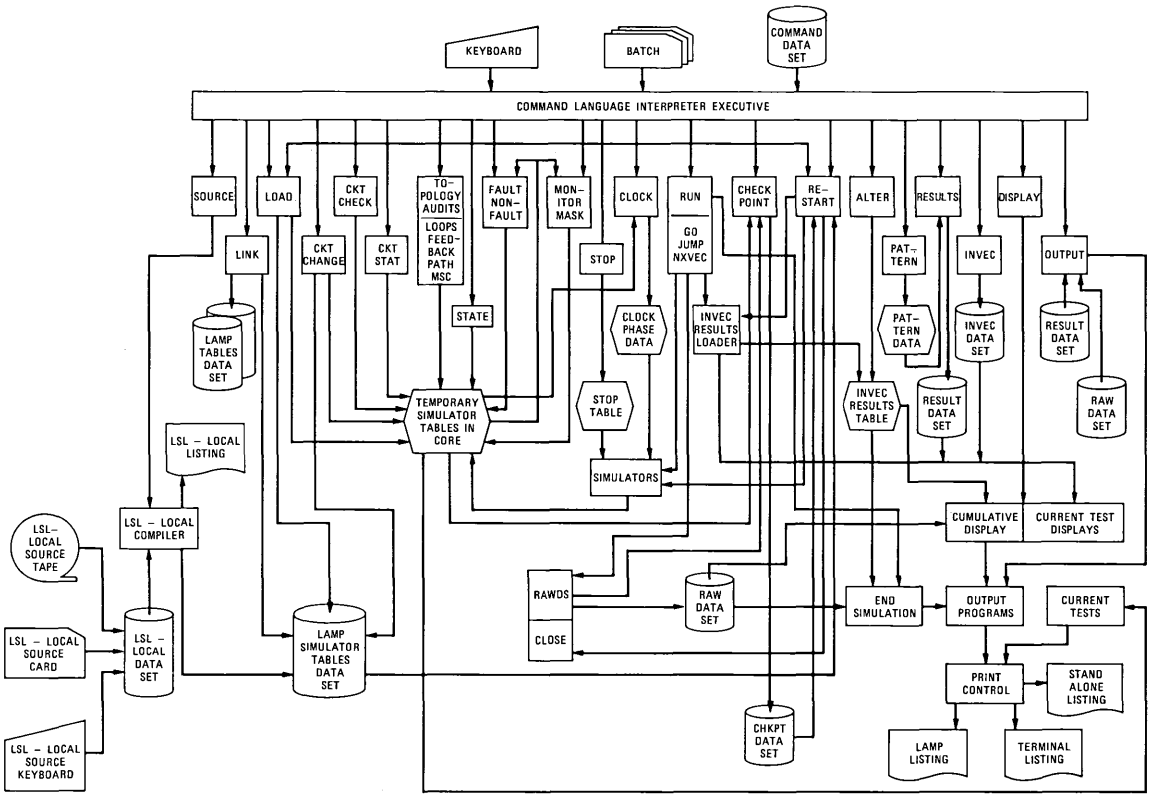


Fig. 3—Examples of commands used in simulation.

command identifies all maximally strongly connected sets of gates within the circuit. All these commands have been proven to be extremely useful, especially in the course of simulating large circuits (e.g., those containing 50,000 gates) under fault conditions.[7]

While the LAMP commands generally assume interactive use of the system (on 360/TSS), they also permit the use of the system in the noninteractive mode (such as 360/TSS batch or 360/370 OS). In these cases, some advance planning must be done to enable the runs to be completed successfully.

3.3 Major tools

There are three major tools in the LAMP system: an automatic-test-generation (ATG) system, a family of simulators, and a system for diagnosability analysis (COMET). Detailed descriptions of these tools are covered in the companion papers.[4–6] The purpose of this section is to describe the salient features of these systems and to briefly describe the interactions among them and the rest of the LAMP system.

3.3.1 Automatic-test-generation (ATG) system

ATG is a system of programs that can automatically produce the test-vector information for a given logic-circuit description. The faults considered are the classical input open, output stuck-at-one, and output stuck-at-zero for each gate in the circuit. There are two major differences between ATG and those test generators that have been discussed in the literature.[9,10] First, ATG is capable of handling both combinational and sequential circuits without the need to identify feedback lines. Second, the system treats logic circuits as an interconnection of unit- and zero-time-delay gates, and thus improves the accuracy of the circuit modeling.

ATG interacts with other parts of the LAMP system via the command-language interpreter (see Fig. 1). A set of about 20 commands is available to the user to set the initial conditions (e.g., loading the circuit description, specifying sequence length of the test), select test-generation strategies, specify output procedure, and direct the general course of action. The fault-detection level achieved by the tests generated by ATG can be evaluated by using the fault simulators available in the LAMP system. If the evaluation results indicate that the detection level is not adequate, ATG can be called again to generate more tests, by using different test strategies and/or changing the sequence length of the tests. This test-generation and evaluation loop

can be repeated several times until a specified level of detectability is achieved.

3.3.2 Controllability, Observability, and Maintenance Engineering Techniques (COMET)

Past experience has indicated that the effectiveness of diagnostic testing depends not merely on the techniques used in deriving tests and test results, but also on the inherent structural diagnosability of the unit.* The ATG system is a tool for aiding the derivation of test vectors for given circuits. The COMET system, on the other hand, employs a technique that enables one to determine for a given circuit the optimal placement of control-access and monitor points for diagnostic testing.

The COMET analysis is initiated by entering the connectivity of the functional blocks of a unit via LSL-LOCAL (see Fig. 1). The control and observation relations among the various functional nodes are automatically created from the connectivity (or simulator) tables. Through the use of the command-language interpreter, the user can then direct COMET to analyze, to examine, and to modify the topological structure of the unit. The modification of the structure for additional control and/or observation is performed automatically, or it can be explicitly directed by the user. Once the design has been COMETized, it enjoys the following advantages:

(*i*) Trouble-location-manual data can be generated and updated without the use of fault simulation.
(*ii*) Multiple faults and all nonclassical faults are locatable if they are detectable.
(*iii*) Diagnostic information can be easily updated in accordance with hardware change(s).
(*iv*) An orderly approach to the implementation of an overall diagnostic design is provided.
(*v*) The fault-location procedure is substantially simplified.

3.3.3 Logic simulators

In the heart of the LAMP system are the logic simulators. These are the programs that actually perform the simulation of the circuit under test. A total of six simulators is available, each of which is designed to

* Depending on the level of integration and the purpose at hand, a unit can be interpreted as a processor, a functional module, a circuit pack, or an LSI chip.

suit a particular condition.* Under the control of the command-language interpreter, one or more of the simulators can be called to simulate a particular circuit. The six simulators available in the LAMP system are:

(*i*) True-value simulator—This simulator simulates only the true-value (or nonfaulted) conditions of the circuit. Simulation is done at the gate level.

(*ii*) Fault simulator—This simulator can simulate the action of classical stuck-at-type faults (input open, output stuck-at-zero, and stuck-at-one) in addition to the true value. This enables one to study the behavior of faulty circuits, to evaluate the fault-detection capability of maintenance-check circuits and tests, and to generate diagnostic data for trouble-location-manual production.

(*iii*) Timing simulator—This simulator allows the specification of individual rise and fall times of all gates in the circuit but does not simulate the effect of the stuck-at faults. It is designed primarily for detailed timing analysis to verify that the circuit will work under worst-case conditions.

(*iv*) Parallel simulator—The features of this simulator are similar to the ones available in the fault simulator. The major difference is that the parallel simulator employs a technique whereby the true value and a small set of faults are simulated concurrently.

(*v*) Shorted-fault simulator—This simulator allows for simulation of nonclassical faults such as crossover shorts and shorts between adjacent paths. It is useful in aiding the design of manufacturing tests for circuit pack check-out.

(*vi*) Functional simulator—This simulator allows one to simulate the circuit behavior at a higher level (e.g., registers, memories, etc.) than at the gate level. Functional simulation is most useful in verifying initial logic design where detailed knowledge of gate-level logic is not available or the function(s) cannot be conveniently modeled by gate-level techniques.

The cost effectiveness of the LAMP system depends on the user's choosing the correct simulator or simulators for use in his application. Consequently, it was found necessary to combine the results of more

* This was found desirable and cost effective especially in a production environment where system performance and accuracy are often weighted against each other in the search for an optimum mix.

than one simulator if the model of one simulator is not sufficient for a particular application. This is accomplished by the output system.

3.4 Output system

In LAMP, a versatile output system is available that enables users to collect simulation and analysis results in one of several different formats (or in user-generated formats). Outputs may be specified at any time during or after the run. The results of several simulation runs may be combined together at some point after the simulation has taken place to produce the desired output. Simulation runs that are so combined may be from different simulators. All these options can be specified by the command language.

Among the various output options available, some of the most commonly used will be described here. To verify the validity of the logic design, the VALUES option is often used, which lists the inputs and outputs along with the (1, 0, and "don't know") values of outputs for a given input test vector. In some cases where one is interested in internal states of the circuit, one can use GATEIO option to display the value of selected gates and their driving and driven gates. This feature is especially useful during a simulation run when the run is temporarily halted or has gone into oscillation; another specific use of this feature is to display circuit connectivity. Another format often used to display the outputs of timing and the true-value simulators is TLGRAPH. TLGRAPH is an oscilloscope-like trace of the signals on the output gates, from the time the test is applied until the time the circuit settles down. Whenever the value of an output gate changes, the time interval is recorded as well as the output gate values. This format has proven to be extremely valuable in studying worst-case timing conditions.

A variety of output formats is also available for studying the completeness, accuracy, and resolution of diagnostic tests. The ATP format lists all the faults that have not been detected for the test sequence simulated. The RAW output format lists the output gate name, each gate's true value as well as the number of faults that causes each gate's true value to be complemented, and a listing of these faults. For a large run where a user is interested in only a summary of the run, the MATRIX output can be used to show the faults detected by each test; the result is presented in the form of a matrix or a fault table. If the user is interested in fault partitioning and diagnosability information, he can choose the TREE output that lists the test results in the form of a diagnostic tree by grouping all those faults causing the circuit to behave in the same manner for a particular test sequence.

Facilities are also provided to allow the user to write his own output processing program. The raw output data set (RAWDS) contains all the raw data on the output gates from a simulation, including information such as the input vector on each test for which raw data are collected, names of inputs and outputs, fault cross-referencing information, fault and nonfault information, and certain circuit statistics. The user can manipulate this information to create the desired output format. The availability of this feature has substantially reduced the burden that otherwise would be imposed on the LAMP system developers to meet the wide variety of user needs.

IV. THE ROLE OF LAMP IN THE DESIGN PROCESS

The process by which a logic design becomes a completed product has become very complex with the advent of integrated-circuit technology. This process is made even more difficult in the telephone industry because of the stringent up-time requirement of the switching machines.[11] The ability to diagnose any equipment failure thus becomes an important consideration in the design and implementation of these machines.

The design and implementation process for a new switching system processor is made feasible by the constant use of computer-aided-design tools. Figure 4 shows the overall implementation process from the initial logic designs through to the completed processor. It also illustrates how the various major features of the LAMP system can be used in each design step.

The start of any major logic design project is the specification of the system architecture along with the basic design decisions. The COMET feature of LAMP helps this process by providing information about the diagnosability of a proposed design. With this tool, the global diagnosability of a system design can be established. Once this overall design step has been completed, the logic can be partitioned into individual circuit packs and detailed circuit designs can begin. In this phase of the design, the designer uses the true-value simulator for design verification, and frequently uses the timing simulator to make sure that the logic-timing functions are correct.

The use of these simulators requires that the logic circuit be encoded in the LSL-LOCAL language. The encoding of the circuit in the LSL-LOCAL language at this point accomplishes two basic functions. The first function is to catch any circuit discrepancies through the use of audits in the language processor and the second is to provide a machine-readable form of the circuit design. This latter function is basic to the entire computer-aided-design function.

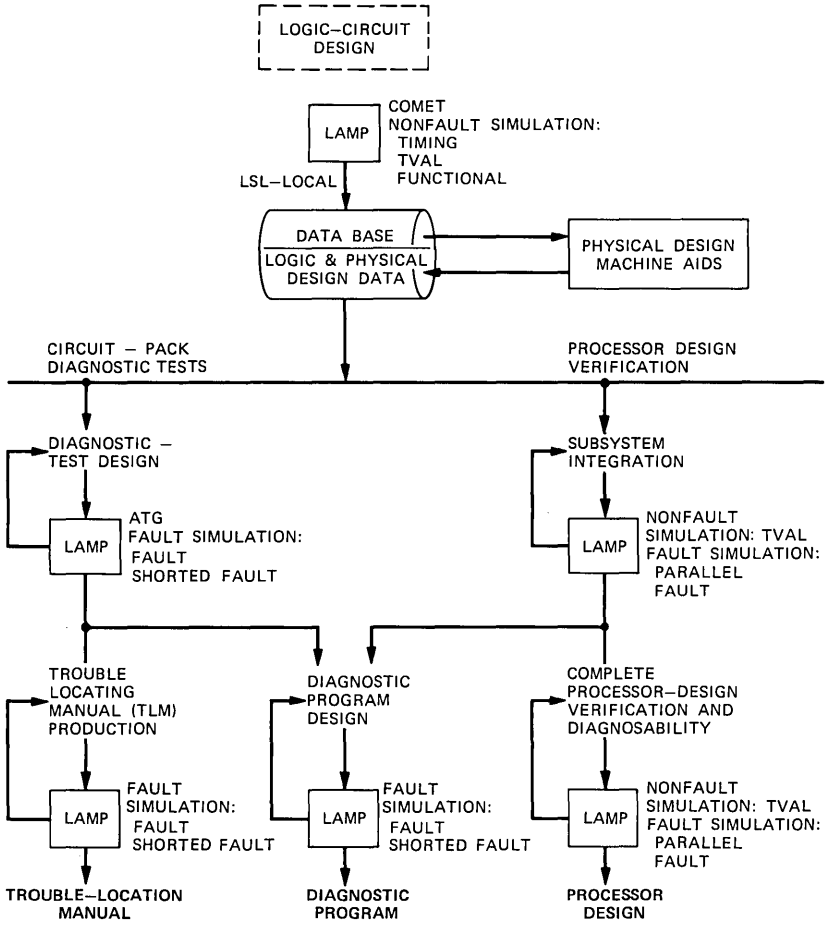


Fig. 4—Diagram of LAMP system use in logic-circuit design.

In addition to the basic circuit information, it is possible to input physical design information through the LSL-LOCAL language. When the designer is satisfied with the design of the circuit on a circuit-pack basis, the verified logic is then used as a base for the physical design process. Here the various additional machine-aided tools are used to perform partitioning, placement, and routing. The successful completion of physical design thus establishes a logical and physical design data base from which other uses of LAMP in the design process may take place. Some examples of these activities are: (*i*) derivation of circuit-pack diagnostic tests for manufacturing check-out purposes;

(*ii*) design verification of the subsystems (which are formed by combining circuit packs) and the complete processor (which is formed by combining the subsystems); and (*iii*) design and verification of diagnostic program(s) and generation of TLM data.

V. EXAMPLES OF LAMP SYSTEM USE

To provide some insight into the use of the LAMP system, a few examples of simple procedures performed with the LAMP system are presented. Because of the large number of ways the LAMP system is utilized, it is impossible to cover more than a small area of the system functions. The examples shown, however, are representative of typical activity.

All user communication with the LAMP system is by use of a command language. Each command represents an action to be taken by the system. In conversational use, the system prompts for the next input by means of a > character. Some commands which require additional information prompt the user with an @ character.

Example 1—Logic Verification Run
(TSS Log-on Procedure)

```
System: LAMP DESIGN AUTOMATION SYSTEM
        ENTER COMMANDS
        >
  User: load expl. tables
System: CKTNAME: EXAMPLE.CIRCUIT VERSION 06/24/73
        >
  User: run tval expl.test.vector,expl.output.results,p
System: LAMP TVAL SIMULATOR—VERSION 2.5
        >
  User: display values,t
System: AT INPUT NO. = 3
        INPUTS:   SA     SB     CA     CB
                  SEN    CEN
        OUTPUTS:  SOUT   COUT
        INPUTS:   100001
        OUTPUTS:  11
        >
  User: end
```

In this example, the user desires to test the "good" operation of his logic design by exciting his circuit with a series of prestored input vectors. The circuit description has been previously compiled from an

LSL-LOCAL encoding into a data set called "expl.tables." The prestored input vectors are located in the data set "expl.test.vectors." Since he is not interested in fault analysis, the TVAL (true-value) simulator is chosen. For nonfaulted operation, this simulator is the most efficient of the six available. The results he needs for his analysis can be obtained in two ways. The bulk of the output is produced via the computer high-speed printer. The particular types of results the user wants are specified by the contents of data set "expl.output.results." The "p" indicates that the results are to be printed as soon as possible. Because the user wants a quick check of some of the results before the other output is available, he instructs the system to display the input and output gate names along with their associated output values on the terminal after the simulation is completed. Satisfied with the results, he ends the simulation.

Example 2—Creation of the Controlling Data Sets
(TSS Log-on Procedure)

```
System: LAMP DESIGN AUTOMATION SYSTEM
        ENTER COMMANDS
        >
  User: source lsllocal expl.source,expl.tables
System: LOCAL LP START
          LOCAL LANGUAGE PROCESSOR—VERSION 3
        LOCAL LP END
        >
  User: results expl.output.results
System: ENTER SIMULATION RESULTS SPECIFICATIONS
        @
  User: after input *every; values
System: @
  User: [default]
System: >
  User: invec expl.test.vectors
System: ENTER INPUT VECTORS
        @
  User: t'101031'
System: @
  User: t'100001'
System: @
  User: [default]
System: >
  User: end
```

In this example, the user creates the data sets used to control the simulation run shown in Example 1. The first action is to compile the logic-circuit description written in LSL-LOCAL that has been stored in data set "expl.source" in a form that the compiler can use. The compiled information is stored in data set "expl.tables." Next the data set ("expl.output.results") that controls the output results is formed by use of the RESULTS command. The information put into this data set will instruct the simulator to print the *values* of the inputs and outputs after every input vector has excited the circuit.

Finally, the series of input vectors used to excite the circuit is created by use of the INVEC command. In this case, a series of these input vectors has been created. The input value "3" signifies a "don't care" value.

Only a few of the available commands and options have been shown. However, these should provide an idea of the ways in which the system can be used. Additional examples will be presented in the other papers of this series to illustrate specific points under discussion.

VI. CONCLUSION

Present and future designs of digital systems require computer aids during all phases of development, from initial architecture specifications to diagnostic-test design. The efficiency of these tools in performing their intended functions is of great importance, from both internal (efficiency of algorithms) and external (user convenience and usefulness) considerations. Viewed in this light, the LAMP system has been an outstanding success. The use of LAMP has been found to be cost effective in that LAMP provides the designers a convenient facility to assure design quality, to expedite error correction, and to reduce design-rework cost. LAMP also offers the designer a versatile tool to evaluate and verify the system diagnostics before hardware is committed. It has become an integral part of the design of new electronic switching systems and has strongly influenced the methodology of their design.

The other papers in the series will give more detailed descriptions of the use and design of selected portions of the LAMP system.

VII. ACKNOWLEDGMENTS

Many of our colleagues have contributed to the development of the LAMP system. Contributions made by R. E. Strebendt, R. E. Michael, and E. A. Rinaldy in the development of LSL-LOCAL, A. B. Marsh in the design of the command system, R. A. Elliott and R. B. Schmidt

in the implementation of output system, and J. R. Burnside, G. A. Raack, R. R. Riser, and F. J. Webb in the development of the OS version of LAMP are gratefully acknowledged. The authors would also like to thank J. A. Harr, W. Ulrich, and R. W. Ketchledge, and the many users for their continuous support and encouragement throughout the development of the system.

REFERENCES

1. S. Seshu and D. N. Freeman, "The Diagnosis of Asynchronous Sequential Switching Systems," IRE Trans. on Elec. Computers, *EC-11* (August 1962), pp. 459–465.
2. S. A. Szygenda, "TGAS2—Anatomy of a General Purpose Test Generation and Simulation System for Digital Logic," Proc. ACM-IEEE Design Automation Workshop (June 1972), pp. 116–127.
3. B. H. Scheff and S. P. Young, "Gate Level Logic Simulation," in Design Automation of Digital Systems, Vol. 1, edited by M. A. Breuer, New Jersey: Prentice-Hall, 1972, pp. 101–172.
4. S. G. Chappell, "LAMP: Automatic Test Generation for Asynchronous Digital Circuits," B.S.T.J., this issue, pp. 1477–1503.
5. H. Y. Chang and G. W. Heimbigner, "LAMP: Controllability, Observability, and Maintenance Engineering Technique (COMET)," B.S.T.J., this issue, pp. 1505–1534.
6. S. G. Chappell, C. H. Elmendorf, and L. D. Schmidt, "LAMP: Logic-Circuit Simulators," B.S.T.J., this issue, pp. 1451–1476.
7. T. T. Butler, T. G. Hallin, J. J. Kulzer, and K. W. Johnson, "LAMP: Application to Switching-System Development," B.S.T.J., this issue, pp. 1535–1555.
8. S. Seshu, "The Logic Organizer and Diagnosis Programs," *Report R-226*, Coordinated Science Laboratory, University of Illinois at Urbana (AD-05927).
9. D. K. Chia and M. Y. Hsiao, "Boolean Difference for Fault Detection in Asynchronous Sequential Machines," IEEE Trans. on Computers, *C-20* (November 1971), pp. 1356–1361.
10. G. R. Putzolo and J. P. Roth, "A Heuristic Algorithm for the Testing of Asynchronous Circuits," IEEE Trans. on Computers, *C-20* (June 1971), pp. 639–647.
11. R. W. Downing, J. S. Nowak, and L. S. Tuomenoksa, "No. 1 ESS Maintenance Plan," B.S.T.J., *43* (September 1964), pp. 1961–2020.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

LAMP:

Logic-Circuit Simulators

By S. G. CHAPPELL, C. H. ELMENDORF, and L. D. SCHMIDT

(Manuscript received February 20, 1974)

The algorithms used for logic-circuit simulation in the Logic Analyzer for Maintenance Planning (LAMP) system are described. Several simulators are available to allow a cost-effective tradeoff between simulation cost and the level of detail needed for a particular application. The true-value simulator provides efficient simulation of fault-free logic circuits. Two fault simulators simulate the classical stuck-at faults as well as shorted-gate-output faults. Hyperactive faults, those faults which cause an inordinate amount of simulation activity, are discussed along with their impact on simulation time. A four-value simulation logic is described which simplifies circuit initialization procedures.

I. INTRODUCTION

The use of digital simulation of logic circuits has been widely accepted in the computer and telephone industries to verify logic-circuit designs, to analyze the behavior of logic circuits in the presence of faults (such as gate outputs permanently stuck at logical 0 or logical 1, open gate inputs and shorted gate outputs), and to aid the generation of fault-detection tests for logic circuits.

Most simulators described in the literature can be divided into three categories. The first category includes the true-value simulators that simulate the circuit in the absence of any faults or, by altering the circuit description, simulate the circuit in the presence of one permanent fault.[1,2] The second category includes the parallel simulators that concurrently simulate the fault-free circuit and the effect on the circuit of a small set of single permanent faults.[2–4] The third category includes the deductive simulators that concurrently simulate the fault-

free circuit and the effect on the circuit of all single permanent faults.[5] The Logic Analyzer for Maintenance Planning (LAMP) system contains simulators from each category.

The LAMP system has been extensively used over the last four years to simulate the No. 1A and No. 4 Electronic Switching Systems to verify the logic design, to aid the generation of diagnostic tests, and to analyze the behavior of the circuits in the presence of faults. Circuits containing 52,000 gates and 23,000 single faults have been simulated using the IBM 370 Model 168 as the host machine.

The simulators in the LAMP system provide a complete range of capabilities for the design of logic circuits. Circuits and subsets of circuits can be simulated at the gate level (NAND, AND, OR NOR, NOT), at the functional level (register, memory, etc.) or at the hybrid level (a combination of gates and functions). At the gate level, gates can be modeled in sufficient detail to account for variations of such parameters as temperature and wiring capacitance. Several different classes of faults can be considered including gate outputs stuck at logical 0 or logical 1, gate inputs open, and shorted gate outputs. Facilities have been provided to help the user debug his logic design and his diagonstic tests.[6]

This paper presents a description of the LAMP simulators. In the first section, the family of simulators are described including an example of their use in the design of a logic circuit. This is followed by a description of the common attributes of the LAMP simulators. In the second section, the basic LAMP simulator for fault-free circuits is described and is the basis for describing the other LAMP simulators. In the next sections, descriptions of the deductive fault simulators and functional simulators are presented. In the seventh section, the detection and elimination of a class of "hyperactive" faults is described. Finally, data on the performance of the various simulators are presented.

II. THE SIMULATOR FAMILY

This section describes the use of the various LAMP simulators during the design of a logic circuit. This is followed by a description of the common attributes of LAMP simulators.

As the level of logic-circuit integration increases, it becomes more difficult to build "breadboard" models. This often means that more emphasis must be placed on the results of logic-circuit simulation. Therefore, it is desirable to use an extremely accurate simulation model of the logic circuit. Unfortunately, as the accuracy (level of detail)

of the model increases, so does the cost of simulation. Since LAMP was designed for large circuits (up to 65,000 gates), cost is an important parameter. One way to partially circumvent this problem is to utilize several different simulators, each of which provides a detailed model especially tailored to optimize the simulation of a physical circuit.

2.1 Use of simulation during circuit design

Consider the design of a small processor. Given the overall specifications for the processor, the designer can create a functional level model of the circuit where the building blocks include registers, memories, decoders and an arithmetic unit. Using the *functional simulator*, the design can be simulated at the functional level to verify the operation and timing of the processor. The processor can now be divided into functional units for detailed logic design of each unit. The functional units may be further divided into circuit packs containing a few hundred gates each.

The detailed logic design of the circuit pack is performed and is verified using the LAMP *true-value simulator*. The true-value simulator simulates only the fault-free circuit by modeling the logic gates as logic elements followed by pure time delays. This is a fast, economical simulator.

If the timing of the signals on the circuit pack is critical, the designer may wish to perform a more detailed timing analysis of his circuit using the LAMP *timing simulator*. The timing simulator[7] allows each gate to be assigned minimum and maximum time delays for the rising and falling signal transitions. The gate output is treated as unknown during the time between the minimum and maximum transition delays. This provides a more detailed analysis of circuit behavior in the presence of variations in gate time delays resulting from such factors as temperature change, gate loading, and capacitance. In addition, gate input pulses of shorter duration than the minimum transition delay are ignored and, therefore, do not affect the gate output value.

Once the designer has verified that his logic-circuit design meets the operational specifications, he must generate manufacturing test vectors (circuit input stimuli) to verify the integrity of the newly manufactured circuit pack. Whether the designer creates the test vectors by hand or uses the automatic-test-generation system,[8] he may use the LAMP *fault simulator* to evaluate the quality of the resulting set of input test vectors. The fault simulator simulates the effect on a

logic circuit of the presence of all single classical (gate input open, gate output stuck-at-zero, and gate output stuck-at-one) faults. This is a deductive simulator[5] that associates with each gate output a *fault list* containing those faults that will complement the correct (true) logic value (logical 0 or 1) of that gate. The fault lists may contain any number of faults, which theoretically allows the simultaneous simulation of all classical faults. Because of the effort required to process the fault information, the fault simulator is considerably more expensive to use than the true-value simulator. Through the use of the fault simulator, tests can be designed, or the circuit can be modified, to attain the desired level of fault detection.

If the number of faults to be simulated is less than a few thousand, it may be more economical to use the LAMP *parallel simulator* instead of the fault simulator. The parallel simulator uses parallel fault-simulation techniques[2–4] to simulate up to 2048 single classical faults in one pass. A variable-width-fault word is utilized so that simulation time and storage are minimized. The relative merits of the parallel and deductive fault simulation techniques are presented in Ref. 9.

After the chip layout and printed-wire routing for the circuit pack is complete, the designer may choose to examine the effectiveness of his classical fault tests against possible shorted faults using the LAMP *shorted-fault simulator*, which simulates the effect on a logic circuit of the presence of single pairs of gate outputs shorted together. If two gate outputs, A and B, are shorted together where gate A has the value logical 1 and gate B has the value logical 0, it is assumed that the logical 0 will dominate and the output of gate A will be forced to logical 0. A user option is available which forces logical 1 to dominate logical 0. Potential shorted faults that may be simulated include shorted adjacent pins on chips, shorted adjacent paths on the printed wiring board, and shorted crossover points on the printed wiring board. These data are obtained from the manufacturing information for each circuit. The shorted-fault simulator uses the deductive simulation technique.

After the circuit packs are designed, the designer can link all the circuit packs together to form the complete processor and perform the same logic verification process on the larger circuit with a few minor differences. The true-value and timing simulators are used both to verify the logic design of the processor and to verify the diagnostic program for the processor. The various fault simulators are used to evaluate the effectiveness of the diagnostic throughout the design-change cycle until the design is complete.

2.2 Common simulator attributes

The common attributes of the LAMP true-value, fault, timing, shorted-fault, parallel, and functional simulators are described below.

(*i*) The version of LAMP that is described is implemented on the IBM 360 Model 67 and IBM 370 Model 168 under the IBM interactive, virtual-memory operating system TSS. A version of LAMP is also available under the IBM operating system OS.

(*ii*) The first version of LAMP (1969) contained only the fault simulator. New simulators have been implemented as needed, and existing simulators have been improved to produce the complete system for logic simulation now available in LAMP.

(*iii*) The simulators can be accessed from an interactive terminal or used in the batch mode via card input or prestored data. Interactive features include the ability to temporarily stop the simulation when any specified gate changes value and the ability to correct from the terminal errors in the circuit design or input data.

(*iv*) Logic circuits are simulated at the gate level (NAND, AND, NOR, OR, and NOT) except in the functional simulator, which also accepts descriptions of higher-level blocks such as memories and registers.

(*v*) Four simulation values (0, 1, 2, and 3) are used to simulate binary-logic circuits. The simulation values 0 and 1 represent the logic values 0 and 1. Values 2 and 3 represent unknown conditions in the logic circuit. This is explained in more detail in Section 3.2.

(*vi*) Conditions that cause the output values of flip-flops to be unpredictable are detected and the flip-flop outputs are forced to the unknown state 3 by a process called *race analysis.* Possible circuit oscillations are detected by a process called *oscillation analysis.* Both procedures will be described in more detail in Sections 3.3 and 3.4.

(*vii*) LAMP uses discrete event simulation where all activity occurs at integral multiples of the basic increment of simulation time. The basic increment definition is arbitrary and may represent such units as nanoseconds, microseconds, or gate delays. Lists, called timing lists, are maintained by each simulator such that one timing list is associated with each increment of simulation time. Each timing list contains a list of gate-

output changes scheduled to occur at that increment of simulation time. The timing list associated with the current increment of simulation time is called the current timing list.

(*viii*) Selective trace is used so that a gate output is computed only if any of the gate's input signals changed value.

(*ix*) The circuit description is contained in a set of two-way, linked-list tables, which include information about each gate such as the driving and driven gates, logic function, time delay, and faults to be simulated. A subroutine, associated with each logic function, examines the gate-input values, computes the new output values, determines whether the output values have changed, and schedules the output change (if any) into some future timing list.

III. THE TRUE-VALUE SIMULATOR

The operation of the true-value simulator will be used as the basis for the presentation of the fault simulators. A simplified flow chart of the operation of the true-value simulator is shown in Fig. 1. This

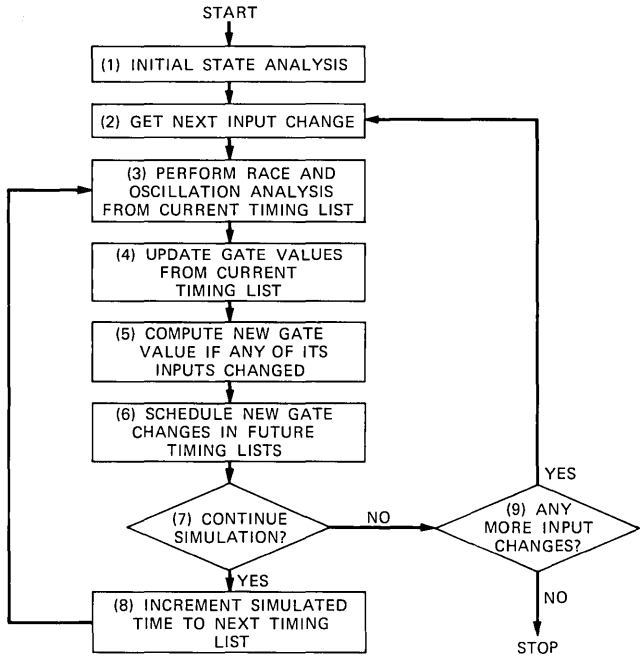


Fig. 1—Simplified simulation flow.

flow chart is also used in Section 3.5 to describe the overall simulator operation.

3.1 The true-value calculation

The LAMP simulators use four logic values, 0, 1, 2, and 3, to simulate the Boolean logic functions. The 0 and 1 are simply the logical 0 and 1 of Boolean algebra. Values 2 and 3 represent nonpropagating and propagating "don't-know" conditions, respectively. The gate output calculation occurs in Step 5 of Fig. 1.

Value 2 is used to allow efficient initialization of the circuit. Prior to a simulation run, all gates are initially assigned a value of 2. Its nonpropagating feature is demonstrated by the following table of a two-input NAND gate:

A	B	$\overline{A \cdot B}$
2	0	1
2	1	Q
2	2	Q
2	3	Q

where Q means no change in the previous true value.

The nonpropagation is necessary to prevent destroying information specified by setting *a priori* the state of the circuit. For example, in Fig. 2, if the state specification sets $C = 0$, nonpropagation is necessary to prevent the true value of C from being overwritten by a don't know. Value 2 allows $C = 0$ to initialize the flip-flop to $C = 0$ and $D = 1$. A more detailed explanation of the behavior of 2s will be presented in the next section.

True-value 3 is a true "don't know" with full propagation features. The truth table for a two-input NAND gate is shown below:

A	B	$\overline{A \cdot B}$
3	0	1
3	1	3
3	2	Q
3	3	3

where Q means no change in the present true value.

In Fig. 2, if all 2s were replaced by 3s, then the output of C and D would become 3 even though the user initialized C to logical 0.

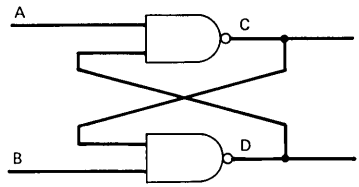


Fig. 2—NAND flip-flop.

3.2 *Initial-state analysis*

The purposes of the initial-state analysis (Step 1 in Fig. 1) are: (*i*) to extract as much information as possible from the user-specified circuit state (if any), and (*ii*) to guarantee that the output of each gate is consistent with its inputs. A flow chart of this procedure is shown in Fig. 3.

True-value 2 is used only during the initial-state analysis which occurs before the first input vector is applied to the circuit. The initial-state analysis is a three-pass procedure that attempts to propagate the effect of any user-specified state through the circuit. Pass 1 has two alternatives. If the user did not set any state, then pass 1 simply changes all of the gates whose output value is 2 to the "true" unknown-value 3 and the simulation of the input vectors begins.

However, if the user has set some initial state, then the initial-state analysis must propagate the effect of that state through the circuit. During pass 1, the circuit contains the logic-value 2 for the "don't-know" condition. The nonpropagation feature of the 2s allows as much information as possible to be extracted by the simulator using only a forward simulation. No attempt is made to set the inputs of

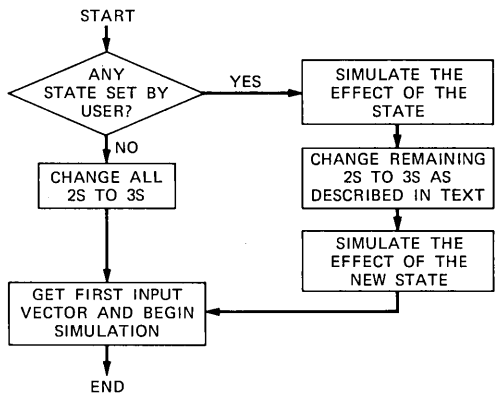


Fig. 3—Initial-state pass.

some NAND gate to logic-value 1 if the output of the gate is logical 0. Thus, LAMP requires that initial states should always be set using the "dominant" value of the particular logic type used. For example, because gate C of Fig. 2 was set to a logical 0, pass 1 would set D to value 1.

Pass 2 goes through the circuit and changes selected remaining gates whose output value is 2 to new output-value 3. This is necessary because the 3s propagate where the 2s do not. Therefore, leaving the 2s in the circuit can cause incorrect simulation results. However, it is only necessary to change to 3 those gates within maximally strongly connected subgroups (MSCs)[10] having output-value 2. This occurs because the circuit inputs are assumed to support any state which the user sets. Therefore, the input gates as well as any combinational circuitry driven by the inputs maintains true-value 2 until it is eliminated by the first input vector or the next pass.

Pass 3 propagates the newly injected 3s as far as possible. This may have the effect of destroying some incomplete state which the user specified because the circuit is unable to support the incomplete state for all possible complete states. If a complete self-supporting (stable) state is specified, no state information will be eliminated.

Initializing the circuit to some known value can introduce simulation inaccuracies during fault simulation. If the circuit is artificially initialized, there is no record of those faults whose presence would prevent the circuit from reaching the initial state specified. Therefore, it is preferable to apply a synchronizing sequence to the circuit to drive it from an unknown state (all gate output values set to 3) to some known state. The facility to artificially initialize the circuit is provided to help the user and to simplify his work.[11]

3.3 True-value race analysis

Race analysis (Step 3 in Fig. 1) is performed on the basic NAND and NOR flip-flop. Previous simulation techniques attempted to treat the flip-flop as a "black box." However, the "black box" approach leads to inaccurate simulations or to unwieldy simulation algorithms. Therefore, the technique used in LAMP is to detect races as invalid conditions on a set of gates. Since both NAND and NOR flip-flops are handled in a similar manner, only the true-value race analysis for the NAND flip-flop will be discussed here. The basic NAND flip-flop is shown in Fig. 2.

The true-value race state for a NAND flip-flop is $A = 1$, $B = 1$, $C = 1$, and $D = 1$ at the same time, t. From this state, it is impossible

to predict (assuming identical gate behavior) whether $C = 1$ and $D = 0$ or $C = 0$ and $D = 1$ when the flip-flop settles. So as not to arbitrarily resolve races, true-value 3 is assigned to the output of both gates in the flip-flop.

To accomplish this, when the flip-flop was in the $A = 1$, $B = 1$, $C = 1$, and $D = 1$ state at time t, the simulator calculates $C = 0$ and $D = 0$ for the new intermediate output to be scheduled into a future timing list. Since the state $C = 0$ and $D = 0$ is impossible unless the previous state was $A = 1$, $B = 1$, $C = 1$, and $D = 1$, both outputs at logical 0 provide an efficient race-detection mechanism.[12] Also, since $C = 0$ and $D = 0$ are unstable, both C and D will be scheduled to change values at the present increment of simulation time. Therefore, the outputs of a NAND flip-flop are set to true-value 3 and a race declared when:

(*i*) The newly calculated, but not yet assigned, outputs of both gates are simultaneously 0.
(*ii*) Both gate outputs are scheduled to be changed at the present time.

If the NAND gates are cross-coupled, as shown in Fig. 2, but are *not* specified as a flip-flop, then race analysis will not be performed. In this case, if the flip-flop is in a race state, the new output $C = 0$ and $D = 0$ will be assigned to the gates in the flip-flop. The next output (assuming the inputs to the flip-flop do not change) will be $C = 1$ and $D = 1$ and the flip-flop will oscillate between $C = 1$, $D = 1$, and $C = 0$, $D = 0$ causing a simulator oscillation.

In addition, because of the behavior of value 3, the condition where the newly calculated output values of the flip-flop are $C = 0$ and $D = 3$ or $C = 3$ and $D = 0$ will cause an oscillation. Therefore, this condition is also detected and declared to be a race. Extensive topological circuit analysis could isolate the undeclared flip-flops, but such analysis is not performed since the circuit designers seldom fail to declare the race-pair gates.

3.4 True-value oscillation analysis

A true-value oscillation (Step 3 in Fig. 1) occurs when the circuit state is unstable as a result of some input conditions. An oscillation is declared if the simulator simulates an arbitrary number, N, of increments of simulation time and the circuit has not stabilized. The value of N is defaulted to be the number of gates in the logic circuit but can be adjusted by the user.

If a true-value oscillation is detected, the old and new true values for every gate B whose output is changing at the present increment of simulated time are compared. If the old and new true values are different for gate B, the new true value is replaced with value 3 since the output of B is changing (i.e., unknown). Value 3 is the new gate output that will be scheduled in some future timing list. When 3s are inserted, the oscillation automatically stops, since a 3 represents both a 0 and a 1.

3.5 The true-value circuit model

The true-value circuit model defines the interactions among the initial-state-analysis, gate-calculation, race-analysis, and oscillation-analysis steps that were presented earlier. Thus, a description of the true-value circuit model is an overall description of the simulator operation.

A simplified flow chart of the basic simulator operation is shown in Fig. 1. The operation includes the following:

Step 1—The circuit is analyzed to check the validity and consistency of any user-supplied initial state, as described in Section 3.1.

Step 2—This step is repeated once for every circuit input vector to be simulated. During this step, the next input vector is obtained and the new input values are assigned to the circuit input leads. The effect of this input vector on the circuit is now propagated through the circuit. Every input gate whose value changed as a result of the new input vector is put into the appropriate future timing list. The future timing lists are examined, as the simulation time is incremented, until the first nonempty timing list is found. This timing list is called the current timing list. Let the present time be t_0 and assume that the set of gates G, $\{G_i, i = 1, 2, \cdots, m\}$, in the current timing list at t_0 contains all the gates whose outputs are changing at time t_0. Steps 3 through 6 are performed once for each timing list.

Step 3—Race analysis is performed for each declared flip-flop formed by two gates, both of which are in G.

Step 4—The new outputs are assigned to every gate in G.

Step 5—After all the new outputs of G have been assigned, the output of each gate H_j, $j = 1, 2, \cdots, n$, which is driven by any G_i whose output has changed, is calculated according to gate model.

Step 6—If the output of some H_k, $1 \leqq k \leqq n$, changed, then H_k is put into the timing list of gates whose output may change at time $t_0 + t_t$, where t_t is the transition time for H_k. If the output of H_k did not change, no further action is taken on the gate. The important feature of this circuit model is that the gates H_j, $j = 1, 2, \cdots, n$, have their inputs calculated based on all new values of the gates in $\{G_i, i = 1, 2, \cdots, m\}$. That is, every change that is going to occur at t_0 occurs before the output of any gate driven by any of the gates in G is calculated.

Step 7—Simulation may be allowed to continue or it may be interrupted to process a change on the input leads (Step 9) or to return to the command language to process user commands.

Step 8—The simulation time is incremented. This makes the timing list at time $t_0 + 1$ the current timing list and the loop continues. Simulation is terminated if there are no more input changes.

IV. THE FAULT SIMULATOR

The fault simulator utilizes Armstrong's[5] fault-list concept to allow concurrent simulation of all open gate input, output stuck-at-one (SA1), and output stuck-at-zero (SA0) faults in one pass per input vector. The input-open fault is assumed to force a nondominant value on that input. For example, for NAND and AND gates, the input open is assumed to force that gate input to logical 1. A number from 0 to $k - 1$ is associated with each of the k faults in the circuit. Each gate G, except the inverter, is assigned $N + 2$ faults, where N is the number of inputs to gate G. The inverter has only the two output SA1 and SA0 faults, since the input-open fault is indistinguishable from the output SA0 fault. These fault numbers are then carried in fault lists associated with each gate. The hard faults, or corresponding fault numbers, in the fault list on gate G represent exactly those faults in the circuit that will cause the true value (logical 0 or 1) of gate G to be complemented. Only gates having 1 and 0 true values can have fault lists. Similarly, the *star faults* in the fault list on gate G represent faults in the circuit for which the value of G is not predictable by the simulation model.

4.1 Fault-simulator gate calculation

The fault-simulator gate calculation (step 5 in Fig. 1) involves the manipulation of the fault lists on each gate using the fault-list algebra.

In the description of the fault-list algebra, each fault list is treated as a set. The three set operations used for fault-list calculation are union ($\cup$), intersection ($\cap$), and difference (Θ).

The union of two fault lists A and B is defined for some fault f to form the output-fault list F:

Union Operation $A \cup B$

		B		
	F	λ	f	$^*f$
A	λ	λ	f	$^*f$
	f	f	f	f
	$^*f$	$^*f$	f	$^*f$

where

$^*f$ = star fault corresponding to fault f,
λ = absence of f and $^*f$ from the set (fault list).

The *intersection* of two fault lists, A and B, is defined for some fault f to form the output-fault list F:

Intersection Operation $A \cap B$

		B		
	F	λ	f	$^*f$
A	λ	λ	λ	λ
	f	λ	f	$^*f$
	$^*f$	λ	$^*f$	$^*f$

The *difference* of two fault lists A and B is defined for some fault f to form the output-fault list F:

Difference Operation $A \Theta B$

		B		
	F	λ	f	$^*f$
A	λ	λ	λ	λ
	f	f	λ	$^*f$
	$^*f$	$^*f$	λ	$^*f$

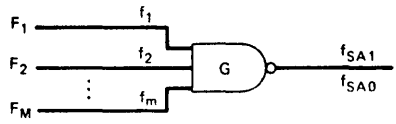


Fig. 4—Fault-list calculation.

For an m input NAND gate G in Fig. 4, let:

F_i = Fault list on the gate driving the ith input of G $(1 \leqq i \leqq m)$.
f_i = Input open on the ith input of G.
f_{SAk} = Output of G stuck at k $(k = 1, 0)$.
F = Resulting fault list on gate G.
$\overset{k}{\bigcup}$ means form the union over all the fault lists on input leads whose true value is logical k, $k = 0, 1$.
$\overset{j}{\bigcap}$ means form the intersection over all the fault lists on input leads whose true value is logical j, $j = 0, 1$.

To calculate the new output fault list F from the input lists F_i, $1 \leqq i \leqq m$, consider the following cases. First, assume all m inputs are logical 1. The output true value is 0 and

$$F = \{\overset{1}{\bigcup} (F_i \ominus \{f_i\})\} \ominus \{f_{SA0}\} \cup \{f_{SA1}\}. \tag{1}$$

This equation means that the output SA1 fault plus any fault on any input, except the respective input-open faults and the output SA0 fault, can cause the correct gate output to change values.

Second, assume that all inputs are logical 0. Then the output value is 1 and

$$F = \{\overset{0}{\bigcap} (F_i \cup \{f_i\})\} \cup \{f_{SA0}\} \ominus \{f_{SA1}\}. \tag{2}$$

This equation means that the output fault list contains the SA0 fault plus any fault present on every input lead. A fault is present on an input lead if it occurs in the lead's fault list or is the lead's input open fault. The output fault list does not contain the SA1 fault.

Third, assume that some inputs are logical 0 (those denoted by i) and the remaining inputs are 1 (those denoted by j). The output value is 1 and

$$F = \{[\overset{0}{\bigcap} (F_i \cup \{f_i\})] \ominus [\overset{1}{\bigcup} (F_j \ominus \{f_j\})]\} \cup \{f_{SA0}\} \ominus \{f_{SA1}\}. \tag{3}$$

The meaning of the equation follows directly from the meaning of eqs. (1) and (2).

Fourth, if there is a value 2 or 3 on any input and a logical 0 on some other input, then the output true value is 1 and $F = \{f_{SA0}\}$ only.

The fault-list computation equations can be derived by considering two input gates. Consider a two-input gate G with inputs A and B. If $A = B = 1$, then eq. (1) can be shown to be true by exhaustive analysis. Similarly, if $A = B = 0$, then eq. (2) is obviously true. Again if $A = 1$ and $B = 0$, then eq. (3) is true. The NAND is simply AND followed by a NOT gate and the AND operation is associative and commutative. Then eqs. (1) and (2) represent simple cascades of pairs of two input gate operations. Similarly, eq. (3) means treat all the logical 0 inputs as one AND gate G_0, then all the logical 1 inputs as an AND gate G_1, and then form the difference of G_0 and G_1. In this explanation, the internal faults were ignored. However, their handling is apparent from eqs. (1) through (3). Equations (1) through (3) describe how the LAMP fault simulator is implemented.

An alternate and more detailed implementation can be achieved by associating two fault lists with each gate whose true value is 3. The fault lists contain those faults that will cause the faulty gate output to be logical k for $k = 0, 1$. This allows more detailed analysis of faulty circuit behavior during initialization. However, this approach will significantly increase the storage required for the fault lists and the CPU time required to perform the simulation. For that reason, eqs. (1) through (3) were chosen as a realistic compromise between detail and cost.

4.2 Fault-simulator race analysis

Race analysis under fault conditions (Step 3 in Fig. 1) is performed on the basic NAND and NOR flip-flop (Fig. 2). An analogous situation to the true-value race can occur because of faults; that is, because of one or more of the faults in the fault list on gate C or D (Fig. 2). Each hard fault f in a fault list on gate G means that if f physically exists in the circuit, then the true value of G will be complemented. Therefore, the behavior of faults is identical to the behavior of true values in the faulty circuit. Then with some modification, the algorithm for detecting true-value races can also be used to detect fault-induced races. A fault f on the output gate(s) of a flip-flop (FF) is a race fault (star fault) if it satisfies all of the following conditions:

(1) Fault f will cause both outputs (D and C) of FF to be 0.
(2) Both gates of FF are scheduled to change at the present increment of simulation time.

(3) Fault f is not:

(a) The input open on D from C or the input open on C from D.
(b) The output of C SA1 or SA0.
(c) The output of D SA1 or SA0.

The first two conditions are the same as the conditions for a true-value race. The third restriction is apparent since, if either of the cross-coupled inputs were open, the gates would not form a flip-flop and could not race. Similarly, either output SA1 or SA0 would make a race impossible since there is no uncertainty about the outcome. As with the true-value race, faults which force $C = 0$ and $D = 3$ or $C = 3$ and $D = 0$ will cause oscillations and are declared as race faults.

Let F_C and F_D be the set of faults (or the fault list) on C and D, respectively. Let F_I represent the set of faults that cannot cause a race on FF [those faults listed in condition (3) above]. Consider three cases:

Case 1: $C = 1$ and $D = 1$; then the race faults F_R are given by: $F_R = (F_C \cap F_D) \ominus F_I$.
Case 2: $C = 1$ and $D = 0$; then the race faults F_R are given by: $F_R = (F_C \ominus F_D) \ominus F_I$.
Case 3: $C = 0$ and $D = 1$; then the race faults F_R are given by: $F_R = (F_D \ominus F_C) \ominus F_I$.

The faults in the set F_R are the star faults. These star faults are then merged into the fault list on gates C and D. That is,

$$F_C \leftarrow (F_C \ominus F_R) \cup {}^*F_R$$
$$F_D \leftarrow (F_D \ominus F_R) \cup {}^*F_R,$$

where F_C and F_D are the fault lists on gates C and D, and F_R is the fault list produced by race analysis. The left arrow ($\leftarrow$) means "is replaced by." The new F_C and F_D are assigned to gates C and D at the same time the other new output values are assigned to their gates.

4.3 Fault-simulator oscillation analysis

A fault oscillation (Step 3 in Fig. 1) is declared if the circuit does not stabilize after N increments of simulation time and no true values are changing. The number N may be set by the user as described earlier.

If a fault oscillation is detected, the old and new fault lists for each gate in the set $\{G_i,\ i = 1,\ 2,\ \cdots,\ m\}$ whose inputs changed during the previous increment of simulation time are compared. Let F_{ni} = new fault list and F_{oi} = old fault list for some gate in $\{G_i\}$. Then the set of faults that are causing the fault-list changes, F_s, is determined as

$$F_s = \bigcup_{i=1}^{m} F_{si}$$

and

$$F_{si} = (F_{ni} \ominus F_{oi}) \bigcup (F_{oi} \ominus F_{ni}).$$

Since single faults are assumed, no fault can cause another fault to be in a fault list. Therefore, the set of faults that alternately appears and disappears in the fault lists must be causing the oscillation. The set of faults causing the oscillation, F_s, is flagged as star faults (or unioned as star faults) in the new list F_{ni}. That is,

$$F_{ni} \leftarrow (F_{ni} \ominus F_{si}) \bigcup {}^*F_{si}.$$

Once a true-value or fault oscillation has been detected, oscillation analysis is performed until the circuit has been stabilized. By adding star faults or adding the value 3, the circuit should eventually stabilize and the oscillation will be resolved.

Figure 5 shows a circuit that illustrates both true-value and fault-list oscillations. If $K1 = 1$, then the circuit will oscillate in true values. However, if $K1 = 0$, the input-open fault from $K1$ on gate $K3$ will cause the circuit to exhibit a fault-list oscillation.

Since the calculations involving the star faults are expensive, a simulator is available (logic simulator) that immediately terminates simulation of any star fault when it occurs. Thus, the logic simulator does not simulate the effect of faults that cause "don't-know" conditions. This approximate simulation yields faster simulation times.

V. OTHER LAMP SIMULATORS

Sections I through IV of this paper explain the fundamental ideas behind logic-circuit simulation in LAMP. In this section, a brief description of the shorted-fault simulator and the functional simulator

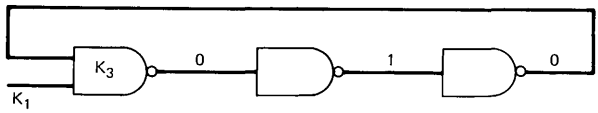


Fig. 5—Oscillating circuit.

is presented. The overall operation of the shorted-fault and functional simulators is similar to the operation of the simulators presented earlier. The fundamental difference lies in the method used to compute the output of the gate or functional element. Therefore, only the basic differences are discussed here.

5.1 The shorted-fault simulator

The shorted-fault simulator uses the deductive technique to simulate the effect on a logic circuit of a single electrical short between two gate outputs, where logical 0 is assumed to dominate logical 1. That is, if two gates, A and B, are shorted together and (in the absence of the short) A has the value 1 and B has the value 0, then in the presence of the short, gate A will have its output forced to logical 0. An option is available that causes logical 1 to dominate logical 0; however, since both cases are similar, only the case dominated by logical 0 is described here.

The shorted-fault simulator is a recent addition to the LAMP system. Because run time was expected to be considerably longer than for the fault simulator, the shorted-fault simulator was implemented to detect and immediately terminate simulation of all star faults.

The operation of the shorted-fault gate calculation requires that two fault lists, the constrained and free fault lists, be associated with each gate. The free fault list for gate A, called F_A, is computed using eqs. (1) through (3). The constrained fault list on gate A, called C_A, reflects the effects of the signals on any gates that can short to gate A. For the computation of the constrained fault lists, consider two gates, A and B, and a fault, s, whose occurrence causes the output of gate A to be shorted to the output of gate B, as shown in Fig. 6. Consider two cases:

(*i*) If $A = B = 1$,

$$C_A \leftarrow C_A \cup \{s \cap (F_A \cup F_B)\} \tag{4}$$

$$C_B \leftarrow C_B \cup \{s \cap (F_A \cup F_B)\}. \tag{5}$$

(*ii*) If $A = 1$ and $B = 0$,

$$C_A \leftarrow C_A \cup \{s \ominus (F_B \ominus F_A)\} \tag{6}$$

$$C_B \leftarrow C_B \ominus \{s \ominus (F_B \ominus F_A)\}. \tag{7}$$

The initial constrained fault list on each gate is exactly the free fault list on that gate. The constrained fault list is then altered as described in eqs. (4) through (7). These equations can be verified by examining

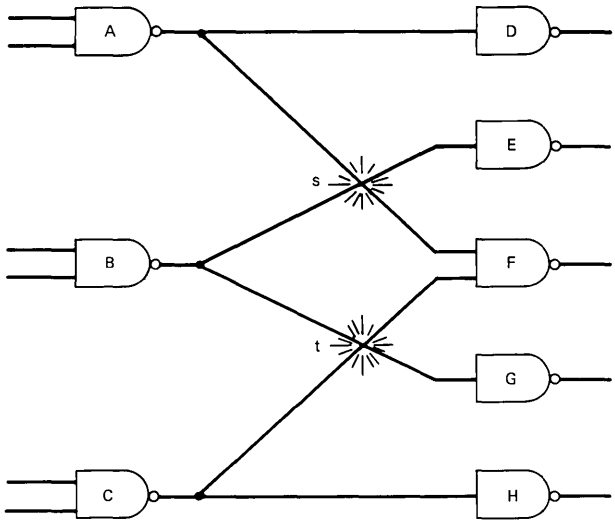


Fig. 6—Two shorted faults.

all eight cases since the only difference between the old and new constrained fault lists is fault s. This procedure must be repeated for every shorted fault that can affect the output of gates A and B (such as fault t in Fig. 6). However, since only one fault is assumed to exist at any time, all the applications of eqs. (4) through (7) are independent.

The constrained fault list on gate G is the "true" fault list for the gate since it reflects the effects of potential shorts to gate G. The free fault list on gate G is used as the starting point to compute the constrained fault list. If the free fault list were discarded after use, it would be necessary to go to the inputs of G and recompute the free fault list on G wherever it was necessary to derive a new constrained fault list on G (e.g., when a gate that could be shorted to G changes values).

The timing considerations are also important. Since the interconnecting paths are assumed to have zero time delay compared to the time delay of the gates, the effect of any shorted fault must immediately be reflected at the outputs of the gates, which may be shorted together. Therefore, the effect of possible shorts on each gate in the current timing list must be considered when the new output values are assigned to the gates (Step 4 in Fig. 1). The effect of the shorted faults may cause gates other than those in the current timing list to change value at the current time. This factor must be considered in Step 6 of Fig. 1 when the gates whose output value changed are scheduled into future timing lists.

The shorted-fault simulator has helped improve the manufacturing tests for circuit packs by aiding the design of sets of test inputs that will detect all shorted faults.

5.2 The functional simulator

The functional simulator allows the simulation of higher-level functional elements, such as clocks, registers, and memories, in conjunction with gate-level simulation. Thus, the functional simulator can be used to evaluate the tentative design of a logic circuit where the entire circuit is described as functional elements. Alternatively, functional memories, registers, and clocks can be added to a gate-level simulation to provide more complete or more efficient simulation of certain blocks by reducing storage requirements and execution time.

The control and data flow within the functional block are described using an "Algol-like" language.[13] Control conditions are described using "if-then-else" statements. Data transfer is accomplished using "Assignment" statements. Such operators as NOT, AND, OR, ADD, SUBTRACT, and SHIFT are allowed. Timing information is conveyed by preceding a statement with an "at time" clause. These statements are compiled into an extended reverse Polish format[14] and executed during simulation.

The functional simulator has significantly increased the capabilities of the LAMP simulators because of the ease of describing a functional unit. It has been used to aid in the logic verification of the No. 1A ESS Central Control.[11]

VI. RUN-TIME DATA

The logic and true-value simulators are the most frequently used LAMP simulators. Hence, more data are available on their run-time characteristics. All data shown were collected using an IBM 360, Model 67.

Table I describes ten typical circuits from a computer system. Since there is no convenient way to measure circuit complexity, two ad hoc measures are used. The number of flip-flops in a circuit provides insight into the circuit complexity on a localized basis while the number (or percentage) of gates in the MSCs[10] provides a more global measure of complexity. These circuits were simulated producing the data shown in Table II and Fig. 7.

Table II shows the simulator CPU time required to simulate the circuits described in Table I using the true-value, logic, and parallel simulators. The data in Fig. 7 show that the average simulator time

Table I — Size and complexity of sample circuits

Circuit	No. of Gates*	No. of Flip-Flops	No. of Gates in MSCs	Percentage of Gates in MSCs to Total Gates
A. Serial-to-Parallel Converter	349	90	224	0.64
B. Error Corrector	340	68	178	0.52
C. Parallel-to-Serial Converter	387	78	184	0.47
D. Decoder and Order Sequencer	311	15	82	0.26
E. Dial-Pulse Sequencer	336	22	112	0.33
F. Decoder and Match Circuit	383	8	44	0.12
G. Arithmetic Unit	6602	234	4378	0.66
H. Core Store Unit II	9359	320	3517	0.37
I. Core Store Unit I	2476	167	1182	0.48
J. Processor	46,012	2149	†	†

* T^2L NANDS are used throughout. There are an average of two inputs per gate for the circuits listed.
† Data not available.

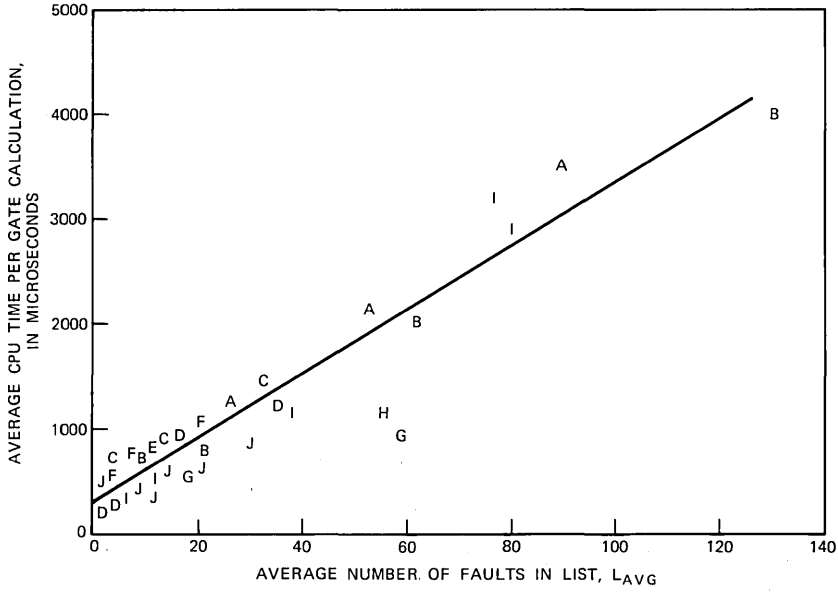


Fig. 7—Simulator time required to calculate gate output.

Table II — Simulation time for three simulators

Circuit	No. of Faults Simulated	No. of Vectors Simulated	Simulation CPU Time (Seconds)		
			Logic	True Value	Parallel
A. Serial-to-Parallel Converter	572	427	433	11	180
B. Error Corrector	894	412	641	9	102
C. Parallel-to-Serial Converter	559	348	253	9	145
D. Decoder and Order Sequencer	886	893	352	17	135
E. Dial-Pulse Sequencer	395	254	32	5	39
F. Decoder and Match Circuit	1065	161	43	3	52
G. Arithmetic Unit	2147	377	510	39	927
H. Core Store Unit II	2582	200	8361	330	*
I. Core Store Unit I	2631	16	326	17	495
J. Processor	9469	134	8673	180	*

* Data not available.

t_d required to compute the output true-value and fault list for one gate (one gate calculation) is a linear function of the length of the average fault list L_{AVG} on that gate. The length of a fault list is the number of faults in the list. The time t_d includes all bookkeeping and overhead involved in the simulation.

Figure 8 shows more data on circuit J in Table I (the No. 1A ESS processor[13]). The two lines represent the CPU time (IBM Model 67) per input vector for execution and read-write tests for the processor as a function of the number of faults being simulated. During the execution tests, the processor is executing instructions. During the read-write tests, the registers of the processor are being written and read by a second computer. The processor contains about 100,000 potential classical faults. These data were collected by simulating a subset of the faults against a subset of the diagnostic tests for the processor. Main memory size (4 megabytes) limits the number of faults that can reasonably be simulated, since it is desirable not to utilize the paging features of the Model 67 virtual memory because of the real-time penalty incurred due to the slow drum and disc accesses. These curves show that simulation time increases linearly with the number of faults simulated for a given set of vectors. However, the curves also show that simulation times are highly dependent on the circuit function being exercised by the input vectors.

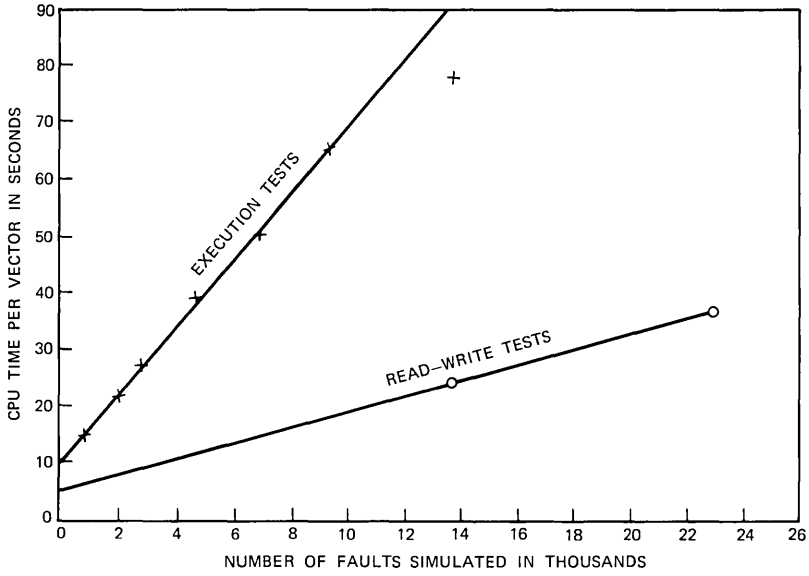


Fig. 8—Simulation times for typical processor diagnostics.

VII. HYPERACTIVE FAULTS

A new phenomenon called hyperactive faults has been found. Hyperactive faults are those faults that cause an inordinate amount of simulation activity. Removal of the hyperactive faults has reduced simulation time by as much as a factor of 8.

The fault simulator typically is more expensive to use than the logic simulator. However, it was discovered that on a 40-vector simulation of a 30,727-gate circuit with 950 faults and 152 star faults (faults which cause a race at some point during the simulation), the logic simulator took 750 seconds of IBM 360, Model 67, CPU time while the fault simulator required 2290 seconds. In an effort to determine the cause of this large discrepancy, the activity count was computed for each fault being simulated. The activity count for a fault is incremented if that fault is in the fault list on some gate at simulated time $t + 1$, but not in the fault list on that gate at simulated time t. The activity count is a measure of the amount of circuit activity caused by each fault.

Figure 9 shows a typical plot of the activity count distribution. For the case mentioned above, there were 14 faults whose activity count was more than 16 times the average activity count for all faults. These

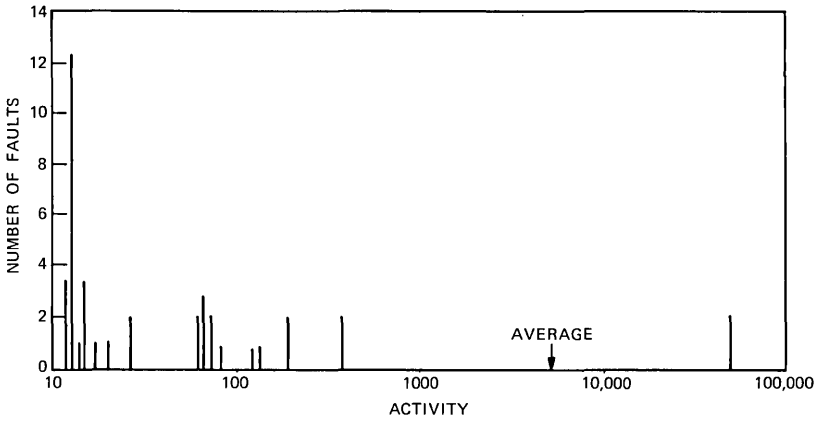


Fig. 9—Fault activity count distribution.

14 faults were removed and the circuit was resimulated with the fault simulator in 695 seconds. Thus, the removal of 1.5 percent of the faults caused approximately a 3-to-1 improvement in simulation CPU time.

Simulator speedups as high as 8 to 1 have resulted from the elimination of faults whose activity counts were excessive. For example, in a 29,696-gate circuit with 642 faults, the run time was 170 seconds per vector. By removing only 14 hyperactive faults (whose activity count was greater than 16 times the average), the run time dropped to 21 seconds per vector for the same vectors.

Two more simulations are of interest. For a 30,727-gate circuit with 1400 vectors and 2318 faults, including 601 race faults, 341 hyperactive faults were removed. On the same circuit with 3500 vectors and 5079 faults, including 1789 race faults, 101 hyperactive faults were removed. Thus, the number of hyperactive faults detected is reasonable.

Hyperactive faults are typically associated with clock circuits, sequencer circuits, and "stop circuitry." The occurrence of a hyperactive fault in the circuit often removes the effectiveness of critical control leads and causes the circuit to "run wild." While the hyperactive faults cause erratic circuit behavior, they do not necessarily cause fault-list oscillations.

The removal of hyperactive faults produces the most dramatic effect in the fault simulator because hyperactive faults are usually a subset of the star faults (race and oscillation faults) discarded by the logic simulator. Thus, the logic simulator is not as sensitive to hyper-

active faults. The removal of hyperactive faults from the fault simulator produces a significant saving in computer resources.

VIII. SUMMARY

The emphasis in the LAMP simulators has been to provide a reasonable level of simulation detail in a cost-effective manner. To achieve this goal, several simulators have been produced, each of which emphasizes some aspect of the cost-versus-detail tradeoff. The true-value simulator provides economical simulation of large logic circuits by using a true-value, integral-delay, gate-level circuit model. The timing simulator is somewhat more expensive since it analyzes minimum and maximum rise and fall delays for each gate as well as performing spike rejection in a gate-level, logic-circuit model. The logic simulator provides a two-value fault simulation using the deductive method, and a gate-level circuit model. The fault simulator is identical to the logic simulator except that it provides a three-value fault simulation. As a result, the fault simulator is more expensive than the logic simulator. Clearly, both fault simulators are more expensive than the true-value simulators.

The LAMP system has been used throughout Bell Laboratories to aid logic-circuit design and analysis. LAMP, and in particular the simulators described in this paper, have been very important in the development of the ESS 1A Processor and the No. 4 ESS.[11] Because of the depth of the simulation capabilities available, LAMP has provided efficient simulation capabilities over a wide range of circuit sizes and device technologies.

IX. ACKNOWLEDGMENTS

We wish to acknowledge the valuable work of G. W. Smith, Jr., R. B. Walford, and R. E. Michael on early versions of the fault simulator. We also wish to acknowledge the work of A. B. Marsh and A. M. Schowe on the functional simulator and the work of E. W. Thompson and D. E. Bzowy on the shorted-fault simulator. In addition, the encouragement and support of W. Ulrich and R. W. Ketchledge are gratefully acknowledged.

REFERENCES

1. J. S. Jephson, R. P. McQuarrie, and R. E. Vogelsberg, "A Three Value Computer Design Verification System," IBM Syst. J., *8*, No. 3 (1969), pp. 178–188.
2. S. A. Szygenda, D. M. Rouse, and E. W. Thompson, "A Model and Implementation of a Universal Time Delay Simulator for Large Digital Nets," Proc. Joint Comp. Conf., AFIPS, Spring 1970, pp. 207–216.

3. S. Seshu, "On an Improved Diagnosis Program," IEEE Trans. Elec. Comp., *EC-14*, No. 1 (February 1965), pp. 76–79.
4. F. H. Hardie and R. J. Suhocki, "Design and Use of Fault Simulation for Saturn Computer Design," IEEE Trans. Elec. Comp., *EC-16*, No. 4 (August 1967), pp. 412–429.
5. D. B. Armstrong, "A Deductive Method for Simulating Faults in Logic Circuits," IEEE Trans. on Comp., *C-21*, No. 5 (May 1972), pp. 464–471.
6. H. Y. Chang, G. W. Smith, and R. B. Walford, "LAMP: System Description," B.S.T.J., this issue, pp. 1431–1449.
7. S. G. Chappell and S. S. Yau, "Simulation of Large Asynchronous Logic Circuits Using an Ambiguous Gate Model," Proc. Joint Comp. Conf., AFIPS, Fall 1971, pp. 651–661.
8. S. G. Chappell, "LAMP: Automatic Test Generation for Asynchronous Digital Circuits," B.S.T.J., this issue, pp. 1477–1503.
9. S. G. Chappell, H. Y. Chang, C. H. Elmendorf, and L. D. Schmidt, "A Comparison of Parallel and Deductive Simulation Techniques," IEEE Trans. Comput., *C-23*, No. 11 (Nobember 1974).
10. C. V. Ramamoorthy, "Analysis of Graphs by Connectivity Considerations," J. Assoc. Comp. Mach., *13*, No. 2 (April 1966), pp. 211–222.
11. T. G. Hallin, K. W. Johnson, and J. J. Kulzer, "LAMP: Application to Switching-System Development," B.S.T.J., this issue, pp. 1535–1555.
12. M. J. Flomenhoft, "A System of Computer Aids for Designing Logic-Circuit Tests," Proc. of SHARE/ACM/IEEE 1970 Design Automation Workshop, pp. 128–131.
13. "Revised Report on the Algorithm Language ALGOL 60," Comm. of ACM, *6*, No. 1 (January 1963), pp. 1–17.
14. D. Gries, *Compiler Construction for Digital Computers*, New York: John Wiley & Sons, 1971.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

LAMP:

Automatic Test Generation for Asynchronous Digital Circuits

By S. G. CHAPPELL

(Manuscript received February 28, 1974)

An automatic test generation system has been developed to detect faults in combinational and sequential circuits. The circuit model treats logic circuits as interconnections of unit- and zero-time-delay gates. A series of time-dependent Boolean equations are derived from the logic network (starting from the network inputs) in terms of sequences of signals (input vectors) on the circuit input leads. These equations account for the effect of specific circuit faults. Many tests, each consisting of a sequence of input signals (input vectors), are needed to detect all single faults in a circuit. Tests are generated from the time-dependent equations using two different strategies: (i) a maximum-cover approach to detect a large number of faults quickly by generating tests for the faults on the circuit-input leads. The fault-detection level achieved by the maximum-cover tests is then evaluated using fault simulation; (ii) tests for individual faults not detected by the maximum-cover approach. ATG has been implemented on the IBM 360, Model 67, and IBM 370, Model 168, computers.

I. INTRODUCTION

The automatic test generation system (ATG) was designed to provide fault-detection tests for single stuck-at faults in combinational and sequential circuits. Since this problem has essentially been solved for combinational circuits,[1–3] this paper concentrates on aspects of automatic test generation for sequential circuits.

The ATG algorithms presented attempt to account for actual circuit behavior as closely as possible. Hence, it is necessary to create

a computer model of the actual gates in the logic circuit. The circuit description used by ATG will utilize a unit/zero time-delay model, where a gate can assume one of three values: logical 0, logical 1, and don't-know X. This model has been widely used for logic-circuit simulation.[4,5] Because the test-generation algorithms described use the same model as many simulators, there are parallels between the simulation and test-generation techniques. These result from the effort to increase the accuracy of test generation to achieve the accuracy of current simulation techniques.

The major drawback of previous algorithms[6–8] for test generation for sequential circuits is the lack of a satisfactory model for the sequential circuit. Previous algorithms use either the Huffman model or an iterative combinational circuit model for sequential circuits. While these models are mathematically convenient, they are hardly accurate representations of real logic circuits. The system to be presented here has the following features:

(*i*) Requires no identification of feedback lines.
(*ii*) Allows gates to have time delays associated with their response to input stimuli.
(*iii*) Resolves races on flip-flops and detects circuit oscillations.
(*iv*) Assumes that an unknown circuit state corresponds to each gate having the unknown value X. [The X corresponds to value 3 in Ref. (5).]
(*v*) Generates a test for a single stuck-at or open-gate input fault, if it exists. The test is guaranteed to detect the fault (subject to the circuit-model assumptions).
(*vi*) Handles gate-level models of sequential circuits containing up to approximately 1000 gates.

For economy, the system allows test generation using two strategies. The first strategy (maximum cover) generates a set of tests designed to detect a large number of single faults without ever explicitly considering a specific fault. The second strategy generates tests for specified single faults. To allow rapid evaluation of the set of tests derived by the first strategy, a fault simulator is needed to simulate all single stuck-at faults. This simulator identifies the undetected set of faults that must be considered by the strategy-2 test generator. To keep the computation time reasonable, a user-specified parameter sets the maximum sequence length that will be considered by the system. The use and operation of the system is shown in the flow diagram in Fig. 1.

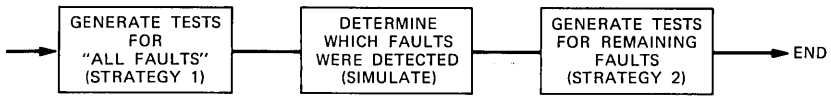


Fig. 1—Overall strategy.

II. MATHEMATICAL BASIS

This section builds the framework for the remainder of the paper. The behavior of some gate G will be described by two equations G^0 and G^1, where G^0 (G^1) describes the input conditions that set gate $G = 0$, ($G = 1$). The gates can assume one of three logical values: logical 0, logical 1, or the don't-know value X. Equations G^0 and G^1, however, are strictly Boolean equations in that the constituent variables of G^0 and G^1 can assume only values of 0 or 1. Similarly, G^0 and G^1 are Boolean variables.

2.1 Definitions

The following definitions are used in this discussion.

(*i*) Input vector: A string of n logical values (0, 1, and X, where X is a don't-know value) that applied to the n corresponding input leads of a circuit. The effect of these values is allowed to propagate through the circuit before the next input vector is applied to the circuit.

(*ii*) Test: A series of input vectors applied in a specific order to the circuit inputs. A test is also sometimes called a sequence. The first vector in each test assumes the circuit is in a completely unknown state. The nth vector ($N > 1$) assumes the state produced by the preceding $N - 1$ input vectors. Many tests may be required to detect all of the detectable faults in a logic circuit. Notice that it is not necessary to allow the circuit to stabilize between input vectors.

(*iii*) Sequence length: The number of input vectors in a test.

(*iv*) Input variables: Associated with each circuit input lead a are two binary input variables a^0 and a^1. The variables a^0 and a^1 can each take on Boolean values 0 and 1 (or "false" and "true"). Together, a^0 and a^1 define the logical value (0, 1, or X) of input lead a as shown in Table I. Hence, if $a^0 = 1$ (disallowing $a^0 = a^1 = 1$), then the logical value of lead a is 0. If $a^1 = 1$, then the logical value of lead a is 1. If neither $a^0 = 1$

Table I — Definition of a^0 and a^1

a^1 Value	a^0 Value	Lead a Logical Value
0	0	X
0	1	0
1	0	1
1	1	Impossible

nor $a^1 = 1$, then the value of input lead a is unknown or X. It is clearly impossible for input lead a to simultaneously have a logical value of 1 and 0. Therefore, $a^0 = a^1 = 1$ is an impossible situation. The variables a^0 and a^1 will often be used as an ordered pair (a^1, a^0). For example, (1, 0) represents the gate value of logical 1.

(*v*) Sequence of input variables: Let a^0i (a^1i), $i = 1, 2, \cdots$, represent the fact that $a = 0$ $(a = 1)$ during the ith input vector of a sequence. If no subscript is used (e.g., a^0 is written), then it is assumed that a represents the first input vector.

(*vi*) Notation: As is traditional, "+" represents logical OR, and "·" represents logical AND. The symbol "−" will be used to represent NOT or complement.

(*vii*) Unknown state: If the circuit is in an unknown state, it is assumed that each gate in the circuit is assigned the unknown output value X.

2.2 Properties of the equations

Some of the properties of input variables a^0 and a^1 are described in this section. Consider a circuit consisting of a two-input AND gate c with inputs a and b. Input leads a and b have associated with them (a^1, a^0) and (b^1, b^0), respectively. The problem is to compute (c^1, c^0). The usual truth table for AND is shown below.

		a		
	AND	0	1	X
	0	0	0	0
b	1	0	1	X
	X	0	X	X

Translating this to the ordered-pair notation, we have:

	AND	(a^1, a^0) (0, 1)	(1, 0)	(0, 0)
	(0, 1)	(0, 1)	(0, 1)	(0, 1)
(b^1, b^0)	(1, 0)	(0, 1)	(1, 0)	(0, 0)
	(0, .0)	(0, 1)	(0, 0)	(0, 0)

Examining these ordered pairs, one finds that $c^0 = 1$ if and only if (iff) $a^0 = 1$ or $b^0 = 1$. Similarly, $c^1 = 1$ iff both $a^1 = 1$ and $b^1 = 1$. Hence, the following relations hold for a two-input AND gate c with inputs a and b:

$$c^0 = a^0 + b^0$$
$$c^1 = a^1 \cdot b^1$$

or

$$(c^1, c^0) = (a^1 \cdot b^1, a^0 + b^0). \tag{1}$$

It is important to note that c^0 is not necessarily the complement of c^1. For example, if $(a^1, a^0) = (0, 0)$ and $(b^1, b^0) = (1, 0)$, then $(c^1, c^0) = (0 \cdot 1, 0 + 0) = (0, 0)$.

A similar set of relations can be derived for a two-input OR gate f with inputs d and e.

$$(f^1, f^0) = (d^1 + e^1, d^0 \cdot e^0). \tag{2}$$

The interpretation of this is that $f = 1$ if either $d = 1$ or $e = 1$ or both. Similarly, $f = 0$ if both $d = 0$ and $e = 0$.

Another relation can be derived for the NOT gate (or inverter) h with input g as follows. Note that the complement of X is still X.

$$(h^1, h^0) = (g^0, g^1). \tag{3}$$

For later use, the relations governing the NAND gate are also presented here. The NAND gate is simply an AND gate followed by a NOT gate. Hence, we have, for a two-input NAND gate w with input y and z:

$$(w^1, w^0) = (y^0 + z^0, y^1 \cdot z^1). \tag{4}$$

The above definitions have been presented for two-input gates. However, since the functions AND and OR are associative, the equations for an N-input gate can easily be derived. For example, for a three-input NAND gate w with inputs p, y, and z, we have:

$$(w^1, w^0) = (p^0 + y^0 + z^0, p^1 \cdot y^1 \cdot z^1). \tag{5}$$

Notice that since a^0 and a^1 are binary variables, they obey all the laws of Boolean algebra. However, the interactions of a^0 and a^1 are not so obvious and are of interest here.

It is significant that in the algorithms presented for computing the output equations for a gate [eqs. (1) through (5)], we have never produced a $\bar{G}^0$, $\bar{G}^1$, or $\bar{G}$ expression, where G is any gate in the circuit. This has occurred for two reasons. First, because gate G can assume three values $\bar{G}$ is not particularly useful. For example, if $G = 1$, then $\bar{G} = 0 + X$. Second, as a practical matter, the computation of $\bar{G}^0$ or $\bar{G}^1$, given G^0 or G^1, is quite time consuming if both input and output are to be in sum-of-products form.

2.3 Some identities and nonidentities

After the operations AND, OR, and NOT have been defined, further properties can be investigated. By simple examination of the definitions for AND, OR, and NOT, the following identities are obvious. Let a represent any gate in the circuit. For ease of understanding, the corresponding theorem of Boolean algebra is written on the same line as the identities but enclosed in brackets.

(*i*) $(0, 1)\cdot(a^1, a^0) = (0, 1)$ $[0\cdot a = 0]$.

(*ii*) $(1, 0)\cdot(a^1, a^0) = (a^1, a^0)$ $[1\cdot a = a]$.

(*iii*) $(a^1, a^0)\cdot(a^1, a^0) = (a^1, a^0)$ $[a\cdot a = a]$.

(*iv*) $(1, 0) + (a^1, a^0) = (1, 0)$ $[1 + a = 1]$.

(*v*) $(0, 1) + (a^1, a^0) = (a^1, a^0)$ $[0 + a = a]$.

(*vi*) $(a^1, a^0) + (a^1, a^0) = (a^1, a^0)$ $[a + a = a]$.

(*vii*) $(a^1, a^0)\cdot(b^1, b^0) = (b^1, b^0)\cdot(a^1, a^0)$ [Commutative].

Proof:
$$(a^1, a^0)\cdot(b^1, b^0) = (a^1\cdot b^1, a^0 + b^0) = (b^1\cdot a^1, b^0 + a^0) = (b^1, b^0)\cdot(a^1, a^0) \quad \text{QED.}$$

Similarly,

(*viii*) $(a^1, a^0) + (b^1, b^0) = (b^1, b^0) + (a^1, a^0)$ [Commutative].

(*ix*) $[(a^1, a^0)\cdot(b^1, b^0)]\cdot(c^1, c^0) = (a^1, a^0)\cdot[(b^1, b^0)\cdot(c^1, c^0)]$ [Associative].

Proof:
$$\begin{aligned}[(a^1, a^0)\cdot(b^1, b^0)]\cdot(c^1, c^0) &= ([a^1\cdot b^1]\cdot c^1, [a^0 + b^0] + c^0) \\ &= (a^1\cdot[b^1\cdot c^1], a^0 + [b^0 + c^0]) \\ &= (a^1, a^0)\cdot[(b^1, b^0)\cdot(c^1, c^0)] \quad \text{QED.}\end{aligned}$$

Similarly,

(*x*) $[(a^1, a^0) + (b^1, b^0)] + (c^1, c^0) = (a^1, a^0) + [(b^1, b^0) + (c^1, c^0)]$ [Associative].

(*xi*) $(a^1, a^0)\cdot(b^1, b^0) + (a^1, a^0) = (a^1, a^0)$ $[ab + a = a]$.

Proof: $(a^1, a^0)\cdot(b^1, b^0) + (a^1, a^0)$
$= (a^1\cdot b^1, a^0 + b^0) + (a^1, a^0)$
$= (a^1\cdot b^1 + a^1, [a^0 + b^0]\cdot a^0)$
$= (a^1, a^0 + a^0b^0) = (a^1, a^0)$ QED.

(*xii*) $a^0\cdot a^1 = a^1\cdot a^0 = 0$.

This is obviously true if a is a circuit input lead. Because the computation of the equations proceeds from gate inputs to gate outputs, this result can be shown inductively. For any valid circuit state (gates have logical values 0, 1, or X), the theorem is true. It is also true for input leads, as mentioned earlier. Then, by examination of eqs. (1) through (5) above, we see that the relationship is preserved when the new gate output equations are computed. Hence, by induction, it follows that the relationship holds for every gate in the circuit.

(*xiii*) $\overline{[(a^1, a^0)\cdot(b^1, b^0)]} = (a^0, a^1) + (b^0, b^1)$ $[\overline{(a\cdot b)} = \bar{a} + \bar{b}]$.

Proof: $\overline{[(a^1, a^0)\cdot(b^1, b^0)]} = \overline{(a^1\cdot b^1, a^0 + b^0)}$
$= (a^0 + b^0, a^1\cdot b^1) = (a^0, a^1) + (b^0, b^1)$ QED.

(*xiv*) $\overline{[(a^1, a^0) + (b^1, b^0)]} = (a^0, a^1)\cdot(b^0, b^1)[\overline{(a + b)} = \bar{a}\cdot\bar{b}]$.

Proof: $\overline{[(a^1, a^0) + (b^1, b^0)]} = \overline{(a^1 + b^1, a^0\cdot b^0)}$
$= (a^0\cdot b^0, a^1 + b^1) = (a^0, a^1)\cdot(b^0, b^1)$ QED.

Again, it is clear that identities (*xiii*) and (*xiv*) can easily be extended to several variables (e.g., $\overline{(a\cdot b\cdot c)} = \bar{a} + \bar{b} + \bar{c}$). These are simply DeMorgan's theorems.

The identities above simply follow the Boolean algebra. The following set of nonidentities results primarily from the three values used to model the gate behavior.

(*i*) $a^0 + a^1 \neq 1$.

Proof by example: $(a^1, a^0) = (0, 0)$.
Clearly, $0\cdot 0 \neq 1$.
This is not unexpected since the only relation between a^0 and a^1 is that $a^0\cdot a^1 = 0$.

(*ii*) $(a^1, a^0)\cdot(b^1, b^0) + (a^1, a^0)\cdot(b^0, b^1) \neq (a^1, a^0)$ $[a\cdot b + a\cdot\bar{b} \neq a]$.

(*iii*) $a\cdot c + \bar{a}\cdot b\cdot c \neq b\cdot c + a\cdot c$.

(*iv*) $a\cdot b + \bar{a}\cdot c + b\cdot c \neq a\cdot b + \bar{a}\cdot c$.

Nonidentities (*ii*), (*iii*), and (*iv*) are easily proved by examining the truth tables, where the variables are allowed to assume three values: logical 0, logical 1, and the don't-know value X.

It is interesting to note that if we required the circuit input leads to have only values 0 and 1, the system presented here would reduce to Boolean algebra with $a^0 = \bar{a}^1$ and $a^1 = \bar{a}^0$. This is a reasonable restriction, since we could always require that any X values generated for the input leads be arbitrarily set to logical 0 or 1. However, it would then be necessary to treat input leads differently from other gates in the circuit, since it is clearly not possible to force every gate in the circuit to a known value (logical 0 or 1). Hence, the generality of allowing circuit input leads to assume the value X is retained in this paper and all gates are treated identically.

III. EQUATION DERIVATION FOR LOGIC NETWORKS

The operation of the ATG has two well-defined steps. The first step is to derive a set of relations (equations) that represent the behavior of the logic circuit. The second step is to derive a set of tests for the circuit based on the equations derived in the first step. In this section the equation-derivation process is described.

The equation derivation process essentially reduces the behavior of a logic circuit to a series of equations. Hence, this reduction process is quite critical. These equations must reflect the true circuit behavior as closely as is possible (or economical). This means that the time delay of gates must be accounted for during the equation-generation process. The equation-generation process will first be presented using a fault-free, unit/zero, time-delay model for each gate. The model will then be extended to account for single stuck-at-one and stuck-at-zero faults.

The method is essentially a dynamic equation-generation process that determines exactly those input sequences that will force each gate to a logical 0 or 1 at each instant of time. The equation-derivation process begins with the circuit inputs and continues through the circuit until the equations are stable; that is, until the output equation on each gate is consistent with the input equation on that gate. The equations are derived in terms of circuit input variables only; no feedback lines need be identified. The input variables may change several times before the circuit finally reaches a stable state.

Since the objective is to generate tests to detect faults in a circuit, the result of this process will be a series of logical values 0, 1, and X (don't know) to be applied to the input leads of the circuit. The output of the circuit will then be observed to determine which classical faults have been detected. That is, the output of the real (perhaps faulty) circuit will be compared to the expected result to determine if the real circuit is performing correctly.

3.1 Fault-free-equation derivation

In this section, the problem of generating equations that represent the behavior of the fault-free circuit is discussed. Because certain simplifications are possible, the equation-derivation process for combinational circuits is discussed first. This is followed by the equation-derivation process for sequential circuits.

3.1.1 Equation derivation for combinational circuits

The derivation of the fault-free equations will be considered here. Consider the NAND gate G shown in Fig. 2. If we assume both inputs to the gate are circuit inputs or other gate outputs, then we have:

$$G^0 = A^1 \cdot B^1$$
$$G^1 = A^0 + B^0.$$

Equation G^0 denotes exactly those input conditions to gate G that force (or set) gate G to logical 0. Implicit in this equation is the unit-delay assumption. If inputs $A = 1$ and $B = 1$ are applied at time t, then the output of G is forced to logical 0 at time $t + 1$. A similar situation exists for G^1. Either $A = 0$ or $B = 0$ (or both) applied at time t to the inputs of G forces its output to be logical 1 at time $t + 1$. This is similar to eqs. (1) through (5) in the previous section, except that the element of time has been added. For most gates, the output of the gate responds to the input stimuli one unit of time later. The gates with one unit of delay are "real" gates, e.g., those containing an active semiconductor device.

In some logic families it is possible to directly connect two (or more) gate output leads together. This connection (called a TIC here, for tied collector) performs a logic function. If the ground level is logical 0, then the TIC function is AND. If the ground level is logical 1, then the TIC function is OR. The TICs may be considered zero-delay gates except for the wire-propagation delay, which is not considered here.

If computation begins at the circuit inputs, which are assumed to be applied at time t, the output of each gate driven by a primary input is reevaluated and the new equation is assigned to the gate output at time $t + 1$. Every gate whose input equation changed at time $t + 1$ is reevaluated and its new output is assigned at time $t + 2$. This process continues until the computation reaches the circuit output

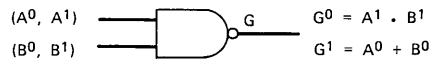


Fig. 2—Equations based on input equations.

gates. At any time $t + i$ after the processing begins, the equations denote those input conditions that force each gate to logical 0 and 1 at i gate delays after application of the vector. In particular, when the circuit has settled to a stable state, the input values that set each gate to logical 0 or 1 are specified. Notice that no assumptions have been made that would preclude the application of this argument to sequential circuits.

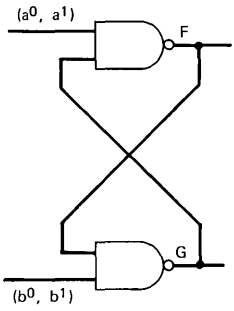
The similarity between this procedure and the actual propagation of electrical signals through the circuit should be evident. In both cases, the input stimuli are applied to the circuit inputs and are allowed to propagate through the circuit.

3.1.2 Equation derivation for sequential circuits

Three points are significant in the discussion of equation derivation for combinational circuits: (*i*) the process assumes all gates have either unit or zero delay, (*ii*) the process starts from the circuit inputs and proceeds through the circuit much as a signal would propagate through the circuit, and (*iii*) the equations define, for each time $t + i$, exactly those input conditions that cause each gate in the circuit to be forced to logical 0 and 1 at that time from the specified initial state. Again, there are no assumptions that limit this technique to combinational circuits.

The primary addition, which must be made to allow the same algorithm to be applied to sequential circuits, is some provision for deciding when to stop the computation. For combinational circuitry, the computation stops when the circuit outputs are reached. However, this is not satisfactory for sequential circuits. The equation derivation yields G^0 and G^1 for each gate G for each time $t + i$. If both G^0 and G^1 at time $t + i$ are equal to G^0 and G^1 at time $t + i + 1$, the gate is in a stable state. Otherwise, each gate driven by gate G must be reevaluated since G changed values (output equations). A detailed flow chart of the equation computation process will be presented later.

Let $a.i$ represent the value of circuit input lead a during the ith vector of the test (or sequence). Similarly, a^0i (a^1i) means make input lead a logical 0 (logical 1) during the ith input vector of the sequence. The first vector in each sequence is number one. If the sequence number is missing, then it is assumed to represent the first vector of the sequence. An example of the application of this algorithm is shown in Fig. 3 where the equations for a flip-flop are calculated. Time runs down the page. The flip-flop is assumed to start from the unknown state since $F^0 = F^1 = G^0 = G^1 = 0$. The inputs are assumed to be



F^0	F^1	TIME
0	0	t
0	a^0	t + 1
$a^1 \cdot b^0$	a^0	t + 2
$a^1 \cdot b^0$	$a^0 + a^0 \cdot b^1 = a^0$	t + 3

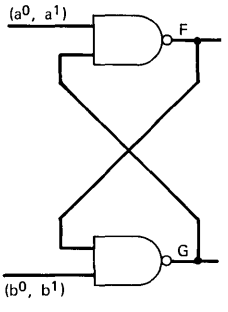
G^0	G^1	TIME
0	0	t
0	b^0	t + 1
$a^0 \cdot b^1$	b^0	t + 2
$a^0 \cdot b^1$	$b^0 + a^1 \cdot b^0 = b^0$	t + 3

Fig. 3—Equations for NAND flip-flop from an unknown state.

applied at time t. At time $t + 1$, only F^1 and G^1 changed values so at time $t + 2$ only G^0 and F^0 are calculated. At time $t + 3$, none of the new output equations changed, so the circuit is stable and computation stops.

A similar computation can be carried out if the circuit is in some known initial state. This is illustrated in Fig. 4 where the circuit is initially set at $F = 0$ ($F^0 = 1$, $F^1 = 0$) and $G = 1$ ($G^0 = 0$, $G^1 = 1$).

The above procedure finds the next state function for a combinational or sequential circuit. That is, given a circuit state (possibly unknown), we can find all possible next states resulting from the application of one input vector.



F^0	F^1	TIME
1	0	t
a^1	a^0	t + 1
a^1	a^0	t + 2
a^1	$a^0 + a^0 \cdot b^1 = a^0$	t + 3

G^0	G^1	TIME
0	1	t
0	1	t + 1
$a^0 \cdot b^1$	$a^1 + b^0$	t + 2
$a^0 \cdot b^1$	$a^1 + b^0$	t + 3

Fig. 4—Equations from $F = 0$, $G = 1$ state.

Clearly, the problem is to determine the state of the circuit as the result of each possible sequence of vectors. If this can be done, then there is no need to select a particular next state since they are all considered simultaneously. A method for doing this is described in the next section.

3.1.3 Sequence derivation for sequential circuits

The algorithm for sequence derivation is based only on the input behavior of the circuit. There is no need to consider any feedback variables. Again, the derivation assumes either unit- or zero-delay gates.

This algorithm is based on the explanation presented in the previous section. The derivation proceeds as follows for a sequence of length M.

(*i*) To the circuit inputs $[a, b, c, \cdots] = I$ apply the variables $[a^*1, b^*1, c^*1, \cdots] = I.1$ (a^* means apply a^1 or a^0) and derive the equations for the circuit (starting from any initial state).

(*ii*) Let $j = 1$.

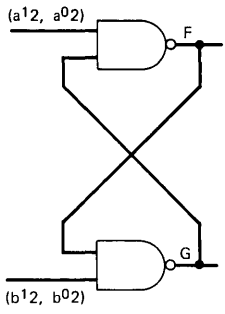
(*iii*) From the circuit "state," as defined by the application of the input vector of variables $I.j = [a^*j, b^*j, c^*j, \cdots]$, apply the input vector of variables $I.j + 1$ and propagate these variables through the circuit, i.e., derive the "equations" for the circuit in terms of $I.j + 1$ and $I.k$ for all $k \leq j$. The effect of $I.j$ need not stabilize before applying $I.j + 1$.

(*iv*) If $j < M$, then let $j = j + 1$ and go to step (*iii*). Otherwise, exit.

This procedure models the behavior of a logic circuit. The input stimuli (variables) are applied to the inputs of the circuit and allowed to propagate through the circuit. The input vector $I.1$ assumes the circuit is in some initial state, which is probably unknown. Input vector $I.2$ produces equations from the initial state produced by $I.1$. In general, the vector $I.j$ starts from the state produced by all $I.k$ where $k < j$.

After application of $I.j$, the effect on the circuit of any sequence of j vectors is known. This is obvious since we have already shown that the application of $I.1$ from any state produces the equations G^0 and G^1 for every gate in the circuit as a result of $I.1$. This extension makes the initial state for $I.j$, $j \geqq 2$, a function of all $j - 1$ vectors.

An application of this algorithm is shown in Figs. 3 and 5 for the NAND flip-flop. Here $I.j = (a^*j, b^*j)$ and the sequence derivation is carried out for sequences of length 2 or less. Figure 3 represents sequences of length 1. Figure 5 represents sequences of length 2. The



F^0	F^1	TIME
$a^11 \cdot b^01$	a^01	t + 3
~~$a^12 \cdot b^01$~~	$a^01 \cdot b^11 + a^02$	t + 4
$a^11 \cdot a^12 \cdot b^01 + a^12 \cdot b^01 \cdot b^02$		RACE
$a^11 \cdot a^12 \cdot b^01 + a^12 \cdot b^02$	$a^01 \cdot b^11 \cdot b^12 + a^02$	t + 5

G^0	G^1	TIME
$a^01 \cdot b^11$	b^01	t + 3
~~$a^01 \cdot b^12$~~	$a^11 \cdot b^01 + b^02$	t + 4
$a^01 \cdot b^11 \cdot b^12 + a^01 \cdot a^02 \cdot b^12$		RACE
$a^01 \cdot b^11 \cdot b^12 + a^02 \cdot b^12$	$a^11 \cdot a^12 \cdot b^01 + b^02$	t + 5

Fig. 5—Result of second vector of sequence.

computation for sequences of length 2 in Fig. 5 begins from the final state of the computation for sequences of length 1 shown in Fig. 3. To illustrate the interpretation of the equations, consider the final state of $G^0 = a^02 \cdot b^12 + a^01 \cdot b^11 \cdot b^12$ shown in Fig. 5. This means that the sequence of length 1, $a = 0$ and $b = 1$ (for $a^02 \cdot b^12$), or the sequence of length 2, $a = 0$ and $b = 1$ followed by $a = X$ and $b = 1$ (for $a^01 \cdot b^11 \cdot b^12$), will set gate G to logical 0.

3.1.4 Equation race analysis

A race occurs on the simple two-NAND-gate flip-flop shown in Fig. 3 when the output state of the flip-flop is unpredictable from the input conditions. Under these circumstances, the outputs of the flip-flop must be set to the unknown value X. Let us examine $F^0 = b^01 \cdot a^12$ and $G^0 = a^01 \cdot b^12$ at time $t + 4$ in Fig. 5. Since $F^0 \cdot G^0 \neq 0$, then both $b^01 \cdot a^12$ and $a^01 \cdot b^12$ could be simultaneously applied to the circuit inputs producing the sequence $a^01 \cdot b^01 \cdot a^12 \cdot b^12$. This represents the application to the flip-flop of the sequence $a = 0$ and $b = 0$ followed by $a = 1$ and $b = 1$. This produces the race state (unpredictable output conditions) for the NAND flip-flop and must therefore be eliminated. The race state for our example is $a = b = F = G = 1$ at some time t. If a race occurs, the values computed and saved for time $t + 1$ are $F = G = 0$. In addition, when unknown states are allowed, a race is also declared if $F = 0$ and $G = X$ or $F = X$ and $G = 0$ at the same instant of time.[5] This implies that when $F = 0$, G cannot be 0 or X. Thus, to eliminate races, it is necessary to demand that if $F = 0$ then $G = 1$ and, similarly, if $G = 0$ then $F = 1$ at the same instant

of time. This is accomplished in our example by forming the new equation F^0n at time $t + 4$ as $F^0n(t + 4) = F^0(t + 4) \cdot G^1(t + 4)$ $= a^11 \cdot a^12 \cdot b^01 + a^12 \cdot b^01 \cdot b^02$. Similarly, $G^0n(t + 4) = a^01 \cdot a^02 \cdot b^12$ $+ a^01 \cdot b^12 \cdot b^11$. This process is called race analysis since it prevents the equations from causing simple flip-flops (basically, two cross-coupled NAND or NOR gates) to race.

Race analysis must be performed at time t if both F^0 and G^1 changed at time t, where F and G are the two gates in a simple flip-flop. While this result is shown here for the NAND flip-flop, the proof can easily be extended to NOR flip-flops. The primary difference is that F^1 and G^1 must be modified for NOR flip-flops while F^0 and G^0 must be modified for NAND flip-flops.

(*i*) If F^0 does not change, then gate F cannot change to logical 0 at time t; therefore, there can be no race.

(*ii*) If G^1 (see Fig. 5) does not change at time t, then $F^0(t)$ was formed by ANDing together $G^1(t - 1)$ and $a^1(t - 1)$. That is, $F^0(t) = G^1(t - 1) \cdot a^1(t - 1)$. But $G^1(t - 1) = G^1(t)$ by assumption. Race analysis would form

$$F^0n(t) = F^0(t) \cdot G^1(t) = F^0(t) \cdot G^1(t - 1)$$
$$= a^1(t - 1) \cdot G^1(t - 1) \cdot G^1(t - 1)$$
$$= a^1(t - 1) \cdot G^1(t - 1) = F^0(t).$$

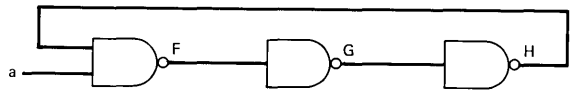
Therefore, the new F^0n resulting from race analysis is the same as the original F^0. Then, there can be no race.

Earlier it was shown that $F^0 \cdot F^1 = 0$ for any gate F at any time. It is easily seen that race analysis does not destroy this property since, if $F^0(t) \cdot F^1(t) = 0$ and $F^0n(t) = F^0(t) \cdot G^1(t)$, then $F^0n(t) \cdot F^1(t)$ $= F^0(t) \cdot G^1(t) \cdot F^1(t) = 0$.

3.1.5 Equation oscillations

It is possible that the equation computation process will never terminate. That is, the old equations on some gate are always different from the new equations on that gate. This situation is known as equation oscillation. If the computation described in Section 3.1.3 proceeds through an arbitrary number (user declared) of timing lists, then an oscillation is declared and the message "equation oscillation" is printed for the user.

An example of an oscillation is shown in Fig. 6. In general, the objective is to stop the oscillation by selecting a stable set of equations. This can usually be done by setting the new equation on an oscillating



F^0	F^1	G^0	G^1	H^0	H^1	TIME
0	1	1	0	0	1	t
a^1	a^0	1	0	0	1	t + 1
a^1	a^0	a^0	a^1	0	1	t + 2
a^1	a^0	a^0	a^1	a^1	a^0	t + 3
0	$a^0 + a^1$	a^0	a^1	a^1	a^0	t + 4
0	$a^0 + a^1$	$a^0 + a^1$	0	a^1	a^0	t + 5
0	$a^0 + a^1$	$a^0 + a^1$	0	0	$a^0 + a^1$	t + 6
a^1	a^0	$a^0 + a^1$	0	0	$a^0 + a^1$	t + 7
a^1	a^0	a^0	a^1	0	$a^0 + a^1$	t + 8

Fig. 6—Equation oscillation.

gate, say $F^0(t)$ equal to $F^0(t) \cdot F^0(t-1)$. This is intended to force the equations on gate F to stabilize by generating equations that make $F^0(t) = F^0(t-1)$. This technique is not guaranteed to resolve all oscillations.

3.1.6 Complete description of equation derivation

The complete algorithm for generating the equations for a sequential circuit is shown in Fig. 7. Only two parts of the flow chart have not been explained previously in this section. One of these parts is the method of handling the zero-delay gates. The output of all the zero-delay gates are calculated before the next list of unit-delay gates is processed. These output equations are assigned to the zero-delay gates immediately.

The remaining unexplained part is the initial-state pass. This pass simply examines the circuit and propagates forward (before the input variables are applied) the effect of any gates set to logical 0 or 1 and any faults. For example, if gate G drives gate H and gate G is set to logical 0, this pass determines that the output of H should be logical 1.

This completes the description of the equation-generation process for fault-free sequential circuits. Next, the algorithm for generating the equations for a sequential circuit in the presence of a single fault is described.

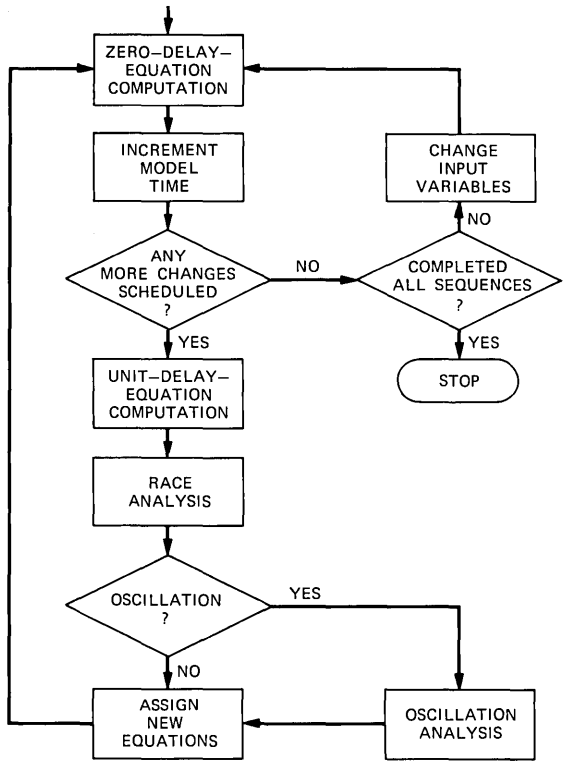


Fig. 7—Equation computation flow chart.

3.2 Equations containing faults

Equations for circuits containing faults may be derived in a way similar to those used for fault-free circuits. This method allows tests to be generated that can detect a specific fault. For efficiency, it is possible to consider several single faults simultaneously. The single faults considered here are the gate outputs stuck-at-one and stuck-at-zero as well as gate inputs open (e.g., open diode or emitter). The input open on a NAND or AND gate will be treated as stuck-at-one while the input open on a NOR or OR will be treated as stuck-at-zero.

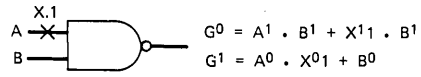
Let the variables x^0i and x^1i represent fault variables. Let x^1i mean the fault $x.i$ is present in the circuit. Similarly, x^0i means the fault is not present in the circuit. For the fault variables, the i does not represent the ith vector in a sequence; rather, it represents the ith fault being considered. (Faults are always denoted by $x.i$ and the

associated variables by x^0i or x^1i.) Since the single faults are assumed to be permanent, any fault $x.i$ will be present during the entire test sequence. The two states for $x.i$ allow the comparison of the faulty and fault-free circuit behavior to derive a test to detect the presence or absence of the fault in the circuit.

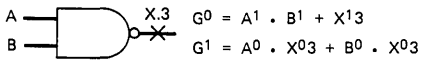
Consider the gate shown in Fig. 2. The fault-free equations are shown. If, however, the input-open fault on gate G from A is being examined, then the equations for gate G are shown in Fig. 8a. It is possible to set gate G to logical 0 either by applying A^1 and B^1 in the presence or absence of fault $x.1$ or by applying B^1 in the presence of fault $x.1$. It is also possible to set G to logical 1 by applying B^0 in the presence or absence of fault $x.1$ or by applying A^0 in the absence of fault $x.1$. Similar analysis for the output stuck-at-zero fault $x.3$ and the output stuck-at-one fault $x.4$ can easily be performed in the manner shown in Figs. 8b and 8c.

Now assume there is only one fault in the circuit and consider the case where the fault propagates around a loop and returns to the site of the failure. If the fault is the input open on gate G from A, then the equations shown in Fig. 8a can be rewritten as shown in Fig. 9a where fault $x.1$ is explicitly considered and D, E, F, $\cdots$ represent sum-of-products equations. Computing G^0 and G^1 yields the equations shown in Fig. 9a. Figure 9b considers the case where the fault exists (x^11) and Fig. 9c considers the case where the fault does not exist (x^01). Comparison of Figs. 9b and 9c with 9a shows that the computations proposed in this section for combinational circuits are also applicable to sequential circuits for the input-open case.

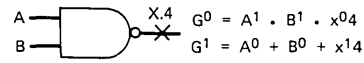
A similar analysis can be carried out for the gate output stuck-at-one and the output stuck-at-zero faults. This demonstrates that the equations shown in Fig. 8 for handling faults in combinational circuits are also applicable to sequential circuits.



(a) INPUT OPEN FAULT EQUATIONS



(b) OUTPUT STUCK–AT–ZERO EQUATIONS



(c) OUTPUT STUCK–AT–ONE EQUATIONS

Fig. 8—Equations for handling faults in combinational circuits.

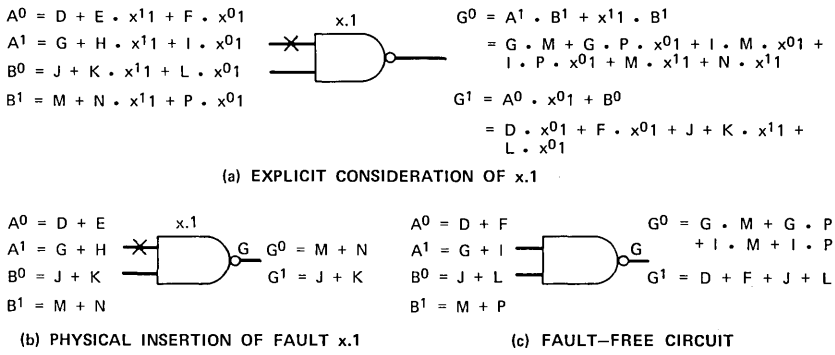


Fig. 9—Equations for handling faults in sequential circuits.

3.3 The halting problem

One problem that must be discussed is how to determine when sequences of sufficient length have been generated. That is, given the equations that represent sequences of length N and the equations that represent sequences of length $N + 1$, will more information be gained by generating sequences of length $N + 2$? The question is answerable[9] if the feedbacks have been identified; however, the maximum sequence length contains factors of the form 2 to the power m, where m is the number of circuit inputs. For 500-gate, 40-input circuits, this is an absurd number.

There does not appear to be any practical method of determining when to halt the equation-generation process. In practice, the maximum sequence length to be considered is supplied by the user. The usual procedure is to start with sequences of length 1 and increase the sequence length until an acceptable level of undetected faults remains using the test-generation schemes presented in the next section. As might be expected, the run time increases significantly with increasing sequence length such that, even if it were simple to determine when to halt, it would probably not be economical. In practice, the halting problem has presented no difficulties. It is, however, an interesting theoretical problem.

In practice, the maximum sequence length required to detect all faults in the circuit provides some measure of the ease with which the circuit can be tested. The shorter the sequence length required, the more easily the circuit can be tested. This fact could be used as a circuit-design constraint by requiring that all circuits be testable with sequences of N or less where N is small. In fact, a 1000-gate, 11-state

sequencer was designed so that the flip-flops representing the state could be written and read directly from circuit inputs and outputs. This produced an easily testable sequential circuit.

3.4 Clocked circuits

The algorithms that have been presented allow the circuit input leads to be treated as variables [e.g., (a^1, a^0)] or as logical values where logical 0 is (0, 1) and logical 1 is (1, 0). It is possible to allow some circuit inputs to be represented by variables and others by logical values. Clearly, it is possible to change the logical values between logical 0 and 1. Then we have the ability to apply a sequence of logical values to an input lead.

For example, suppose the circuit being considered has a clock lead whose normal operating waveform is 1–0–1–0 and all other input leads are static during this cycle. Then it is possible to apply variables to all but the clock lead and to supply the waveform (1, 0) – (0, 1) – (1, 0) – (0, 1) to the clock lead. In this way, ATG does less work since we have considered a sequence of length 4 on the clock lead and sequences of length 1 on all other leads. This is considerably more economical than computing sequences of length 4 over all input leads.

In a similar way, user-specified initialization sequences can be applied to the circuit to place it in some desired state before allowing ATG to select the next input sequence. This is an effective way of using ATG.

3.5 Self-initializing circuits

Certain classes of sequential circuits are self-initializing in that, regardless of the initial state, the circuit always assumes a known state when power is applied. A simple example of such a circuit is shown in Fig. 10. Because the flip-flop always initializes to $C = 0$, $D = 1$ or $C = 1$, $D = 0$, gate F will always be logical 0 forcing the flip-flop to the $C = 1$, $D = 0$ state.

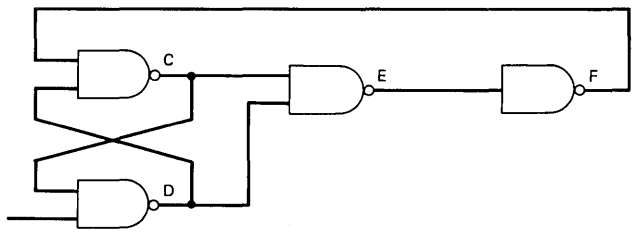


Fig. 10—Self-initializing circuit.

If this circuit is assumed to be in an unknown state ($B = C = D = E = F = X$), then because $\bar{X} = X$, the basic ATG algorithm does not determine the required initial state. The operation of ATG requires that gates be forced to some initial state by the application of input vectors from some initial state. Hence, self-initializing circuits require that the proper initial state be specified by the user. This has not proved to be a problem in practice since most circuits contain initializing leads.

IV. TEST GENERATION FROM THE EQUATIONS

Two different schemes for generating tests are described in this section. The first scheme described is the generation of tests to detect single faults, where the equations are derived in terms of these faults. However, in a 500-gate, 2000-fault circuit it is not economical to attack all 2000 faults on a one-at-a-time basis. The second method for test generation is aimed at detecting large numbers of faults as easily as possible. It attacks the problem by essentially attempting to detect the stuck-at-one and stuck-at-zero fault at each circuit input lead by observing each circuit output lead. This is called the maximum-cover strategy. This scheme typically detects around 90 percent of the classical faults if the equations reasonably describe the circuit—that is, if the sequence length used is long enough.

4.1 Generating a test to detect a fault

To detect fault $x.i$, it is necessary to select a test (input sequence) that will force some output of the circuit to have the value k for $k = 0, 1$ in the presence of the fault $x.i$ and to have the value $\bar{k}$ in the absence of fault $x.i$ starting from the given initial state. Let one output gate G of a circuit have the following equations (by simple factoring):

$$\begin{aligned} G^0 &= A + B \cdot x^1 i + C \cdot x^0 i \\ G^1 &= D + E \cdot x^1 i + F \cdot x^0 i, \end{aligned} \tag{6}$$

where $A, B, \cdots, F$ are also sum-of-products expressions. This means that the terms in $A(D)$ are the only terms that set $G = 0$ ($G = 1$) regardless of the presence or absence of fault $x.i$. The tests to detect fault $x.i$ at gate G are given by $B \cdot F + C \cdot E$. This is proven as follows.

Since the fault either exists or does not exist, $x^1 i \cdot x^0 i = 0$. First consider the case in which $k = 0$. Since $G^1(G^0)$ represents exactly those conditions that set $G = 1$ ($G = 0$), then $G^1(x^0 i = 1) = D + F$ represents those conditions that set $G = 1$ in the absence of fault $x.i$. Simi-

larly, $G^0(x^1i = 1) = A + B$ represents those conditions that set $G = 0$ in the presence of the fault. Hence, every condition (input vector) that makes the good output of $G = 0$ and makes the faulty output of $G = 1$ is given by $(A + B)\cdot(D + F) = A\cdot D + B\cdot D + A\cdot F + B\cdot F$. Examination of the terms of this equation reveals $A\cdot D = B\cdot D = A\cdot F = 0$. Term $A\cdot D = 0$ because, if it were not zero, then there would be some term in $A\cdot D$ that could set $G = 1$ and $G = 0$ simultaneously. This is clearly impossible. Similarly, $B\cdot D \neq 0$ $(A\cdot F \neq 0)$ implies that in the presence (absence) of the fault, there is some term in $B\cdot D(A\cdot F)$ that can set $G = 1$ and $G = 0$ simultaneously. Therefore, any term that can set $G = 0$ in the presence of the fault and $G = 1$ in the absence of the fault must be in $B\cdot F$.

For the case in which $k = 1$, the test must be a term of $(D + E)\cdot(A + C) = D\cdot A + D\cdot C + E\cdot A + E\cdot C$. By similar analysis, $A\cdot D = D\cdot C = A\cdot E = 0$. Therefore, a term that sets $G = 1$ in the presence of the fault and $G = 0$ in the absence of the fault must be in $E\cdot C$.

Since the problem is to detect fault $x.i$ without regard to the output value of G, any term in $B\cdot F + E\cdot C$ is a valid test. Therefore, all tests to detect fault $x.i$ at gate G can be expressed as

$$\text{Detection Tests} = B\cdot F + E\cdot C.$$

If $B\cdot F + E\cdot C = 0$, there is no test that will detect fault $x.i$ at gate G. It is then necessary to examine each remaining circuit output to determine if $x.i$ is detectable. If $x.i$ is not detectable at any circuit output, then there exists no test to detect $x.i$ for the sequence length specified.

Clearly, this algorithm generates every test that will detect fault $x.i$ at each output. Since it is probably necessary to detect the fault only once, the first valid test found usually terminates the process.

4.2 The maximum-cover strategy

The maximum-cover strategy has been quite successful. In most cases, it has detected from 85 to 100 percent of the faults in the circuit that are detectable with the maximum sequence length specified. For highly sequential circuits, a short-maximum-sequence length may detect few faults because the circuit cannot be exercised completely without using a long sequence of input vectors.

The maximum-cover strategy operates on the fault-free equations derived for the circuit according to the maximum sequence length specified. The basic idea is simply to attempt to detect each primary circuit input fault at each circuit output. Factoring the output equa-

tions as before yields:

$$F^0 = A + B \cdot a^1 j + C \cdot a^0 j$$
$$F^1 = D + E \cdot a^1 j + F \cdot a^0 j, \tag{8}$$

where A, B, C, D, E, and F are sum-of-product terms. This case attempts to detect the input faults on a at the output F. A test is formed in a manner similar to that used for detecting faults, except that it is necessary here to specify the value to be assigned to input lead $a.j$.

$$\text{Maximum-Cover Test} = (E \cdot C + F \cdot B) \cdot (a^1 j + a^0 j).$$

This process is repeated until an attempt has been made to test each circuit input fault at each output lead. This scheme actually produces every test that satisfies the above equation. The shortest test (fewest input leads set to logical 0 or 1) is selected in each case.

The time spent performing this computation is usually much less than that required to derive the equations. Also the time and results of the maximum-cover operation must be weighed against the cost of detecting additional faults on a one-at-a-time basis. Thus, while maximum cover is an expensive heuristic (when compared to, say, random-number test generation), it provides a set of tests that is usually good enough so that one can economically attack the remaining faults on a one-at-a-time basis. As a general rule, about 5 to 10 faults can be detected using the one-at-a-time strategies for the same cost as one pass of the maximum-cover strategy which inherently tries to detect all faults.

V. EXPERIMENTAL RESULTS

The final measure of an automatic test generation system is how well it does its job on real circuits. The ATG system has been programmed and is being used at several locations in Bell Laboratories. The algorithms presented here are generally not useful for hand computation. The version of ATG used by Bell Laboratories on the IBM 360, Model 67, collects certain data each time it runs successfully. The data collected include the execution CPU time, number of test vectors generated, number of faults detected, number of gates in the circuit, and number of flip-flops in the circuit.

This implementation of ATG requires about 100,000 bytes for program storage. Other storage, used during execution, depends on the characteristics of the circuit being run. As the equations get longer,

the storage requirements increase. Generally speaking, ATG requires from one to five megabytes of virtual storage. This implementation allows only unit- and zero-gate delays, handles single stuck-at-one and stuck-at-zero faults, and generates fault-detection tests for single faults as well as the maximum-cover test-generation strategy.

The data that have been collected indicate that ATG has been primarily used to generate tests via the maximum-cover strategy. In a few uses of ATG, the user attempted to detect only specified faults; these data are not included in this paper.

The data collected represent only successful ATG runs. If the same circuit was run several times, then only the run that produced the fewest undetected faults (e.g., used the longest sequence length) is included. This is consistent with the recommended operational procedure, which starts with a short sequence length and increases it until an acceptable level of fault detection is reached. Faults in unused gates are included both in the undetected faults and in the total number of faults in the circuit.

The results of 300 ATG runs on 120 circuits using the maximum-cover strategy are summarized in Figs. 11 through 15. The average circuit contained about 270 gates including about 10 flip-flops in the sequential circuits. Thirty-two circuits were combinational. ATG produced an average of 94 vectors in an average of 43 seconds of IBM 360, Model 67, CPU time resulting in an average detection level of 88 percent of the total number of faults in the circuit. However, the median percentage of undetected faults was only 7 to 8 percent. The longest sequence length used for these circuits was 5. Unfortunately, there is almost no correlation between the five parameters plotted in Figs. 11 through 15. The data correlate only in the extreme cases. For example, the circuit with 32 flip-flops produced a large

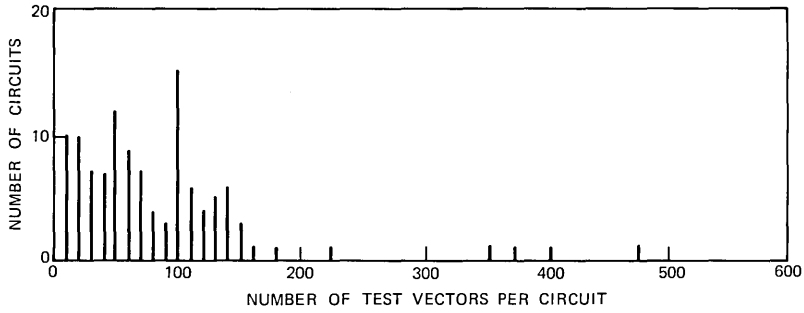


Fig. 11—Distribution of number of test vectors per circuit.

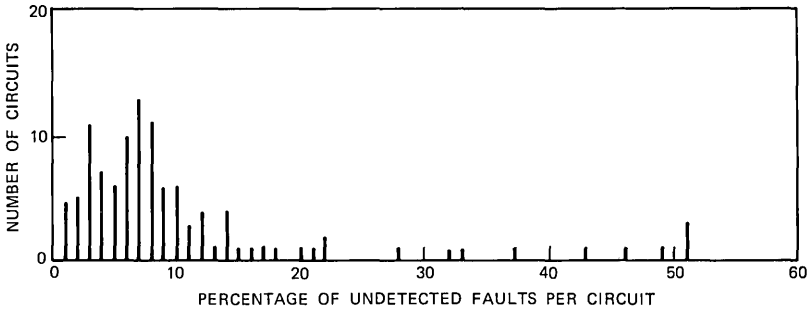


Fig. 12—Distribution of percentage of undetected faults per circuit.

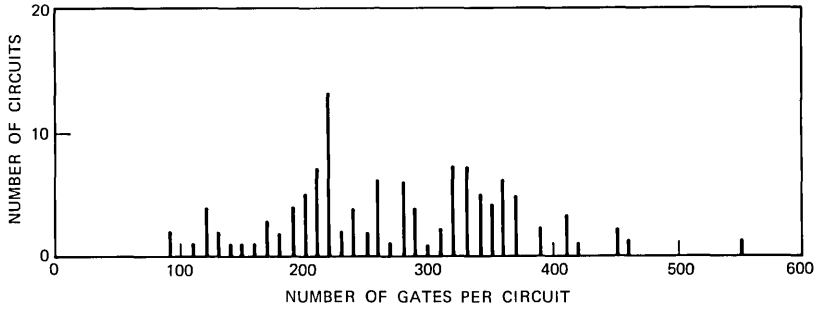


Fig. 13—Distribution of number of gates per circuit.

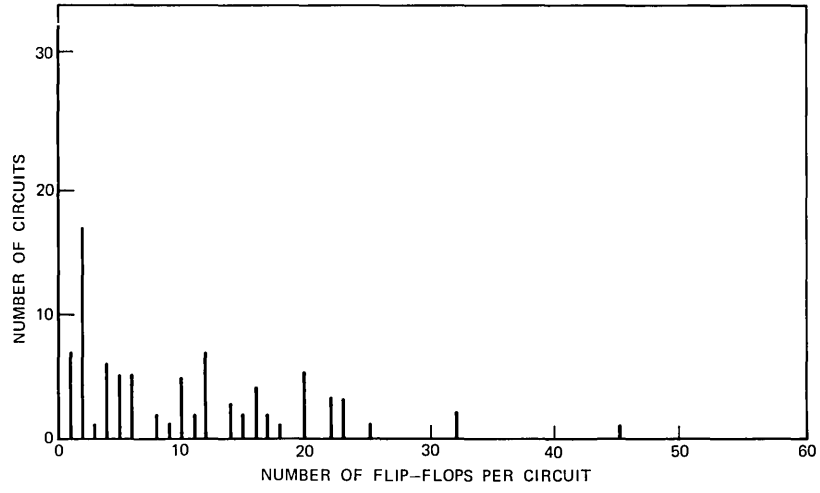


Fig. 14—Distribution of number of flip-flops per circuit.

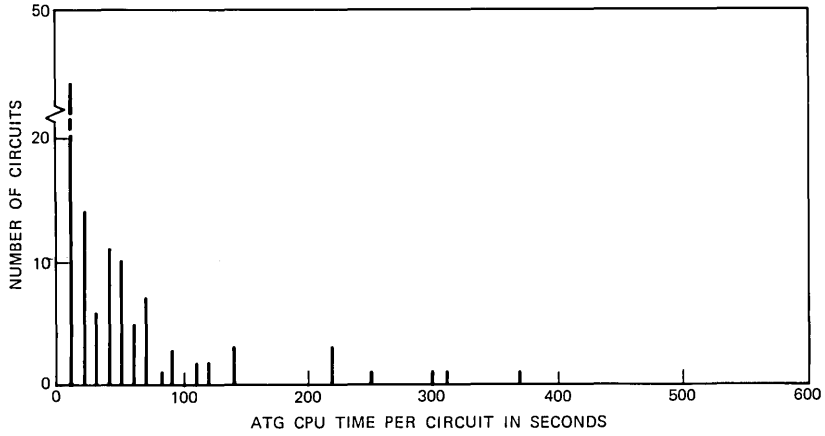


Fig. 15—Distribution of CPU time per circuit.

percentage of undetected faults. For most of the data, none of the parameters correlates significantly.

ATG did not produce acceptable results on all circuits. In general, ATG is limited by the length of the equations generated. As these equations become long, the execution time increases and ATG may not terminate successfully due to excessive run time and/or storage requirements. The equations can become long as a result of long sequence lengths (e.g., shift registers and counters) or as a result of the function of the logic circuit (e.g., parity trees and adders). In addition, circuits such as parity trees produce quite long equations and ATG generates more vectors than the minimum required.

One circuit recently run on ATG using the maximum cover strategy is worthy of special mention. The circuit is a 1000-gate, 11-state sequencer plus input, output, and transition logic. The sequencer state, represented by four *D*-flip-flops, can be read and written from circuit outputs and inputs respectively. Extensive use is made of the system clock to control transitions and gating. The clock waveforms were supplied to ATG by the user. ATG, using the clock and sequences of length one, generated 770 vectors in about 800 seconds, detecting about 95 percent of the faults in the circuit. The success of ATG here is partially due to the "easily testable design" which allows the sequencer state to be directly read and written.

In practice, while ATG will not efficiently handle all circuits, it appears to be an economical tool for automatic test generation for "mildly" sequential circuits containing around 500 gates. The design

of circuits in an "easily testable" manner greatly eases the work required to automatically generate test vectors for the circuit.

VI. SUMMARY AND CONCLUSION

In summary, the method used to generate tests is as follows:

(*i*) Set the maximum sequence length $k = 1$.
(*ii*) Generate equations for the logic circuit with sequence length k.
(*iii*) Generate tests using maximum-cover strategy.
(*iv*) Simulate the tests. If the percentage of undetected faults is less than, say, 10 percent, proceed to step (*v*). Otherwise, set $k = k + 1$ and return to step (*ii*).
(*v*) Generate tests for remaining undetected faults (one fault at a time) detectable with sequence length k.
(*vi*) If an acceptable percentage of undetected faults remains, stop. Otherwise, set $k = k + 1$ and return to step (*v*).

In practice, most users of ATG have been satisfied with the ATG results without trying steps (*v*) or (*vi*).

The algorithms treat a logic network as an interconnection of gates which are assigned some fixed time delay. The technique generates two equations, $F^0(t)$ and $F^1(t)$, for each gate in the circuit. These equations denote the input conditions required to set gate F to logical 0 and 1 respectively at time t. Because the technique starts from the circuit inputs and proceeds forward through the circuit (like the signal flow), it is not necessary to identify feedback leads. Therefore, both combinational and sequential circuits can be handled by the same algorithm.

The primary difference between combinational and sequential-circuit test generation is that several input vectors may be required in a sequential circuit to set the desired state, detect some fault, and then propagate the fault to some output lead. The number of input vectors required to perform some test on the circuit is called the sequence length of the test. A sequence length of one is sufficient to generate all tests for a combinational circuit since it has no memory. The maximum sequence length to be considered is supplied by the user.

The test-generation algorithms first generate the equations for the circuit, taking into consideration the gate delays and the maximum sequence length specified. These equations also take into account the effect of various single stuck-at-one, stuck-at-zero, or open-gate input faults when tests are being generated for specific faults. Then, from these equations, the algorithms will generate a test for any of the above faults if such a test exists within the sequence length specified.

Equations may also be generated that represent only the fault-free circuit. It is then possible to generate tests from these equations which exercise the circuit in such a way that many faults are detected. This has been a popular feature because it produces good results economically.

These algorithms have been implemented and are currently being used to generate tests for circuits containing around 500 gates. Quite good results have been produced using the maximum-cover technique. A median of 7 to 8 percent undetected stuck-at faults was reached in less than 1 minute of IBM 360, Model 67, CPU time on a sample of some 120 circuits. Because of the success of the maximum-cover techniques, very little use has been made of the "single–fault" techniques.

In conclusion, ATG is a production system that has been found to be a valuable tool for the generation of circuit pack tests.

VII. ACKNOWLEDGMENTS

The author gratefully acknowledges the guidance of Professor S. S. Yau of Northwestern University during the course of this work. The work of G. F. Shuttleworth of Bell Laboratories on the nonvirtual memory version of ATG is also acknowledged. In addition, the support of W. Ulrich and R. W. Ketchledge during the course of this work is appreciated.

REFERENCES

1. D. B. Armstrong, "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets," IEEE Trans. on Computers, *EC-15*, No. 1 (February 1966), pp. 66–73.
2. F. F. Sellers, Jr., M. Y. Hsiao, and L. W. Bearnson, "Analyzing Errors With the Boolean Difference," IEEE Trans. on Computers, *EC-17*, No. 7 (July 1968), pp. 676–683.
3. J. P. Roth, W. G. Bouricius, and P. R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEEE Trans. on Computers, *EC-16*, No. 5 (October 1967), pp. 567–580.
4. S. A. Szygenda, D. W. Rouse, and E. W. Thompson, "A Model and Implementation of a Universal Time Delay Simulator for Large Digital Nets," Proc. AFIPS Spring Joint Computer Conference, 1970, pp. 207–216.
5. S. G. Chappell, C. H. Elmendorf, and L. D. Schmidt, "LAMP: Logic-Circuit Simulators," B.S.T.J., this issue, pp. 1451–1476.
6. S. S. Yau and Y. S. Tang, "Generation of Shortest Test Sequences for Individual Faults of Sequential Circuits," to be published.
7. G. R. Putzolu and J. P. Roth, "A Heuristic Algorithm for the Testing of Asynchronous Circuits," IEEE Trans. on Computers, *C-20*, No. 6 (June 1971), pp. 639–647.
8. M. Y. Hsiao and D. K. Chia, "Boolean Difference for Fault Detection in Asynchronous Sequential Machines," IEEE Trans. on Computers, *C-20*, No. 11 (November 1971), pp. 1356–1361.
9. E. F. Moore, "Gedanken Experiments on Sequential Machines," Automata Studies, Princeton: Princeton University Press, 1956, pp. 129–153.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

LAMP:

Controllability, Observability, and Maintenance Engineering Technique (COMET)

By H. Y. CHANG and G. W. HEIMBIGNER

(Manuscript received February 28, 1974)

A new technique has been developed for organizing (or reorganizing) system design to enhance fault diagnosability. This technique is called the controllability, observability, and maintenance engineering technique, or COMET. Using graph-theoretical analysis, one can systematically apply COMET to a proposed or an existing digital system to determine the placement of control, access, and monitor points for diagnostic testing. In addition, it provides a means of studying the trade-offs between fault resolvability and the cost of maintenance hardware and/or packaging.

COMET offers an orderly approach to implementing an overall diagnostic design by providing guidelines in early design stages. A design developed using COMET has the following advantages: trouble location manual data can be generated without the use of fault simulation, multiple faults and/or nonclassical faults are locatable if they are detectable, and diagnostic or trouble-location information can be easily updated in accordance with hardware changes. Studies indicate that applying COMET to an existing processor design would require a modest increase in hardware of less than 10 percent.

I. INTRODUCTION

Recent advances in integrated-circuit technology offer the circuit and system designers many opportunities to explore new, low-cost, high-performance design techniques. The increased operational speed and the logic complexity of many medium-scale-integration (MSI) and large-scale-integration (LSI) designs, however, also present acute problems in factory testing and field maintenance. For

factory testing, it becomes increasingly difficult to diagnose faults in an LSI package, partly owing to equipment packaging constraints and partly to inadequate fault isolation technique(s) for nonclassical and/or multiple faults. In field maintenance, while the isolation of faults to a component or chip level is unimportant, the problem of quickly, and automatically, detecting and recovering from faults is further compounded by increases in circuit size and complexity. In addition, the cost of using fault-simulation techniques to generate data for the trouble-location manual[1] (TLM) may become economically prohibitive, especially for large systems. Finally, the problem of accurately updating trouble-location data whenever circuit or design changes occur remains important but unresolved.

Many designers of fault-tolerant computer systems have studied these problems.[2] Some have proposed design approaches with built-in automatic-fault-detection hardware.[2,3] Others have explored diagnosable design concepts purely from a structural standpoint based on graph-theoretical techniques.[4,5] Unfortunately, the search for practical methods of generating (and updating) TLM data for large systems has been largely unsuccessful.

This paper describes a technique called COMET (controllability, observability, and maintenance engineering technique) for organized system design and system reorganization to enhance diagnosability. COMET enables a designer to systematically establish the diagnosability of a system by combining circuit design, physical arrangement, and maintainability considerations. It also offers an efficient and practical way to generate TLM data.

In Section II, the concept of COMET is described, followed by a detailed discussion of the technique and its relation to fault location. Possible methods of implementation are then discussed, along with the results of applying this technique to a small self-checking processor.[3] Lastly, the long-term impact of COMET on system design is pointed out.

II. DESCRIPTION OF CONCEPT AND TECHNIQUES

2.1 Philosophy and characteristics

The design of a fault-location procedure involves several steps. For a given processor or circuit, a set of tests capable of detecting all the assumed faults is first derived. The usual assumptions are that faults are solid and are of the "stuck-at" type. This step is called the test-derivation phase. These tests are then verified either by sample fault simulation or by a complete fault simulation to determine if they are

indeed a good set of tests. The next step is to derive for each fault in the fault set the corresponding test results. This is done by simulating each fault with respect to the tests that are designed to detect this fault. The test results obtained by this technique are then processed to form a TLM. These two steps are called TLM data generation and data processing.

Suppose the processor has only one circuit pack. Every time a fault occurs and is detected, this circuit pack is replaced. It will no longer be necessary to distinguish faults in this processor; the fault-location problem is eliminated and the TLM data-generation and processing steps disappear. The only step required is to derive the test capable of detecting all faults and to record the test results of the "good machine."

Now if the processor is composed of more than one circuit pack, the following conceptual approach may be used. At the beginning of diagnosis, half of the processor is disabled. This can be done, for example, by physically removing half of the circuit packs in a processor. Diagnostic tests are run only on the enabled portion and only the pass-or-fail data of the tests are recorded. Thus, if a fault exists in the enabled portion of the processor, the test result will give a failure indication, meaning that the fault is not in the disabled portion. However, if the test result gives a pass indication, this means that the fault is in the portion that has been disabled or removed.

Based on the pass-or-fail indication, one can further partition the enabled portion (in the case where the fault is in the portion that remained enabled during testing), or the disabled portion (in the case where the fault is in the disabled portion) to allow further testing. A general flow diagram of this procedure is shown in Fig. 1. Disabling means that the circuit packs associated with the disabled portion are

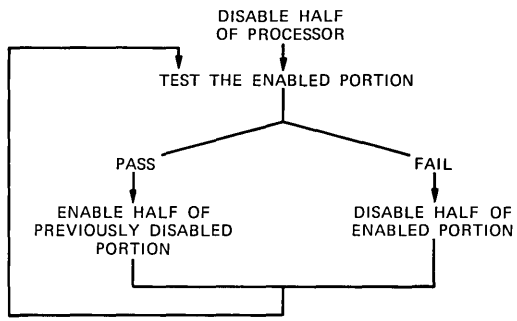


Fig. 1—Flow diagram of disabling process.

either physically disconnected from the processor or are logically in a passive state.

Figure 2 shows an example of how faults are located using this technique in a processor composed of four circuit packs (A, B, C, and D). Assume that the fault is in circuit pack B. Diagnostic tests are first performed with circuit packs C and D disabled or removed. The tests, which are designed to detect all the faults in packs A and B, are run and a failure is indicated. It can then be concluded that some failure exists in either pack A or B. Next, the diagnostic tests are run on circuit pack A with circuit pack B disabled. This time, however, because the fault (which is in circuit pack B) has been masked, the test result will show a pass indication. It can then be concluded that a fault is in circuit pack B because it gives a fail and then a pass signature. In other words, by successively reducing the circuitry under test and by only recording the fail/pass results in each step, the locations of all faulty circuit packs are uniquely identified.

It is quite apparent that the basic difference between this technique and the conventional approach is that in the latter one must, for isolation purposes, distinguish the faults not only from the good machine but also from every faulty machine. In other words, the important consideration is where the fault is. In the proposed approach, however, we are not required to distinguish the various faulty machines; we are only interested in whether the test result from the good machine differs from that of the faulty machine. Resolution is obtained by successively reducing the circuitry under test.

For each element of the partition (e.g., packs A and B of the first partition in Fig. 2, or circuit pack A of the second partition in Fig. 2),

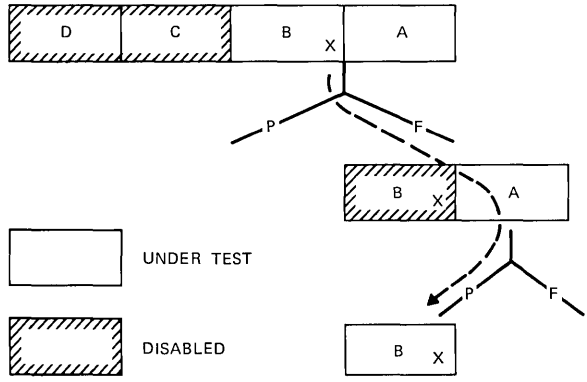


Fig. 2—Proposed method of fault location.

only a pass or fail indication is required. This means that the test result of the good machine for each partition is all that is required; there is no need to simulate faults to generate sufficient information for the distinguishability of the faults within the element of partition. It can be seen that each faulty circuit pack is identifiable by a unique pattern of pass-and-fail indications; the pass-and-fail numbers are in a one-to-one correspondence to the circuit packs in the processor. This means that the number of trouble numbers in the TLM is drastically reduced.

Another characteristic of the proposed technique is that multiple faults on a circuit pack are locatable as long as they are detectable by the applied tests. This should improve accuracy since the requirement for consistency of test signatures for TLM lookup no longer exists.

2.2 Description of techniques

2.2.1 Controllability and observability

There are some problems that must be solved. First, one must design a disabling process that allows circuit packs to be selectively disabled or removed from the processor. Second, *controllability* of the various circuit packs or functional blocks must be established. For example, as shown in Fig. 3a, gate G' must be operational to test gate G. That is, G' must not be disabled when G is being tested. If, however, G' is in the portion that has been disabled, the testing of G and therefore the test results become meaningless because G is not controllable from G'.

Similarly, a proper ordering of the various circuit packs or functional blocks in relation to *observability* must be derived. As shown in Fig. 3b, to observe the test results of a fault (marked ✕) associated with gate G, the output of G must not be in the logic block that has been disabled. Otherwise, the test results will show an all-tests-pass

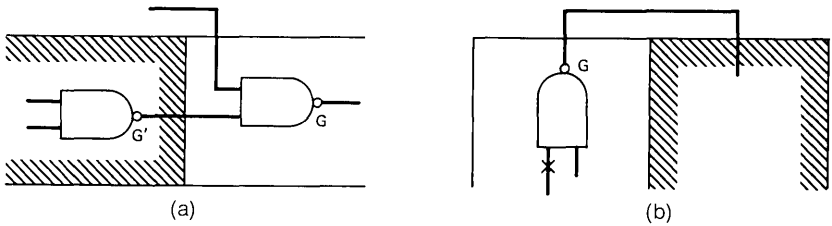


Fig. 3—(a) Controllability-ordering relations. (b) Observability-ordering relations.

outcome even though the fault is present on gate G. Thus, the necessary condition for this technique is the establishment of an ordering relation of controllability and observability such that a partitioning procedure can be used for fault isolation. If the elements of partition i do not depend on elements of partition j (where $j > i$) for controllability and observability, one can successively apply the partitioning and testing process to locate the faulty circuit pack, providing that the disabling technique of circuit packs is available.

2.2.2 Logic-disabling technique

There are a number of disabling techniques possible. The first and the most obvious one is to physically remove the circuit pack(s) from the processor. This is only feasible if a reliable connector is available and the number of circuit packs involved is small. An alternative is to physically disconnect the input and output leads of a circuit pack by some mechanical device. A third alternative in some cases is simply to remove the power and ground leads of a circuit pack, thus putting the circuit pack in the passive state. A fourth alternative is the logic-disabling technique illustrated in Fig. 4.

If control lead C_i (Fig. 4) goes to 0, it forces output of the output gates to logical value 1, which for NAND logic is the passive state. After the circuit pack has been disabled, error symptoms caused by any fault or faults in the circuit pack (marked ✕ in the illustration) cannot propagate beyond the circuit-pack outputs.* Thus, if each circuit pack i is modified by adding disable control lead C_i, each circuit pack in the processor can be enabled or disabled selectively.

2.2.3 Partitioning techniques

Once a practical way of disabling circuit packs is obtained, the next step is to devise a technique of ordering the circuit packs or the functional blocks based on the observability and controllability relations. The controllability and observability relations can best be understood by the example shown in Fig. 5.

We define a functional node as a functionally well-defined logic circuit, such as a rotate circuit, an adder, etc. In some instances, a functional node is also defined as a logically or physically related block of circuitry, such as bits 0 through 7 of the X, Y, Z registers. To test a functional node, one must apply signals via control inputs and observe

* Note that the stuck-at-0 faults on the disabled gates can still propagate. Treatment of these faults is discussed in Section 2.2.4.

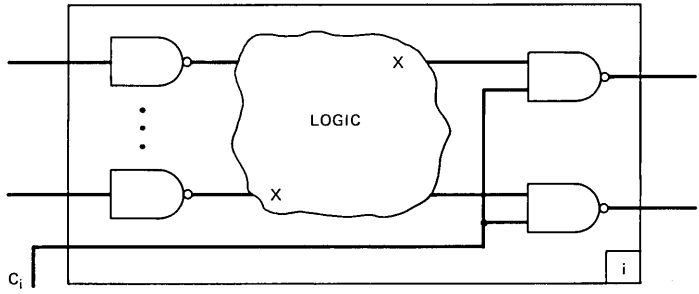


Fig. 4—Circuit-pack logic-disabling process.

test results via some observable outputs. Thus, in Fig. 5, it can be concluded that to test functional node R, the inputs are obtained from functional node A, and control is obtained from functional node C. Functional nodes A and C, therefore, must be operational when functional node R is being tested. Similarly, to observe the test results of R, functional nodes O_1 and O_2 are used. In other words, functional nodes A and C *control* R, and functional nodes O_1 and O_2 *observe* R; they must not be in the portion that is disabled when node R is being tested.

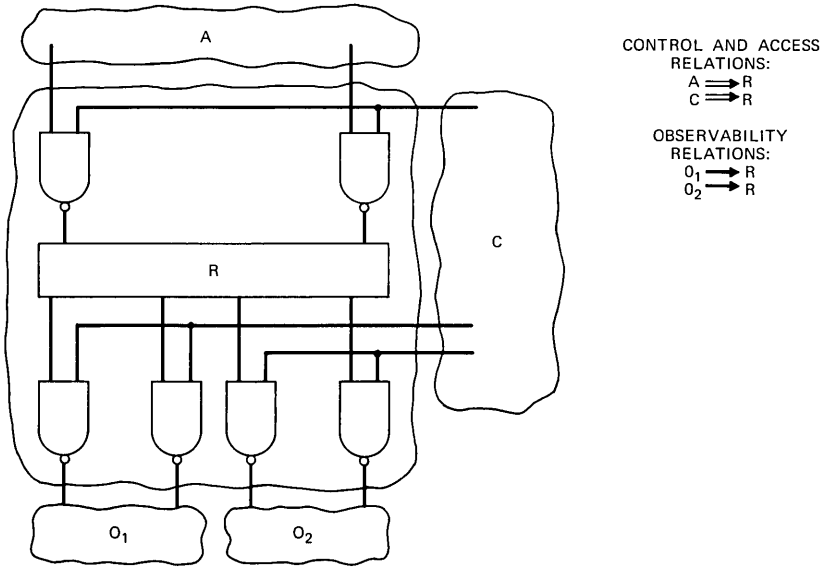


Fig. 5—Controllability and observability relations.

A set of relations can now be defined as follows: node A *controls* node B, if node B requires control from A to be fully testable. This relation is expressed by $A \Rightarrow B$. Similarly, node A *observes* node B, if node A is required to observe the test results of node B. This is represented by $A \rightarrow B$.

From a set of properly defined functional nodes, the controllability and the observability relations among neighboring nodes can be obtained. These relations are conveniently represented by a directed graph where the nodes of the graph correspond to the functional nodes and the edges of the graph correspond to the controllability and/or the observability relations. If the resultant graph is loop-free, all the nodes can be arranged in a partially ordered list so that the higher-order* nodes can be fully tested independently of any lower-order node(s). This guarantees a relation of controllability and observability among all the functional nodes such that we are able to partition the system. However, if the resultant graph is not loop-free, a conflict exists from a controllability and observability viewpoint.

For example, if some functional node F_6 controls functional node F_1, which in turn controls F_5, and if F_5 observes F_6, then a loop exists containing F_6, F_1, and F_5. This loop must be "broken" to test these nodes. In this context, breaking loops means that additional control or observation must be added in appropriate places to obtain a loop-free graph.

The process of systematically ordering the nodes and identifying conflicts in the directed graph makes use of graph-theory techniques.[4] The directed graph of the controllability and observability characteristics of functional nodes can be represented by a connectivity matrix $C = [c_{ij}]$, where $c_{ij} = 1$ if there is a directed edge from i to j. In other words, if node i controls node j, the entry $c_{ij} = 1$ in the connectivity matrix for the controllability relation. Similarly, if node k observes node p, the entry $c_{kp} = 1$ in the connectivity matrix for observability.

A node j is *reachable* (controllable or observable) from node i if and only if there is at least one directed path from i to j. A graph is *strongly connected* if and only if every node is reachable from any other node. This means that in a strongly connected graph, every node is in at least one loop. A *maximal strongly connected* (MSC) subgraph is one that includes all possible nodes that are strongly connected with each

---

* The term *higher* refers to the location in a diagnostic procedure. The higher nodes are verified first.

other. This designates a maximum set of functional nodes that are in conflict from a controllability and observability viewpoint. A *link* subgraph of a graph is one that contains no strongly connected subgraphs or unconnected subgraphs in it. The link subgraph is loop free and therefore all the nodes in it can be arranged in a partially ordered list.

Generally speaking, all functional nodes are either in some MSC subgraph or in some link graph from the observability and controllability viewpoint. The objectives are, therefore, to locate the MSCs (i.e., areas of conflicts) in the directed graph and to add additional controllability or observability points in order to break the MSCs and arrive at a partial ordering of the nodes. The following is an algorithm for performing this function.

(*i*) Construct connectivity matrix of the functional nodes of the processor with respect to the controllability and observability relations.
(*ii*) Locate all MSCs and represent them as pseudonodes.
(*iii*) Establish the order of the nodes in the new directed graph.
(*iv*) Any MSCs left? If yes, go to step (*v*). If no, exit.
(*v*) Pick an MSC of the highest order and apply MSC breaking technique; then return to step (*iii*).

The connectivity matrix is arrived at by merging the connectivity matrices of the controllability relations and observability relations.

All MSCs found in the directed graph are located and temporarily identified as pseudonodes so that the ordering process of step (*iii*) can be carried out. Ordering means that all nodes having only primary inputs are considered to be of the highest order (i.e., first order), and a node is of ith order if all of its inputs are of order $i - 1$ or less and at least one of the inputs is of order $i - 1$. In an ordered list, nodes of the ith order do not depend on those of jth order, for $j \geqq i$ for controllability and observability. The kth order is said to be higher than the ith order if $k < i$. Once the ordering process is performed, we then proceed to break the MSCs, if necessary.* The process of breaking MSCs is similar to the one described by Ramamoorthy.[4] First, the entry nodes of a given MSC are identified. An entry node having the highest ratio of number of incoming edges to number of outgoing edges is selected. All edges entering it are deleted; this means that additional

* This is necessary only if the required resolvability is one faulty circuit pack and there exist some MSCs that cannot be packaged on one circuit pack.

control and/or monitor points are added to those nodes associated with these edges.

The result of this process [steps (*i*) through (*v*)] is a loop-free directed graph or a partially ordered list of nodes. Nodes of the *i*th order are completely testable using only those nodes of orderings higher than *i*. At this point, packaging considerations can be incorporated in order to arrive at a reasonable set of partially ordered lists of circuit packs. The general guide-lines for packaging are:

(*i*) Group only nodes of the same ordering on one circuit pack or set of circuit packs.
(*ii*) If this is not possible, then group a node (or a set of nodes) of the *i*th order with those of order $i - 1$ or $i + 1$, but not both.

These guidelines assume that the resolution is to be to one circuit pack. For example, the groupings of functional nodes shown in Figs. 6a and 6b are acceptable. In Fig. 6b, the ordering of circuit packs P_j and P_{j+1} are irrelevant because the two circuit packs are equivalent. But the ordering of circuit packs P_{j-1} and P_j or P_{j-1} and P_{j+1} is im-

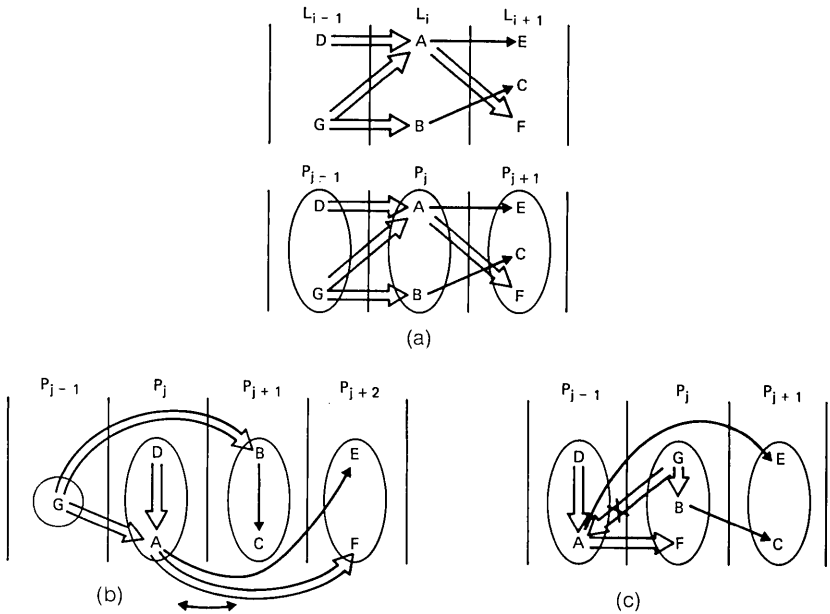


Fig. 6—Packaging considerations.

portant. This is because functional node G controls both nodes A and B, and therefore G must be of a higher ordering than nodes A or B. Figure 6c illustrates a conflict because G controls A but A controls F. Suppose functional nodes G, B, and F are packaged on one circuit pack. A conflict then exists because packs P_{j-1} and P_j cannot be separated for testing purposes. In other words, the resolution has been degraded from one pack to two packs in this case.

Once the nodes have been packaged and all the packaged circuits have been ordered, a partitioning procedure can be applied to the packaged circuit packs. The following example shows the process of ordering and partitioning of nodes.

Example: Suppose the directed graph shown in Fig. 7a represents the controllability and the observability relations of a circuit. Each node in the graph represents a functional entity; each edge represents either a controllability relation (denoted by $\Rightarrow$) between the two nodes or an observability relation (denoted by $\rightarrow$) between the two nodes. For example, node D observes node E; node E controls node H. The information represented by this graph is equivalent to a connectivity matrix which can be constructed by examining each functional node in the circuit and its observability and controllability relations with its neighboring functional nodes.

To arrive at a partially ordered list of nodes, the first step is to locate all MSCs and represent them as pseudonodes. In this case, there is one MSC (as indicated by the dotted line in Fig. 7a) denoted by v. The reduced graph is then ordered by applying the ordering process. For example, to completely test and observe node C which is of order L_2, it is only necessary that nodes of higher order, i.e., nodes A and B of order L_1, be available for control and observation.

If all the nodes can be packaged at this point according to the previous guidelines, a partial ordering of nodes has been obtained. However, if, for example, the pseudonode v contains too many functions to be packaged, the pseudonode v must be further decomposed by breaking the MSC it represents. The entry nodes to this MSC are nodes D and G. Node G is chosen and the edge D–G is broken by adding control to node G. At this point there is still another MSC in the graph so the process is repeated (see Fig. 7b).

A new MSC denoted by a pseudonode v_1 is identified, and the graph is ordered once again. The MSC denoted by v_1 is again "broken" after having decided to add an observable point to the functional node D. The final ordered list of functional nodes is shown in Fig. 7c; these

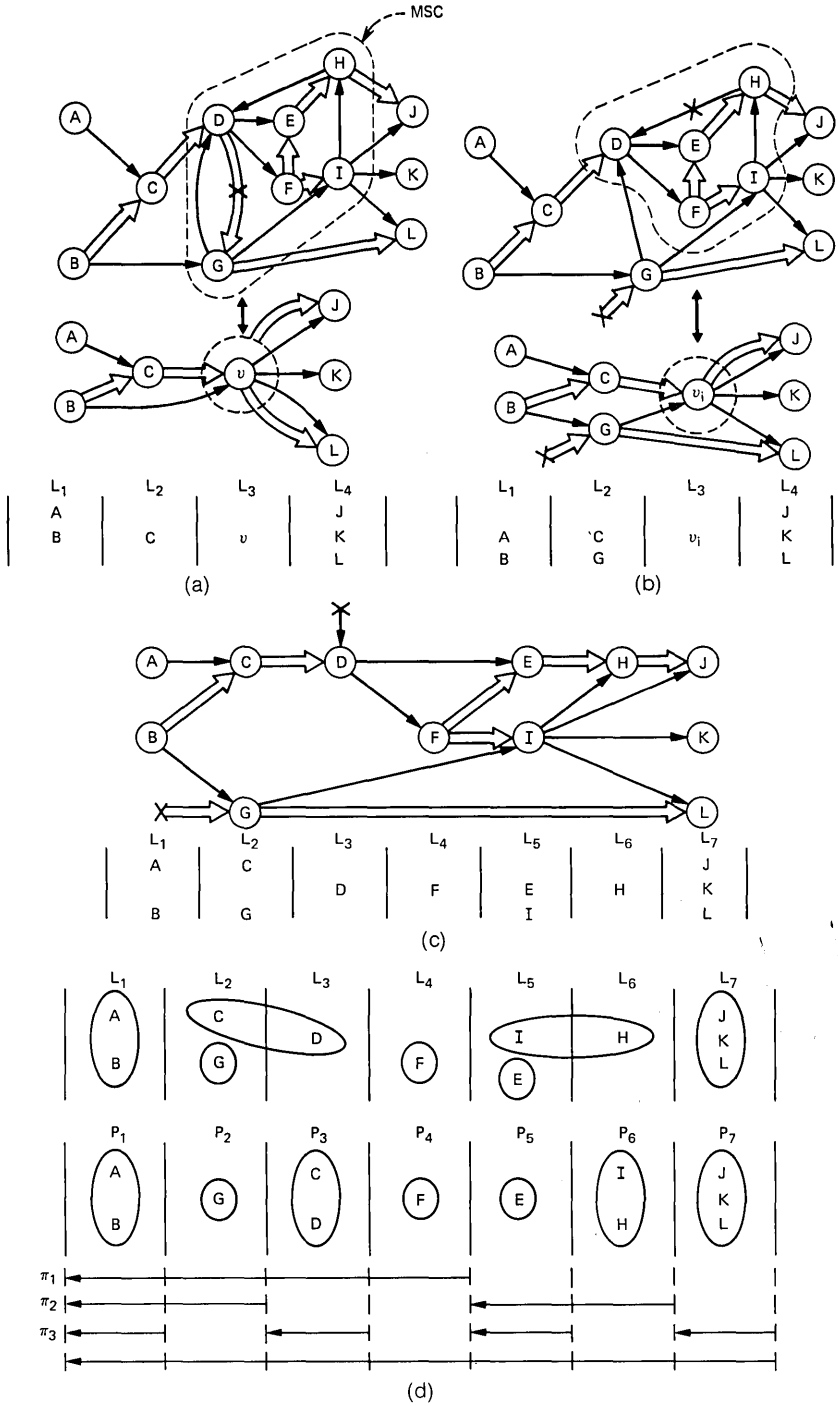


Fig. 7—Partitioning and ordering of functional nodes.

nodes form a partially ordered list in that nodes of order L_i are testable using only those nodes of an ordering higher than L_i.

To demonstrate how fault(s) can be isolated using this process, the functional nodes are assumed to be packaged as shown in Fig. 7d. Without loss of generality, a binary partitioning process will be used in the following cases.

Case 1: Single-fault location. Suppose a fault occurs on circuit pack P_5. Diagnosis will be performed first on portions of circuitry consisting of P_1, P_2, P_3, and P_4, while circuit packs P_5, P_6, and P_7 are disabled. The test results are valid because the testing of circuit packs P_1 through P_4 does not depend on circuitry associated with circuit packs P_5 through P_7 due to the proper partial ordering requirement. The first test result ($\pi 1$) will yield a pass indication implying that the fault is not on circuit packs P_1, P_2, P_3, or P_4. Following the process discussed in Section 2.1 (see Fig. 1), the diagnosis ($\pi 2$) is then performed on circuit packs P_5 and P_6, using circuitry associated with P_1 through P_4, which has been previously verified, while disabling circuit pack P_7. This time the tests will show a fail indication; the failure is isolated to circuit packs P_5 and P_6. In the next step ($\pi 3$), circuit packs P_6 and P_7 are disabled while running tests on circuit pack P_5 using circuitry associated with P_1 through P_4. The test result again shows a fail indication and, thus, identifies the faulty pack to be P_5.

Case 2: Multiple-fault location. Now suppose that a fault exists on circuit pack P_5 and another fault exists on circuit pack P_3. The initial diagnosis of partition $\pi 1$ shows that a failure is in circuit packs P_1 through P_4 with circuit packs P_5 through P_7 disabled. The failure symptom caused by the fault of P_5 will not interact with the testing of P_1 through P_4 because it has been disabled.

Next, circuit packs P_1 and P_2 are tested in partition $\pi 2$; this test indicates a pass indication. The final test ($\pi 3$) is on P_3 using circuits associated with P_1 and P_2, and disabling circuit packs P_4 through P_7. This test identifies the fault on P_3. Once this faulty circuit pack is identified and replaced, a complete check is run. The presence of the fault on P_5 will now indicate a test failure. The diagnostic process described previously is repeated again to isolate and identify the second fault.

It can be seen that this process enables us to systematically isolate faults one at a time until all faults in the circuits are identified and repaired. Any fault that gives a test result that is different (regardless of the nature of the fault) from the true-value signature is detectable. In other words, the single-fault assumption and the classical stuck-at-0 and stuck-at-1 assumptions on failure modes are no longer necessary with this approach.

2.2.4 Global Feedback

The constraint of being able to fully disable the outputs of *all* circuit packs is an overly restrictive condition. The only reason for disabling is to prevent propagation of fault symptoms *into* the portion of the machine under test. Thus, at any partition only those leads crossing from the lower-order levels into the higher-order levels are of major importance. In a general case, this should represent only about half of the leads crossing the boundary. In a machine organized using COMET, it could be expected that the portion of leads crossing from low levels to high levels would be less than half, reflecting the attempt COMET makes to break things into a tree structure.

For the purpose of discussion, "global feedback" will be defined on the linear ordering of functional nodes. A global feedback is any directed *wire* going from a lower-order node to a higher-order node in the list of partially ordered functional nodes shown in Fig. 8. The distinction between a wire and a controllability/observability edge is important. Optimization of the latter (i.e., control by gate A) may remove a connectivity matrix entry while the wire still remains. It can be seen that it is only necessary to be able to disable all leads that fit the definition of global feedbacks. This is a considerably less stringent requirement than being able to disable *all* circuit-pack outputs.

The concept of logic disabling is a method of emulating the physical removal of circuit packs (i.e., leaving circuit-pack-output gates in the all 1's state). However, if the stuck-at-0 output of the gate is disabled, it presents a potential problem. This can be analyzed as follows. First, the stuck-at-0 output may feed to a higher-order node. If the ordering has been carefully observed, there must be an alternate method of controlling the lead. In this case the highest circuit pack fed by the stuck-

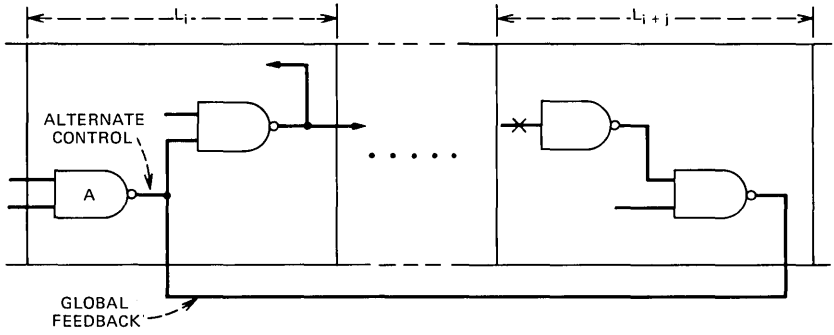


Fig. 8—Example of global feedback.

at-0 output is identified as bad. If ordering has not been carefully observed and no special control has been added, the result is unpredictable. If the stuck-at-0 output feeds only lower nodes and control or observation has not been added, the highest of the lower nodes are identified as bad. If observation has been added, the fault should be properly isolated. In the cases where the ordering has been observed, the pack identified as bad is either the proper one or is fed by the stuck-at-0 output. A simple check is to remove the circuit pack identified by diagnosis. Next, all packs feeding the removed circuit pack are disabled. At this point, all connector pins should be logical 1's. A test connector can then be inserted into the vacant slot and examined. Assuming that the basic connectivity information is available on a per-pack basis, the actual faulty pack is now identified by correlating any grounded pins with a faulty pack.

The test procedure is highly automatic. Diagnosis proceeds to locate a suspect pack. The craftsman replaces this pack with the test connector. The machine disables the necessary packs, scans the connector, and locates any grounds. The location of the grounds can be combined with connectivity information stored on bulk storage to uniquely identify the bad circuit pack. This procedure is much less susceptible to manual errors than previous diagnostic techniques.

III. FEASIBILITY STUDY—APPLICATION OF COMET TO A SMALL PROCESSOR

To verify the feasibility of the COMET procedure, a small self-checking processor was selected for study. This choice was made because a simulation model of the processor existed. This allows verification by simulating the ability of COMET to locate faults. In addition, the processor is complex enough to present a good sampling of "real-life" problems.

3.1 Brief description of processor

The processor is a stored program machine,[3] composed of approximately 4,400 logic gates. It features a microprogram control and a general-register structure. It is fully self-checking and does not rely on matching for fault detection. From a system view, the active and standby processors are linked to the outside world by the local maintenance center.[6] The LMC is responsible for the diagnosis of an off-line central control. In terms of COMET, control of the disabling and actual testing is exercised by the LMC.

The interface between the processor and the LMC is detailed in Fig. 9. The major ports for controlling the processor are the input bus,

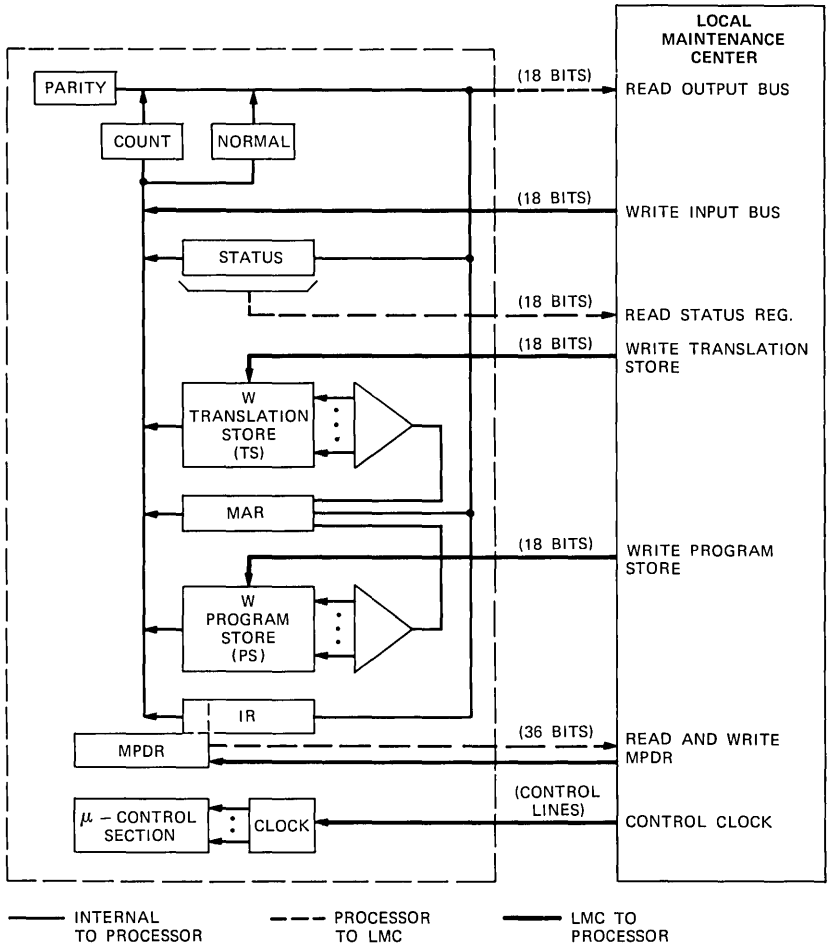


Fig. 9—Interface between processor and local-maintenance center.

the microprogram data register (MPDR), the clock, and the various stores. The ports prescribe external controllability and observability for the processor. The internal structure of the processor is a register-bus structure with 16 addressable registers.

The control of the machine centers around a microprogram unit. This unit is composed of a 36-bit data register and two major decoders, the "to" and "from" decoders. It is highly self-checking as is the entire machine. Nearly any hardware error causes immediate notification of the LMC via the status register.[6]

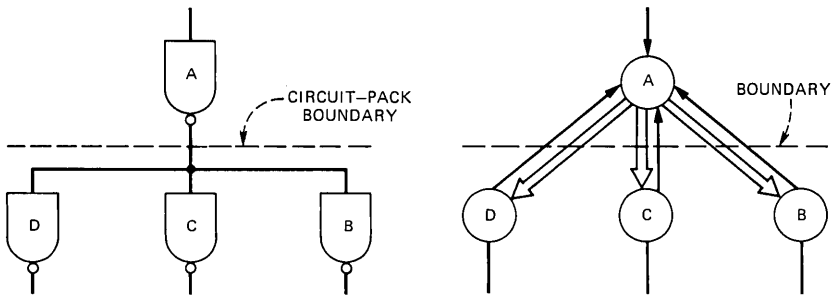
3.2 Formation of the connectivity matrix

The first step in deriving the processor connectivity matrix is to define a reasonable set of functional nodes. In this case, the functional node definitions parallel the processor block diagram quite closely. There are a total of 41 functional nodes under consideration. These vary considerably in size. The LMC, for instance, is nearly as large as an entire processor while the decision logic is only a few gates.

With a preliminary set of functional nodes defined, the controllability and observability relations can be derived. These relations are derived on a local (i.e., node-by-node) basis. In other words, the set of relationships for a single node can be written by only considering its neighboring nodes. In general, these relationships bear a close association to the physical connections in a circuit. In fact, in the limit, every physical connection between two nodes could generate both a control and an observation edge.

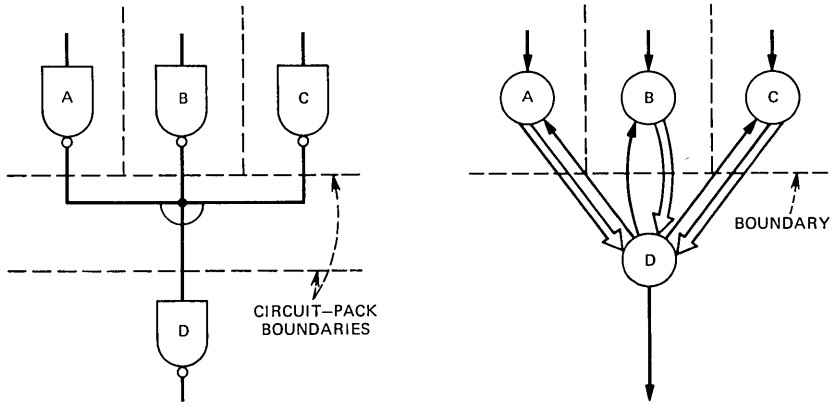
The mechanical construction of the connectivity matrix by only physical connection information will yield a sufficient set of conditions for leveling. This will be equivalent to doing a bit-by-bit OR of the control connectivity matrix with its transpose to formulate the combined control and observation matrix. However, in practice, simplification can often be achieved without resorting to the seemingly brute-force approach.

The method of simplification is based on two logic features not properly represented by the graph-theory model. The first and most important feature is the existence of fanout. A gate from a register may fan out to several points, as shown in Fig. 10a. The resultant directed graph for proper controllability and observability relations was derived strictly based on physical connection information



(a) GATE–LEVEL CIRCUIT (b) GRAPH–THEORY MODEL

Fig. 10—Fanout considerations in modeling.



(a) GATE–LEVEL CIRCUIT (b) GRAPH–THEORY MODEL

Fig. 11—Collector-tie considerations.

shown (see Fig. 10b). The interpretation is that all nodes B and C and D are required to observe node A. In fact, however, it is only necessary to have node B or C or D to observe node A. Rather than have entries in the observability matrix for all three, only one entry is needed. The determination of which entry to retain is usually not completely arbitrary. In general, it is desirable to retain only the highest-order node (B, C, or D) for observation.

The other logical feature that allows simplification is the collector tie (see Fig. 11a). This effect on the control-connectivity matrix is analogous to the effect of fanout on the observation-connectivity matrix. It is only necessary to be able to control one of the inputs to a collector tie to check its validity with respect to the gates feeding it (Fig. 11b).*

3.3 Ordering and partitioning of nodes

The 41 nodes may now be analyzed to arrive at a partial ordering. If they can be leveled at this point, an order is established. In general, this will not be the case. Analysis programs are then used to identify and locate controllability and observability MSCs. There are several ways to treat these problems. From a diagnostic point of view, the MSCs should be broken down into link-graph structures. This implies adding control or observation points and reanalyzing the graph. This

* The resolution of a ground on a collector-tied node is a well-known classical problem.

approach will yield a circuit that is diagnosable to one functional node (which may be one circuit pack or part of a circuit pack).

In practice, adding sufficient control or observation to break all MSCs may be expensive. In many cases the acceptance of reduced resolution is more attractive than the addition of much hardware. For example, it may not be economical to resolve a circuit pack output stuck-at-1 fault and an input-diode-open fault of the pack it drives, even if these two packs form a two-node MSC. Whenever possible one should always attempt to put the connected nodes on a single package to reduce (and eliminate) MSCs. This is equivalent to admitting that faults in the nodes are indistinguishable. However, this is of no importance if they are detectable and on a single package.

In attempting to partially order the processor nodes, several controllability and observability MSCs were discovered. The most obvious concern is the status register. The states of all check circuits in the machine are sampled and trapped in the status register. Any error indications cause an immediate maintenance interrupt of the processor. The LMC has the ability to directly read all bits of this register and to clear the register. However, it did not originally have controlled write access to the register. The situation that existed is shown in Fig. 12. This can be written as follows:

STATUS REGISTER → "TO" DECODER 1/N CHECK → "TO" DECODER ⇒ STATUS REGISTER

and indicates the presence of a loop. The flip-flops of the status register are not testable in a "start-small" diagnosis. The solution to this problem is rather simple. The LMC is given controlled write access to the status register, as shown in Fig. 12. This allows removal of the control link specifying:

"TO" DECODER ⇒ STATUS REGISTER

and replacing it with:

LMC ⇒ STATUS REGISTER.

In addition, there is a two-node loop between the MPDR and the microprogram store (MPS). This is solved by careful merging of the LMC access to the MPDR. The collector-tie access will restrict fault isolation to two packs (one in the LMC and one in the MPS) in the stuck-at-0 case but will allow resolution of the stuck-at-1 problems to one pack. The decision logic and MPDR are also connected by a loop. The decision logic controls bit 0 of the MPDR and is observed by the MPDR. The stuck-at-0 on the decision logic output would only

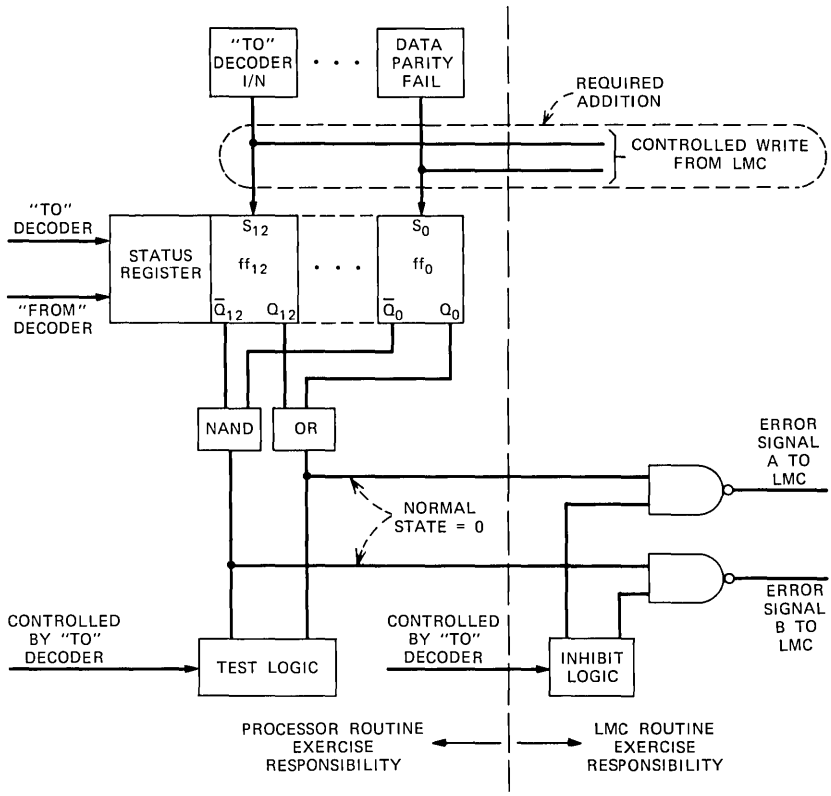


Fig. 12—Status register and its local-maintenance-center interface.

be resolvable to two packs. The stuck-at-1 output is not distinguishable from the set-bit-0 input open. The provision of a single control-write lead from the LMC eliminates the latter resolution problem.

Another loop exists between the decoders and the $1/N$ check circuits. Again, either two-pack resolution must be accepted or the two nodes must be packaged together if additional hardware is not provided. In this case, the decoder functions can be halved into an even parity and an odd parity and part of the check circuitry packaged with each half. The tradeoff again is one of engineering judgment and cost effectiveness.

The process of ordering the processor nodes can be completed by consciously resolving any conflict that appears. The term "consciously" is a key point. Assuming that the control and observation relations are complete, the analysis procedure will point out diagnostic

problems. To proceed to a fully ordered graph, these problems must be examined and treated. Thus, the existence of a fully ordered graph should insure consideration of diagnostic problems. A by-product of the ordering process will be the addition of maintenance hardware or a generated list of graph edges having some degree-of-resolution problems. The nature of this list of edges is such that it should give a reasonably good measure of the diagnostic resolution possible with the proposed design.

The final ordering of the functional nodes (Fig. 13) requires seven levels to encompass the 41 functional nodes. In this case there are fewer circuit packs than nodes.[3] This indicates that more than one node will be packaged on a circuit pack, as expected. The diagram gives some guidelines as to which nodes can conveniently be packaged together and which should not be packaged together.

From the ordered control/observation graph, a diagnosis strategy can be derived. Here a linear-partitioning process is used. In a linear-partitioning process, all circuit packs but one of the highest order are first disabled. Tests are run on this circuit pack and, if a failure is detected, the fault is in the circuit pack. If the failure is not detected,

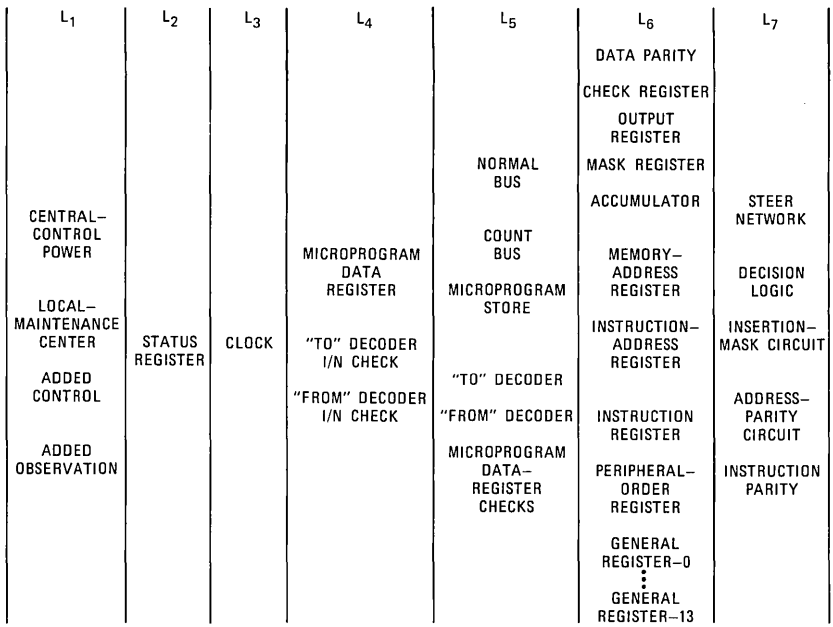


Fig. 13—Partial ordering of processor nodes.

further disabling is in order. In a second partition circuit, packs 1 and 2 (of order 1 and 2) are enabled, whereas the rest of the circuit packs are disabled. Tests are then run on circuit packs 1 and 2. If they pass the tests, then the fault(s) would be in the rest of the disabled proportion; if they fail the tests, then the faulty circuit pack is in the portion of circuit under test. Since circuit pack 1 was verified to be good in the previous partition the faulty circuit would be circuit pack 2. This process is repeated for each partition in a linear fashion until the faulty pack is located.

In this example, the diagnostic strategy is to verify the first five levels in order. The power (L_1) is verified from the LMC. Next, the status register (L_2) is verified by controlled write and read access. With the status register now available as an observation port, the clock (L_3) can be verified. Next, controlled read and write access are used to verify the MPDR and the decoder check circuits (L_4). The MPDR and the status register provide sufficient control and observation to diagnose the MPDR check circuits (L_5). Cycling through the MPS and observing the results with the MPDR will verify the microstore (L_5). The decoders (L_5) are now checked using the MPDR and clock as inputs and the status register as outputs. Note that the combination of the previously discussed packaging reasons could obviously modify L_5 and L_6. The buses (L_5) are verified by controlled read and write access. This leaves a skeleton processor capable of executing microinstructions. The nodes of L_6 and L_7 must be arbitrarily converted into a form adaptable to linear enabling and diagnosis; an example is shown in Fig. 14. It was largely these functional nodes that were used for the simulation experiments.

3.4 Fault location using COMET

To verify the results that are expected from COMET, several simulation experiments were performed. A 2700-gate* simulation model of the processor was used. The model, which excluded the various stores, the peripheral communication and the LMC interfaces, provides a good vehicle for checking the COMET technique.

The procedure for checking the control section of the machine is quite straightforward with the aid of the LMC. Thus, simulation was performed assuming that the microcontrol section had been tested and found good.

* The major difference in gate count between the simulation model (2700 gates) and the entire processor (4400 gates) is due to the inclusion of only two of the general-purpose registers in the simulation model.

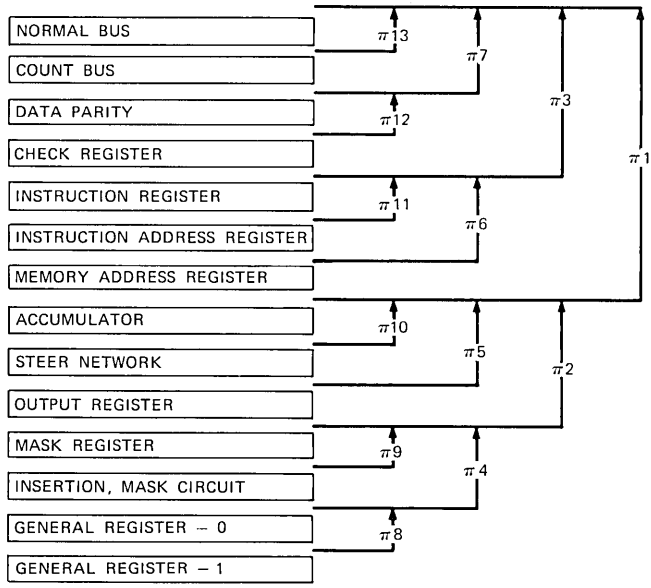


Fig. 14—Functional nodes included in simulation experiments.

Selected faults were inserted into the model and a simplified set of diagnostic tests were run. The functional node arrangement and the test (πi) procedure are shown in Fig. 14 in which a binary-partitioning technique is used. In actual applications, this would be replaced with a linear-partitioning arrangement.

The first experiment consisted of inserting a stuck-at-0 fault in bit 10 of the first level of the combinational data-rotation network.[3] The sequence of tests and their results, as derived by simulation, are shown in Table I. Tests $\pi 1$ and $\pi 10$ passed, whereas $\pi 2$ and $\pi 5$ failed. The pass/fail number of $\pi 1\ \pi 2\ \pi 5\ \pi 10 = (1001)$ uniquely points to a functional node which, in this case, is the rotation or "steer."

The second experiment was to insert a stuck-at-0 fault in the instruction address register (IAR). The results of running the test phases on this fault are shown in Table II. Again, a unique functional node is isolated by the pass/fail number $\pi 1\ \pi 3\ \pi 6\ \pi 11 = (0101)$.

For a third experiment, both the fault in the rotation logic and the fault in the IAR were inserted simultaneously. As expected, the IAR (i.e., the highest) fault was isolated independent of the other fault. Upon correction of the IAR fault, the steer fault would be isolated. If the replacement circuit pack that is to correct the IAR fault had been

Table I — Simulation results—steer network fault

Machine	Test (See Fig. 14)	Input Vector	Output Vector (Binary)
Good	$\pi 1$	# 7	0000 0000 0000 0000 00
Faulty	$\pi 1$	# 7	0000 0000 0000 0000 00
Good	$\pi 1$	#12	1111 1111 1111 1111 11
Faulty	$\pi 1$	#12	1111 1111 1111 1111 11
Good	$\pi 1$	#14	0000 0000 0001 0000 11
Faulty	$\pi 1$	#14	0000 0000 0001 0000 11
Good	$\pi 1$	#16	1000 0000 0001 0000 10
Faulty	$\pi 1$	#16	1000 0000 0001 0000 10
Good	$\pi 2$	# 8	0000 0000 0000 0000 00
Faulty	$\pi 2$	# 8	0000 0000 0010 0000 00
Good	$\pi 2$	#13	1111 1111 1111 1111 11
Faulty	$\pi 2$	#13	1111 1111 1111 1111 11
Good	$\pi 5$	# 6	0000 0000 0000 0000 00
Faulty	$\pi 5$	# 6	0000 0000 0010 0000 00
Good	$\pi 5$	# 9	1111 1111 1111 1111 11
Faulty	$\pi 5$	# 9	1111 1111 1111 1111 11
Good	$\pi 10$	# 6	0000 0000 0000 0000 00
Faulty	$\pi 10$	# 6	0000 0000 0000 0000 00
Good	$\pi 10$	# 9	1111 1111 1111 1111 11
Faulty	$\pi 10$	# 9	1111 1111 1111 1111 11

faulty itself, it would also have been isolated. This demonstrates the capability of isolating multiple faults.

The final simulation experiment involved altering the basic IAR circuit to reflect a signal short between bits 4 and 14 of the register. The test set used has sufficient fault-detection capability to recognize this nonclassical fault. The COMET approach makes it possible to isolate the fault, as indicated by the simulation results shown in Table III.

These experiments bear out the results expected for isolating single, multiple, and/or nonclassical faults. They also point out the dependence on a good set of tests. The signal short is an example of this dependence. Normally, a set of tests capable of detecting all stuck-at-1 or stuck-at-0 faults might not detect the short. However, if the fault is detectable, it is isolated by using COMET. Considerations such as these will obviously have some impact on the diagnostic program design.

IV. TRADE-OFFS

Most of the discussion thus far has considered diagnostic resolution to a single circuit pack. If COMET dictated an all-or-nothing approach to the problem, its usefulness would be severely affected. The cost of

Table II — Simulation results—instruction address register classical fault

Machine	Test (See Fig. 14)	Input Vector	Output Vector (Binary)
Good	$\pi 1$	# 7	0000 [0000] 0000 0000 00
Faulty	$\pi 1$	# 7	0000 [1000] 0000 0000 00
Good	$\pi 1$	#12	1111 1111 1111 1111 11
Faulty	$\pi 1$	#12	1111 1111 1111 1111 11
Good	$\pi 1$	#14	0000 [0000] 0001 0000 11
Faulty	$\pi 1$	#14	0000 [1000] 0001 0000 11
Good	$\pi 1$	#16	1000 [0000] 0001 0000 10
Faulty	$\pi 1$	#16	1000 [1000] 0001 0000 10
Good	$\pi 3$	# 4	0000 0000 0000 0000 00
Faulty	$\pi 3$	# 4	0000 0000 0000 0000 00
Good	$\pi 3$	# 5	1111 1111 1111 1111 11
Faulty	$\pi 3$	# 5	1111 1111 1111 1111 11
Good	$\pi 3$	# 7	0000 0000 0001 0000 11
Faulty	$\pi 3$	# 7	0000 0000 0001 0000 11
Good	$\pi 3$	# 8	1000 0000 0001 0000 10
Faulty	$\pi 3$	# 8	1000 0000 0001 0000 10
Good	$\pi 6$	# 6	0000 [0000] 0000 0000 00
Faulty	$\pi 6$	# 6	0000 [1000] 0000 0000 00
Good	$\pi 6$	#10	1111 1111 1111 1111 11
Faulty	$\pi 6$	#10	1111 1111 1111 1111 11
Good	$\pi 11$	# 5	0000 0000 0000 0000 00
Faulty	$\pi 11$	# 5	0000 0000 0000 0000 00
Good	$\pi 11$	# 8	1111 1111 1111 1111 11
Faulty	$\pi 11$	# 8	1111 1111 1111 1111 11

removing *all* control/observation MSCs is in general quite high. There are a number of alternatives that can and should be evaluated.

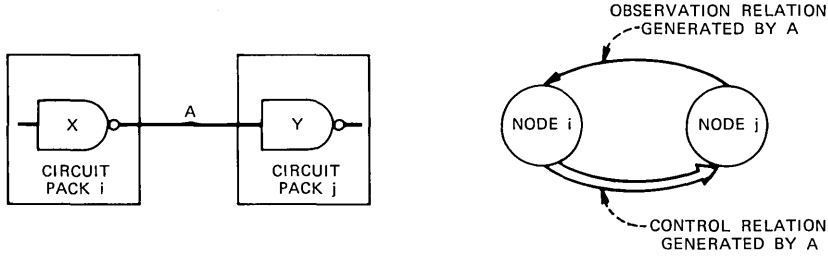
First, packaging must be considered as a trade-off parameter. It is possible that toleration of a slight reduction in packaging density could result in a significant reduction in the number of MSCs left to be broken. Procedures could be derived to evaluate the possible packaging trade-offs, but for now it appears to be a matter of engineering judgment.

Second, the possibility of accepting reduced resolution must also be considered. A vast majority of the loops in the controllability and observability connectivity matrix contain leads going from one pack to another. An example of those leads is shown in Fig. 15a. The *wire* marked A will generate a control relation and an observation relation, as shown in Fig. 15b. COMET analysis of this situation will reveal that to distinguish between an output stuck-at-1 fault (on gate X) and an input-diode-open fault (on gate Y), one must add control or

Table III — Simulation results—instruction address register nonclassical fault

Machine	Test (See Fig. 14)	Input Vector	Output Vector (Binary)				
Good	$\pi 1$	# 7	0000	0000	1111	1111	11
Faulty	$\pi 1$	# 7	0000	1000	1111	1111	11
Good	$\pi 1$	#12	1111	1111	0000	0000	11
Faulty	$\pi 1$	#12	1111	1111	0000	0010	11
Good	$\pi 1$	#15	0000	0000	0001	0000	11
Faulty	$\pi 1$	#15	0000	0000	0001	0000	11
Good	$\pi 1$	#19	1000	0000	0001	0000	10
Faulty	$\pi 1$	#19	1000	0000	0001	0000	10
Good	$\pi 3$	# 4	0000	0000	1111	1111	11
Faulty	$\pi 3$	# 4	0000	0000	1111	1111	11
Good	$\pi 3$	# 5	1111	1111	0000	0000	11
Faulty	$\pi 3$	# 5	1111	1111	0000	0000	11
Good	$\pi 3$	# 8	0000	0000	0001	0000	11
Faulty	$\pi 3$	# 8	0000	0000	0001	0000	11
Good	$\pi 3$	#10	1000	0000	0001	0000	10
Faulty	$\pi 3$	#10	1000	0000	0001	0000	10
Good	$\pi 6$	# 6	0000	0000	1111	1111	11
Faulty	$\pi 6$	# 6	0000	0000	1111	1111	11
Good	$\pi 6$	#10	1111	1111	0000	0000	11
Faulty	$\pi 6$	#10	1111	1111	0000	0000	11
Good	$\pi 11$	# 5	0000	0000	1111	1111	11
Faulty	$\pi 11$	# 5	0000	1000	1111	1111	11
Good	$\pi 11$	# 8	1111	1111	0000	0000	11
Faulty	$\pi 11$	# 8	1111	1111	0000	0010	11

observation to line A. However, if two-pack resolution is tolerable, one can ignore this two-node loop generated by wire A. Experience has shown that a very large number of the problems that graph-theoretic analysis points out are of this type. It is reasonably simple to handle these problems if reduced resolution is tolerable. When



(a) CIRCUIT REPRESENTATION

(b) CONTROL/OBSERVATION REPRESENTATION

Fig. 15—Resolution vs cost trade-off.

COMET analysis points out one of these loops, the engineer can remove either the control or observation entry in the connectivity matrix and place it in a reduced resolution list. Later, after leveling the graph, he will go back and determine a set of suspect packs to be identified when a pack is found to be bad. Thus, the most probably faulty pack is named along with packs that could have stuck-at-1 outputs. The information to do this is available in the previously created reduced-resolution list.

It is likely that "hybrid" techniques are also possible. A small number of faults that would be costly to handle by COMET could be fault simulated. The simulation of this small set of faults is much less expensive than the totally fault simulated case. This approach, however, is open to the same criticism as the traditional exact-match TLM. Consistent results may be hard to come by and one also must update the fault simulation as any changes are made.

COMET is obviously of most value when applied completely. Engineering considerations may point to this being undesirable. In this case, a number of trade-offs and less costly variations on COMET can be used.

V. IMPACT OF COMET

5.1 Design for maintainability

The major impact of COMET should be a partial redefinition of design philosophies for digital systems. COMET is, after all, an *engineering technique.* It is capable of identifying diagnostic resolution problems before the design has been frozen and allows the design engineer to consciously evaluate the possible problems. To proceed in the analysis, the engineer must determine what is to be done about the problem. His decision may lead to the compilation of a table-of-resolution problems. This list can give an idea of the overall fault resolution possible for a design. Iteration of the process will allow full evaluation of the cost/resolution trade-off before the design is committed to hardware. With this tool, diagnosability takes its place as a primary design parameter with a reasonably well-defined set of trade-offs for evaluation.

Efficient application of COMET depends in part on the orderly application of the technique. Analysis should first consider rather global functional blocks. The information gained at this stage of analysis provides the basis for optimization of the control and observation relations. The global analysis will usually determine which of the observation edges to retain. It can also help prescribe necessary

external control and observation of the system. In many cases, the ability to provide only necessary external access can lead to minimization of the interaction of systems and associated sanity-preservation problems.

Once the global analysis phase has been completed, the expansion of the global functional nodes on a local (node-by-node) basis can take place. The task is to expand the global functional node into identifiable entities. COMET is then used within the global node to specify an organization that is diagnosable. The problem has been converted to one of modularization. By carefully considering the amount of connectivity of nodes, packaging problems can also be anticipated.

One outgrowth of the proposed design approach should be an increased awareness of the functional aspects of a system. This, in turn, should lead to more attempts at functional testing and an increased ability to do localized test design. COMET requires conscious consideration of test design and conscious consideration of the controllability and observability of an entity. It further requires that diagnosis be ordered such that these input/output ports are verified prior to testing the entity in question. This information may be sufficient to allow test design and verification on a local basis. This would lend itself to algorithmic test generation and inexpensive verification. The ordering process, if adhered to, could also substantially reduce the problem of having to simulate faults in large blocks of circuitry for test-design purposes.

Probably a more important aspect of the functional orientation of diagnosis is the emphasis on detection. Traditional diagnostic design relies on detection of all single stuck-at types of faults. In integrated-circuit technology, this may be a somewhat restricted subset of all possible failures. The nonclassical failures may cause trouble in traditional diagnosis, even though they are nearly always detectable. With COMET, detection is the only thing that is important.

5.2 Other applications of COMET

COMET has a number of advantages over normal diagnostic procedures. In the simulation experiments, the ability to diagnose multiple independent faults was touched upon briefly. This ability means that a machine designed using COMET can be tested upon installation with the regular diagnostic program. The procedure would be to insert a selected number of the highest-order nodes and run the diagnostic. The missing packs appear to be disabled. The initial packs can be diagnosed and, if no faults are found, the process can be con-

tinued. Otherwise, the faulty pack or packs are replaced and the process repeated. The ability to isolate nonclassical faults was also touched upon briefly. The impact of this on the diagnostic program must be evaluated.

It should be noted that COMET can be used at various levels of design and analysis. For instance, the first analysis may assume functional nodes to be of the subsystem size. At this level of detail, the overall system diagnostic philosophy may be evaluated and improved. Next, each functional node of subsystem size may be subdivided into functional nodes of circuit-pack size. Assuming that the circuit packs may eventually be composed of LSI chips on a ceramic, it may be feasible to consider subdividing the circuit-pack-sized functional nodes into LSI-chip-sized functional nodes. The potential benefit here is in the area of factory repair. If COMET is applied at the chip level it becomes possible to isolate the faulty LSI chip using factory test apparatus. Faulty chip location is done by automatically disabling chip outputs and testing the remaining chips. Conceptually, this could simplify the problems of repair of LSI packages.

5.3 Further work

There are several aspects of COMET that will require further work. Most apparent is work intended to automate a large part of the analysis and synthesis phases. This work appears to be of an open-ended type. A system that iteratively seeks a near optimum solution while considering physical design and other constraints would need to be a sophisticated system. The system of program aids that has been implemented as part of LAMP represents a start in this direction.

In addition, some thought must also be directed to obtaining the functional node definitions. These do not of necessity have to parallel the function performed, but can reflect such techniques as bit slicing and other packaging considerations.

The idea of relying on routine exercise procedures to check failures in logic disabling is not very appealing. It is possible that improvements can be made in this area.

VI. SUMMARY

The concept of controllability and observability for organized system design to enhance diagnosability has been introduced. It makes use of logic disabling of circuit modules (or packs) and the proper ordering of functional nodes of a system. Both binary and linear ordering can be used. Other arrangements are possible but have not

been investigated. Graph-theoretic algorithms for determining the optimum placement of control and access and monitor points for diagnostic testing were presented.

A feasibility study was performed by applying COMET to an existing processor. The capability of locating single, multiple, and non-classical faults was demonstrated, based solely on the pass/fail indication of test partitions. Additional control and monitor circuitry was added to satisfy the requirements of COMET. The added hardware, which included both the logic for disabling and for modifications is modest (e.g., less than 10 percent).

For a given set of tests, significant savings in simulation time to generate TLM data are achievable for a design using COMET. Furthermore, if a laboratory or a field model of the system is available, the TLM data generation effort will be almost totally eliminated. Data can be generated by actually running the diagnostic program on the machine for which a TLM is to be produced. This approach also makes TLM update easy and therefore drastically reduces many problems caused by machine differences and circuit changes. Finally, it offers the possibility to tailor-make TLMs that can be stored on-line.

Application of COMET is best suited for new designs where the trade-offs among circuit design, logic packaging, and diagnosability can be jointly considered early in the design stage. The use of machine aid tools would make this feasible and could greatly facilitate the design process. Retrofiting COMET to existing systems may be feasible on many designs, but may not be economical on others. Engineering judgment must be exercised to study the impact.

REFERENCES

1. H. Y. Chang, E. G. Manning, and G. Metze, *Fault Diagnosis of Digital Systems*, New York: John Wiley and Sons, 1970.
2. IEEE Trans. on Computers, Special Issue on Fault-Tolerant Computing, *C-22*, No. 11 (November 1971), pp. 1241–1435.
3. H. Y. Chang, R. C. Dorr, and D. J. Senese, "The Design of a Microprogrammed Self-Checking Processor of an Electronic Switching System," IEEE Trans. on Computers, *C-22*, No. 5 (May 1973), pp. 489–500.
4. C. V. Ramamoorthy, "A Structural Theory of Machine Diagnosis," AFIPS Conference Proceedings, 1967 Spring Joint Computer Conference, *30*, 1967, pp. 743–756.
5. C. V. Ramamoorthy and L. C. Chang, "System Segmentation for the Parallel Diagnosis of Computers," IEEE Trans. on Computers, *C-20*, No. 3 (March 1971), pp. 261–270.
6. H. Y. Chang, G. W. Heimbigner, D. J. Senese, T. L. Smith, "Maintenance Techniques of a Microprogrammed Self-Checking Control Complex of an Electronic Switching System," IEEE Trans. on Computers, *C-22*, No. 5 (May 1973), pp. 501–512.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

LAMP:

Application to Switching-System Development

By T. T. BUTLER, T. G. HALLIN, J. J. KULZER, and K. W. JOHNSON

(Manuscript received March 26, 1974)

Specific attempts have been made at Bell Laboratories to shorten development intervals, to improve the quality of system design, and to improve unit, manufacturing, and system testing by widespread application of the LAMP system during development of the 1A Processor and No. 4 Electronic Switching System. LAMP has played a major role in two areas of the design process—design verification and fault simulation. Although ten major digital subsystems of No. 4 ESS and the 1A Processor are now being simulated on the LAMP system, this paper describes the experience gained during development of two of the 1A Processor subsystems, central control and program/call store.

I. INTRODUCTION

Computerized development aids have become an integral part of the development of large, complex, electronic systems. One such aid, the LAMP system, is finding widespread application in the development of the 1A Processor,[1] a stored program processor, and in No. 4 ESS,[2] a new switching system that employs a solid-state, time-division, digital-switching network.

The need for computerized aids in the development of advanced switching systems is vital for several reasons. Vast amounts of engineering and manufacturing information must be generated. Complicated design decisions coming from engineers of diverse disciplines must be coordinated, and with the increasing complexity of systems and electronic technology comes the need for more thorough and consistent testing at all stages of design and manufacture. As the computer increases in power, it plays a greater role in reducing manual design effort, enhancing design quality, improving the accuracy of information transfer, and making more complex designs economically feasible.

For these reasons, the LAMP system has been used on TSS 360/67 to simulate ten major subsystems in No. 4 ESS and the 1A Processor. The experience gained during the development of two of these subsystems, central control (the heart of the 1A Processor, which provides program execution and overall executive control) and program/call store (a magnetic core memory unit that provides storage for ESS programs and temporary call-related data) is detailed here.

To facilitate maintenance of design information and provision of adequate testing for the project, computerized data bases have been implemented for all design information. The combination of common data bases for hardware and software design information, the LAMP simulator, and the conversational features of an interactive host computer have proven quite effective in the hardware and test design for the project.

There are two major ways in which the LAMP system has been used in the development process: design verification and fault simulation. These will be discussed separately and in detail in Sections II and III, respectively, of this paper. Briefly, design verification consists of demonstrating that the unit being simulated performs the functions it was designed to perform, with no faults present. Fault simulation, on the other hand, consists of inserting faults into the simulated unit and testing the ability of maintenance programs to detect and isolate those faults.

Three categories of test programs are used in the simulation of the previously mentioned units. They are:

(*i*) Circuit pack level tests, which are used in design verification, fault simulation, and pack testing.

(*ii*) Diagnostic tests, which are the primary tool for both design verification and fault simulation at the complete unit level. The diagnostic tests are written in a high-level language, concurrent with the design of the hardware. They are intended for factory and installation tests as well as for on-line fault detection and repair in an operating system. Total fault detection is the ideal primary goal, with good resolution the secondary goal.

(*iii*) "Special" test programs, which are used only for design verification and are intended to test the functional capability of the simulated unit (e.g., its ability to execute the program), and to test complex interactions between different portions of the unit.

Initially, LAMP was used to simulate each digital circuit pack of the 1A Processor (*i*) to verify the design, (*ii*) to verify that sufficient

access was available on the pack to detect all classical faults (output stuck high or low, open input), and (*iii*) to verify that the tests were capable of detecting these faults. This circuit pack level of simulation continued throughout the development process as changes were made to circuit packs and new packs were issued.

A second, temporary phase for some units was the simulation of a functional part of a unit before the complete design was available. This allowed design verification through simulation to begin while other functions were being designed. At this level, many special test programs described previously were used.

Finally, as the complete design became available, complete unit simulation was begun. At this level, the diagnostic tests were used, and the majority of design verification and fault simulation was done.

II. DESIGN VERIFICATION

2.1 Circuit pack design verification

This section describes the use of LAMP simulation in the design verification of 1A Processor circuit packs. This is differentiated from design verification at the complete unit level and from fault simulation of circuit packs, which will be discussed in later sections.

2.1.1 Objectives

Substantial time and expense are required to produce an artmaster for a 1A Processor circuit pack (100 to 400 logic gates) and then to produce the first hardware version of the pack. It becomes important, therefore, to verify the accuracy of the design before this process begins. The purpose is to test the ability of the design to perform the intended functions as completely as possible. In addition to new designs for circuit packs, changes inevitably must be made during the course of the project. Again, it is important that the design of these changes be verified before the time-consuming process of modifying the hardware begins. For this reason, design verification of circuit packs, through simulation, is done not only early in the development process but throughout its course.

2.1.2 Circuit pack simulation

The mechanics of circuit pack design verification by simulation consist of building a LAMP model of the pack, devising a set of tests, running the tests and interpreting the results, and updating the model so that proposed corrections of design errors may be tested immediately. Building and updating the model are described under complete unit simulation (Section 2.2.2).

2.1.3 Circuit pack tests

Design verification at the circuit pack level uses tests written by the pack designer or another person familiar with the design and information generated automatically via the automatic test generation (ATG) program.[3] The handwritten tests consist of sets of inputs (vectors) to be applied to the circuit pack. During simulation of the vectors, the outputs of the pack are continuously monitored and, thus, may be compared to a set of expected results. An effort is made to make a complete "active" test of every output, i.e., to ensure that each output is active when the inputs are selected to make it active. However, "inactive" tests of each output (insuring that the output is not active when it should not be) are necessarily limited to those the test designer feels to be high-probability cases. An exhaustive set of inactive tests can be prohibitively large for even a relatively simple function.

The ATG program is intended primarily for the generation of factory tests for the circuit packs, but it proves useful for design verification as well. This program approaches test generation from the same standpoint as the LAMP simulator, as is discussed in detail in a companion paper.[3] One result of this approach to test generation is that the program effectively determines the Boolean function for each output as it actually exists on the pack. By printing these functions in a form so that they can be compared to the functions intended by the pack designer, ATG is an effective tool for design verification of combinational and many sequential packs. By recreating the function from the gate level information, ATG effectively makes both "active" and "inactive" tests of the function design.

2.1.4 Experience

The simulation of the 1A Processor circuit packs for design verification was not a one-time occurrence, but continued throughout the design process. During this simulation procedure, some errors were found on a majority of circuit pack codes. Without circuit pack simulation, these errors would not have been found until complete unit simulation, or perhaps not until testing of the first hardware model of the unit. In most cases, even if the errors were detected at the complete unit level of simulation, the expensive task of generating the artmaster for the pack would already have been completed.

The tests developed for circuit pack design verification served as a substantial portion of the factory tests for these packs. The process of achieving complete fault detection capability uncovers redundancies,

thereby enhancing design verification. Section III describes the fault simulation of circuit packs.

2.2 Complete unit design verification

Inherent in major unit simulation is the need to first verify the accuracy of the simulated unit (or simulation model). This is done by running the diagnostic tests on the simulated unit with no faults, then comparing the simulation results with the expected results for each test. This verification procedure proves useful in three ways: it verifies the functional design of the unit, it verifies the accuracy of the design data base, and it verifies the design and expected results of the diagnostic tests.

2.2.1 Objectives

In the design of large digital units, a time lag exists between the completion of the initial design and the arrival of the first manufactured units. Before the advent of high-speed integrated circuits, the unit was breadboarded, in many cases, to allow continuing design feedback while waiting for the manufactured unit to arrive. However, with the increasing complexity and the higher integration levels of digital systems, it may no longer be feasible to breadboard a digital unit for design verification purposes. LAMP simulation now provides an alternative to breadboarding for units as large as 40,000 gates. LAMP was selected over other alternatives for speed, flexibility, economy, and capability.

The use of simulation has significantly decreased the design interval. Once the initial design is completed, it is easier to make a LAMP model of the unit than to manufacture it. Thus, design verification can begin well before the factory model arrives. As is discussed later, features in LAMP make testing the simulation model comparable to testing the hardware unit itself.

Simulation reduces the number of changes that must be made after the unit is manufactured. Because integrated circuits are being used, changes no longer involve just adding or deleting wires from a wire-wrapped backplane. Now changes may require difficult modifications to printed wire or thin-film circuits or to multilayer backplanes. As the changes become more complex, they also take more time. Critical changes may halt all other debugging until the new change can be designed and implemented. The change facilities in LAMP permit fixes involving a small number of gates and wires (under 50 changes) to be implemented almost immediately. Larger changes can be pre-

pared in less than a day, when required. This means that, with simulation, debugging can continue without significant delay when changes are encountered. By simulating early, many changes can be found and incorporated into the design before the first unit is manufactured.

Diagnostic tests are required for the first and all subsequent units in manufacturing. One objective of simulation is to verify the diagnostic test design before testing the first unit being manufactured. The unit test interval can be reduced significantly if it is known in advance that the tests are correct and that the problem is a malfunction in the unit being tested.

The remainder of this section discusses how simulation is used for logic design and diagnostic test verification, and how it fulfills the objectives of decreased design interval and reduction in change activity after manufacture.

2.2.2 Model building and updating

Figure 1 is a diagram of the simulation process. As the hardware design moves toward completion, the information is encoded into a design data base. This data base is used to generate all information for the LAMP model, the circuit pack and interconnection drawings, the artmasters for the circuit packs, and wiring information for the backplane. When the data base is complete, the simulation model is constructed from the information in the data base. This is done in two stages. First, an LSL-LOCAL[4] description of the unit is generated

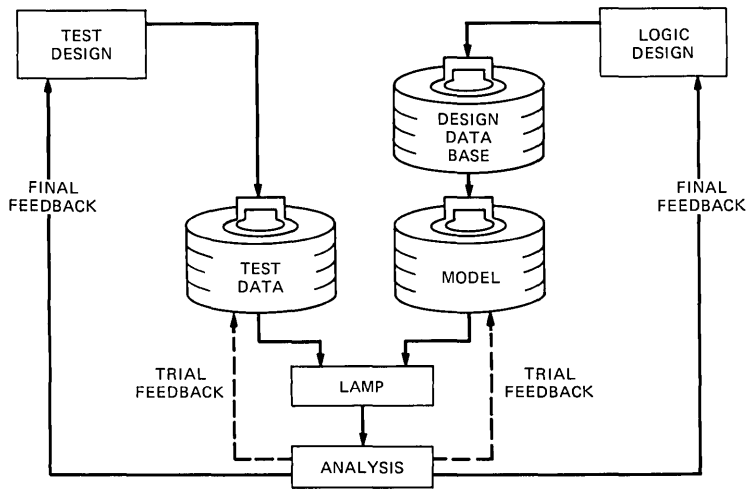


Fig. 1—Design verification.

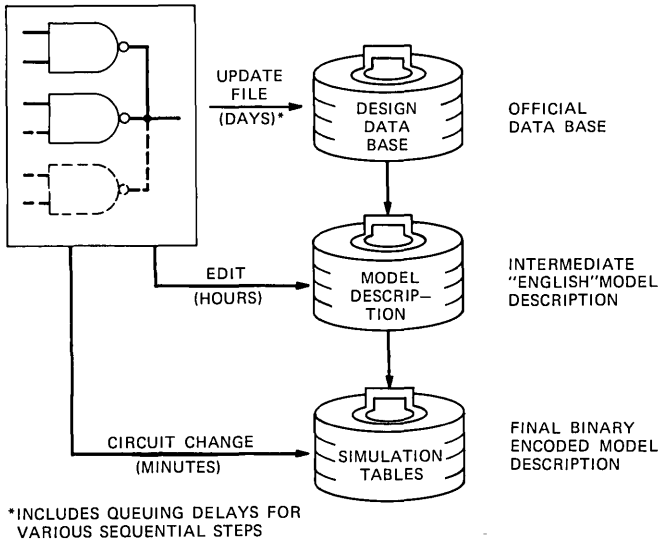


Fig. 2—Model updating process.

from the data base. This LSL-LOCAL description is then compiled into a data set of simulation tables for the model.

The verification of the data base begins as the simulation model is being constructed. Error diagnostics in the programs that create the simulation model find some inconsistencies in the data base. Once the model has been constructed, more errors can be found by running the diagnostic tests.

To make this process practical and easy, methods to change the model must be available. Figure 2 illustrates the change process. To have the change officially issued into the central data base and then to create a new model is a lengthy task, a result primarily of queuing delays in the various sequential steps. The data base has to be updated and a new model must be generated. To shorten the model updating time and to verify changes before they are officially issued, two other methods are used to modify the LAMP model. First, a text editor can be used to update the LSL-LOCAL description of the circuit, which is then compiled into a new set of LAMP tables. Second, the LAMP CKTCHANGE[4] command can be used to modify the existing tables directly. For small units, text editing and recompilation is as fast as using CKTCHANGE. Therefore, this procedure is used for small units, while for large units, because of its speed advantage, CKTCHANGE is used. Any change required is first put into the model by

```
WRITE WØRD (XR), DATA (ϕ(52525252))
READ WØRD (XR), EXPECT (ϕ(52525252))
WRITE WØRD (YR), DATA (ϕ(77777777))
READ WØRD (YR), EXPECT (ϕ(77777777))
RUN CYCLE (2)
READ WØRD (XR), EXPECT (0)
```

Fig. 3—Sample diagnostic test.

CKTCHANGE or by editing the LSL-LOCAL description. The change is then verified by further simulation. If corrections are found necessary after simulation, the change is updated and retried. Only when the change is correct will it be added to the official file. This results in fewer changes being made to the official design data base, thus lessening the chance for error.

2.2.3 Simulation procedure

Once a model has been built, tests are needed to verify the correctness of the model. The diagnostic tests are chosen as the main logic verification tests since the test design schedule closely parallels the logic design and, thus, most tests are ready at the time the simulation model is generated. The tests are written in a high-level language from which they are easily compiled into the input language for the simulator. Each test includes an explicit expected result so, while simulating, it is easy to ascertain if the test being simulated is passing or failing. Figure 3 is an example of a simple test. The X and Y registers are initialized and then read to verify the initialization. The run statement executes the test and is followed by a read to determine if the proper action occurred during the test.

Given the LAMP model for the circuit, a set of tests to simulate, and the ability to make quick changes, design verification may begin. The standard procedure is to simulate a complete functional block of tests called a phase. During the simulation, a list of test failures is automatically produced by LAMP. These failures could be hard errors or logic 3 (output in unknown state) propagated to the output. The data from the failing tests are analyzed by the logic designer to determine the source of the errors. Frequently, the solution to a problem is obvious from the test results. In other instances, a follow-up run is required to determine the exact cause of the error.

LAMP provides two basic debugging facilities, an oscilloscope-like timing trace and a stop-and-display feature. Some follow-up runs in-

volve simulating the failing test and generating an oscilloscope-like output trace covering a large number of points. Typically, 100 to 500 points may be traced. This type of trace is effective when the designers know where the error is likely to be. The timing trace is also used extensively with special tests of critical timing functions and complex sequencer interactions. This output has proved to be the most effective technique for uncovering circuit timing problems.

Another type of follow-up run involves simulating the phase in the conversational mode and imbedding stops in the simulator. The stops are activated when a preselected gate changes to a specified value. When the stop is activated, the simulation is suspended, and the logic designer can look at the state of any gate at that instant in time. This allows the designer to trace the trouble back to its source. Imbedded stops are used if the time of occurrence of the problem is known or if the problem is isolated to a particular gating lead changing state for an unknown reason. In the first case, a stop is planted at a particular time. The simulation is run up to this point and then stopped. Using the DISPLAY facility in LAMP, the designer then displays the states of critical gates. The DISPLAY command presents the current value of the gate along with its fan-in or fan-out gates, plus their logical level (value). The values are those at the time the simulation stopped. If these gates are in the wrong state, then the values of the inputs are checked to trace back the problem. The tracing continues until the source of the error is determined. In other cases, the value of a gating or data lead is used to activate the stop. This is used when it appears that the problem is caused by a function changing state at the wrong time. Again, the display facility is used to track the problem back to the source.

Once the trouble is isolated by oscilloscope-like tracing or imbedded stops, either the circuit is changed, if there is an error in the logic design or in the data base, or the diagnostic test is modified, if there is an error in the tests. The simulation is then rerun to verify the correction. These two procedures are so effective that, for some problems, the designers have preferred debugging the logic and tests via simulation even after hardware models are available. Running tests on simulation is slower than running the tests on the unit; however, debugging is facilitated by the ability to suspend the simulation and to observe large numbers of points internal to a circuit pack that are not observable on the hardware model. This has significantly reduced the design verification interval and, by finding the errors during simulation, many costly changes have been eliminated.

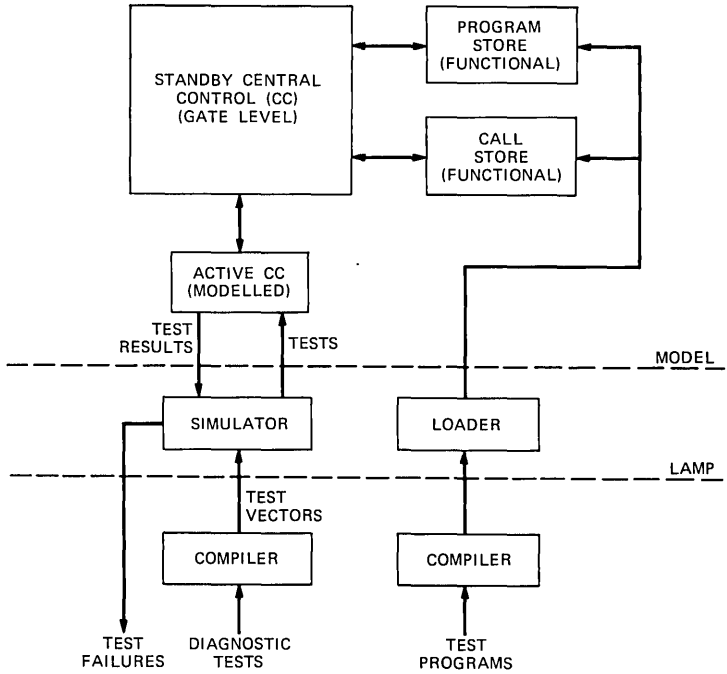


Fig. 4—Central control simulation.

2.2.4 Experience

The central control for the 1A Processor was the first and largest unit whose logic was extensively verified using LAMP. The experience gained in verifying the central control is presented here to illustrate the effectiveness of LAMP for logic and diagnostic test verification. Figure 4 is a functional block diagram of the simulation models and interfaces. The 1A Processor contains duplicate central controls. When diagnostic testing is performed, the active central control tests the standby central control through an interface port. In the simulation process, the LAMP model for the complete standby central control was created. A simplistic model of the active central control was created to take the tests and apply them to the standby central control at the proper time. The active central control model contains an oscillator that could be started and stopped, allowing the circuit to settle and a new test to be applied at the appropriate time. In the active central control model, a comparator circuit was built in to check the actual results of a test with the expected results. Whenever the error flag gate became active, a message was printed listing the failing test. For conversational

simulations, this allowed termination of simulations if the number of failures became too large.

In addition to the diagnostic tests, a series of special test programs was simulated that was designed to test program execution functionally. Since these programs were not written in the high-level language of the diagnostic tests, they were not simulated in the same way. Instead, using LAMP features, a functional model consisting of a program store and a call store was constructed and connected to the gate model of the central control. The programs were compiled and loaded into the memories using a special memory loader facility. Special LAMP vectors were written to initialize the standby central control model and then to release the oscillator. This allowed the clock to run continuously and to simulate actual program fetching and execution. Conversational LAMP control procedures were used to stop the simulation when the program transferred to the error address or when it reached the return address. These programs allowed operational testing in addition to the diagnostic testing.

For the central control verification, approximately 20,000 diagnostic tests and 4000 words of program were simulated. The simulation of all the special tests and most of the diagnostics was completed before the first unit arrived from the factory. Through simulation, approximately 85 percent of the logic and 80 percent of the diagnostic tests were verified. The areas remaining to be tested were primarily the circuitry and tests that interconnect the central control with its system environment. At the present time, this type of testing is beyond the capability of LAMP, mainly because of the large number of gates required to model all the units connecting with the 1A central control.

Many diagnostic and logic changes were generated as a result of simulation. LAMP is an indispensable part of the development process. It has proven effective in reducing development intervals and in reducing the number of circuit modifications.

III. FAULT SIMULATION

3.1 Purposes

Fault simulation is necessary to determine (and enhance) the detection level of the diagnostic tests and can be viewed as an extension of the design verification process. The true behavior of a circuit is studied during verification. The set of other possible behavioral responses can be ascertained with fault simulation. This process is systematic; responses are derived on a fault-by-fault basis. This type of information is essential to the 1A Processor subsystems because (*i*) it contributes

to meeting stringent factory test requirements, and (*ii*) the complete characterization of subsystem behavior is necessary to satisfy long-term in-service system maintenance requirements.

Fault simulation at the circuit pack level is not complicated and is discussed in the next section. On the other hand, unit level simulation is quite complex. Many factors are involved, including large gate counts, precise timing, complex fault modeling considerations, and limited computer resources. These are discussed in the remaining sections.

3.2 Circuit pack simulation

The size of 1A Processor circuit packs (100 to 400 gates) makes fault simulation on LAMP an easily managed process. Simulation models of packs are extracted from the central design data base. These models seldom require any additional modeling changes. Classical faults are simulated for every gate in the circuit. Tests are designed to detect every simulated fault, if possible. A mixture of manual test design and automatic test generation via ATG is used.[3] The process is facilitated through the use of interactive LAMP on the IBM 360/67 or 370/168 computer.

Each test consists of one or more input vectors and an expected set of outputs. The input vectors are simply strings of 1 and 0 combinations that are applied to the inputs of the pack. The tests need not be functionally arranged, but may be independent of each other. Testing is not "clocked," inputs are applied simultaneously, and outputs are examined well after the circuit has settled down. Test minimization is not stressed, since the cost of simulating each fault per test is small, and redundant test sequences may improve detection of nonclassical and/or multiple faults.

As described in Section 2.1, verification tests were generated using a combination of manual and ATG techniques. These tests were then run in the fault simulation mode, achieving an average of 75 percent fault detection. Additional tests were generated with manual and ATG techniques to achieve 100 percent fault detection.

The ATG program was used on about 50 percent of the packs to increase fault detection to about 90 percent although, on some combinational packs (about 10 percent of total), ATG immediately produced 100 percent detection. Manual techniques were required to provide detection of the last 10 percent of the faults on most packs. During this process, design redundancies and other bugs were uncovered. Overall, fault simulation at the circuit pack level produced debugged

factory tests and helped uncover design bugs, thus significantly reducing the intervals required for manufacture.

3.3 Unit modeling

The logic model used for unit level design verification forms the basis for the fault simulation model. In some cases, it may be necessary or desirable to make alterations. Simulated logic used only to support verification studies may be removed to save simulation time. It may be necessary to manually append additional models of specialized circuitry not kept in the design data base. Typical devices are discrete analog components such as operational amplifiers, current drivers, and signal buffers. This circuitry communicates with the logic and affects its state, but is nonlogic in nature. Often, many one-of-a-kind models must be constructed through truth table and timing diagram studies. In some cases, circuitry exists that is not modeled. The 1A Processor call/program store core memory is an example. While functional simulation can be used to model the memory for design verification at the present time, LAMP cannot support fault list propagation into and out of a functional model.

Once the fault model is completed, circuit initialization must be considered prior to simulation. Ideally, simulation should begin with the circuit in an unknown state. This represents the most accurate approximation to the physical circuit whose state prior to testing is not necessarily predictable. The unknown state approach is normally used for design verification simulation, but not for fault simulation. This is because starting from a known state significantly reduces the excessive simulation CPU time caused by the potential for large fault list buildup during initialization. The resulting loss in accuracy is small.

A known state is normally achieved by applying an initializing sequence to the circuit using true value simulation. Fault simulation is conducted starting with this true-value state and a set of "null" fault lists.

3.4 Test selection

Test selection is an important consideration for large unit fault simulation because it is costly to simulate an input vector, and each test usually expands into a series of from two to ten such vectors.

The decision concerning which tests to simulate is influenced by the particular objective, the circuit model, and the available computer resources. Fault simulation to evaluate early tests or to support the

improvement of tests (test enhancement) is concerned with the detection of classical faults. This objective permits the exclusion of tests designed to improve fault isolation or to detect lead shorts (such as a walk of 1 through a field of 0's). Test exclusion is also necessary to size the tests according to the capabilities of the simulation model. This is most evident in tests that deal with the system environment, such as tests of "interunit" communication buses or of nondigital circuitry such as memory. A set of tests may also be simulated versus only a fraction of the faults. This is discussed further in Section 3.6, under "simulation strategy."

How the tests are sequenced is also important. Functional ordering (grouping tests according to the circuitry being tested) is essential to make the tests useful as a repair vehicle. The test phase is considered the basic functional entity whose predesigned sequence must be maintained. In some cases, tests may be deleted from a phase during simulation, but the order of remaining tests is preserved. During test evaluation, phase ordering is not essential. How or when a fault is detected is of little consequence at this stage. Functional phase ordering assumes greater importance when data are collected to support trouble location dictionaries. Also, tests excluded from earlier stages of simulation are included here wherever possible.

3.5 Fault selection

The 1A Processor uses conventional TTL integrated circuits for logic function implementation. Discrete circuitry augments this logic where required. The major subsystems use LAMP to simulate "stuck-at" (input open, output stuck at 0 and at 1) logical faults on TTL gates. Shorted fault simulation has, to date, not been used at the unit level. The restriction to stuck-at fault simulation is an engineering decision. In an actual circuit, the possible failure modes are much more extensive, including not only lead shorts but also timing changes, voltage changes, intermittents, etc. It is assumed that the majority of these faults will behave as stuck-at faults for a portion of the tests. Experience with previous electronic switching systems has shown this assumption to have validity.

Since there are several hundred thousand possible stuck-at faults in 1A Processor subsystems, the fault selection process must be facilitated by various options available in LAMP. Depending upon application, faults are selected on an individual basis by gate name, by circuit pack, or to a certain extent by hardware function.

When detection information is being sought, random sampling is used. A sufficiently large random sample has been found to predict detection levels accurately. According to requirements, samples are selected on a uniform, localized, or stratified basis.

Several automatic fault administration options in LAMP have reduced 1A Processor simulation costs significantly. Early termination is used extensively during test evaluation and enhancement. Under this option, a fault is terminated (removed from simulation) after one "hit," or detection. Typically, this option saves from 50 to 75 percent of simulation time. Fault collapsing removes $n - 1$ out of n logically equivalent faults from the fault set being simulated. For example, if five faults cause the same effect upon a logic circuit, LAMP simulates one of the five. Typically, this option reduces the fault set by 25 to 50 percent. Undetectable fault elimination, which removes faults such as those on unused gates, reduces fault sets up to 10 percent.

Even after every attempt to prune the fault set has been exhausted, it is still necessary to form fault partitions for typical large unit simulation runs. The degree of partitioning necessary depends upon the size of the unit, the tests, the circuit topology, and the computer resources. The primary factor influencing fault partitioning is the available computer main memory. Using an IBM 360/67, typical fault sets are in the 3000- to 6000-fault range. Using an IBM 370/168, sets as large as 30,000 faults have been simulated.

On the other hand, test partitioning (the smallest partition being a phase) is primarily influenced by simulation time limits. Some partitions may require 1 or 2 hours of processor time.

For protection, the LAMP CHKPOINT/RESTART facility is used to permit rollback in the event of a computer, simulator, or procedural failure.[4]

3.6 Simulation strategy

An interactive simulation procedure is used in the 1A Processor for test enhancement. A random sample of faults is selected and simulated with the diagnostic tests. The results are analyzed to reveal undetected faults for which additional tests are designed. A new sample is then selected, consisting of the previous undetected faults plus an additional random sample, and the process is repeated. The size of a fault sample is generally larger than the minimum required to predict detection levels. This increases the probability of revealing classes of undetected faults. Ultimately, large classes are eliminated, and a point of diminish-

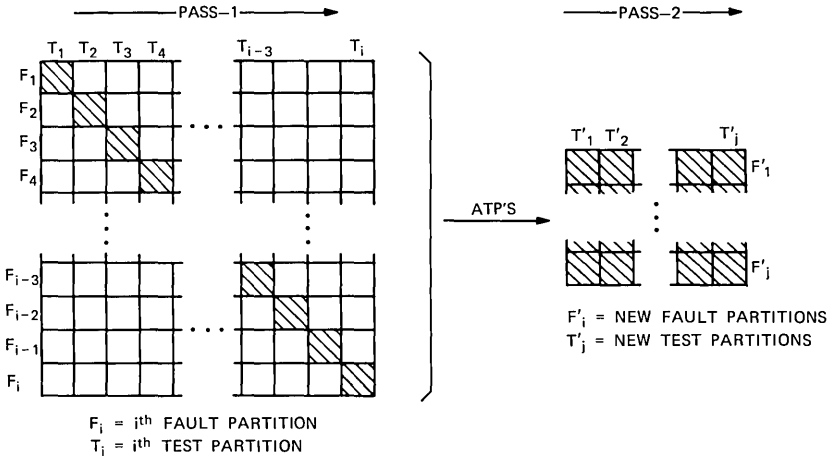


Fig. 5—Two-pass diagonal simulation.

ing returns is reached. At this stage, complete fault detection information is secured by simulating all remaining faults with the complete set of tests.

Selection of several fault partitions (such as a random sample divided into four parts) that must be simulated against many phases, a phase at a time, requires the execution of many successive simulation runs. The most obvious way to execute these runs is to simulate the various combinations of fault and test partitions in succession. This approach does not prove efficient, and significant cost reductions are possible through an alternate strategy called "two-pass diagonal simulation." Figure 5 is an illustration of this method. In the ideal situation, i mutually exclusive fault partitions and i test partitions are selected. Each fault partition is chosen in the area of circuitry that a corresponding test partition is designed to exercise. This correspondence significantly increases the probability of fault detection per second of processing time. Pass 1 simulation then consists of i runs, not i^2. Early termination is used to remove detected faults prior to pass 2. In pass 2, undetected faults are collected and repartitioned into a smaller set of j runs, and all combinations of partitions are simulated. The effectiveness of this method lies in the fact that, for a minimum of resources, the great majority of faults are detected in pass 1. From a practical point of view, mutually exclusive fault sets with clear test associations are hard to produce. Instead, overlapping fault partitions are usually selected. Pictorially, this means that, in Fig. 5, regions off the major diagonal would be lightly shaded. This reduces slightly the

efficiency of the method, but the savings are still significant. Results have shown that tightly connected circuits benefit least from this technique because the concept of localized fault detection tends to break down.

The method just described is used primarily for simulation where detection information is being sought. In other applications such as data collection for trouble location dictionaries, the procedure of Fig. 5 is not used since it produces incomplete fault behavior information. In such cases, the simple strategy of simulating all fault and test partitions may be utilized with corresponding CPU time increases.

3.7 Experience

A composite LAMP model of the 1A Processor call/program store was constructed to support design evaluation and diagnostic development. Figure 6 is a block diagram showing interrelations of distinct portions of this model.

The model contains approximately 10,000 LAMP gates. This count is about 40 percent higher than the real gate count because of attendant logic controlling bus interaction and specially modeled "nonlogic" circuitry. This model was verified in the manner described in Section II. A preliminary diagnostic was designed consisting mainly of functional exercise tests, which translated into about 3000 LAMP input

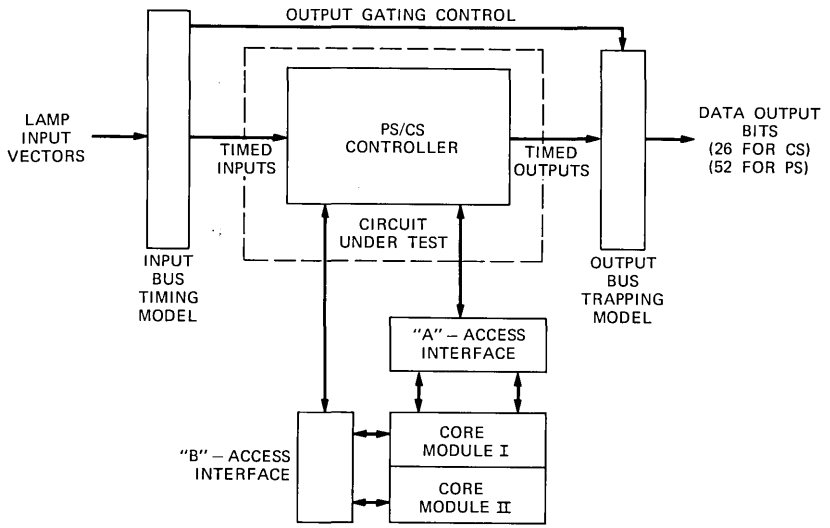


Fig. 6—Program/call store lamp model.

Table I — Program store simulation data

Statistic	15,000 Faults	580-Fault Sample
Number Detected*	10,400	400
Number Race Faults	2200	83
Number Oscillation Faults	80	3
Detection with Races Removed	81%	80%
Detection Estimate Following First Test Enhancement Iteration	90%	90%
Estimate of Ultimate Detection Level	96%	96%

* Using initial set of tests.

vectors distributed over a number of phases. At the time, the inability to model the core memories with LAMP gates necessitated omission of several test phases.

A study was conducted using the model of Fig. 6 and these tests to meet the following objectives:

(*i*) Determine the detection power of the tests.
(*ii*) Provide data to support test enhancement.
(*iii*) Evaluate proposed methods of data collection for use with other units (this was a pilot study for large unit fault simulation).

The model contained a set of 15,000 meaningful classical faults. As a first experiment, this set was simulated using the two-pass diagonal simulation method of the previous section. As a second experiment, a random sample of 580 faults was selected and simulated against all tests. In the first experiment, the number of faults in each partition was chosen to optimize use of the host computer. For the random sample, it was possible to simulate the 580 faults in one partition. About eight partitions were selected for the 15,000-fault experiment, each containing on the average 4000 faults. Significant overlapping of fault partitions was necessary because the minimum faultable unit was selected as a circuit pack (out of convenience) which often encompassed several functions.

About 24 CPU hours of IBM 360/67 computer time were required to complete the above experiments. This included the time required to restart several runs because of procedural errors and system crashes. Through some additional runs, it was possible to show that diagonal simulation saved about 50 percent of the CPU time that would have been required to simulate all tests versus all faults.

Table I contains some simulation results and estimates for the two

experiments. It is important to emphasize the fact that the tests were an initial cut used to support early factory testing of frames. The resulting detection level of 81 percent was about as expected at this stage of design. The most significant result was the relatively large number of race faults encountered. The LOGIC simulator[5] was used for the study to save computer time. This resulted in race faults (those causing indeterminate gate or output states) and oscillation faults (those leading to circuit oscillation) being removed during simulation when encountered. Subsequent studies using the FAULT simulator[5] showed that most race faults were in reality noncritical races that did not cause indeterminate output states. Furthermore, the race faults were actually detected at about the same level as normal faults. In Table I, it was, therefore, reasonable to remove them from the sample in computing the actual detection level.

Oscillation faults are a serious problem that demand careful study. They potentially jeopardize system operation by causing interference with other units on the system buses. Although such faults would probably be detected in the real system, they were not considered as such in Table I.

The undetected faults in the random sample were carefully analyzed and led to some interesting results. Fourteen major classes of undetected faults (with respect to the 15,000 fault sets) were categorized. In most cases, these classes consist of similar faults on repetitive functions, such as successive bits of a register, which require a few tests for detection. In some cases, other faults were revealed that implied the design of a class of new tests. Several one-of-a-kind faults, each requiring unique tests, were also revealed by the analysis. About 2 percent of the faults were undetectable for various reasons. Sixty percent of the faults were associated with operational logic, and the rest with maintenance circuitry.

It is estimated in Table I that new tests designed because of the results of the random sample analysis will reduce the undetected fault set to about 10 percent. It is also estimated that, through the use of LAMP in this iterative process, no more than three iterations will be required to achieve an ultimate detection level of 96 percent of the classical faults.

Resolution of the remaining 4 percent, which are truly undetectable, pose a problem. Some represent true circuit redundancies (in the simplest case, a single gate output feeding a gate twice). Others, more subtle, reside in circuitry used to improve noise or electrical margins.

These might be detected under worst-case conditions by existing tests. A third class deals with system constraints (inputs constrained not to assume certain state combinations).

It is impractical to consider complete removal of these faults through design changes. In any event, LAMP has done its job by categorizing these faults. Maintenance information can at least be provided to help deal with the possibility of their existence throughout the life of the system.

IV. CONCLUSION

When LAMP was first introduced, it received almost immediate acceptance and support from circuit and diagnostic program design groups, although many growing pains were involved with its use. On countless occasions, the user community taxed both LAMP and its host computer resources to their limits. In response to this, LAMP has grown and matured, making significant improvements in capacity, speed, and capability.

The use of LAMP to verify the paper design of 1A Processor subsystems significantly reduced laboratory debugging intervals and provided major cost reductions. Logic design errors were located and corrected prior to the construction of initial hardware. In association with this, the "first iteration" of diagnostic design, the debugging of functional tests using LAMP simulation, was completed prior to the availability of system laboratories. These tests were then used to test the frames in the factory environment.

The availability of interactive LAMP has been a significant aspect of design verification. The option to freeze the state of a logic simulation in order to examine internal nodes has proved so powerful that circuit designers have sometimes preferred this facility to the actual unit as a debugging tool.

The LAMP fault simulator has been essential to the development of complete circuit pack test vectors for the 1A Processor. The extension of fault simulation to large subsystems is just beginning. Initial fault simulation studies using the 1A Processor program/call store are encouraging. Iterative test enhancement using LAMP will insure the detection or classification (as to reason for not being detected) of every stuck-at logical fault. This is of primary importance because of the very stringent maintenance requirements of the 1A Processor.

In the future, the trend toward higher scales of logic integration will increase the use of LAMP for design verification and factory test

development. The use of LAMP as a breadboard will become a practical necessity.

REFERENCES

1. R. E. Staehler, "1A Processor—A High-Speed Processor for Switching Applications," International Switching Symposium Record, June 1972.
2. H. E. Vaughan, "An Introduction to No. 4 ESS," International Switching Symposium Record, June 1972.
3. S. G. Chappell, "LAMP: Automatic Test Generation for Asynchronous Digital Circuits," B.S.T.J., this issue, pp. 1477–1503.
4. H. Y. Chang, G. W. Smith, Jr., and R. B. Walford, "LAMP: System Description," B.S.T.J., this issue, pp. 1431–1449.
5. S. G. Chappel, C. H. Elmendorf, and L. D. Schmidt, "LAMP: Logic-Circuit Simulators," B.S.T.J., this issue, pp. 1451–1476.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

Rain-Induced Cross-Polarization at Centimeter and Millimeter Wavelengths

By T. S. CHU

(Manuscript received March 13, 1974)

Rain-induced cross-polarization is an important factor in design of dual-polarization microwave radio communication systems. We present current estimates of this effect based upon calculated differential characteristics of canted oblate raindrops and their relationship to experiments. Measured differential attenuation and cross-polarization, mainly at 18 GHz, are used to determine two empirical parameters: an effective average of the absolute value of the canting angle and a measure of the imbalance between positive and negative canting angles. We can then provide estimates for median values of cross-polarization discriminations at other frequencies; these are found to agree fairly well with available measured data.

Differential phase shift is the dominant factor in the rain-induced cross-polarization at frequencies below about 10 GHz, and differential attenuation becomes increasingly important at higher frequencies. For a given rain fading, the cross-polarization decreases with increase in frequency and is relatively insensitive to the rain rate, whereas for a given amount of rain the cross-polarization increases with frequency up to about 35 GHz. The cross-polarization discrimination of circularly polarized waves is much poorer than that of linearly polarized waves. When the angle α between the direction of propagation and the axis of symmetry of oblate raindrops is not equal to $\pi/2$, as on earth-space paths in satellite communication systems, the differential attenuation and differential phase shift can be approximated by $\sin^2 \alpha$ times those for $\alpha = \pi/2$, which is the condition for terrestrial paths.

I. INTRODUCTION

Understanding depolarization properties of the transmission medium is of crucial importance in planning frequency reuse by employing orthogonal polarizations in a radio communication system. The rain-induced depolarization, which concurs with heavy rain attenuation,

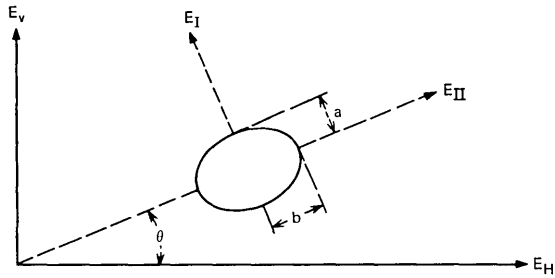


Fig. 1—Canted oblate spheroidal raindrop.

has attracted considerable theoretical and experimental efforts.[1–9] A mathematical model of canted oblate spheroidal raindrops as shown in Fig. 1 has been assumed in most theoretical investigations. Cross-coupling between vertical and horizontal polarizations occurs as a result of the differential attenuation and differential phase shift between two polarizations parallel and perpendicular to a major axis of the oblate raindrops. Recently, Morrison et al.[10,11] have given the calculated differential characteristics for various rain rates throughout the microwave range from 4 to 100 GHz. These results are based upon numerical solutions[12] of the scattering of a plane electromagnetic wave by oblate spheroidal raindrops using a point-matching procedure or perturbation about an equivolumic sphere. The modified perturbation scheme[11] offers a substantial improvement over Oguchi's previous perturbation calculations.[1] Very recently, Oguchi[13,14] also used a point-matching procedure to make similar calculations. The purpose of this paper is to assess our current understanding of the rain-induced microwave depolarization in view of these new calculations and their relationship to the measured data.

Table I — Attenuation and phase shift at 4 GHz with $\alpha = 90°$

Rain Rate in mm/hr	A_I in dB/km	A_{II} in dB/km	Φ_I in deg/km	Φ_{II} in deg/km
0.25	1.825×10^{-4}	2.013×10^{-4}	1.413×10^{-1}	1.487×10^{-1}
1.25	7.806×10^{-4}	8.931×10^{-4}	5.494×10^{-1}	5.894×10^{-1}
2.5	1.506×10^{-3}	1.759×10^{-3}	9.933×10^{-1}	1.077
5.0	2.980×10^{-3}	3.570×10^{-3}	1.806	1.984
12.5	7.581×10^{-3}	9.435×10^{-3}	3.995	4.478
25.0	1.610×10^{-2}	2.089×10^{-2}	7.377	8.417
50.0	3.547×10^{-2}	4.836×10^{-2}	13.83	16.10
100.0	8.038×10^{-2}	1.164×10^{-1}	26.15	31.10
150.0	1.324×10^{-1}	2.021×10^{-1}	38.35	46.23

Table II — Attenuation and phase shift at 11 GHz with $\alpha = 90°$

Rain Rate in mm/hr	A_I in dB/km	A_{II} in dB/km	Φ_I in deg/km	Φ_{II} in deg/km
0.25	2.428×10^{-3}	2.669×10^{-3}	3.985×10^{-1}	4.195×10^{-1}
1.25	1.592×10^{-2}	1.820×10^{-2}	1.579	1.697
2.5	3.787×10^{-2}	4.399×10^{-2}	2.880	3.127
5.0	9.144×10^{-2}	1.076×10^{-1}	5.266	5.783
12.5	2.907×10^{-1}	3.470×10^{-1}	11.69	13.06
25.0	6.898×10^{-1}	8.293×10^{-1}	21.32	24.18
50.0	1.605	1.945	38.94	44.93
100.0	3.586	4.392	70.25	82.58
150.0	5.605	6.919	99.26	118.3

The considerable uncertainty about the canting-angle distribution can be characterized by two parameters as suggested by Thomas.[2] The first parameter, which is an effective average of the absolute value of the canting angle, can be determined by comparing the calculated values with the measured differential attenuation between vertical and horizontal polarizations. The second parameter, which is a measure of the imbalance between positive and negative canting angles with respect to the vertical direction, can be determined by comparison between the calculated and measured cross-polarizations. The reliability of empirical parameters will be much improved by recent developments in theory and experiment. Then the theoretical model provides systematic extrapolation of the measured data.

Section II tabulates the details of the calculated attenuation and phase shift that were abbreviated in Ref. 10. The normalized differential characteristics with respect to rain fading give physical insight and the main features of the rain-induced cross-polarization. Section III describes the calculation of depolarization for both linearly and

Table III — Attenuation and phase shift at 18.1 GHz with $\alpha = 90°$

Rain Rate in mm/hr	A_I in dB/km	A_{II} in dB/km	Φ_I in deg/km	Φ_{II} in deg/km
0.25	9.797×10^{-3}	1.071×10^{-2}	6.674×10^{-1}	7.029×10^{-1}
1.25	6.483×10^{-2}	7.275×10^{-2}	2.608	2.801
2.5	1.458×10^{-1}	1.658×10^{-1}	4.680	5.078
5.0	3.205×10^{-1}	3.702×10^{-1}	8.362	9.182
12.5	8.927×10^{-1}	1.055	17.78	19.90
25.0	1.874	2.273	31.33	35.63
50.0	3.869	4.846	55.24	63.88
100.0	7.696	10.04	97.39	114.1
150.0	11.5	15.39	137.0	161.4

Table IV — Attenuation and phase shift at 30 GHz with $\alpha = 90°$

Rain Rate in mm/hr	A_I in dB/km	A_{II} in dB/km	Φ_I in deg/km	Φ_{II} in deg/km
0.25	3.347×10^{-2}	3.660×10^{-2}	1.102	1.160
1.25	1.960×10^{-1}	2.221×10^{-1}	4.134	4.428
2.5	4.121×10^{-1}	4.757×10^{-1}	7.220	7.789
5.0	8.506×10^{-1}	1.001	12.51	13.57
12.5	2.179	2.634	25.26	27.56
25.0	4.321	5.321	42.54	46.31
50.0	8.444	10.58	71.16	76.72
100.0	15.96	20.25	118.6	125.2
150.0	23.05	29.37	161.3	167.7

circularly polarized waves. Section IV determines empirical parameters of the canting-angle distribution. Estimates are made for the median values of the rain-induced depolarization at centimeter and millimeter wavelengths. An approximation for the case of oblique propagation is also examined.

II. DIFFERENTIAL ATTENUATION AND DIFFERENTIAL PHASE SHIFT

The ratio of minor to major axes of the oblate spheroidal raindrop is assumed to be linearly dependent upon the radius $\bar{a}$ (in centimeters) of the equivolumic spherical drop; specifically, $a/b = 1 - \bar{a}$. This relationship is a simple approximation for the experimental data of the drop shape.[15] Morrison and Cross[12] have used a least-squares-fitting procedure to calculate the complex forward scattering functions $S_I(0)$ and $S_{II}(0)$[16] for the two polarizations I and II parallel and perpendicular to the plane containing the axis of symmetry of the raindrop and the direction of propagation of the incident wave. They have given numerical tables of forward scattering functions for all the raindrop sizes

Table V — Attenuation and phase shift at 30 GHz with $\alpha = 70°$

Rain Rate in mm/hr	A_I in dB/km	A_{II} in dB/km	Φ_I in deg/km	Φ_{II} in deg/km
0.25	3.372×10^{-2}	3.648×10^{-2}	1.108	1.160
1.25	1.984×10^{-1}	2.215×10^{-1}	4.170	4.433
2.5	4.183×10^{-1}	4.746×10^{-1}	7.293	7.803
5.0	8.664×10^{-1}	9.996×10^{-1}	12.66	13.61
12.5	2.230	2.633	25.61	27.67
25.0	4.440	5.326	43.18	46.53
50.0	8.712	10.60	72.26	77.19
100.0	16.53	20.33	120.3	126.1
150.0	23.91	29.50	163.4	168.9

Table VI — Attenuation and phase shift at 30 GHz with $\alpha = 50°$

Rain Rate in mm/hr	A_I in dB/km	A_{II} in dB/km	Φ_I in deg/km	Φ_{II} in deg/km
0.25	3.433×10^{-2}	3.617×10^{-2}	1.127	1.161
1.25	2.045×10^{-1}	2.199×10^{-1}	4.272	4.447
2.5	4.343×10^{-1}	4.719×10^{-1}	7.501	7.840
5.0	9.068×10^{-1}	9.959×10^{-1}	13.07	13.70
12.5	2.362	2.632	26.57	27.93
25.0	4.746	5.339	44.90	47.11
50.0	9.398	10.67	75.17	78.36
100.0	17.97	20.52	124.7	128.3
150.0	26.10	29.84	168.8	172.0

of the Laws and Parsons distribution. The rain-induced attenuation and phase shift are obtained from the forward scattering functions as[16]

$$A_{I,II} = 0.434 \frac{\lambda^2}{\pi} \sum \operatorname{Re} S_{I,II}(0) n(\bar{a}) \qquad \text{(dB/km)} \qquad (1)$$

$$\phi_{I,II} = -36 \frac{\lambda^2}{4\pi^2} \sum \operatorname{Im} S_{I,II}(0) n(\bar{a}) \qquad \text{(deg/km)}, \qquad (2)$$

where λ is the wavelength in centimeters and $n(\bar{a})$ is the number of drops with equivolumic radius $\bar{a}$ per cubic meter. For $\alpha = 90°$, the attenuation and phase shift for various rain rates at 4, 11, 18.1, and 30 GHz have been listed in Tables I to IV. Here α is the angle between the direction of propagation and the axis of symmetry of the raindrop. The

Table VII — Index of refraction of water at 20°C (computed from a recent empirical equation in Ref. 17)

Frequency in GHz	Index of Refraction
4	$8.77 + 0.915i$*
5	$8.685 + 1.195i$
6	$8.574 + 1.399i$
8	$8.319 + 1.761i$
11	$7.884 + 2.184i$
14	$7.437 + 2.477i$
18.1	$6.859 + 2.716i$
20	$6.614 + 2.780i$
24	$6.151 + 2.849i$
30	$5.581 + 2.848i$
40	$4.886 + 2.725i$
60	$4.052 + 2.393i$
100	$3.282 + 1.864i$

* Since the calculations at 4 GHz were made at an earlier date, the 4-GHz refractive index was taken from the older literature (Ref. 18).

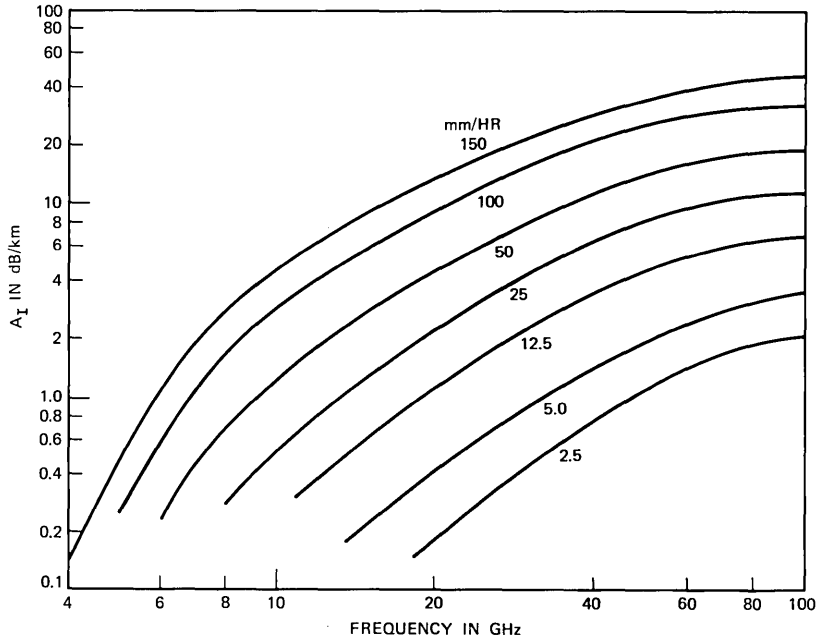


Fig. 2—Calculated attenuation coefficients of polarization I from perturbation about an equivolumic sphere.

cases $\alpha = 50°$ and $70°$ at 30 GHz have been listed in Tables V and VI. The differential attenuation and the differential phase shift were presented graphically in Ref. 10, whereas the above tables are documented here for the reader interested in more details. Most results of this paper belong to the case $\alpha = 90°$, which is pertinent to terrestrial microwave relay systems. However, we discuss later an approximation for other values of α that are of interest in satellite systems.

Since the results of a modified perturbation scheme showed close agreement with those of the least-squares-fitting procedure, the rain-induced differential attenuation and differential phase shift based upon that perturbation scheme have been graphically presented[11] for various rain rates from 4 to 100 GHz. As a supplement to Ref. 11, the refractive indices used for the calculations are listed in Table VII, computed from an equation in Ref. 17. For a given rain rate, the differential attenuation increases with frequency until about 35 GHz, whereas the differential phase shift peaks around 20 GHz and then decreases sharply to negative values for millimeter wavelengths. The

cross-polarization is expected to increase with frequency until about 35 GHz for a given amount of rain.

As reference, Fig. 2 presents A_I obtained from the modified perturbation scheme. Combining Fig. 2 with the differential attenuation data of Ref. 11 results in A_{II}. In comparison with Setzer's data,[18] we note that the attenuation by equivolumic spherical drops lies between A_I and A_{II}, but closer to A_{II}. Since the average of the absolute value of the canting angle is about 25°, as demonstrated later, the attenuation of vertical polarization will be greater than A_I, whereas the attenuation of horizontal polarization will be less than A_{II}.

A communication system is usually designed for a certain margin of fading. In propagation experiments, attenuation and cross-polarization

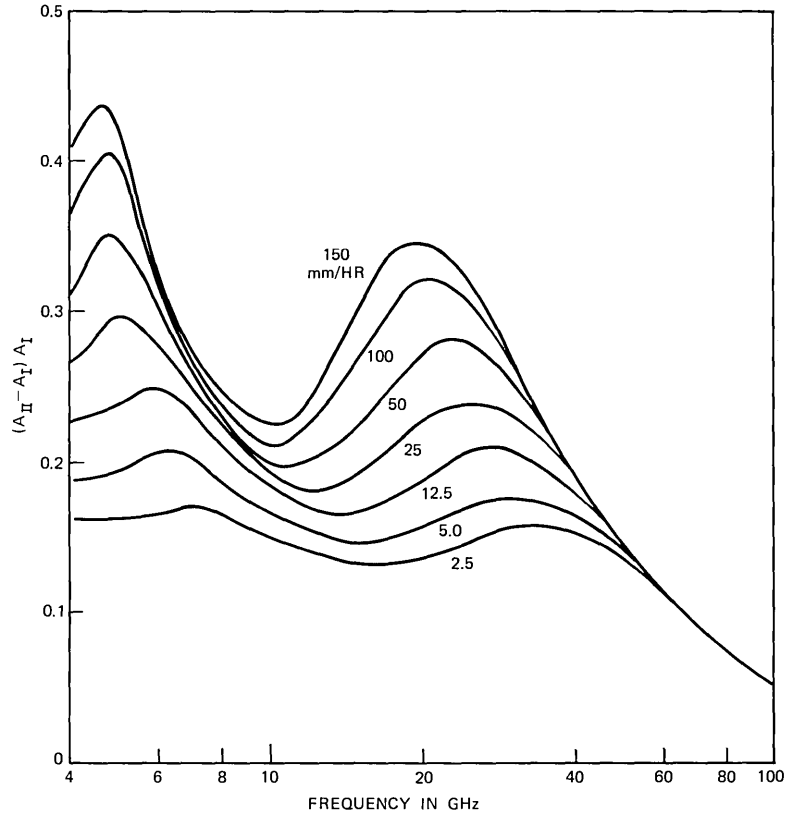


Fig. 3—Normalized differential attenuation with respect to A_I.

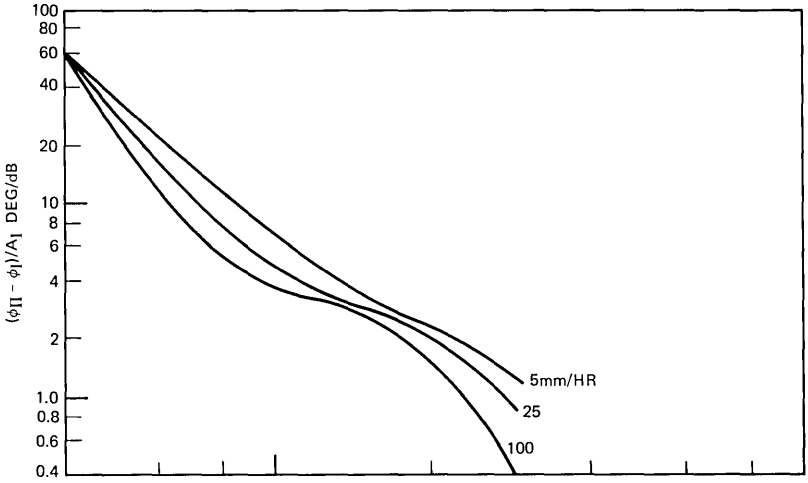


Fig. 4a—Normalized differential phase shift with respect to A_I (4 to 30 GHz).

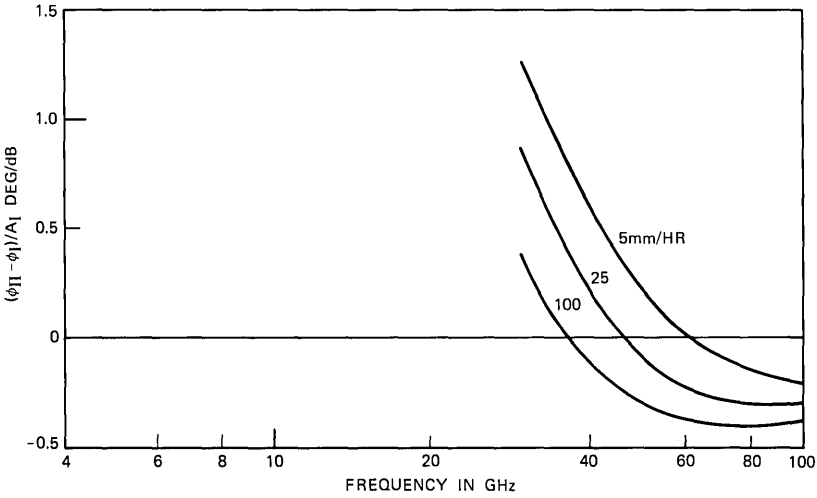


Fig. 4b—Normalized differential phase shift with respect to A_I (30 to 100 GHz).

are often simultaneously measured for correlation with each other. The above two considerations suggest the normalization of both differential attenuation and differential phase shift with respect to the attenuation of polarization I as shown in Figs. 3, 4a, and 4b. These presentations invite a number of observations, as described below.

It is well known that the rain attenuation of longer microwaves such as 4 GHz is slight. Therefore, the relatively high ratio for the nor-

malized differential attenuation in this frequency region will make little contribution to depolarization. However, the very high differential phase shift at 4 GHz in Fig. 4a indicates that significant depolarization is possible even for only a few decibels of attenuation that can occur on a long path, say 40 km between two repeaters. Barnett[8] has reported experimental observations of 4-GHz depolarization during heavy rain. Here, the differential phase shift is indeed the dominant cause of rain-induced depolarization.

For a given fade, the differential phase shift declines sharply with the increase in frequency, whereas the normalized differential attenuation is relatively insensitive to frequency. Therefore, the differential attenuation becomes increasingly important as the frequency increases. The differential attenuation in nepers and the differential phase shift in radians are about the same at about 20 GHz. The sharp descent of the differential phase shift also implies less depolarization at higher microwave frequencies for a given fade.

We note that the decline of the differential phase shift continues into negative values at millimeter wavelengths as shown in Fig. 4b. Although the absolute accuracy of the differential phase shift above 30 GHz is somewhat doubtful in view of its dependence on the cancellation between large numbers, we should expect small differential phase shift per decibel of fading at millimeter wavelengths. Furthermore, the normalized differential attenuation also becomes small because smaller rain drops of less ellipticity are contributing heavily to the attenuation at shorter wavelengths. Delange et al.[19] observed only 2-dB differential attenuation between vertical and horizontal polarizations with a rain fading of 40 dB at 60 GHz.

As the rain rate decreases, the normalized differential attenuation of each frequency decreases. On the other hand, the differential phase shift per decibel of fading generally increases with the decrease of the rain rate. These two opposite trends tend to keep the depolarization relatively insensitive to the rain rate, as is shown in Section 4.2.

III. CALCULATION OF DEPOLARIZATION

Let the axis of symmetry of the oblate raindrop be oriented with respect to the vertical direction at an angle θ, called the canting angle, which is discussed later. We now calculate the depolarization resulting from the anisotropy described in the preceding section. In practice, dual-polarization radio communication systems employ either two orthogonal linear or circular polarizations. The orthogonal linear polarizations are usually aligned in the vertical and horizontal direc-

tions. The polarization transformation of an anisotropic medium can be conveniently obtained by listing the following matrix operations:

$$\begin{pmatrix} H_0 \\ V_0 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} T_2 & 0 \\ 0 & T_1 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} H_i \\ V_i \end{pmatrix}, \quad (3)$$

where T_2 and T_1 are the transmission coefficients over a path of length L for polarizations II and I,

$$T_2 = \exp\,[-\,(\alpha_2 - j\beta_2)L], \quad (4)$$

$$T_1 = \exp\,[-\,(\alpha_1 - j\beta_1)L]. \quad (5)$$

Carrying out the multiplication of the matrices yields the relationship between input (transmitted) and output (received) polarizations:

$$\begin{pmatrix} H_0 \\ V_0 \end{pmatrix} = \begin{pmatrix} a_{hh} & a_{hv} \\ a_{vh} & a_{vv} \end{pmatrix} \begin{pmatrix} H_i \\ V_i \end{pmatrix}, \quad (6)$$

where

$$a_{hh} = T_2 \cos^2\theta + T_1 \sin^2\theta, \quad (7)$$

$$a_{vv} = T_1 \cos^2\theta + T_2 \sin^2\theta, \quad (8)$$

$$a_{hv} = a_{vh} = \left(\frac{T_2 - T_1}{2}\right) \sin 2\theta. \quad (9)$$

The following expressions are given for the convenience of computation:

$$|a_{hh}|^2 = \exp\,[-\,(\alpha_1 + \alpha_2)L](D + C\cos^2 2\theta - E\cos 2\theta), \quad (10)$$

$$|a_{vv}|^2 = \exp\,[-\,(\alpha_1 + \alpha_2)L](D + C\cos^2 2\theta + E\cos 2\theta), \quad (11)$$

$$|a_{hv}|^2 = \exp\,[-\,(\alpha_1 + \alpha_2)L]\; C\sin^2 2\theta, \quad (12)$$

where

$$C = \cos^2\beta L \sinh^2\alpha L + \sin^2\beta L \cosh^2\alpha L, \quad (13)$$

$$D = \cosh^2\alpha L \cos^2\beta L + \sinh^2\alpha L \sin^2\beta L, \quad (14)$$

$$E = 2\cosh\alpha L \sinh\alpha L. \quad (15)$$

The differential attenuation coefficient and the differential phase shift coefficient are $2\alpha = \alpha_2 - \alpha_1$ and $2\beta = \beta_2 - \beta_1$, respectively. When αL (in nepers) and βL (in radians) are small, eq. (13) may be approximated by $C = (\alpha L)^2 + (\beta L)^2$.

Since both a_{hh} and a_{vv} are independent of the sign of θ, an average of the absolute value of the canting angle can be obtained by comparing the calculated $|a_{hh}/a_{vv}|$ for various θ with the measured differential attenuation between vertical and horizontal polarizations. The ex-

pressions for $a_{hv} = a_{vh}$ indicate that the cross-polarized components resulting from positive and negative canting angles tend to cancel each other. Furthermore, Saunders[9] found from photographs of falling raindrops that the canting angle is not far from an even distribution between the positive and the negative senses. Then the cross-polarization coefficient $|a_{hv}|$ or $|a_{vh}|$ should be reduced by a factor ϵ which is a measure of the imbalance between positive and negative canting angles. Having first obtained the average of the absolute value of the canting angle, ϵ can be determined by comparison between the calculated and measured cross-polarization data.

Then, when transmitting horizontal and vertical polarizations the crosstalk discriminations are

$$\text{XTDH} = \epsilon^2 \left| \frac{a_{hv}}{a_{hh}} \right|^2 = \frac{\epsilon^2 C \sin^2 2\theta}{D + C \cos^2 2\theta - E \cos 2\theta} \tag{16a}$$

for receiving horizontal polarization, and

$$\text{XTDV} = \epsilon^2 \left| \frac{a_{vh}}{a_{vv}} \right|^2 = \frac{\epsilon^2 C \sin^2 2\theta}{D + C \cos^2 2\theta + E \cos 2\theta} \tag{16b}$$

for receiving vertical polarization. The crosstalk discriminations are important information for communication systems. In propagation experiments, we often receive both horizontal and vertical polarizations with one transmitted polarization. The cross-polarization discriminations are

$$\text{XPDH} = \epsilon^2 \left| \frac{a_{vh}}{a_{hh}} \right|^2 = \frac{\epsilon^2 C \sin^2 2\theta}{D + C \cos^2 2\theta - E \cos 2\theta} \tag{17a}$$

for transmitting horizontal polarization and

$$\text{XPDV} = \epsilon^2 \left| \frac{a_{hv}}{a_{vv}} \right|^2 = \frac{\epsilon^2 C \sin^2 2\theta}{D + C \cos^2 2\theta + E \cos 2\theta} \tag{17b}$$

for transmitting vertical polarization. The crosstalk discriminations in (16) are numerically the same as the cross-polarization discriminations in (17). Experimental confirmation of this theoretical equivalence based on the simple model has been reported by Watson and Arbabi.[20]

The attenuation is greater in horizontal than in vertical polarization because the effective average of the absolute value of raindrop canting angle is estimated to be about 25°, as shown in the next section. Then XPDH will be poorer than XPDV. Since $a_{hv} = a_{vh}$, the difference between XPDH and XPDV for the same rain is expected to be the

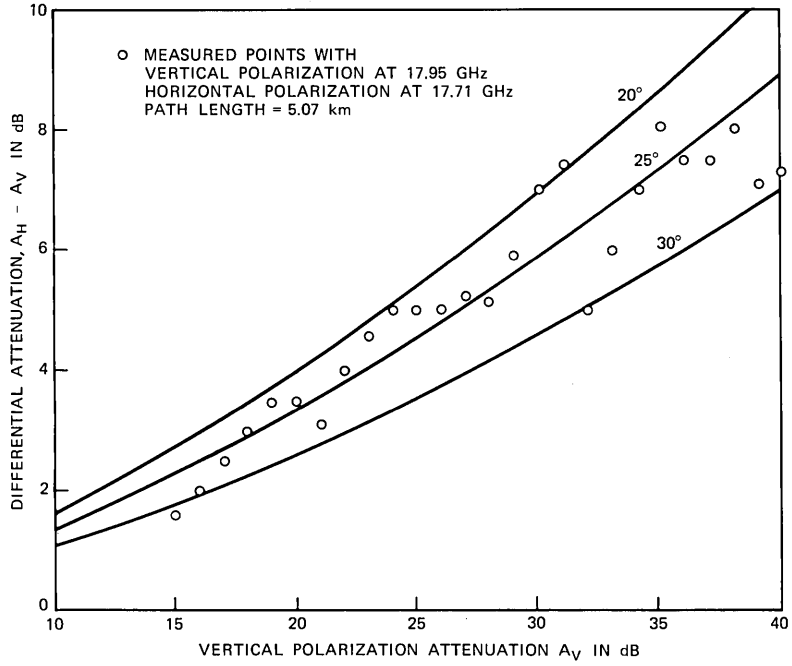


Fig. 5—Comparison between calculated and measured differential attenuation at 18 GHz.

same as the differential attenuation between horizontal and vertical polarizations.

Now let us examine the case of circular polarization. The polarization ratio V/H is first obtained from the following transformation:

$$\begin{pmatrix} \mathrm{H} \\ \mathrm{V} \end{pmatrix} = \begin{pmatrix} a_{hh} & a_{hv} \\ a_{vh} & a_{vv} \end{pmatrix} \begin{pmatrix} 1 \\ j \end{pmatrix}. \tag{18}$$

The ratio of the desired circular polarization to the undesired rain-induced circular polarization is[21]

$$\frac{1 - j\mathrm{V/H}}{1 + j\mathrm{V/H}} = \frac{T_2 + T_1}{T_2 - T_1} e^{-j2\theta}. \tag{19}$$

The cross-polarization discrimination of circular polarization becomes

$$\mathrm{XPDC} = \left| \frac{T_2 - T_1}{T_2 + T_1} \overline{e^{j2\theta}} \right|^2 = (|a_{vh}/a_{hh}|^2_{\theta=45°}) \, |\overline{e^{j2\theta}}|^2, \tag{20}$$

where the mean value of $e^{j2\theta}$ is taken over the canting angle distribu-

tion. If all the oblate raindrops are oriented at a single canting angle, XPDC is equal to $|a_{vh}/a_{hh}|^2$ with $\theta = 45°$. In view of the uncertainty of the canting angle distribution, comparison between the measured and calculated cross-polarizations will be used to estimate this reduction factor $|\overline{e^{j2\theta}}|^2$. We also note that the cross-polarization discriminations of two circular polarizations should be equal to each other.

IV. NUMERICAL RESULTS AND DISCUSSIONS

4.1 XPD of vertical and horizontal polarizations

We first estimate an effective average of the absolute value of the canting angle by comparing the calculated differential attenuation with the measured data. Such comparisons are shown in Figs. 5 and 6 for 18 and 30 GHz. The calculated curves assume that the canting

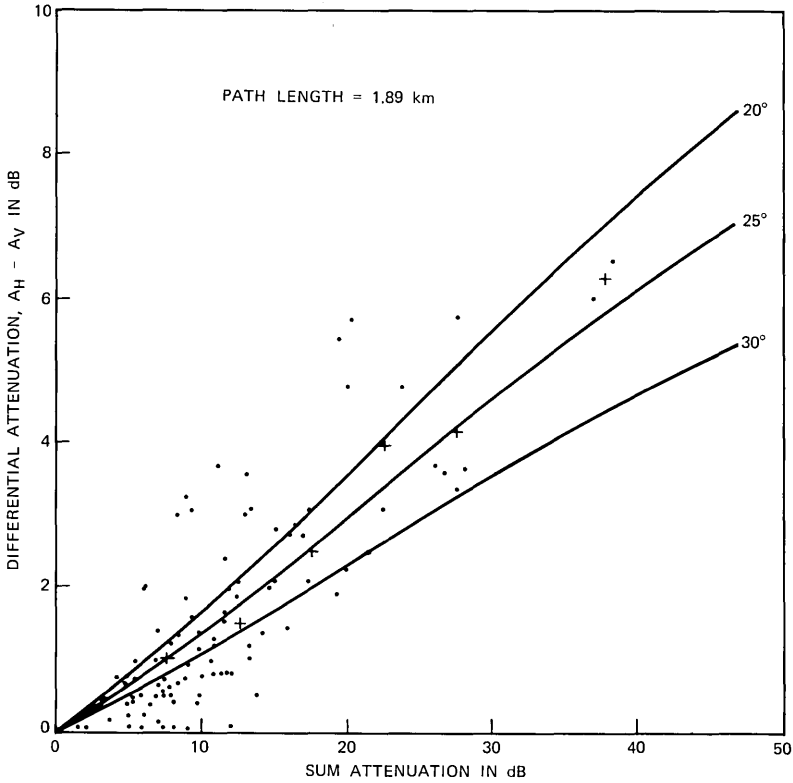


Fig. 6—Comparison between calculated and measured differential attenuation at 30 GHz.

angles of all raindrops are oriented at 20°, 25°, and 30° from the vertical direction. A uniform rain rate has been also imposed over each measured path in the calculations.

The 18-GHz data in Fig. 5 are obtained by W. T. Barnett[7] and S. H. Lin[22] in Georgia. R. A. Semplak's 30-GHz measurement[5] in New Jersey is shown in Fig. 6, where the dots indicate the scatter of the original data and the crosses are median values at 5-dB intervals. The abscissa in Fig. 5 is simply the attenuation of vertically polarized waves, whereas the one in Fig. 6 is the sum attenuation, $\frac{1}{2}(|a_{hh}|^2 + |a_{vv}|^2)$, which represents the received power sum of the horizontal and vertical components of a transmitted wave linearly polarized at 45° from the vertical direction. It is evident from the comparisons in Figs. 5 and 6 that 25° is a good estimate for an effective average of the absolute value of the canting angle. Substituting $|\theta| = 25°$ in (10) and (11) yields the calculated attenuation of vertically and horizontally polarized waves as shown in Fig. 7. We note that an effective average of the

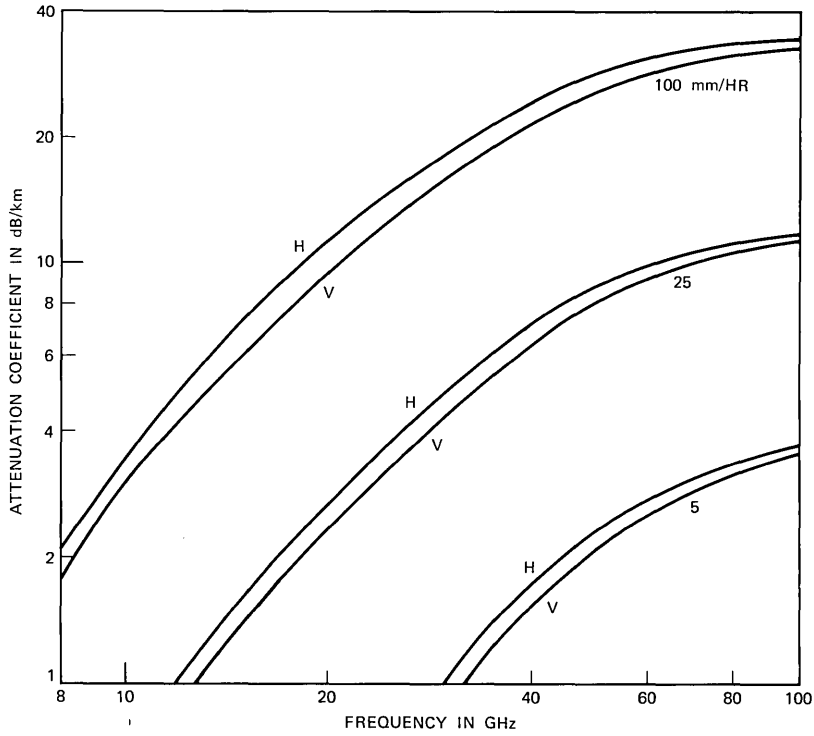


Fig. 7—Attenuation coefficients of vertically and horizontally polarized waves.

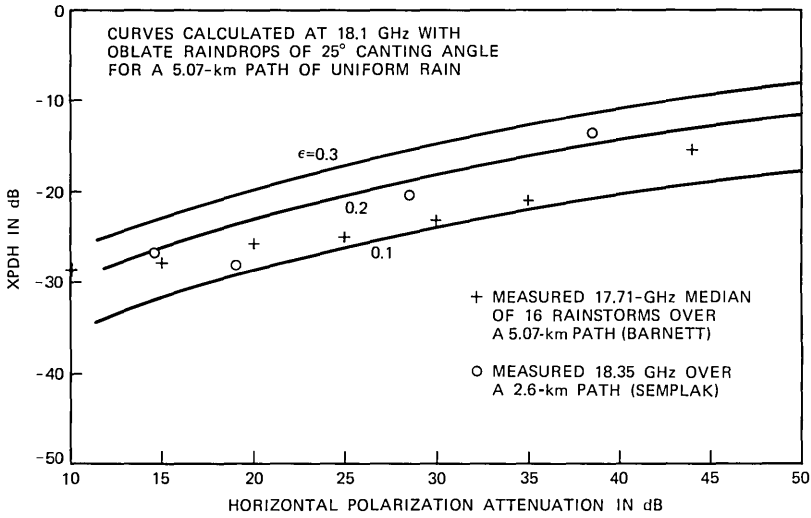


Fig. 8—Comparison between calculated and measured cross-polarization discriminations at 18 GHz.

absolute value of the canting angle should not be mistaken for the mean canting angle,[23] which is much smaller.*

Next, an estimate is made of the imbalance between positive and negative canting angles by comparing the calculated cross-polarization discrimination with the measured data. The calculated curves in Figs. 8 and 9 are computed from (17) with $|\theta| = 25°$ and ϵ assumed as 0.1, 0.2, and 0.3. Uniform rain rates have been assumed over the propagation path in the calculations. The crosses in Fig. 8 are Barnett's measured 17.71-GHz median XPDH of 16 rain storms in Georgia.[7,21] The dots from Semplak's 18.35-GHz measurements[24] over a 2.6-km path are given in Fig. 8 to provide a check. The measured XPDV points in Fig. 9 are deduced from polarization rotation measurements[25] at 30.9 GHz, where the effect of the differential phase shift has been assumed to be negligible. The assumption is approximately valid at this frequency for heavy rain. Figures 8 and 9 show that the measured data are largely confined within the curves with $\epsilon = 0.1$ and 0.2, and hence suggest the geometric mean 0.14 as a good estimate for the median value of ϵ.

Having determined the parameters $|\theta| = 25°$ and $\epsilon = 0.14$, we can use (16) to calculate crosstalk ratios vs attenuation of transmitted

* If the canting angle distribution is simulated by a gaussian model, then the mean will be 2 to 3° and the standard deviation will be 30 to 40°.

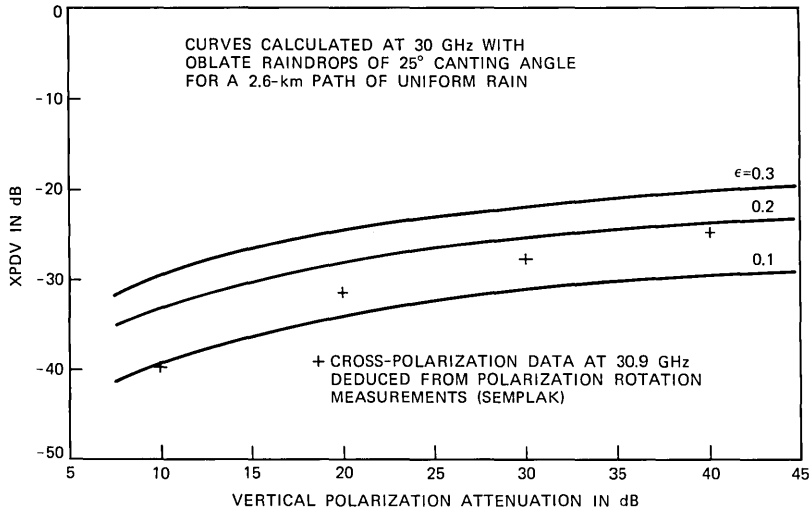


Fig. 9—Comparison between calculated and measured cross-polarization discriminations at 30 GHz.

signal for various frequencies at 100 mm/hr rain rate, as shown in Fig. 10. The solid and dashed curves are expected to be median values at 100 mm/hr rain rate for XPDH and XPDV, respectively. For a given rain fading, the cross-polarization increases with decreasing frequency. However, 4- and 6-GHz communication systems seldom experience rain attenuation of more than a few decibels. Although it takes a slightly heavier rain for the vertically polarized signal to suffer the same fading as the horizontally polarized signal, XPDV is generally less than XPDH for the same fading, except at lower frequencies such as 4 GHz, where the differential phase shift dominates the cross-polarization excitation.

In addition to explaining the measured cross-polarization at 18 and 30 GHz, the predicted cross-polarization discriminations in Fig. 10 also agree fairly well with measured data at 4 GHz of Barnett,[8] at 11 GHz of Watson and Arbabi[26] and Evans and Thompson,[27] at 20 GHz of Yamamoto et al.,[28] and at 60 GHz of Delange et al.[19] The lack of precise agreement stems not only from the imperfection of the theoretical model but also from the measuring error of the experiments. Ground reflection and antenna depolarization often limit the cross-polarization discrimination of measuring systems to around −35 dB in clear weather.

Measurements show considerable variance of cross-polarization discrimination at a given rain attenuation. This factor must be kept in mind when median values of cross-polarization discriminations at given rain fades are used for the design of a dual-polarization radio communication system. The worst cross-polarization discrimination can be 5 to 10 dB above the median values, whereas it is also possible to have very little cross-polarization when almost perfect cancellation occurs among the raindrop canting-angle distribution, i.e., $\epsilon \rightarrow 0$. For more precise predictions of radio channel reliability, the joint statistics of rain fading and cross-polarization discrimination should be considered.

4.2 Effect of rain rate

Measured statistics will always be needed for the design of communication systems. In order to extrapolate statistical results from

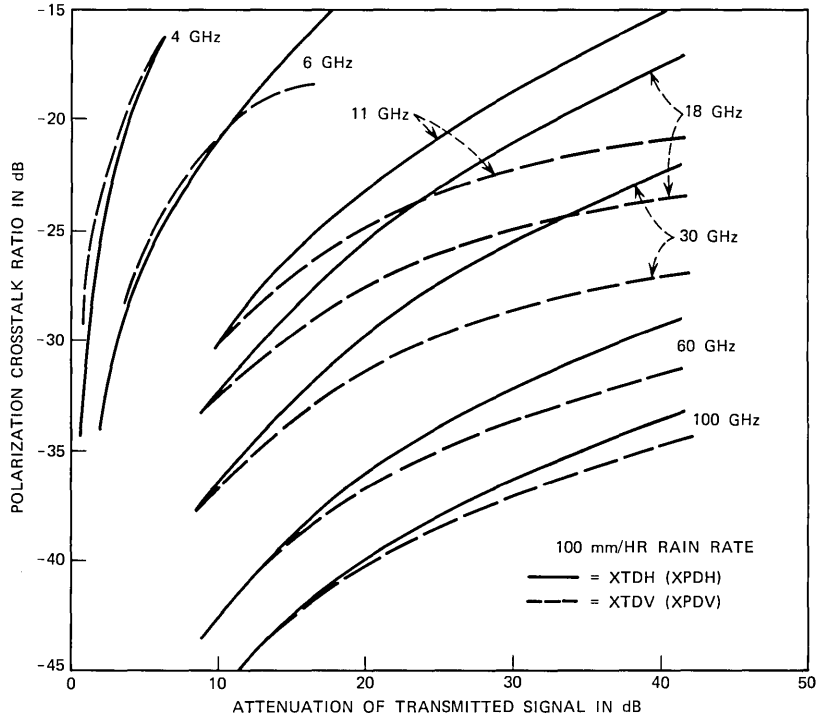


Fig. 10—Calculated rain-induced cross-polarization of horizontally and vertically polarized waves.

one measured path to another path of different length, it is essential to know the effect of rain rate on depolarization for a given fading. Qualitative discussions in Section II already suggest that the depolarization is relatively insensitive to rain rate. This observation is now confirmed in Fig. 11, where XPDH versus frequency are plotted for 20-dB fading at three rain rates of 100, 25, and 5 mm/hr.

4.3 XPD of circular polarization

Equation (20) has been used to calculate the rain-induced cross-polarization of circularly polarized waves as shown in Fig. 12. To obtain agreement between the calculated curves and the measured median values of Semplak[6] at 18 GHz, the canting-angle reduction factor $|\overline{e^{j2\theta}}|^2$ has been empirically determined as 8 dB. On the other hand, Saunders' measured canting-angle distribution yields a reduction factor of 6 dB. In view of the variability of the rain storm, the above discrepancy does not seem unreasonable. Comparison of Figs. 10 and 12 shows that the rain-induced depolarization of circularly

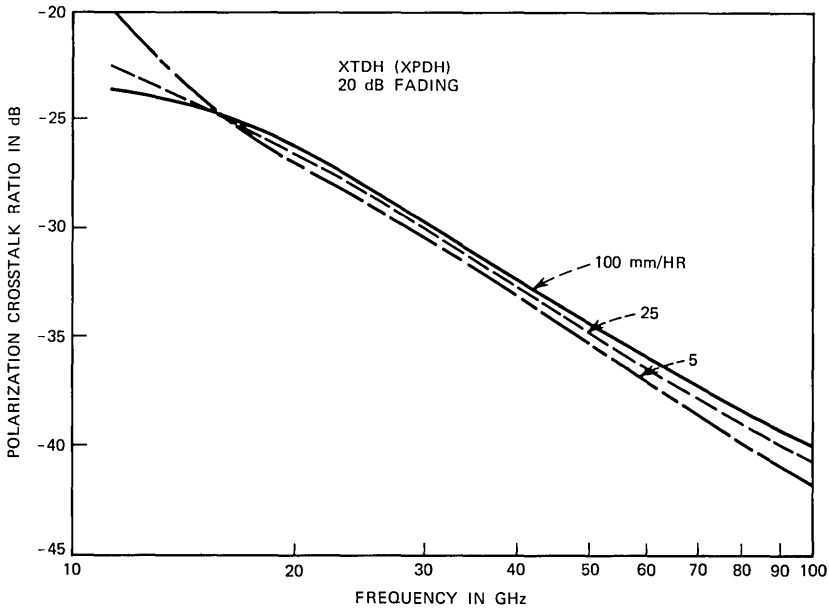


Fig. 11—Cross-polarization at 20-dB fading for various rain rates.

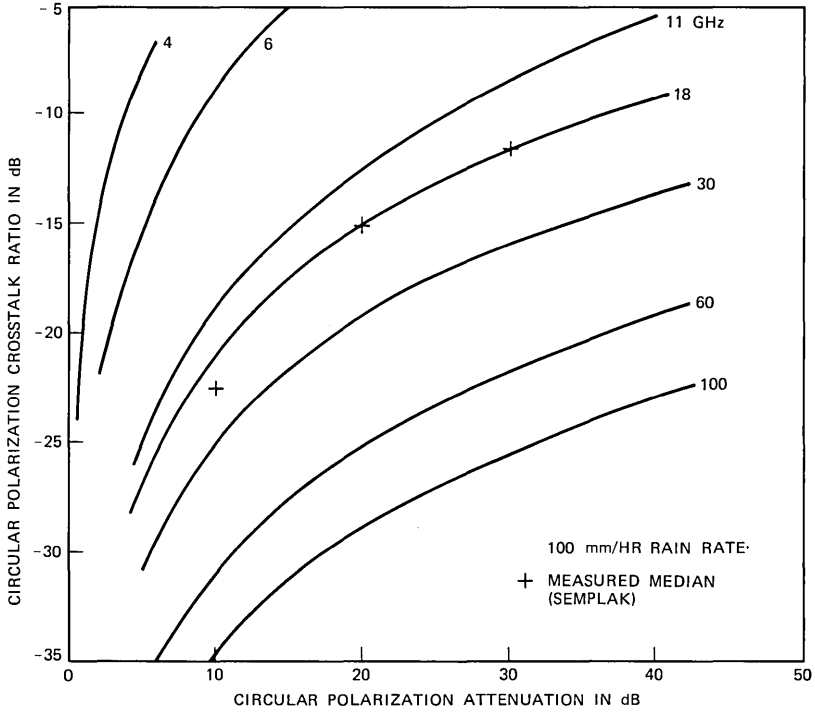


Fig. 12—Calculated rain-induced cross-polarization of circularly polarized waves.

polarized waves is about 10 dB worse than that of a horizontally polarized wave.*

4.4 Oblique propagation

The above numerical results have been confined to the case of $\alpha = \pi/2$, where α is the angle between the direction of propagation and the axis of symmetry of oblate raindrops. This case corresponds to terrestrial microwave relay systems, whereas other values of α are of interest in satellite systems. Limited data for the latter cases are also available from point-matching scattering solutions.[10,14] However, the following approximate relations exist:

$$(A_{II} - A_I)_\alpha = (A_{II} - A_I)_{\pi/2} \sin^2 \alpha, \tag{21}$$

$$(\Phi_{II} - \Phi_I)_\alpha = (\Phi_{II} - \Phi_I)_{\pi/2} \sin^2 \alpha. \tag{22}$$

* The depolarization of a signal linearly polarized at 45° from vertical is expected to be about the same as that of a circularly polarized wave (Ref. 29).

Table VIII — 30-GHz differential attenuation in dB/km

p in mm/hr	$\alpha = 90°$	$\alpha = 70°$	$\alpha = 50°$
0.25	0.00313	0.00276(0.00276)*	0.00184(0.00184)
1.25	0.0261	0.0231(0.0230)	0.0154(0.0153)
2.5	0.0637	0.0563(0.0562)	0.0376(0.0374)
5.0	0.150	0.133(0.133)	0.0890(0.0886)
12.5	0.455	0.403(0.402)	0.270(0.267)
25.0	1.00	0.886(0.883)	0.593(0.587)
50.0	2.14	1.89(1.89)	1.27(1.26)
100.0	4.29	3.80(3.79)	2.54(2.52)
150.0	6.32	5.60(5.58)	3.74(3.71)

* Numbers in parentheses are $\Delta A_{\alpha=90°} \sin^2 \alpha$.

The above approximation was also suggested by Evans and Troughton.[30] A simple derivation in the appendix illustrates the underlying assumption. This low-frequency approximation has been tested by comparison with 30-GHz point-matching results of $\alpha = 70°$ and $\alpha = 50°$.[10] Tables VIII and IX show excellent agreement except for the differential phase shift of heavy rain rates at $\alpha = 50°$, where cancellation between large numbers is involved.

Making use of eqs. (21) and (22), we can predict the cross-polarization discriminations of satellite systems by simply dividing the abscissa scale by $\sin^2 \alpha$ in Fig. 10. Here, the incident linear polarization from the satellite has been assumed to be parallel or orthogonal to the plane containing the direction of propagation and the local gravity direction of the ground station. When this assumption is violated in the fringe area of an area coverage satellite antenna, higher cross-polarization is expected.

Table IX — 30-GHz differential phase shift in deg/km

p in mm/hr	$\alpha = 90°$	$\alpha = 70°$	$\alpha = 50°$
0.25	0.0585	0.0521(0.0517)*	0.0346(0.0343)
1.25	0.294	0.263(0.260)	0.175(0.173)
2.5	0.569	0.510(0.502)	0.339(0.334)
5.0	1.06	0.952(0.936)	0.632(0.622)
12.5	2.30	2.06(2.03)	1.36(1.35)
25.0	3.77	3.36(3.33)	2.21(2.21)
50.0	5.56	4.93(4.91)	3.20(3.26)
100.0	6.59	5.77(5.82)	3.59(3.87)
150.0	6.43	5.53(5.68)	3.23(3.77)

* Numbers in parentheses are $\Delta\phi_{\alpha=90°} \sin^2 \alpha$.

V. ACKNOWLEDGMENT

This paper is built upon the basic work of J. A. Morrison and M.-J. Cross.[12] The author is indebted to J. A. Morrison, D. C. Hogg, and M. J. Gans for helpful discussions, and to D. Vitello for assistance with the computation. He also wishes to thank W. Y. S. Chen and R. P. Slade for reviewing the manuscript.

APPENDIX

Approximation for Oblique Propagation

If the axis of symmetry of each oblate spheroidal raindrop is aligned in the same direction and the raindrop radius is small compared with wavelength, then a uniform rain can be approximately characterized as an anisotropic medium with the relative dielectric constant given in the Cartesian coordinates (XYZ):

$$\bar{\bar{\epsilon}} = \begin{pmatrix} \epsilon_1 & 0 & 0 \\ 0 & \epsilon_2 & 0 \\ 0 & 0 & \epsilon_2 \end{pmatrix} = \begin{pmatrix} n_1^2 & 0 & 0 \\ 0 & n_2^2 & 0 \\ 0 & 0 & n_2^2 \end{pmatrix}. \tag{23}$$

The governing equation for a plane wave $e^{\bar{\gamma}\cdot\bar{r}}$ is simply

$$\bar{\gamma} \times (\bar{\gamma} \times \bar{E}) = k^2 \bar{\bar{\epsilon}} \cdot \bar{E}. \tag{24}$$

In view of the symmetrical property of the medium, the arbitrary direction of propagation, $\bar{\gamma}$, may be confined to the XZ-plane without loss of generality. Let the angle between the direction of propagation and the axis of symmetry (x-axis) be denoted by α. Now we consider two polarizations with subscripts I and II designating the electric fields parallel and perpendicular to the plane containing the axis of symmetry and the direction of propagation. Polarization II has only one electric field component, E_y. Substituting $\bar{E} = E_y\hat{y}$ into (24) immediately yields

$$\gamma_{\text{II}}^2 = -k^2 n_2^2. \tag{25}$$

Polarization I has two electric field components, E_x and E_y. Substituting them into eq. (24) gives two equations:

$$(k^2 n_1^2 + \gamma_{\text{I}z}^2) E_x - \gamma_{\text{I}z}\gamma_{\text{I}x} E_z = 0 \tag{26}$$

$$-\gamma_{\text{I}z}\gamma_{\text{I}x} E_x + (k^2 n_2^2 + \gamma_{\text{I}x}^2) E_z = 0. \tag{27}$$

For the above two equations to be compatible, we need the following condition:

$$\frac{k^2 n_1^2 + \gamma_{\text{I}z}^2}{\gamma_{\text{I}z}\gamma_{\text{I}x}} = \frac{\gamma_{\text{I}z}\gamma_{\text{I}x}}{k^2 n_2^2 + \gamma_{\text{I}x}^2}. \tag{28}$$

Substituting $\gamma_{Ix} = \gamma_I \cos \alpha$ and $\gamma_{Iz} = \gamma_I \sin \alpha$ into the above equation yields

$$\gamma_I^2 = \frac{-k^2 n_1^2 n_2^2}{n_1^2 \cos^2 \alpha + n_2^2 \sin^2 \alpha}. \tag{29}$$

We note that

$$\gamma_I^2 = -k^2 n_1^2 \qquad \text{when } \alpha = \frac{\pi}{2}. \tag{30}$$

Subtracting (29) from (25),

$$\gamma_{II}^2 - \gamma_I^2 = \frac{-k^2 n_2^2 (n_2^2 - n_1^2) \sin^2 \alpha}{n_1^2 \cos^2 \alpha + n_2^2 \sin^2 \alpha}. \tag{31}$$

Since $n_1 \approx n_2 \approx 1$ and $\gamma_I \approx \gamma_{II} \approx jk$,

$$\gamma_{II} - \gamma_I \approx jk(n_2 - n_1) \sin^2 \alpha. \tag{32}$$

REFERENCES

1. T. Oguchi, "Attenuation of Electromagnetic Waves Due to Rain with Distorted Raindrops," J. Radio Res. Labs. (Tokyo), Part I in *7*, No. 33 (September 1960), pp. 467–485; Part II in *11*, No. 53 (January 1964), pp. 19–44.
2. D. T. Thomas, "Cross Polarization Discrimination in Microwave Radio Transmission Due to Rain," Radio Science, *6*, October 1971, pp. 833–839.
3. D. C. Hogg, "Depolarization of Microwaves in Transmission Through Rain," AGARD Conference, Telecommunications Aspects on Frequencies Between 10 and 100 GHz, September 1972, Preprint CPP-107, p. 6–1.
4. P. A. Watson and M. Arbabi, "Rainfall Cross Polarization at Microwave Frequencies," Proc. IEE, *120*, April 1973, pp. 413–418.
5. R. A. Semplak, "Effect of Oblate Raindrops on Attenuation at 30.9 GHz," Radio Science, *5*, March 1970, pp. 559–564.
6. R. A. Semplak, "The Effect of Rain on Circular Polarization at 18 GHz," B.S.T.J., *52*, No. 6 (July–August 1973), pp. 1029–1031.
7. W. T. Barnett, "Some Experimental Results on 18 GHz Propagation," Conference Record of the 1972 National Telecommunications Conference, Houston, Texas; IEEE Publication 72 CHO 601-5-NTC, pp. 10E-1 to 10E-4.
8. W. T. Barnett, "Deterioration of Cross-Polarization Discrimination During Rain and Multipath Fading at 4 GHz," Conference Record of the 1974 International Conference on Communications, Minneapolis, Minn., IEEE Publication 74 CHO 859-9-CSCB, pp. 12D-1 to 12D-4.
9. M. J. Saunders, "Cross Polarization at 18 and 30 GHz Due to Rain," IEEE Transactions on Antennas and Propagation, *AP-19*, March 1971, pp. 273–277.
10. J. A. Morrison, M.-J. Cross, and T. S. Chu, "Rain-Induced Differential Attenuation and Differential Phase Shift at Microwave Frequencies," B.S.T.J., *52*, No. 4 (April 1973), pp. 599–604.
11. J. A. Morrison and T. S. Chu, "Perturbation Calculations of Rain-Induced Differential Attenuation and Differential Phase Shift at Microwave Frequencies," B.S.T.J., *52*, No. 10 (December 1973), pp. 1907–1913.
12. J. A. Morrison and M.-J. Cross, "Scattering of a Plane Electromagnetic Wave by Axisymmetric Raindrops," B.S.T.J., *53*, No. 6 (July–August 1974), pp. 955–1019.
13. T. Oguchi, "Attenuation and Phase Rotation of Radio Waves Due to Rain: Calculations at 19.3 and 34.8 GHz," Radio Sci., *8*, No. 1 (January 1973), pp. 31–38.
14. T. Oguchi and Y. Hosoya, "Differential Attenuation and Differential Phase Shift of Radio Waves Due to Rain: Calculations at Microwave and Millimeter Wave Regions," presented at the I.U.C.R.M. Colloquium, October 1973, Nice, France.

15. H. R. Pruppacher and R. L. Pitter, "A Semi-Empirical Determination of the Shape of Cloud and Rain Drops," Journal of the Atmospheric Sciences, *28*, January 1971, pp. 86–94.
16. H. C. Van de Hulst, *Light Scattering by Small Particles*, New York: John Wiley, 1957.
17. P. S. Ray, "Broadband Complex Refractive Indices of Ice and Water," Appl. Opt., *11*, August 1972, pp. 1836–1844.
18. D. E. Setzer, "Computed Transmission Through Rain of Microwave and Visible Frequencies," B.S.T.J., *49*, No. 8 (October 1970), pp. 1873–1892.
19. O. E. Delange, A. F. Dietrich, and D. C. Hogg, "An Experiment on Propagation of 60-GHz Waves Through Rain," to be published in B.S.T.J., January 1975.
20. P. A. Watson and M. Arbabi, "Cross Polarization Isolation and Discrimination," Elec. Lett., *9*, November 1, 1973, pp. 516–517.
21. V. H. Rumsey et al., "Techniques for Handling Elliptically Polarized Waves with Special Reference to Antennas," Proc. IRE, *39*, May 1951, pp. 533–552.
22. S. H. Lin, private communication.
23. G. C. McCormick and A. Hendry, "Polarization Properties of Transmission through Precipitation over a Communication Link," presented at the I.U.C.R.M. Colloquium, October 1973, Nice, France.
24. R. A. Semplak, "Simultaneous Measurements of Depolarization by Rain Using Linear and Circular Polarizations at 18 GHz," B.S.T.J., *53*, No. 2 (February 1974), pp. 400–404.
25. R. A. Semplak, "Measurements of Rain-Induced Polarization Rotation at 30 GHz," Radio Science, *9*, April 1974, pp. 425–429.
26. P. A. Watson and M. Arbabi, "Rainfall Cross-Polarization, Comparison of Theory and Measurement," presented at the I.U.C.R.M. Colloquium, October 1973, Nice, France.
27. B. G. Evans and P. T. Thompson, "Cross-Polarization due to Precipitation at 11.6 GHz," presented at the I.U.C.R.M. Colloquium, October 1973, Nice, France.
28. H. Yamamoto et al., "Experimental Considerations on 20 GHz High-Speed Digital Radio-Relay System," Conference Record of 1973 IEEE International Conference on Communications, pp. 28–37 to 28–42.
29. C. W. Bostian et al., "Millimeter Wave Rain Depolarization: Some Recent 17.65 GHz Measurements," 1973 G-AP International Symposium Digest, pp. 289–292.
30. B. G. Evans and J. Troughton, "Calculation of Cross-Polarization Due to Precipitation," IEE Conference on Propagation of Radio Waves at Frequencies above 10 GHz, April 1973, pp. 162–171.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

Design Considerations for a Two-Phase, Buried-Channel, Charge-Coupled Device

By J. McKENNA, N. L. SCHRYER, and R. H. WALDEN

(Manuscript received March 29, 1974)

The design of a two-phase, buried-channel (or bulk-channel) charge-coupled device is presented. Directionality is obtained by using a stepped-oxide structure. The basic operation of the device is explained, and the effect that changes in various design parameters have on its operation is examined in some detail. A set of roughly optimal parameters are found that yield an extremely fast and efficient device. We estimate a charge-transfer time of 1.8 ns and a charge capacity of 4.1 × 10^{11} (electrons/cm²). Only existing technology is necessary for its fabrication.

This paper presents some design considerations for a two-phase, buried-channel (or bulk-channel) charge-coupled device (BCCD). The concept of the BCCD has been presented previously,[1,2] and operation of three-phase BCCD's has been demonstrated.[3–6] Two-phase surface charge-coupled devices (CCD's) have advantages over three-phase surface CCD's in many applications, and several designs have been discussed.[7–12] Therefore, it seems important and timely to consider the design of two-phase BCCD's.

We present here a brief review of the basic n-channel BCCD structure. Figure 1 shows the CCD electrode configuration originally proposed for the buried-channel device.[1] Beneath the charge-transfer electrodes are successively a layer of silicon dioxide about 1200 Å thick, a layer of n-type single-crystal silicon, and finally the substrate of lightly doped p-type silicon. By depleting the entire n-region and part of the adjacent p-substrate of mobile carriers with the aid of a reverse-biased diode at the end of the channel, a potential configuration is obtained like the one shown schematically in Fig. 2.[1] Here we plot the negative of the electrostatic potential, i.e., the potential energy of

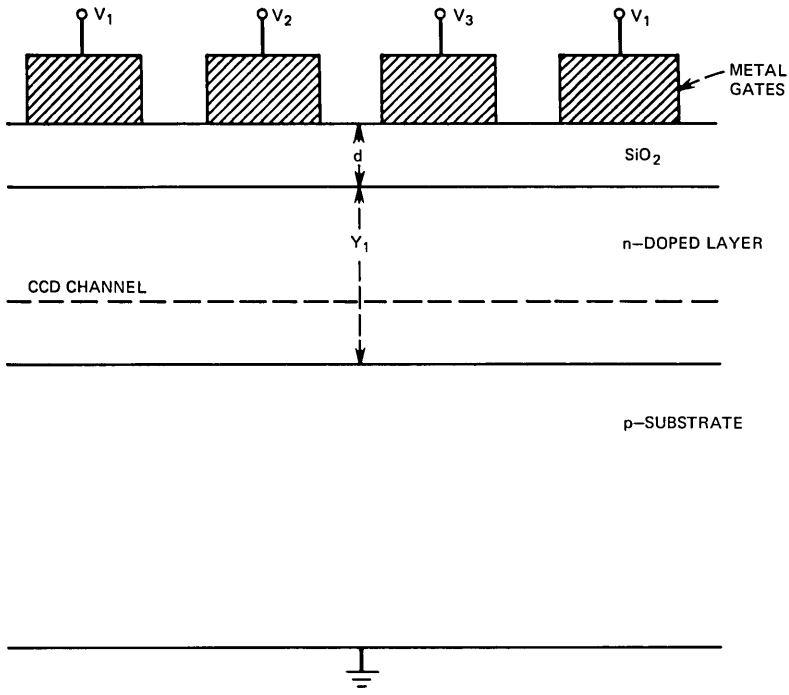


Fig. 1—Schematic diagram of a three-phase buried-channel CCD.

an electron. The distinguishing feature is that the potential energy minimum is located away from the semiconductor-insulator interface; this means that any mobile charge being transferred down the channel travels in bulk silicon, and the transfer should be free of losses associated with interface states. It also means that the free carrier mobility may have a value close to that for bulk. Both these factors were expected to increase transfer efficiency relative to surface CCD's, but at the expense of a reduced charge-carrying capability resulting from the reduced capacitance associated with the increased separation between the metal gates and the channel.

However, there are necessarily gaps between adjacent electrodes in this three-phase device. Since the image charge in the metal plays an important role in controlling the channel potential, the finite gaps give rise to local potential wells, which store charge between the electrodes.[1,13] The amount of charge in each well is not constant; it depends on the values of the clock voltages on neighboring gate electrodes. Thus, charge can be exchanged between the signal and the well. This can

lead to extremely inefficient transfer.[1] It has been shown that this problem can be eliminated by ensuring that the potential between electrodes varies monotonically as a function of distance between plates.[3,13,14] The gap problem can also be alleviated by using a fabrication procedure that reduces the interelectrode gap to zero.[15] If we have a zero-gap two-phase device, there will be operational and fabricational simplifications relative to three-phase devices.

The stepped-oxide structure illustrated in Fig. 3 not only can be operated as a two-phase BCCD, but it also has essentially zero gaps between the electrodes.[10,11] This is the basic configuration studied in this paper. Other studies of this configuration have been made.[16–18] Techniques now exist for fabricating the device. The n-type layer, which has a uniform surface concentration, can be obtained by doing an ion implant in the required channel region before the oxide steps are defined. The definition of metallization and oxide steps can be accomplished by using either the undercut isolation scheme[10] or an overlapping-gate technology.[11]

Our purposes here are to investigate the principles of operation of the device, to study the effect of varying certain of its design parameters, and to attempt to make a reasonably optimal choice of these parameters.

In Fig. 3, we see that the width of the electrode over the thick-oxide step is w_1 and over the thin-oxide step is w_2. The thickness of the thick step is d_1 and of the thin step d_2; the permittivity of the oxide

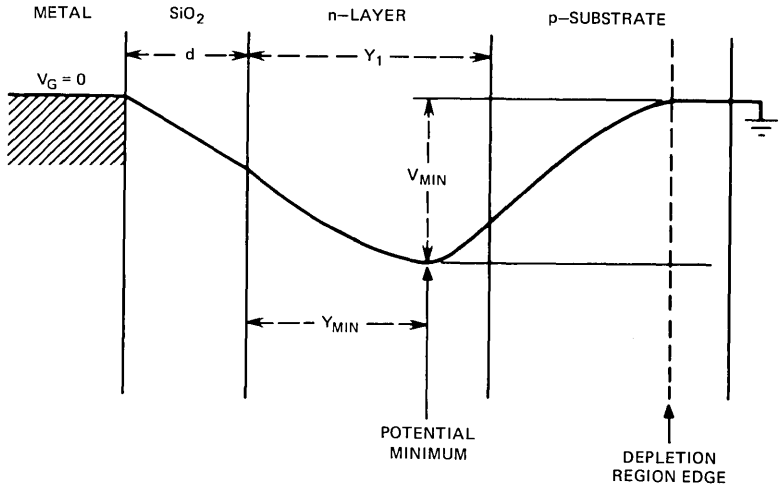


Fig. 2—Schematic potential diagram of a buried-channel CCD.

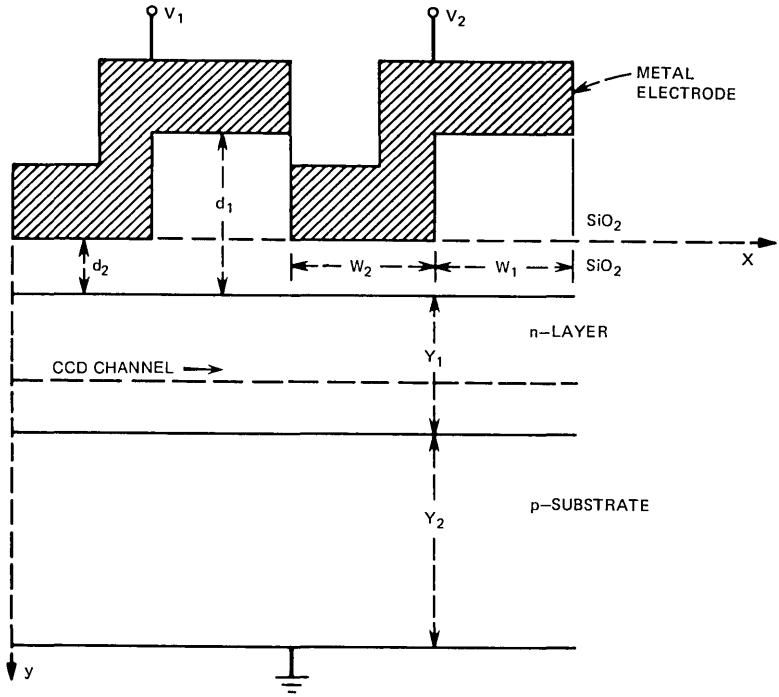


Fig. 3—Schematic diagram of a two-phase, stepped-oxide BCCD.

is ϵ_{0x}. The n-type layer has thickness Y_1 and permittivity ϵ_s, and the donor density N_D is assumed to vary with position as[19]

$$N_D(y) = c_s \exp\left\{-\left(\frac{y - d_2}{Y_1}\right)^2 \ln\frac{c_s}{N_A}\right\} - N_A, \qquad (1)$$

where y is distance measured from the top of the thin-oxide step, c_s is the number density of donor ions at the upper surface of the n-type layer, and N_A is the acceptor number density of the p-type substrate. Finally, the uniformly doped p-type substrate has thickness Y_2 and permittivity ϵ_s. Of these, the design parameters are d_1, Y_1, and the total implanted charge in the n-layer.

The operation of the device can be qualitatively explained on the basis of a simplified one-dimensional model with constant n-layer doping N_D, which is discussed in the appendix. As shown there, the depth of the potential energy well, shown schematically in Fig. 2, increases with increasing oxide thickness. This means that the region under the thin oxide in Fig. 3 will act as a barrier to charge flow while that under the thick oxide will store charge. Interestingly, this is just

the opposite of the case for a surface CCD, and consequently the direction of transfer in a two-phase BCCD is opposite to that of a surface device. The device acts as a BCCD provided the electrode voltage does not exceed a limiting value V_{lim}, where

$$V_{\text{lim}} = \frac{N_D}{2N_A}\left(1 + \frac{N_D}{N_A}\right)\frac{eN_A Y_1^2}{\epsilon_s}. \tag{2}$$

If the plate voltage exceeds V_{lim}, then the potential minimum is located exactly at the insulator-semiconductor interface. Typically, V_{lim} has a value of several hundred volts.

We wish to choose the design parameter values so that the potential well under the thick oxide is deep enough to store as much signal charge as possible, and yet the potential barrier between two wells can be overcome by applying reasonable potentials to the plates to obtain complete transfer of this charge.

To obtain more quantitative information about the device, we turn to a two-dimensional calculation. We use a model described in an earlier paper[13] to calculate the electrostatic potential $\varphi(x, y)$ in the absence of any mobile charge. For the purpose of the two-dimensional

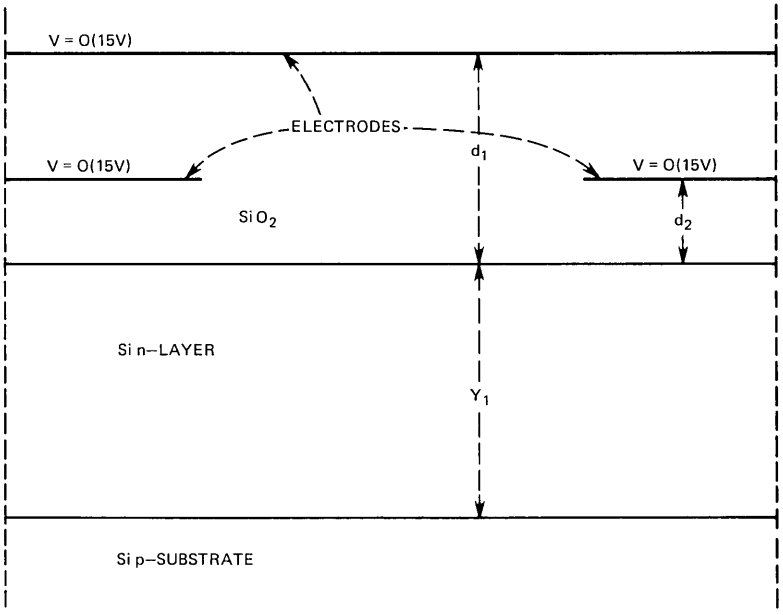


Fig. 4—Single cell of the model used to calculate curves of Figs. 5 and 6 and numbers of Table I.

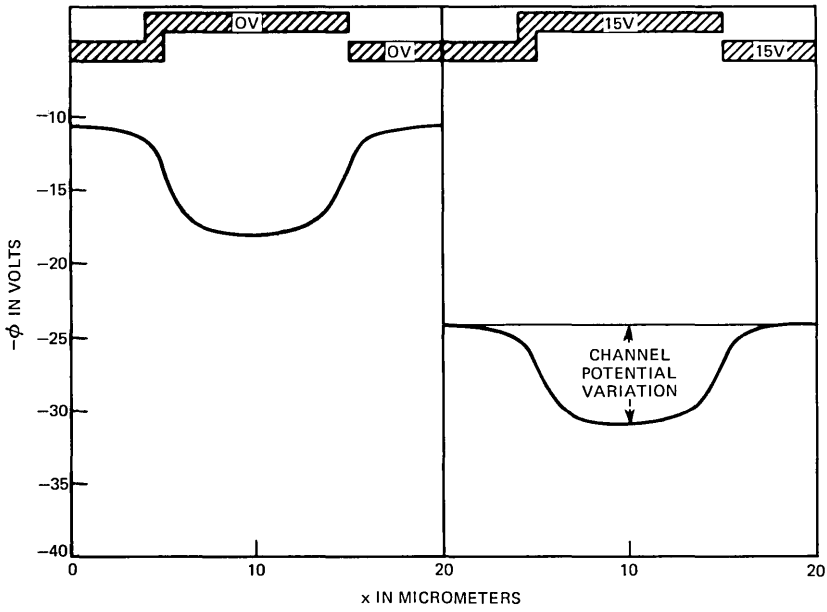


Fig. 5—Channel potential in a two-phase BCCD with all plates either at 0 or 15 volts. Parameters are $d_1 = 0.3$ μm, $d_2 = 0.12$ μm, $Y_1 = 1.2$ μm, $Q_n = 1.5 \times 10^{12}$ cm^{-2}, $N_A = 5 \times 10^{14}$ cm^{-3}.

potential calculation, the plate widths w_1 and w_2 are kept fixed at 10 μm throughout the discussion, as is the thin-oxide thickness at $d_2 = 0.12$ μm. The uniform doping of the n-type substrate is also held fixed at $N_A = 5 \times 10^{14}$ cm^{-3}, and it is assumed that $Y_2 \geqq 50$ μm, which is at least twice the maximum-depletion-region width at any voltage considered. These values are similar to those commonly used in most MOS technologies. We somewhat arbitrarily put an upper limit of 15 volts on the potential difference between electrodes, which we are willing to use to transfer charge from one potential well to a neighboring one.

First, we consider the case in which the electrodes are all at the same potential, either 0 or 15 volts, and we approximately model the device by the configuration in Fig. 4. The potentials and fields were calculated for combinations of the following parameter values: thick-oxide thicknesses (d_1) of 0.3 and 0.6 μm; n-layer thicknesses (Y_1) of 0.4 and 1.2 μm, and total n-layer doping charges (Q_n) of 0.5×10^{12} cm^{-2}, 1.5×10^{12} cm^{-2}, and 4.5×10^{12} cm^{-2}. The spatial distribution of doping in the n-layer is assumed to be given by (1). It can be shown[20]

that c_s, Y_1, and Q_n are related by

$$Q_n = Y_1 \left\{ \frac{\sqrt{\pi}}{2} \left\{ \frac{c_s}{\sqrt{\ln (c_s/N_A)}} \right\} \operatorname{erf} [\sqrt{\ln (c_s/N_A)}] - N_A \right\}, \qquad (3)$$

where erf (z) is the error function.[21] Equation (3) was used to calculate c_s, given the other parameter values.

Figures 5 and 6 give plots of the channel potential $\varphi_c(x)$ as a function of distance parallel to the oxide semiconductor interface for two sets of parameter values shown in the figure captions. If $\varphi(x, y)$ is the electrostatic potential in the device, then

$$\varphi_c(x) = - \max_{d_2 \leq y \leq d_2 + Y_1} \varphi(x, y). \qquad (4)$$

We summarize our calculations in Table I. Of particular interest is the variation of the channel potential with the thick-oxide thickness. The variation is defined as the voltage difference between the minimum and the maximum, as shown in Figs. 5 and 6. For $Y_1 = 1.2$ μm,

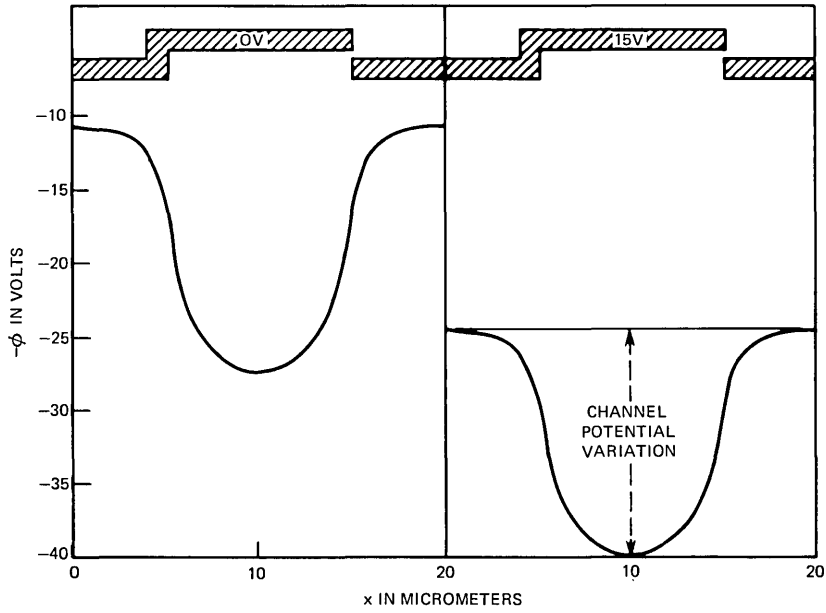


Fig. 6—Channel potential in a two-phase BCCD with all plates either at 0 or 15 volts. The parameters are $d_1 = 0.6$ μm, $d_2 = 0.12$ μm, $Y_1 = 1.2$ μm, $Q_n = 1.5 \times 10^{12}$ cm^{-2}, $N_A = 5 \times 10^{14}$ cm^{-3}.

Table I

d_1 in μm	Y_1 in μm	$10^{-12}Q_n$ in cm^{-2}	Plate Voltage in V	Minimum Channel Potential in V	Maximum Channel Potential in V	Minimum Channel Depth in μm	Plate Voltage Limit V_{lim}
0.6*	0.4	1.5	0	−25.28	−7.56	0.106	343
0.6*	0.4	1.5	15	−38.23	−22.02	0.099	343
0.3	0.4	1.5	0	−15.62	−7.93	0.140	343
0.3	0.4	1.5	15	−28.90	−21.87	0.124	343
0.6*	1.2	1.5	0	−27.27	−10.78	0.381	352
0.6*	1.2	1.5	15	−39.83	−24.51	0.340	352
0.3	1.2	1.5	0	−17.89	−10.61	0.398	352
0.3	1.2	1.5	15	−30.88	−24.23	0.398	352
0.6*	0.4	0.5	0	−6.78	−2.35	0.096	39
0.6*	0.4	0.5	15	−18.58	−16.01	0.050	39
0.3	0.4	0.5	0	−4.42	−2.34	0.125	39
0.3	0.4	0.5	15	−17.12	−15.99	0.058	39
0.6*	1.2	4.5*	0	−91.3	−35.63	0.398	3088
0.6*	1.2	4.5*	15	−104.74	−49.90	0.390	3088
0.3	1.2	4.5*	0	−58.57	−34.45	0.398	3088
0.3	1.2	4.5*	15	−72.25	−48.58	0.398	3088

* Unacceptable values.

$Q_n = 1.5 \times 10^{12}$ cm^{-2}, and a plate voltage of 15 volts, the values of the variation are 6.65, 11.75, and 15.32 volts, corresponding respectively to d_1 values of 0.3, 0.45, and 0.6 μm.

Note that the physical depth of the channel (the distance of the potential minimum below the oxide interface) is less by as much as a factor of 2 when the doping profile is given by (1) than when it is constant, equal to the average doping, which has been pointed out elsewhere.[22]

Although actual operation of the device involves having different voltages on successive electrodes, a first screening of the possible parameter values can be made on the basis of the calculations described in the preceding paragraphs. A criterion for total charge transfer from under the plate at 0 volt to the plate at 15 volts is that the minimum channel potential under the 0-volt plate be greater than the maximum channel potential under the plate at 15 volts, i.e., the barrier potential in the receiving region should be less than that of the potential well in the sending region. Table I shows that all cases of $d_1 = 0.6$ μm or $Q_n = 4.5 \times 10^{12}$ cm^{-2} violate this condition. The cases of $Q_n = 4.5 \times 10^{12}$ cm^{-2} and $Y_1 = 0.4$ μm are not shown because they would also violate this condition. These parameter choices were re-

jected. The parameter value $Q_n = 0.5 \times 10^{12}$ cm^{-2} was also rejected because the minimum channel depth is small.

There remain the parameter values $d_1 = 0.3$ μm and $Q_n = 1.5 \times 10^{12}$ cm^{-2}, and either $Y_1 = 0.4$ μm or $Y = 1.2$ μm. Since there seems to be little difference between these two cases on the basis of the calculations so far, we also consider the case in which d_1 and Q_n are as stated above and $Y_1 = 0.8$ μm. We now examine the device in which one plate is at 0 volt and the adjacent one at 15 volts. The device was modeled by the configuration of Fig. 7, and the calculations are again based on the model of Ref. 13, in which there is no mobile charge. In all the calculations to be discussed now, we took $d_1 = 0.3$ μm, $d_2 = 0.12$ μm, $Y_1 = 0.4$, 0.8, or 1.2 μm, $Q_n = 1.5 \times 10^{12}$ cm^{-2}, $N_A = 5 \times 10^{14}$ cm^{-3}, and $N_D(y)$ given by (1). Figure 8 plots some results for one cell of such a BCCD for the case $Y_1 = 0.8$ μm; φ_c is the channel potential,

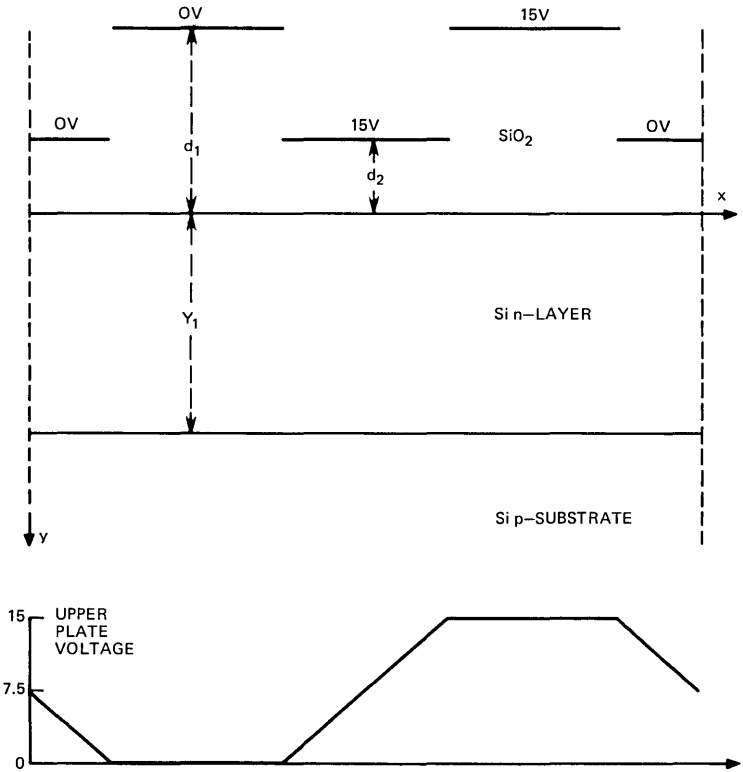


Fig. 7—Two cells of the model used to calculate the curve of Fig. 8. Assumed potential variation along the second level of metallization is shown below.

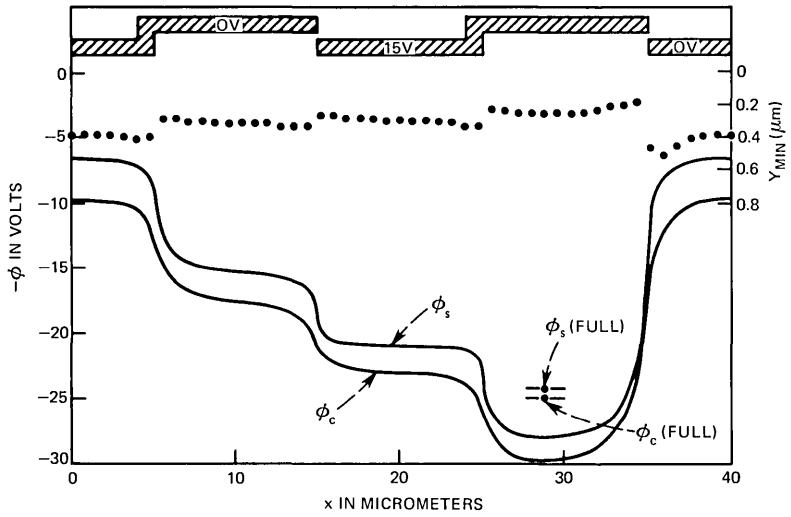


Fig. 8—Channel potential φ_c and potential at the semiconductor oxide interface φ_s in two cells of a two-phase BCCD. There is no inserted charge, plate potentials are as shown, and parameters are $d_1 = 0.3$ μm, $d_2 = 0.12$ μm, $Y_1 = 0.8$ μm, $Q_n = 1.5 \times 10^{12}$ cm^{-2}, $N_A = 5 \times 10^{14}$ cm^{-3}. The dashed curve shows the position of the channel below the oxide-semiconductor interface. φ_c and φ_s at the potential minimum under the receiving plate are also indicated when the device is full of charge.

φ_s is the potential at the oxide-semiconductor interface, and the dotted curve is the position of the potential minimum below the oxide-semiconductor interface.

The amount of charge that can be carried in this BCCD was estimated using a one-dimensional analysis in the well. Charge was added to the one-dimensional well until the minimum potential in the well just equalled the barrier potential; that is, the potential under the thin-oxide step part of the 15-volt plate of Fig. 8. The values obtained ($3–5 \times 10^{11}$ cm^{-2}) indicate that practical quantities of charge can be handled by the BCCD. The method of this calculation[23] is similar to one carried out by Kent.[24]

It is of interest to consider the relative values of surface potential and channel potential for empty and full wells. Figure 8 shows the results of the two-dimensional calculation for both potentials with no free charge; a potential difference of approximately 1.75 volts is maintained along the channel in the receiving well. As the well is filled with charge, this difference is reduced to 0.825 volt, as is indicated in the diagram. These last data were obtained with the aid of the one-dimensional calculation described above.[23] The 0.825-volt

differential ensures that the carrier concentration at the silicon-silicon-dioxide interface will be a negligible fraction of that in the channel which, in turn, indicates that device performance will be essentially unhindered by surface effects.

Table II contains a list of charge-carrying capacities and fringing field values as a function of Y_1. Notice that the capacity falls off relatively slowly with increasing Y_1, while the fringing fields increase at a somewhat more rapid rate. Two columns give field strengths; the left-hand column refers to the minimum horizontal field in the channel under the "sending" well, and the right-hand column refers to that under the "receiving" barrier. Notice that charge transport will be mainly limited by the fields under the latter. The situation would reverse if the maximum clock voltage were increased somewhat beyond the 15 volts used here. It is shown below, however, that a field strength of 710 V/cm is sufficient to ensure extremely rapid charge transfer. The data in Table II show that the ultimate choice of Y_1 is one involving a tradeoff between capacity and fringing field and would depend on the particular device requirements.

Both from the simple model in the appendix and from our two-dimensional calculations, we estimate that, for our choice of parameter values, the electric field at the semiconductor surface never exceeds 1.8×10^5 V/cm and at the p–n junction never exceeds 10^5 V/cm. These fields are below the avalanche breakdown fields for these conditions ($3–4 \times 10^5$ V/cm). It can be shown that the field at the semiconductor surface increases with increasing Q_n, so if Q_n is too large, this field will exceed the avalanche breakdown field. In fact, our calculations show that in the case $Q_n = 4.5 \times 10^{12}$ cm^{-2}, $Y_1 = 1.2$ μm, which we rejected for other reasons, the surface field is about 5.8×10^5 V/cm, which indeed exceeds the avalanche breakdown field for that case.

Finally, we estimated the speed with which the device of Fig. 8 can transfer charge from one well to the neighboring well. A technique of

Table II

Y_1 in μm	Charge Capacity in cm^{-2}	Fringe Field Under Well in V/cm	Fringe Field Under Barrier in V/cm
0.4	4.8×10^{11}	1395.	482.
0.8	4.1×10^{11}	1755.	710.
1.2	3.4×10^{11}	1955.	845.

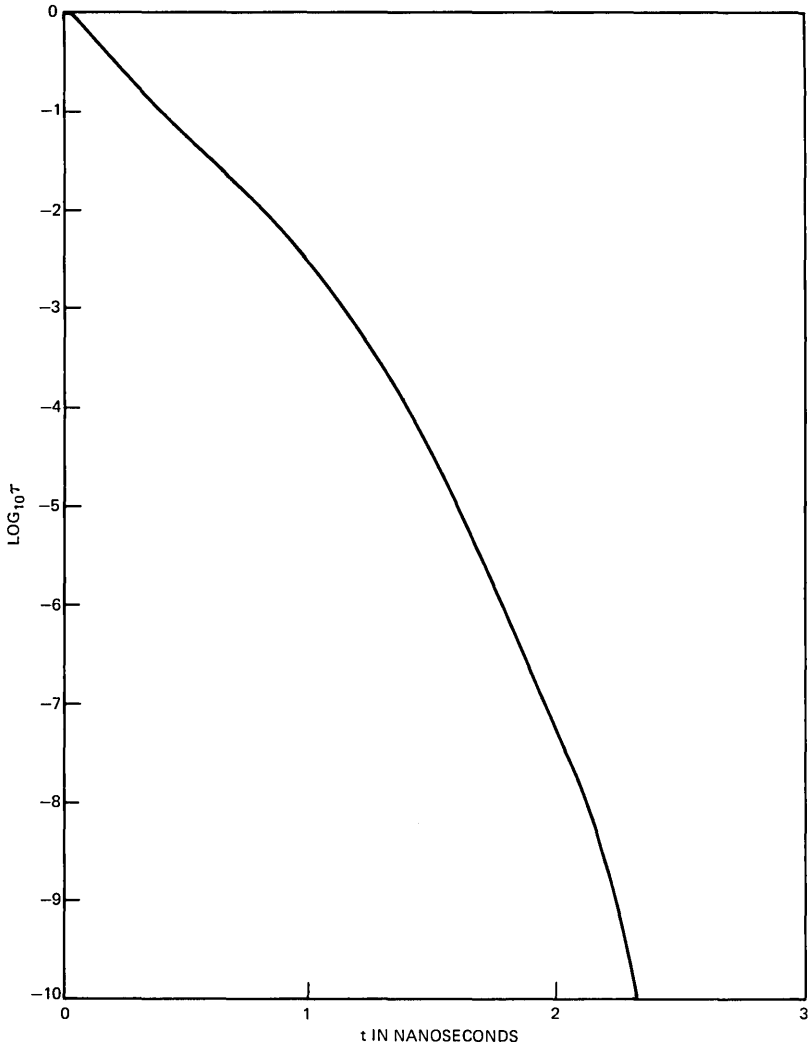


Fig. 9—Plot of $\log_{10} \tau(t)$ as a function of t for the BCCD of Fig. 8.

Strain and Schryer[25] has been adapted to cases such as the present one. Initially, we assumed the plate voltages were the opposite of those shown in Fig. 7, and the charge was all stored in the left-hand well ($5\,\mu \leqq x \leqq 15\,\mu$). Then at $t = 0$, the voltages were instantaneously reversed to the configuration shown in Fig. 7, and the charge flowed from the left-hand well to the right-hand well ($25\,\mu \leqq x \leqq 35\,\mu$). The

calculation[26] based on a one-dimensional analysis of the charge flow in the channel showed that if the well initially contained 10^6 electrons, then essentially *all* the charge transfers in 1.8 ns (see Fig. 9). Let $Q_g(t)$ denote the total charge in the left-hand well at time t, and define the transfer ratio $\tau(t)$ by

$$\tau(t) = Q_g(t)/Q_g(0). \tag{5}$$

Figure 9 plots $\log_{10} \tau(t)$ as a function of t. Figure 10 plots the charge density in the channel (in dimensionless units) as a function of position for $t = 0$, 0.18 ns, and 2.56 ns. By referring to Fig. 8, it is seen that the two deep depressions in the curve for $t = 0.18$ ns are due to the very strong field-aided transfer at those points. Note in Table II that the minimum field under the receiving barrier is 710 V/cm, while the minimum field under the sending well is 1755 V/cm. This accounts for the bunching effect at $t = 0.18$ ns shown in Fig. 10. This bunching effect can be reduced by increasing the most positive electrode potential.

By taking advantage of the capabilities of either self-aligned gate technology or undercut isolation schemes and of ion implantation technologies, the preceding paragraphs have shown the design param-

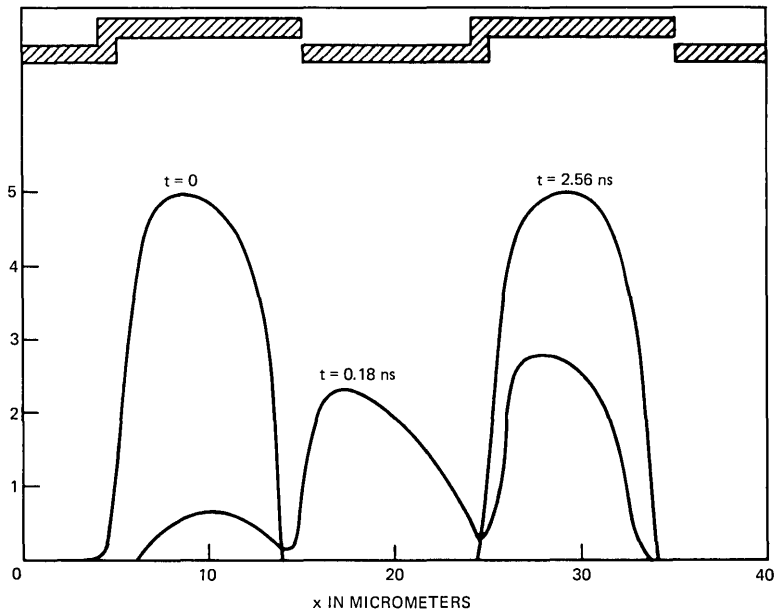


Fig. 10—Charge distribution (in dimensionless units) along the channel for three different times.

eters required in the fabrication of an extremely fast and efficient two-phase, buried-channel, charge-coupled device. This device should have the advantages of convenient operation because of two-phase operation and high transfer efficiency because the buried channel eliminates surface trapping and surface scattering of the transferring carriers and introduces strong fringing fields. Further, by careful design, the charge capacity of this device, while lower, can be competitive with surface devices.

APPENDIX

This appendix briefly derives some results using a simplified, one-dimensional model of a BCCD and the well-known depletion-layer approximation.[20] The oxide layer has thickness d and permittivity ϵ_{0x}. The n-type layer has a thickness Y_1, permittivity ϵ_s, and is *uniformly* doped with donor density N_D. The p-type substrate is assumed to be infinitely thick, with permittivity ϵ_s and acceptor density N_A. The electrostatic potential is denoted by $\varphi(x)$.

We introduce dimensionless quantities as follows. All distances are measured in terms of Debye lengths λ_D,

$$\lambda_D = (\epsilon_s kT/e^2 N_A)^{\frac{1}{2}}, \tag{6}$$

where k is Boltzmann's contant, T is the absolute temperature, and e is the magnitude of the electronic charge. Then we define

$$z = y/\lambda_D, \quad h = d/\lambda_D, \quad z_1 = Y_1/\lambda_D. \tag{7}$$

In addition, we define the dimensionless electrostatic and electrode potentials

$$\psi(z) = e\varphi(y)/kT, \quad V_0 = eV_G/kT, \tag{8}$$

and the dimensionless ratios

$$\eta = \epsilon_{0x}/\epsilon_s, \quad \sigma = N_D/N_A. \tag{9}$$

Then $\psi(y)$ is the solution of the equations

$$\psi''(z) = 0, \qquad 0 \leqq z \leqq h, \tag{10a}$$

$$\psi''(z) = -\sigma, \qquad h \leqq z \leqq h + z_1, \tag{10b}$$

$$\psi''(z) = 1, \qquad h + z_1 \leqq z \leqq h + z_1 + R, \tag{10c}$$

$$\psi(z) \equiv 0, \qquad h + z_1 + R < z, \tag{10d}$$

which satisfies the electrostatic boundary conditions

$$\psi(0) = V_0, \tag{11a}$$

$$\psi(h-) = \psi(h+), \qquad \eta\psi'(h-) = \psi'(h+), \tag{11b}$$

$$\psi(h + z_1-) = \psi(h + z_1+), \qquad \psi'(h + z_1-) = \psi'(h + z_1+), \tag{11c}$$

$$\psi(h + z_1 + R) = \psi'(h + z_1 + r) = 0. \tag{11d}$$

The thickness of the depletion layer, R, is an unknown to be determined from the boundary conditions. The solution can be determined easily.

$$\begin{aligned}\psi(z) &= V_0 + (\sigma z_1 - R)\frac{z}{\eta}, \qquad 0 \leqq z \leqq h,\\ &= -\tfrac{1}{2}(1+\sigma)(z-h-z_1)^2 + \tfrac{1}{2}(z-h-z_1-R)^2,\\ &\qquad\qquad h \leqq z \leqq h + z_1,\\ &= \tfrac{1}{2}(z-h-z_1-R)^2, \qquad h + z_1 \leqq z \leqq h + z_1 + R,\end{aligned} \tag{12}$$

where

$$R = -\left(\frac{h}{\eta} + z_1\right) + \sqrt{(1+\sigma)\left(\frac{h}{\eta} + z_1\right)^2 - \sigma\left(\frac{h}{\eta}\right)^2 + 2V_0}. \tag{13}$$

The electrostatic field is obtained from (12) by differentiation.

To determine the position, y_m, of the electrostatic potential maximum, we first set $\psi'(z_m) = 0$ in $h < z < h + z_1$ and obtain

$$z_m = h + \frac{1}{\sigma}(\sigma z_1 - R) = h + z_1 - \frac{R}{\sigma}. \tag{14}$$

Thus the position of this maximum occurs in $h < z < h + z_1$ if and only if $\sigma z_1 - R > 0$. If $\sigma z_1 - R \leqq 0$, it is easy to show that $\psi(z) \leqq V_0$, and the device would operate as a surface CCD, since the potential maximum in the semiconductor would be at the oxide-semiconductor interface. It is easy to show that $\sigma z_1 - R > 0$ if and only if

$$V_0 < \tfrac{1}{2}\sigma(1+\sigma)z_1^2. \tag{15}$$

When it is written in terms of dimensional quantities, we obtain inequality (2).

Inequality (15) is a rough criterion that places an upper limit on the plate voltages that may be used in a BCCD.

Assuming (15) is satisfied, the value of the electrostatic potential maximum is

$$\psi(z_m) = \frac{1}{2}\left(1 + \frac{1}{\sigma}\right)R^2. \tag{16}$$

It is straightforward to show that

$$\frac{dR}{dh} = \frac{\sigma z_1 - R}{h + \eta(R + z_1)}. \tag{17}$$

Thus, as long as $\sigma z_1 - R > 0$, $dR/dh > 0$. Consequently, from (16) $d\psi(z_m)/dh > 0$ as long as (15) is satisfied. In other words, for electrode voltages within the operating range of a BCCD, the value of the electrostatic potential maximum is greater under the thick-oxide step than it is under the thin-oxide step. This is just the opposite of the case in a surface, stepped-oxide CCD. From (14) it follows that

$$\frac{d}{dh}(z_m - h) = -\frac{1}{\sigma}\frac{dR}{dh}. \tag{18}$$

Thus, within the operating range, the position of the electrostatic maximum is closer to the oxide-semiconductor interface under the thick-oxide step than it is under the thin-oxide step.

REFERENCES

1. R. H. Walden, R. H. Krambeck, R. J. Strain, J. McKenna, N. L. Schryer, and G. E. Smith, "The Buried Channel Charge Coupled Device," B.S.T.J., *51*, No. 7 (September 1972), pp. 1635–1640. Also presented at the Device Research Conference, Ann Arbor, Mich., June 1971.
2. L. J. M. Esser, "Peristaltic Charge-Coupled Device: A New Type of Charge-Transfer Device," Elec. Ltrs., *8*, No. 25 (December 1973), pp. 620–621.
3. C. K. Kim, J. M. Early, and G. F. Amelio, "Buried Channel Charge Coupled Devices," Conference Proceedings NEREM, Part 1 (October 1972), pp. 161–164.
4. C. C. Kim and R. H. Dyck, "Low Light Level Imaging With Buried Channel Charge Coupled Devices," Proc. IEEE, *61*, No. 8 (August 1973), pp. 1146–1147.
5. K. C. Gunsager, C. K. Kim, and J. D. Phillips, "Performance and Operation of Buried Channel Charge Coupled Devices," IEDM Washington, Technical Digest (December 1973), pp. 21–23.
6. L. J. M. Esser, M. G. Collet, and J. G. von Sonten, "The Peristaltic Charge Coupled Device," IEDM Washington, Technical Digest (December 1973), pp. 17–20.
7. R. H. Krambeck, R. H. Walden, and K. A. Pickar, "Implanted-Barrier, Two-Phase Charge-Coupled Devices," Appl. Phys. Ltrs., *19*, No. 12 (December 1971), pp. 520–522.
8. R. H. Krambeck, R. H. Walden, and K. A. Pickar, "A Doped Surface Two-Phase CCD," B.S.T.J., *51*, No. 8 (October 1972), pp. 1849–1866.
9. W. F. Kosnocky and J. E. Carnes, "Charge Coupled Digital Circuits," IEEE J. Solid State Circuits, *SC-6*, No. 2 (October 1971), pp. 314–322.
10. C. N. Berglund, R. J. Powell, E. H. Nicollian, and J. T. Clemens, "Two-Phase Stepped Oxide CCD Shift Register Using Undercut Isolation," Appl. Phys. Ltrs., *20*, No. 11 (June 1972), pp. 413–414; C. N. Berglund, E. H. Nicollian, and J. T. Clemens, "Undercut Isolation—A Technique for Closely Spaced and Self-Aligned Metallization Patterns for Integrated Circuits," J. Electrochem. Soc., *120*, No. 9 (September 1973), pp. 1255–1260.
11. W. E. Engeler, J. J. Tiemann, and R. D. Baertsch, "Surface Charge Transport in Silicon," Appl. Phys. Ltrs., *17*, No. 11 (December 1970), pp. 469–472.

12. D. Kahng and E. H. Nicollian, "Monolith Semiconductor Apparatus Adapted for Sequential Charge Transfer," U. S. Patent 3, 651, 349, applied for February 16, 1970, issued March 21, 1972.
13. J. McKenna and N. L. Schryer, "The Potential in a Charge-Coupled Device With No Mobile Minority Carriers," B.S.T.J., *52*, No. 10 (December 1973), pp. 1765–1793.
14. J. McKenna and N. L. Schryer, "The Potential in a Charge Coupled Device With No Mobile Minority Carriers and Zero Plate Separation," B.S.T.J., *52*, No. 5 (May–June 1973), pp. 669–696.
15. W. J. Bertram, A. M. Mohsen, F. J. Morris, D. A. Sealer, C. H. Sequin, and M. F. Tompsett, "A Three-Level Metallization Three Phase CCD," IEDM Late News Paper (1973).
16. R. H. Walden, "Buried Channel Charge Coupled Apparatus," U. S. Patent applied for, April 27, 1973.
17. A. M. Mohsen, R. Bower, and T. C. McGill, "Overlapping-Gate Buried Channel Charge Coupled Device," Elec. Ltrs., *9*, No. 17 (August 1973), pp. 396–398.
18. D. M. Erb, W. Kotyczka, S. C. Su, C. Wong, and G. Clough, "An Overlapping Electrode Buried Channel CCD," IEDM Washington, Technical Digest (December 1973), pp. 24–25.
19. A. S. Grove, "Physics and Technology of Semiconductor Devices," New York: John Wiley, 1967, pp. 49–50.
20. J. McKenna and N. L. Schryer, "On the Accuracy of the Depletion Layer Approximation for Charge Coupled Devices," B.S.T.J., *51*, No. 7 (September 1972), pp. 1471–1485.
21. M. Abramowitz and J. A. Stegun, "Handbook of Mathematical Functions," Washington, D. C.: National Bureau of Standards, 1964, p. 297.
22. L. J. M. Esser, "The Peristaltic Charge-Coupled Device for High Speed Charge Transfer," ISSCC Philadelphia Digest of Technical Papers (February 1974), pp. 28–29.
23. N. L. Schryer, unpublished work.
24. W. H. Kent, "Charge Distribution in Buried-Channel Charge-Coupled Devices," B.S.T.J., *52*, No. 6 (July–August 1973), pp. 1009–1024.
25. R. J. Strain and N. L. Schryer, "A Nonlinear Diffusion Analysis of Charge-Coupled-Device Transfer," B.S.T.J., *50*, No. 6 (July–August 1971), pp. 1721–1740.
26. N. L. Schryer and J. McKenna, "An Analysis of Field-Aided Charge-Coupled Device Transfer," unpublished work.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

Pulse Spreading in Multimode, Planar, Optical Fibers

By J. A. ARNAUD

(Manuscript received March 8, 1974)

A dielectric slab can keep optical beams confined transversely in its plane if it is tapered, with the slab thickness having a maximum along some straight line. When the square of the local wave number of the slab (k^2) is a quadratic function of the transverse coordinate (y), the rays in the plane of the slab are sinusoids whose optical length is almost independent of the amplitude. For thin slabs ($2d \ll \lambda$) as well as for thick slabs ($2d \gg \lambda$), pulse spreading is large because the ratio of the local phase to group velocity is strongly dependent on the distance (y) from axis. We show that pulse spreading is almost negligible, however, if the thickness of the slab is properly chosen. For example, if the slab thickness on axis is 2.5 micrometers and the refractive index of the slab is 1 percent higher than that of the surrounding medium, pulse spreading is only 0.05 nanosecond per kilometer at a wavelength of 1 micrometer. Pulses in clad fibers having the same width (0.2 millimeter) and carrying the same number of modes (15) spread 50 times faster. Splicing and matching to injection lasers may be easier with planar fibers than with conventional fibers. Low-dispersion planar fibers are therefore attractive when used in conjunction with sources that are multimoded in one dimension. Closed-form expressions are given for square-law and linear-law profiles.

I. INTRODUCTION

This introduction gives first a brief review of the general concepts of pulse transmission in multimode waveguides,[1,2] and subsequently considers the case of planar structures that ensure transverse confinement of the optical beams.

The most important parameters of optical fibers for communication are loss (perhaps a few decibels per kilometer) and pulse spreading (perhaps a few tens of nanoseconds per kilometer). Given these two

parameters, the maximum repeater spacing and the transmission capacity of the fiber are pretty much determined, considering the limitations that presently exist in source power (L.E.D. or injection lasers) and detector sensitivity (avalanche photodiodes). If the loss is the limiting factor, a reduction in bandwidth allows an increase in repeater spacing because of the increased receiver sensitivity, but only by a modest distance. Inversely, baseband equalization allows the transmission capacity to be increased at the expense of optical power, but not by a very large factor. In this paper, we consider only the problem of pulse spreading.

Consider first a single-mode waveguide; for instance, a rectangular waveguide whose width is less than a wavelength. The wave number β may be a rapidly varying function of ω, particularly near cut-off. The transit time of a pulse of radiation is equal to the ratio $Ld\beta/d\omega$ of the path length L and the group velocity $d\omega/d\beta$. Because a pulse of small duration has a broad frequency spectrum, some components arrive ahead of the others if $d\beta/d\omega$ varies with ω; that is, if $d^2\beta/d\omega^2 \neq 0$. The pulse duration, τ, is of the order of $(Ld^2\beta/d\omega^2)^{\frac{1}{2}}$. If the waveguide is filled with a material having dispersion, the phenomenon remains essentially the same. Single-mode pulse spreading is small at optical frequencies when the carrier is almost monochromatic (e.g., injection lasers) because, for a given kind of waveguide, single-mode pulse spreading is inversely proportional to the square root of the frequency; that is, it is 100 times smaller at optical frequencies than at microwave frequencies. This effect can therefore be neglected.

A quite different mechanism for pulse spreading is found in multimode waveguides (with modes of the order $m = 0, 1, 2, \cdots$) excited by multimode sources. In most waveguides, different modes have different group velocities. Thus, a pulse decomposes into a train of pulses, one for each mode, having times of arrival $Ld\beta_m/d\omega$, $m = 0, 1, 2, \cdots$. This effect has similarities with the multipath effects observed in open space. Multimode pulse spreading is observed even when a single mode is excited because, soon after, the power is transferred to other modes and back to the first mode, as a result of the irregularities of the fiber or of the bends (see Ref. 2 and references therein). In this paper, we assume that the fiber is perfectly straight and uniform, and investigate ways of minimizing the dependence of $d\beta_m/d\omega$ on m.

To appreciate the magnitude of the problem, let us consider first a nondispersive homogeneous dielectric slab with refractive index n close

to unity. By comparing the length of rays at the critical angle (θ_o) to the length of axial rays, we find that the pulse spreading is $\Delta T = (L/c)[(\cos\theta_o)^{-1} - 1] \approx (L/c)(n-1)$. This pulse spreading can be written as a function of M, the number of modes that we want to transmit (a characteristic of the source used) and of the slab width Y: $\Delta T = 400\, M^2(\lambda/Y)^2$ ns/km. For example, if we want to transmit 20 modes and $Y = 70\,\lambda$, pulse spreading is 33 ns/km, a value that seriously limits the transmission capacity for long-distance applications. The guide width Y cannot be increased very much because the bending losses would rapidly increase and because it is difficult to fabricate clad fibers with very small differences in refractive index.

The difficulty is solved in principle if the permittivity ϵ of the medium varies as the square of the transverse coordinate y: $\epsilon(y) = 1 - y^2$. In a square-law medium, the optical length of the rays is almost independent of their amplitude. If the permittivity has the form $\epsilon(y) = (\cosh y)^{-2} \approx 1 - y^2 + \frac{2}{3}y^4 + \cdots$, rays have in fact all *exactly* the same optical length.[1,3–7] Because most glasses have negligible dispersion, such media exhibit very small pulse spreading.* Multimode square-law fibers are certainly attractive. However, it may prove difficult to obtain with sufficient accuracy the desired variation of permittivity. Furthermore, the losses (impurities and scattering) are usually higher for heterogeneous material than for homogeneous material such as fused quartz. It is therefore interesting to investigate whether a dimensional change can replace the continuous change in the refractive index considered above.

A proposal to that effect was first made by Kawakami and Nishizawa.[1] They have shown that optical beams can be confined transversely in the plane of a slab if the slab thickness has a maximum along some straight line (z-axis). This can be understood from a geometrical optics point of view. The slab thickness can be considered a constant over a small interval of the transverse coordinate y. Various modes can propagate in this uniform slab. Let $\mathbf{k}$ denote the wave vector of one of them, e.g., the H_1 mode. Because of isotropy, the magnitude k of $\mathbf{k}$ is the same in all directions. Once the local properties of the waveguide characterized by the wave number $k(y)$ have been obtained, the propagation of optical beams can be found, in the semiclassical approximation. We need deal only with $k(y)$. For instance, if $k^2(y)$ is a quadratic function of y, e.g., $k^2(y) = k_o^2 - \Omega^2 y^2$, the rays are sinusoids and they have almost all the same optical length. Diffraction effects in the

* The properties of graded-index fibers that depart somewhat from a quadratic law have also been investigated (Refs. 8 to 10).

yz plane can be taken into account, to some extent, as the Hamiltonian theory of beam modes shows.[11] For the quadratic variation considered above, for example, the modes of propagation are Hermite-gauss,[1,11] regardless of the physical origin of the variation of k with y (that is, whether the variation of k with y results from a genuine variation in refractive index or from a change in slab thickness). Because we are interested in highly multimoded fibers, we consider only the geometrical optics field. In that approximation, a mode is represented by a manifold of rays $y(z + \zeta)$, $0 < \zeta < Z$, where Z denotes the ray period. The main result of this representation is that the axial propagation constant (k_z) of the guide is the value assumed by k at the turning point $y = \xi$ of the trajectory. Therefore, we need only solve a ray equation.

The preceding discussion is applicable to the propagation of waves at one angular frequency, ω_o. To obtain information concerning the propagation of optical pulses, we need to know, not only $k(y)$, but also the variation of the local group velocity u with y. If the ratio $(\omega_o/k) \cdot (\partial k/\partial \omega)$ of the local phase velocity ($v = \omega_o/k$) and group velocity ($u = \partial\omega/\partial k$) happens to be independent of the y coordinate, the time of flight of a pulse along a ray trajectory is proportional to the optical length of that ray. In that case (but only in that case), equal optical lengths imply equal times of flight. The above condition (v/u independent of y) is rather well satisfied for most materials with low dispersion, such as fused quartz, whose refractive index is changed slightly by such processes as ion implantation. (For normal quartz $n = 1.4564$, $dn/d\lambda = -0.27 \times 10^{-5}$ at $\lambda = 0.6563$ μm.) In cases where there is a physical change in the refractive index, it is sufficient to consider the optical lengths of rays with different amplitudes to obtain with good approximation the value of the pulse spreading. For a homogeneous dielectric slab, however, the ratio of the local phase to group velocities is strongly dependent on the slab thickness ($2d$), and therefore on y, when either $2d \gg \lambda$* or when $2d \ll \lambda$. (The latter approximation is made in Ref. 1; pulse spreading for tapered slabs is not discussed in Ref. 1). We will show that small pulse spreading is obtained only for a precise value of the slab thickness on axis. For simplicity, we have considered only quadratic and linear dependences of k^2 on y. The optimum profile may be different, however. In Section II we give the essential formulas for the ray trajectories and times of flight in structures with

* We are indebted to E. A. J. Marcatili for pointing out that pulse spreading in thick, quadratically tapered slabs is almost as large as in clad slabs. This observation, at first surprising, stimulated our interest in the problem.

known local phase and group velocities. In Section III we consider in detail the case of tapered slabs and given design values for low pulse spreading. General results are given in Appendix A, and analytic solutions for square-law and linear-law tapers are given in Appendix B.

II. GENERAL RESULTS

The local value k of the wave number of a slab mode is given in Section III. In the present section we assume that the local wave number $k \equiv \omega_o v$ and the inverse $\partial k/\partial\omega$ of the local group velocity u are known functions of y at the operating angular frequency (ω_o). We give the general form of the ray equations and the time of flight of a pulse in a mode m, in the geometrical optics (J.W.K.B.) approximation. The derivations are given in Appendix A.

In a medium that is isotropic, time-invariant, and independent of the axial coordinate (z), that is, in a uniform fiber, the ray equations $y(z)$ are most convenient in the form

$$k_z^2 = k^2(\omega, y) - k_y^2, \tag{1a}$$

$$dy/dz = -\partial k_z/\partial k_y = k_y/k_z, \tag{1b}$$

$$dk_y/dz = \partial k_z/\partial y = \tfrac{1}{2}(\partial k^2/\partial y)/k_z, \tag{1c}$$

$$dt/dz = \partial k_z/\partial\omega = \tfrac{1}{2}(\partial k^2/\partial\omega)/k_z. \tag{1d}$$

Because of the t and z invariance of the medium, ω and k_z are constant along any given ray (constants of motion). The x coordinate is ignored. The first equation, (1a), says that, because of local isotropy, $k_z^2 + k_y^2$ is equal to k^2. In (1b) to (1d), k_z is considered a function of k_y, ω, and y. Equations 1(b) and (1c) are the ray equations. They give the increments in ray position (dy) and momentum* (dk_y) for an increment dz of z. As indicated before, k_z characterizes a ray trajectory, that is, it is different from one ray to another, but it remains the same along any given ray. We can eliminate k_y from (1b) and (1c) by differentiation. We obtain

$$d^2y/dz^2 = \tfrac{1}{2}(\partial k^2/\partial y)/k_z^2. \tag{2}$$

We first select, as an initial condition, the angle θ_o that the ray makes with the z axis at the origin of the coordinate system ($y = z = 0$). We then evaluate the constant of motion k_z from

$$k_z = k(0)\cos\theta_o. \tag{3}$$

* The ray momentum is the transverse component of the wave vector. Ray momenta and photon momenta ($\hbar\mathbf{k}$) are essentially equivalent concepts.

The ray trajectory $y(z)$ is obtained step by step from (1a) and (1b),

$$y_{i+1} = y_i + [k^2(y_i)/k_z^2 - 1]^{\frac{1}{2}}\Delta z, \tag{4}$$

Δz being the increment in z, and $y_0 = 0$. Note that, because of symmetry, it is sufficient to evaluate $y(z)$ from $y = 0$ to the turning point $y = \xi$, with ξ given by $k(\xi) = k_z$.

To any given value of θ_o (or k_z) we can associate a mode number m. The mode number is the area enclosed in phase space (k_y, y) by a ray trajectory, divided by 2π minus $\frac{1}{2}$ (see Appendix A). Thus, if the integration is stopped at the turning point $y = \xi$ (one-fourth of the ray trajectory), we have

$$m = (2/\pi)\int_0^\xi k_y dy - \tfrac{1}{2}. \tag{5}$$

Strictly speaking, only those values θ_{om} of θ_o should be considered that make m an integer in (5). However, because we are interested in modes of high order, m can be considered a continuous parameter. An approximate value for m is $\pi\theta_o\xi/\lambda$, where λ denotes the wavelength on axis $[k(0) \equiv 2\pi/\lambda]$.

The time of flight T of a pulse is, for a unit length, the inverse $1/v_g$ of the axial group velocity. We show in Appendix A that T is obtained most easily by integrating along a ray ds/u, where $ds = (k/k_y)dy = (k/k_z)dz$ denotes the elementary ray arc length, and $1/u = \partial k/\partial\omega$ the inverse of the local group velocity. Thus,

$$T = Z^{-1}\oint ds/u = (2/Z)\int_0^\xi (\partial k^2/\partial\omega)(k^2 - k_z^2)^{-\frac{1}{2}}dy. \tag{6}$$

Near the turning point $(k = k_z)$, the integrand in (6) is singular. It is therefore preferable from a computational point of view to set $ds = (k/k_z)dz$ and integrate over z rather than over y. We have [also directly from (1d)]

$$T = (2/Zk_z)\int_0^{Z/4} (\partial k^2/\partial\omega)dz. \tag{7}$$

The purpose of this paper is to find ways to minimize the variation ΔT of T for $0 < m < M$, where m is given in (5) and M is the number of modes that we want to transmit. It is interesting to compare this variation to the variation ΔT_c for a clad fiber having the same width $Y \equiv 2\xi$ and the same number of modes M. The latter is, as we have seen in the introduction,

$$\Delta T_c = (1/32)M^2(\lambda/\xi)^2 c. \tag{8}$$

Thus, we want to maximize a quality factor Q defined as

$$Q \equiv \Delta T_c/\Delta T = (1/32)M^2(\lambda/\xi)^2/c\Delta T. \tag{9}$$

Note that, since ΔT and ΔT_c are times of flight for unit lengths, they have the dimensions of inverses of velocities. For given $k(y)$ and $(\partial k/\partial\omega)(y)$, integration of (4), (5), and (7) gives $Q(\theta_o)$ in (9). As θ_o is increased, Q increases and reaches a maximum $Q_{\max}$, which characterizes the pulse spreading properties of an optical waveguide for a given profile. The best profile is the one that maximizes $Q_{\max}$, provided other specifications (number of modes, channel width, $\cdots$) are met.

III. TAPERED DIELECTRIC SLABS

Let us now consider the tapered dielectric slab shown in Fig. 1b. We consider only the H_1 mode of the slab. A similar discussion would be applicable to the E_1 mode (and to higher-order modes if the slab is thick enough to support them). Of course, a profile that is optimum for the H_1 mode need not be optimum for the E_1 mode, for example, unless $\epsilon = n^2$ is very close to unity. Let us first give expressions applicable to slabs with constant thicknesses. We assume that the medium is the same on both sides of the slab. (For dissymmetrical media, the formulas in Ref. 13 would be helpful.)

The dispersion equation $k(\omega)$ for H_1 modes in a slab with relative permittivity ϵ and thickness $2d$ is, as is well known,

$$(kd)^2 - \left(\frac{\omega}{c}d\right)^2 = \phi^2 \tan^2\phi, \tag{10a}$$

$$\phi^2 \equiv \epsilon\left(\frac{\omega}{c}d\right)^2 - (kd)^2. \tag{10b}$$

From (10) we obtain at $\omega/c = 2\pi$ (that is, $\lambda = 1$ μm, using the μm as the unit of length), by straightforward substitutions and differentiations,

$$d = (1/2\pi)(\epsilon - 1)^{-\frac{1}{2}}\phi/\cos\phi, \tag{11a}$$

$$k^2 = (2\pi)^2[1 + (\epsilon - 1)\sin^2\phi], \tag{11b}$$

$$D \equiv \frac{c}{2}\frac{\partial k^2}{\partial\omega} = \frac{2\pi(\epsilon\phi\tan\phi + \epsilon\sin^2\phi + \cos^2\phi)}{(\phi\tan\phi + 1)}. \tag{11c}$$

Thus, the quantities k^2 and $\frac{1}{2}\partial k^2/\partial\omega$ that enter in our previous expressions are explicit functions of the parameter ϕ, related to d by (11a). The parameter ϕ varies from $\pi/2$ for $d = \infty$ to 0 for $d = 0$. The varia-

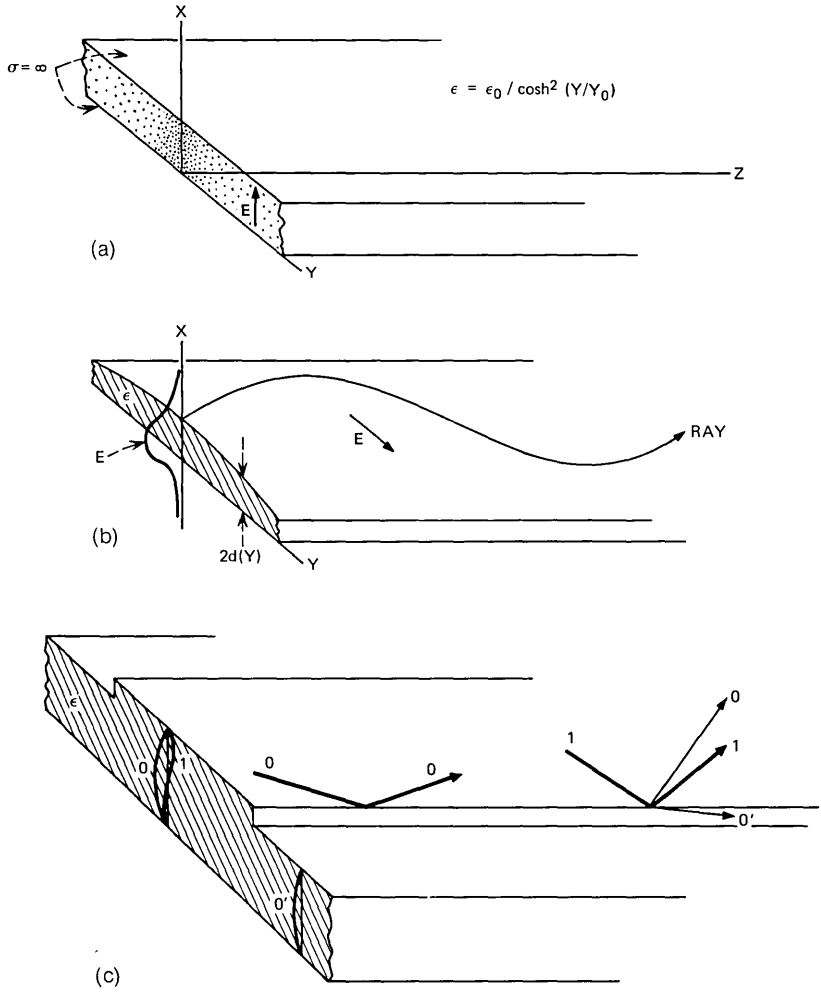


Fig. 1—Planar fibers. (a) Fiber with constant thickness and variation of the permittivity of the form $1/\cosh^2(y)$. (b) Tapered dielectric slab. The field is shown for the H_1 slab mode. (c) Coupling between the various slab modes (H_1, H_2, $\cdots$, E_1, E_2, $\cdots$) cannot be neglected when the thickness $2d(y)$ varies abruptly. This coupling can eliminate the higher-order modes (H_2, E_2, $\cdots$) for suitable dimensions (see Ref. 12).

tion of

$$\frac{v}{u} \equiv \left(\frac{2\pi}{k^2}\right)\frac{c}{2}\frac{\partial k^2}{\partial \omega} = \frac{\epsilon\phi \tan\phi + \epsilon \sin^2\phi + \cos^2\phi}{[1 + (\epsilon - 1)\sin^2\phi](\phi \tan\phi + 1)} \qquad (12)$$

is plotted in Fig. 2 as a function of ϕ for various values of ϵ. For quad-

ratic $k^2(y)$, the optimum value ϕ on axis if close to the maximum of the curves, shown by a dotted line, because, near this maximum, times of flight are proportional to optical lengths (see Section II). Thus, we have for that case a rule for the selection of the slab thickness on axis, $2d_o \equiv 2d(0)$. The optimum value of d_o may be slightly different, however, than the one given by the maximum of the curves in Fig. 2, because we want to minimize the variations of T over a finite range of m.

Instead of specifying the slab profile $d(y)$ or the square of the wave number law $k^2(y)$, we find it convenient, for the ease of computations, to specify $\phi(y)$. If ϕ is quadratic in y, both $k^2(y)$ and $d(y)$ are quadratic in y to first order. Thus, we set

$$\phi = \phi_o - Ky^2, \tag{13}$$

where K denotes a constant, in (11), and substitute in the ray equations, (1b) and (1c), eq. (5) for m and eq. (7) for T.

The variation of the time of flight as a function of the angle θ_o that the ray makes with the z axis at the origin is shown in Fig. 3 for $\phi_o = 1.5$ to 0.2 and $n = 1.45$, $\lambda = 1$ μm. Large pulse spreading is observed

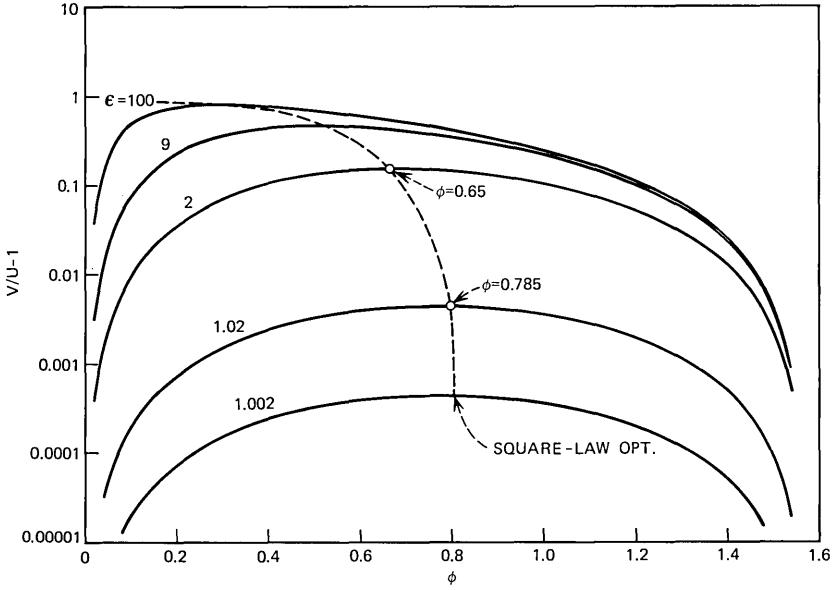


Fig. 2—Variation of the ratio of phase to group velocity in a dielectric slab for different relative permittivities, as a function of the characteristic angle ϕ. The optimum points of operation for low pulse spreading in square-law tapered slabs are shown by a dashed line ($\lambda = 1$ μm).

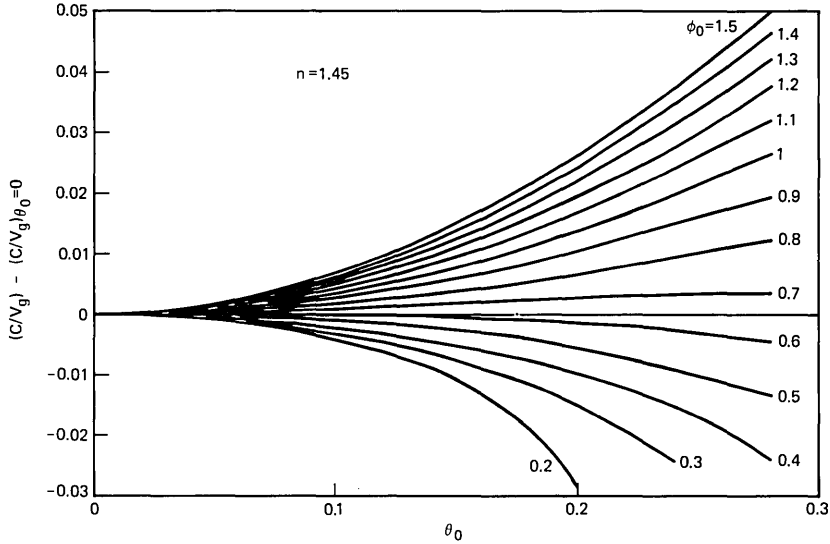


Fig. 3—Ratio of vacuum to axial group velocities (c/v_g) as a function of the ray angle (θ_o) at the origin, for a tapered dielectric slab with $n = 1.45$ and a quadratic variation of the characteristic angle $\phi: \phi = \phi_o - 4 \times 10^{-5}y^2$, for various values of ϕ_o. Group delay is related to c/v_g by $T = (10^4/3)c/v_g$ ns/km. The characteristic angle on axis $\phi_o = 0.65$ is seen to give a small variation of c/v_g over a large range of values of θ_o ($\lambda = 1$ μm).

when the slab is very thick on axis ($\phi_o = 1.5$) or very thin ($\phi_o = 0.2$). Optimum values are between 0.6 and 0.7. Detailed results will be given for the case $n = 1.01$ (refractive index of the slab is 1 percent higher than that of the surrounding medium), which seems of greater practical importance.

For $n = 1.01$, $\lambda = 1$ μm, and $\phi = \phi_o - 10^{-5}y^2$, we see in Fig. 4 that the tapered slab can be 50 times superior to the equivalent clad fiber (factor Q). The profile of this fiber is shown in Fig. 6 (curve *a*), the thickness on axis being equal to 2.5 μm. The results for the case of a linear law $\phi = \phi_o - 5 \times 10^{-3}|y|$ are shown in Fig. 5 and the corresponding profile in Fig. 6 (curve *b*). For both quadratic and linear laws, we note that a trade-off has to be made between the quality factor Q and the mode number M. (Note that the results are meaningful only when M is large compared with unity.)

In conclusion, tapered dielectric slabs can exhibit very low pulse spreading if properly dimensioned. If the slab material has a refractive index 1 percent higher than that of the surrounding medium, the thickness should be of the order of 2.5 ± 0.2 μm at a wavelength of 1 μm.

The waveguide width would be in that case of the order of 0.2 mm. Pulse spreading does not exceed 0.05 ns/km for 15 modes. These optical waveguides are attractive because they can be stacked for multichannel operation (a possible arrangement is shown in Fig. 7) and splicing would perhaps be easier than with conventional fibers (a good angular alignment, however, is required for planar fibers). Further technological researches are needed to settle this point.

IV. ACKNOWLEDGMENT

The author expresses his thanks to E. A. J. Marcatili for stimulating discussions.

APPENDIX A

Times of Flight in the J.W.K.B. Approximation

The purpose of this appendix is to derive the ray equations and the time-of-flight equations from general principles. We start from the Hamilton equations in space-time both for conceptual clarity and to

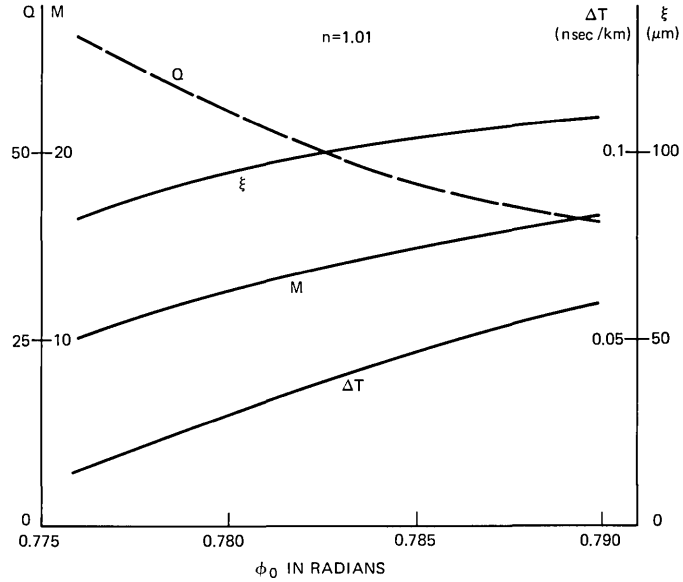


Fig. 4—Variation with the characteristic angle on axis ϕ_o (or slab thickness on axis $2d_o$) of the quality factor Q (defined as the ratio of pulse spreading for an equivalent clad fiber ΔT_c to the actual pulse spreading ΔT) for $\epsilon = 1.02$ ($n = 1.01$) and $\phi = \phi_o - 10^{-5}y^2$. ξ denotes the maximum ray excursion, M the total number of modes, and ΔT the pulse spreading. The ray period is 14 mm and θ_o is equal to 2.6° for $\phi_o = 0.785$ ($\lambda = 1$ μm).

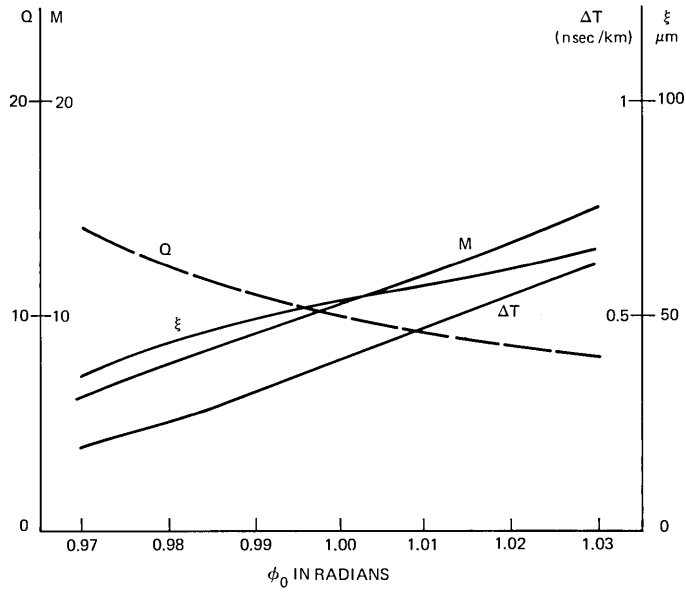


Fig. 5—Variation with the characteristic angle on axis ϕ_o (or slab thickness on axis $2d_o$) of the quality factor Q for $\epsilon = 1.02$ ($n = 1.01$) and $\phi = \phi_o - 5 \times 10^{-3}|y|$. The variations of ξ, M, and ΔT are also given. The ray period is 5.6 mm and θ_o is equal to 4° for $\phi_o = 1$.

facilitate generalizations to anisotropic or time varying media (which are not discussed in detail in the main text, but are of potential interest).

A general medium is described by a function of ω, $\mathbf{k}$, t, $\mathbf{x}$

$$H(\omega, \mathbf{k}, t, \mathbf{x}) = 0. \tag{14a}$$

The space-time trajectories (world lines) of particles or wave packets, $[t(\sigma), \mathbf{x}(\sigma)]$ or $\mathbf{x}(t)$, are obtained by integrating the Hamilton equations

$$\begin{aligned} dt/d\sigma &= -\partial H/\partial\omega, \\ d\mathbf{x}/d\sigma &= \partial H/\partial\mathbf{k}, \\ d\omega/d\sigma &= \partial H/\partial t, \\ d\mathbf{k}/d\sigma &= -\partial H/\partial\mathbf{x}, \end{aligned} \tag{14b}$$

where σ denotes an arbitrary parameter.* These equations are in a suit-

* If we define $\mathbf{X} \equiv \{\mathbf{x}, ict\}$, $K \equiv \{\mathbf{k}, i\omega/c\}$, the Hamilton equations (14b) are: $d\mathbf{X}/d\sigma = \partial H/\partial\mathbf{K}$ and $d\mathbf{K}/d\sigma = -\partial H/\partial\mathbf{X}$. The latter follows from the first (see Ref. 11) because $H = 0$ and $\mathbf{K} = \nabla S$. The dynamical significance of the Hamilton equations follows from the expression of the canonical stress-energy tensor (Ref. 14):

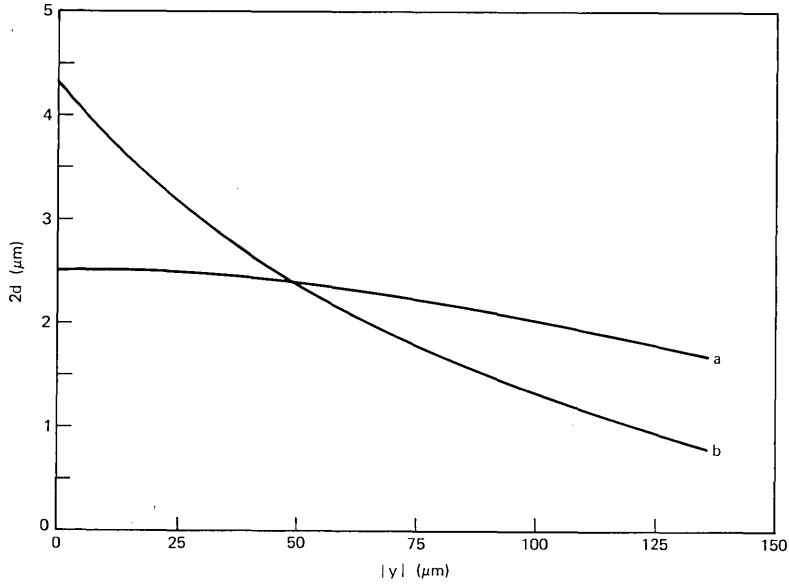


Fig. 6—Slab profiles for $n = 1.01$. (a) Quadratic case $\phi = 0.785 - 10^{-5}y^2$. (b) Linear case $\phi = 1 - 5 \times 10^{-3}|y|$.

able form for numerical integration. The initial conditions must, of course, be consistent with (14a). Then (14a) remains satisfied at all σ because, from (14b), $dH/d\sigma = 0$.

For time-invariant media, the form

$$\omega = \omega(\mathbf{k}, \mathbf{x}) \tag{15}$$

is more useful. The motion $\mathbf{x}(t)$ of a wave packet is a solution of the Hamilton equations

$$\begin{aligned} d\mathbf{x}/dt &= \partial\omega/\partial\mathbf{k}, \\ d\mathbf{k}/dt &= -\partial\omega/\partial\mathbf{x}. \end{aligned} \tag{16}$$

If we are interested only in ray trajectories at some fixed ω, we can rewrite (15)

$$h(\mathbf{k}, \mathbf{x}) = 0, \tag{17a}$$

$\mathbf{T} = \mathbf{K}\partial\mathfrak{L}/\partial\mathbf{K}$, where $\mathfrak{L}$ denotes the averaged Lagrangian density. $\partial\mathfrak{L}/\partial\mathbf{K}$ is the (conserved) wave action, and $\mathbf{T}$ is conserved in time-invariant homogeneous media. The equality of group and energy velocities readily follows from this expression for $\mathbf{T}$. Note that these results are applicable to any linear wave (e.g., matter waves, acoustical waves, or optical waves).

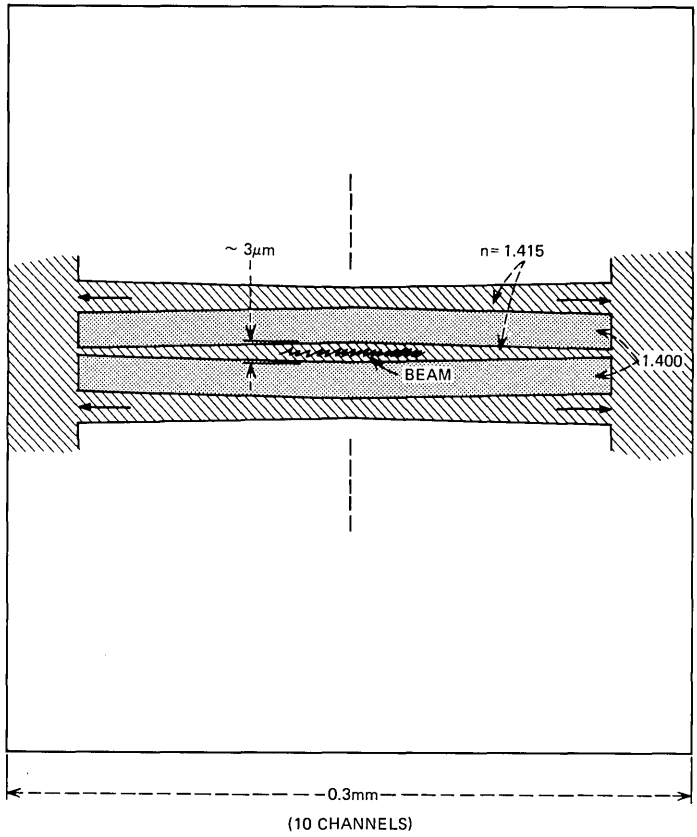


Fig. 7—Stacked tapered dielectric slabs. Adjacent slabs are separated by slabs with inverted slope and high index to minimize crosstalk caused by scattering.

and obtain the rays from

$$\begin{aligned} d\mathbf{x}/d\sigma &= \partial h/\partial \mathbf{k}, \\ d\mathbf{k}/d\sigma &= -\partial h/\partial \mathbf{x}, \end{aligned} \tag{17b}$$

where σ is again an arbitrary parameter. Equation (17) is the reduction of (14) to three dimensions. Note that the Fermat principle (in three dimensions) is applicable to rays $\mathbf{x}(\sigma)$ at a constant frequency ω. It is unrelated to the time of flight of wave packets, except for nondispersive media. It is important for our study that the time of flight of a pulse be carefully distinguished from the transit time of the crest of a time-harmonic wave (optical length). The latter is the integral of the ray index along the ray path, evaluated at a fixed frequency ω.

We need the Hamilton equations in one more form, in which the z axis is singled out. For media that are invariant in the z direction, it is convenient to solve (14a) for k_z. Ignoring the x coordinate, we have

$$H \equiv k_z - k_z(\omega, k_y, y) = 0, \tag{18a}$$

and the ray equations are, from (14b),

$$\begin{aligned} dy/dz &= -\partial k_z/\partial k_y, \\ dt/dz &= \partial k_z/\partial \omega, \\ dk_y/dz &= \partial k_z/\partial y, \end{aligned} \tag{18b}$$

where ω and k_z are constants of motion. If the surface y, z is isotropic, $\mathbf{k}$ enters only through its magnitude k. Thus,

$$k_z^2 = k^2(\omega, y) - k_y^2, \tag{19}$$

and (18) becomes

$$dy/dz = k_y/k_z, \tag{20a}$$

$$dk_y/dz = \tfrac{1}{2}(\partial k^2/\partial y)/k_z, \tag{20b}$$

$$dt/dz = \tfrac{1}{2}(\partial k^2/\partial \omega)/k_z. \tag{20c}$$

These are the expressions used in the main text. Equations (20a) and (20b) give the rate of change of the ray position and momentum as a function of z. Equation (20c) gives the time of flight of a pulse by direct integration. We now show that this result can be obtained from the J.W.K.B. approximation of the wave optics solution.

The scalar Helmholtz equation is obtained from the substitution

$$k_y \rightarrow -i\partial/\partial y \tag{21}$$

in (19). We obtain

$$[\partial^2/\partial y^2 + k^2(\omega, y)]\psi_m = k_{zm}^2\psi_m, \tag{22}$$

where $m = 0, 1, 2, \cdots$, for trapped modes. Given $k(\omega, y)$, we look for solutions of (22) that are square-integrable and obtain the time of flight of a pulse in a mode m over a unit length by differentiating k_{zm} with respect to ω,

$$T = 1/v_g = \partial k_{zm}/\partial \omega. \tag{23}$$

Instead of solving (22) for k_z and differentiating with respect to ω, we may use the Hellmann-Feynman (H.F.) theorem.[15] Let $\mathfrak{K}$ be a self-adjoint operator depending on a parameter ω,

$$\mathfrak{K}(\omega)\psi_m = E_m\psi_m. \tag{24}$$

Premultiplying both sides of (24) by ψ_m we obtain

$$E_m = \langle \psi_m \mathfrak{K}(\omega) \psi_m \rangle / \langle \psi_m \psi_m \rangle, \tag{25}$$

where

$$\langle ab \rangle \equiv \int_{-\infty}^{+\infty} a^*(y) b(y) dy. \tag{26}$$

It is not difficult to show that E_m is stationary with respect to a small change in ψ_m. Thus, when we differentiate (25) with respect to ω (or ω^2), we can ignore the dependence of ψ_m on ω (or ω^2). We have for ψ a real

$$\frac{dE_m}{d\omega^2} = \frac{\langle \psi_m (d\mathfrak{K}/d\omega^2) \psi_m \rangle}{\langle \psi_m \psi_m \rangle}. \tag{27}$$

In our case, (22),

$$\mathfrak{K}(\omega) \equiv d^2/dy^2 + k^2(\omega, y). \tag{28}$$

Thus, by application of the H.F. theorem we obtain

$$(c/v_g)_m = (k_o/k_{zm})(dk_{zm}^2/dk_o^2) = (k_o/k_{zm}) \times \int_{-\infty}^{+\infty} (\partial k^2/\partial k_o^2) \psi_m^2 dy \Big/ \int_{-\infty}^{+\infty} \psi_m^2 dy \equiv (k_o/k_{zm}) \langle \partial k^2/\partial k_o^2 \rangle_m. \tag{29}$$

The J.W.K.B. method shows that, for large m, a mode can be represented by a manifold of rays satisfying the Bohr-Sommerfeld condition

$$\oint k_y dy = (m + \tfrac{1}{2}) 2\pi, \tag{30}$$

where the integral on the lefthand side in (30) is the area enclosed in phase space (k_y, y) by a ray trajectory. Equation (30) expresses the uniqueness of the phase of the field. At the turning point, $k_y = 0$, $y = \xi_m$, we have from (19)

$$k_{zm} = k(\omega, \xi_m). \tag{31}$$

An alternative way of obtaining the time of flight of a pulse in a mode m is to integrate ds/u from $z = 0$ to 1 along a ray of the manifold. The arc length is denoted by $ds = (k/k_z)dz$ and $u^{-1} \equiv \partial k/\partial \omega$ is the inverse of the local group velocity. Thus,

$$T = \int_0^1 \left(\frac{\partial k}{\partial \omega} \right) \left(\frac{k}{k_z} \right) dz \equiv c^{-1} \left(\frac{k_o}{k_z} \right) \left\langle \frac{\partial k^2}{\partial k_o^2} \right\rangle. \tag{32}$$

This expression, (32), in which $\langle\ \rangle$ denotes an average taken along a ray period, is the semiclassical analog of the Hellmann-Feynman theorem eqs. (27) and (29), and is used in the main text. It can be obtained

alternatively by noting that the group velocity in a waveguide is the ratio of the total energy flow to the energy stored per unit length. (This is a special case of the theorem derived in Ref. 16 for periodic bi-anisotropic media. To obtain the result applicable to open waveguides, we only have to let the periods go to infinity.) The result, (32), follows by integrating the energy density along a ray pencil bounded by the rays $y(z)$ and $y(z + dz)$. Let us sketch the proof. If Pdz denotes the energy flow in this ray pencil, the total energy flow in the waveguide is PZ. The energy density, on the other hand, is $P/u \sin\theta$, where θ is the angle that the ray makes with the z axis. Thus, the energy per unit length is obtained by integrating Pds/u along the ray, in agreement with (32).

APPENDIX B

Square-Law and Linear-Law Media

In this appendix we work out the case of square-law and linear-law media because they lend themselves to exact analytical expressions that are useful for comparison with computed solutions. The case in which the wave number k varies quadratically with y is also useful to obtain first-order solutions. Let us consider this case first.

$$k^2(\omega, y) = k_o^2(\omega) - \Omega^2(\omega)y^2, \tag{33}$$

where the functions $k_o(\omega)$ and $\Omega(\omega)$ are arbitrary. The wave equation, (22), is

$$(\partial^2/\partial y^2 + k_o^2 - \Omega^2 y^2) = k_z^2\psi, \tag{34}$$

where ψ represents, for instance, the y component of the electric field for H modes in a dielectric slab. This equation has the well-known eigenvalues

$$k_z^2 = k_o^2 - (2m + 1)\Omega. \tag{35}$$

Thus,

$$\begin{aligned} T = 1/v_g = dk_z/d\omega &= [k_o\dot{k}_o - (m + \tfrac{1}{2})\dot{\Omega}][k_o^2 - (2m + 1)\Omega]^{-\frac{1}{2}} \\ &= \dot{k}_o + (\Omega/k_o)(\dot{k}_o/k_o - \dot{\Omega}/\Omega)(m + \tfrac{1}{2}) + (\Omega^2/k_o^3) \\ &\quad \times [(\tfrac{3}{2})\dot{k}_o/k_o - \dot{\Omega}/\Omega](m + \tfrac{1}{2})^2 + \cdots, \end{aligned} \tag{36}$$

where upper dots denote differentiation with respect to ω. The condition for the removal of the first-order terms in (36), $\dot{k}/k_o = \dot{\Omega}/\Omega$, is the same as the condition of stationarity of $v/u \equiv \omega k^{-2}\frac{1}{2}(\partial k^2/\partial\omega)$ given in the main text. (Note that m is proportional to θ_o^2. Thus, first-order terms in m correspond to θ_o^2 terms.)

Let us now show that this result can be derived from the ray equations. Equations (19) and (20) are

$$dy/dz = k_y/k_z, \tag{37a}$$

$$dk_y/dz = -\Omega^2 y/k_z, \tag{37b}$$

$$d^2y/dz^2 + (\Omega/k_z)^2 y = 0. \tag{37c}$$

The solution of these equations is straightforward. We obtain

$$y = (k_{yo}/\Omega) \sin [(\Omega/k_z)z], \tag{38a}$$

$$k_y = k_{yo} \cos [(\Omega/k_z)z], \tag{38b}$$

where

$$k_{yo} \equiv k_o^2 - k_z^2, \tag{38c}$$

if we specify, for simplicity, that $y(0) = 0$, and use (33). The quantum condition, (30), is therefore

$$k_{yo}^2 = (2m + 1)\Omega. \tag{39}$$

Thus, setting $k_{yo}/\Omega \equiv \xi$, the axial wave number is given by

$$k_z^2 = k^2(\omega, \xi) = k_o^2 - \Omega^2\xi^2 = k_o^2 - (2m + 1)\Omega, \tag{40a}$$

in exact agreement with (35) (the agreement needs to be exact only for square-law media).

The ratio of the optical length of a ray period (period Z) to the corresponding length on axis is

$$\begin{aligned} R &= (k_o Z)^{-1} \int_0^Z k ds = (k_o k_z Z)^{-1} \int_0^Z k^2 dz \\ &= (k_o k_z Z)^{-1} \int_0^Z \{k_o^2 - k_{yo}^2 \sin^2 [(\Omega/k_z)z]\} dz \\ &= (1 - \tfrac{1}{2} \sin^2 \theta_o)/\cos \theta_o = 1 + \theta_o^4/8 + \cdots, \end{aligned} \tag{40b}$$

where θ_o denotes the angle between the ray and the z axis at the origin. By comparison, we have for a clad slab

$$R_c = 1/\cos \theta_o \approx 1 + \theta_o^2/2 + \cdots. \tag{40c}$$

Thus, for small θ_o, $R - 1$ is much smaller than $R_c - 1$, as discussed in the introduction. The above results, (40b) and (40c), are significant in the problem of pulse spreading in graded-index fibers if the material has low dispersion, but they are not relevant to tapered dielectric slabs. They are given here only for comparison.

Let us now evaluate the group velocity by integrating ds/u along a ray of the manifold, following (32). We have

$$v_g^{-1} = (k_z Z)^{-1} \frac{1}{2} \int_0^Z \left(\frac{\partial k_o^2}{\partial \omega} - \frac{\partial \Omega^2}{\partial \omega} y^2 \right) dz, \tag{41}$$

where $Z = 2\pi k_z/\Omega$ denotes the spatial period, and $y(z)$ is given in (38a). The integration is straightforward. Using (39), a result identical to (36) is obtained. Note that the above results are exact; the paraxial approximation was not made. We have shown in Appendix A that it is legitimate to evaluate pulse spreading by integrating the inverse of the local group velocity along rays representing the modes of propagation, in the limit of large mode numbers. The agreement is now found to be exact for square-law media.

For a linear-law medium with

$$k^2(\omega, y) = k_o^2(\omega) - 2a(\omega)|y|, \tag{42}$$

we shall only give the results. The rays are, from (20),

$$y(z) = \tan \theta_o z \mp (a/2k_o^2 \cos^2 \theta_o) z^2 \begin{cases} 0 < z < Z/2 \\ Z/2 < z < Z, \end{cases} \tag{43}$$

with a period

$$Z = 4k_o^2 \sin \theta_o \cos \theta_o / a. \tag{44}$$

The ratio of the optical length of a ray to the length on axis is

$$\int k ds \Big/ \int k_o dz = (1 - \tfrac{2}{3} \sin^2 \theta_o)/\cos \theta_o = 1 - \theta_o^2/6 + \cdots. \tag{45}$$

The situation is opposite to that of a clad fiber: The optical length *decreases* as θ_o increases. Therefore, we may in that case have a small increase of v/u when the slab thickness is reduced, that is, work on the right side of the dotted line in Fig. 2. This leads to a thicker slab than in the case of square-law profiles. These theoretical results are confirmed by the curves in Figs. 5 and 6. We note that the optimum ϕ_o is about 1, the maximum of the v/u curve being at only 0.8. The time of flight is, using (32),

$$T = 1/v_g = (\cos \theta_o)^{-1} dk_o/d\omega - (23/6)(k_o/a)(\sin^2 \theta_o/\cos \theta_o)(da/d\omega).$$

Thus, T is independent of θ_o for small θ_o (no terms in θ_o^2) if $k_o(\omega)$ and $a(\omega)$ in (42) are related by

$$(dk_o/d\omega)/k_o = (23/3)(da/d\omega)/a. \tag{46}$$

It can be shown that this condition corresponds to an increase of v/u with $|y|$, in agreement with the previous discussion.

REFERENCES

1. S. Kawakami and J. Nishizawa, "Proposal of a New Thin Film Waveguide," Research Inst. of Elec. Comm. Tech. Rep., TR-25, Oct. 1967 and "An Optical Waveguide with the Optimum Distribution of the Refractive Index with Reference to Waveform Distortion," *op. cit.*, TR-24.
2. H. E. Rowe and D. T. Young, "Transmission Distortion in Multimode Random Waveguides," I.E.E.E. Trans. of Microwave and Technique, *M.T.T. 20*, No. 6 (June 1972), pp. 350–365.
3. L. B. Slichter, "The Theory of the Interpretation of Seismic Travel-Time Curves in Horizontal Structures," Physics, *3*, December 1932, pp. 273–295 (this reference was given by A. H. Carter in an unpublished work).
4. R. K. Luneburg, *The Mathematical Theory of Optics*, lectures at Brown University, 1944, published by University of California Press, Berkeley, 1964, p. 180.
5. L. D. Landau and E. M. Lifshitz, *Quantum Mechanics Non-Relativistic Theory*, 2nd ed., London: Pergamon Press, 1965, pp. 72–73.
6. E. T. Kornhauser and A. D. Yaghjian, "Modal Solution of a Point Source in a Strongly Focusing Medium," Radio Science, *2* (March 1967), pp. 299–310.
7. E. G. Rawson, D. R. Herriott, and J. McKenna, "Analysis of Refractive-Index Distributions in Cylindrical Graded Index Glass Rods Used as Image Relays," Appl. Opt., *9* (March 1970), pp. 753–759.
8. D. Marcuse, "The Impulse Response of an Optical Fiber with Parabolic Index Profile," B.S.T.J., *52*, No. 7 (September 1973), pp. 1169–1173.
9. S. E. Miller, "Delay Distorsion in Generalized Lens-Like Media," B.S.T.J., *53*, No. 2 (February 1974), pp. 177–193.
10. D. Gloge and E. A. J. Marcatili, "Impulse Response of Fibers with Ring-Shaped Parabolic Index Difference," B.S.T.J., *52*, No. 7 (September 1973), pp. 1161–1168.
11. J. A. Arnaud, "Hamiltonian Theory of Beam Mode Propagation," in *Progress in Optics*, Vol. XI, E. Wolf, ed., Amsterdam: North Holland, 1973.
12. E. A. J. Marcatili, "Slab-Coupled Waveguides," B.S.T.J., *53*, No. 4 (April 1974), pp. 645–674.
13. H. Kogelnik and V. Ramaswamy, "Scaling Rules for Thin-Film Optical Waveguides," Appl. Opt. *13*, No. 8 (August 1974), pp. 1857–1862.
14. G. B. Whitham, "Two-Timing, Variational Principles and Waves," J. Fluid Mech., *44*, Part 2, pp. 373–395, 1970.
15. E. Merzbacher, *Quantum Mechanics*, New York: John Wiley, 1970, p. 442.
16. J. A. Arnaud and A. A. M. Saleh, "Theorems for Bi-anisotropic Media," Proc. of the I.E.E.E., *60*, No. 5 (May 1972), pp. 639–640.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

Theory of the Single-Material Fiber

By D. MARCUSE

(Manuscript received March 6, 1974)

The term "single-material fiber" describes a dielectric optical waveguide made of only one type of glass. The theory of this waveguide is simplified by placing the structure between two perfectly conducting planes that have very little influence on the properties of the low-order modes.

The field distribution and propagation constant of the lowest-order mode are investigated and compared to an approximate theory.

I. INTRODUCTION

A dielectric optical waveguide made entirely of one type of material is called a "single-material fiber."[1] Figure 1 shows such a structure schematically. It may be regarded as a rectangular dielectric waveguide supported by two infinitely extended slabs made of the same material. Such a structure has been shown to be capable of supporting modes that are concentrated near the enlarged section of the waveguide and that do not lose power by energy seepage into the slabs.[1,2] Single-material fibers are usually made of pure fused silica. Since no other material is needed to form a waveguide, the low-loss properties of pure fused silica can be fully utilized.[3]

The single-material fiber has been described by means of an approximate theory by Marcatili.[1] The theory presented here serves the purposes of proving that truly guided modes do indeed exist in single-material fibers and of providing more precise solutions for comparison with the approximate theory.

An analysis of the guided modes of the single-material fiber is presented in this paper. The mode field is expressed as a superposition of the guided modes as well as the radiation modes of the two types of slabs. The enlarged region, henceforth called the core, can be regarded as a slab joined by narrower support slabs on either side. Since the radiation modes of the slabs have a continuous spectrum of eigenvalues[4,5] (propagation constants), their contribution to the total field

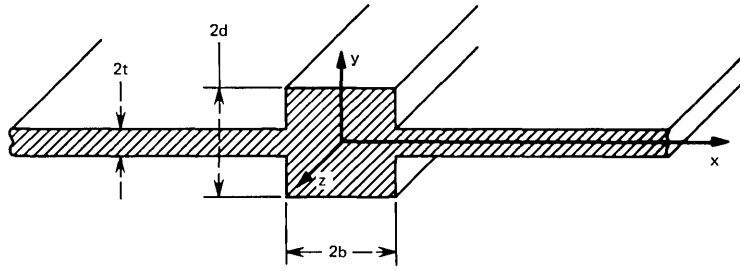


Fig. 1—Single-material fiber showing the rectangular core attached to its support slab.

consists of an integral that must be approximated by a sum for purposes of numerical analysis. In addition, the mathematical expression describing the guided and radiation modes are different so that the analysis becomes rather complex.

To simplify the analysis, it is convenient to consider the single-material fiber enclosed between two perfectly conducting planes, as shown in Fig. 2. Since the fields of the guided modes of the slabs, and hence the field of the guided mode of the fiber, are very tightly confined near the dielectric structure, the presence of the perfectly conducting planes does not appreciably influence the shape of horizontally polarized fields. However, the simplification of the analysis is considerable, since the modes that correspond to the guided modes of the open slab and the waveguide modes of the parallel plate system (corresponding to the radiation modes of the open slab) are now described by one analytical expression and belong to a system of discrete eigenvalues. There is, therefore, no need to worry about a suitable approximation to the integral over the radiation modes of the slabs. Vertically polarized fields (polarized in the y-direction) are strongly

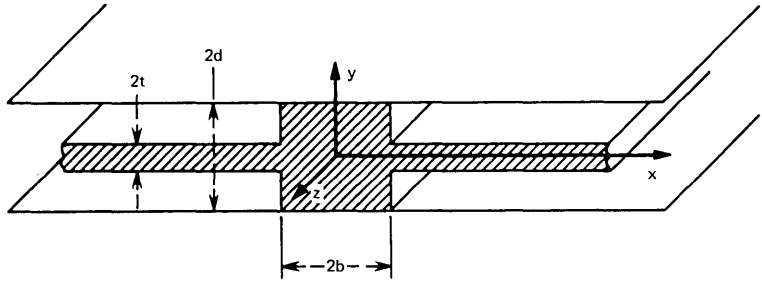


Fig. 2—For our analysis, the single-material fiber is placed between two perfectly conducting planes.

influenced by the presence of the metal plates, since the normal field components reach the metal plates with maximum intensity. For this reason, we limit the study of the single-material fiber to horizontally polarized modes. Vertically polarized modes could be treated if the perfectly conducting planes were replaced by magnetic short circuits. (An explanation of the negligible influence of the perfect conductors is given in the appendix.)

After formulating the exact solution to our problem, we present numerical approximations for the field distributions and the solution of the eigenvalue equation. The theory is compared to an approximate analysis.

II. CALCULATION OF THE MODES OF THE SINGLE-MATERIAL FIBER

The electric and magnetic fields of the modes of the single-material fiber are expressed as

$$\mathbf{E}^{(i)} = \sum_{\nu=1}^{\infty} c_\nu^{(i)} \mathcal{E}_\nu^{(i)} \tag{1}$$

and

$$\mathbf{H}^{(i)} = \sum_{\nu=1}^{\infty} c_\nu^{(i)} \mathcal{H}_\nu^{(i)}. \tag{2}$$

The script symbols indicate modes of the slabs. The superscript i assumes the values 1 and 2. Value 1 indicates modes of the wider slab that forms the core of the single-material fiber, while value 2 indicates the modes of the narrower supporting slabs.

The modes in the core region are those of a metallic parallel plate waveguide. These modes can be designated as TE modes with $\mathcal{E}_z^{(1)} = 0$ and TM modes with $\mathcal{H}_z^{(1)} = 0$. We have for the TE modes in region 1

$$\mathcal{E}_{z\nu}^{(1)} = 0 \tag{3}$$

and

$$\mathcal{H}_{z\nu}^{(1)} = \frac{A_\nu}{\omega\mu_0} \cos(k_{x\nu}x) \sin(k_{y\nu}y) e^{-i\beta z}. \tag{4}$$

We use odd integers, $\nu = 1, 3, \cdots$, to label these modes. In addition to the sine and cosine functions appearing in (4) we could also use the other three possible combinations. We restrict ourselves to the modes shown here, thus limiting ourselves to the study of fiber modes of a certain symmetry. All other modes can be obtained similarly.

The other field components can be obtained from $\mathcal{E}_z^{(1)}$ and $\mathcal{H}_z^{(1)}$ by differentiation (see, for example, page 13 of Ref. 4 or page 51 of Ref. 5).

The parameters appearing in (4) are related as

$$n^2k^2 = k_{x\nu}^2 + k_{y\nu}^2 + \beta^2, \tag{5}$$

with

$$k = \omega\sqrt{\epsilon_0\mu_0} = \frac{2\pi}{\lambda}. \tag{6}$$

The refractive index of the single-material fiber is designated by n and the refractive index of the medium outside the fiber is taken to be unity. The angular frequency is ω, and ϵ_0 and μ_0 are the dielectric permittivity and the magnetic permeability of vacuum.

The TM modes are labeled by even integers, $\nu = 2, 4, \cdots$, and are obtained from the field components

$$\mathcal{E}_{z\nu}^{(1)} = \frac{B_\nu}{\omega\epsilon_0} \sin(k_{x\nu}x) \cos(k_{y\nu}y)e^{-i\beta z} \tag{7}$$

and

$$\mathfrak{IC}_{z\nu}^{(1)} = 0. \tag{8}$$

TE and TM modes must satisfy the boundary conditions that $\mathcal{E}_{z\nu}^{(1)}$ and $\mathcal{E}_{x\nu}^{(1)}$ vanish at $y = \pm d$. These conditions are met if we use

$$k_{y\nu} = (2\mu_\nu - 1)\frac{\pi}{2d}. \tag{9}$$

Equations (5) and (9) are the same for TE modes (odd values of ν) and TM modes (even values of ν). The integers μ_ν assume the values

$$\begin{array}{llll} \mu_\nu = 1 & \text{for} & \nu = 1, 2 \\ \mu_\nu = 2 & \text{for} & \nu = 3, 4 \\ \mu_\nu = 3 & \text{for} & \nu = 5, 6 \\ \text{etc.} & & \end{array} \tag{10}$$

The TE and TM modes are mutually orthogonal. Their amplitude coefficients can be related to the amount of power in the core region by means of the equation

$$P = \frac{1}{2}\int_{-b}^{b} dx \int_{-d}^{d} dy(\mathcal{E}_\nu \times \mathfrak{IC}_\nu^*)_z. \tag{11}$$

The asterisk indicates complex conjugation, and the subscript z labels the z component of the vector. From (3), (4), and (11) we obtain

$$A_\nu = \left\{\frac{2\omega\mu_0(k_{y\nu}^2 + k_{x\nu}^2)^2 P}{\beta d\left|(k_{y\nu}^2 + k_{x\nu}^2)b + (k_{y\nu}^2 - k_{x\nu}^2)\dfrac{\sin 2k_{x\nu}b}{2k_{x\nu}}\right|}\right\}^{\frac{1}{2}}. \tag{12}$$

From (7), (8), and (11) follows also

$$B_\nu = \left\{ \frac{2\omega\epsilon_0(k_{y\nu}^2 + k_{x\nu}^2)^2 P}{n^2\beta d\left|(k_{y\nu}^2 + k_{x\nu}^2)b - (k_{y\nu}^2 - k_{x\nu}^2)\dfrac{\sin 2k_{x\nu}b}{2k_{x\nu}}\right|} \right\}^{\frac{1}{2}}. \tag{13}$$

It is apparent from (5) and (9) that $k_{x\nu}^2$ can assume positive as well as negative values.

We choose P equal to the unit of power. With this normalization, $|c_\nu^{(i)}|^2$ appearing in (1) and (2) measure directly the power carried by each mode.

Next, we turn to the modes of the narrower support slabs. Since the perfectly conducting planes do not touch the support slabs, their modes are more complicated. Strictly speaking, we do not have TE or TM modes with reference to the z direction. However, if we refer the labels TE or TM to the direction of propagation of the modes in the x–z plane, we do indeed have transverse electric and transverse magnetic modes. Used in this sense, we obtain the following z components of the TE modes of the support slabs:

$$\mathcal{E}_{z\nu}^{(2)} = \begin{cases} C_\nu e^{-i\sigma_{x\nu}(|x|-b)} \cos(\sigma_{y\nu}y) e^{-i\beta z} & |y| \leqq t \\ -C_\nu \dfrac{\cos\sigma_{y\nu}t}{\sin[\rho_\nu(d-t)]} e^{-i\sigma_{x\nu}(|x|-b)} & \\ \qquad \times \sin[\rho_\nu(|y|-d)] e^{-i\beta z} & t \leqq |y| \leqq d \end{cases} \tag{14}$$

and

$$\mathcal{H}_{z\nu}^{(2)} = \begin{cases} \dfrac{\sigma_{y\nu}\beta}{i\sigma_{x\nu}\omega\mu_0} C_\nu e^{-i\sigma_{x\nu}(|x|-b)} \sin(\sigma_{y\nu}y) e^{-i\beta z} & |y| \leqq t \\ \dfrac{\sigma_{y\nu}\beta}{i\sigma_{x\nu}\omega\mu_0} C_\nu \dfrac{y}{|y|} \dfrac{\sin\sigma_{y\nu}t}{\cos\rho_\nu(d-t)} e^{-i\sigma_{x\nu}(|x|-b)} & \\ \qquad \times \cos[\rho_\nu(|y|-d)] e^{-i\beta z} & t \leqq |y| \leqq d. \end{cases} \tag{15}$$

For TE modes, we have $\mathcal{E}_{y\nu}^{(2)} = 0$.

Maxwell's equations are satisfied if the parameters appearing in these field expressions satisfy the following relations:

$$n^2k^2 = \sigma_{x\nu}^2 + \sigma_{y\nu}^2 + \beta^2 \tag{16}$$

and

$$k^2 = \sigma_{x\nu}^2 + \rho_\nu^2 + \beta^2. \tag{17}$$

We again use odd integers ν to label the TE modes.

The field expressions satisfy the condition of vanishing tangential electric fields at the perfectly conducting planes. To satisfy the boundary conditions at the surface of the slab, the x dependence of the field expressions must be identical inside as well as outside the slab. For

this reason, the parameter σ_{xv} is common to the fields for $t \geqq |y|$ and $t \leqq |y| \leqq d$. Since the fields must also satisfy boundary conditions along the planes $x = \pm b$ for all values of z, the parameter β must be the same for all field expressions, where β is the propagation constant for the mode of the single-material fiber that is yet to be determined.

The requirement of continuity of the field components $\mathcal{E}_{zv}^{(2)}$, $\mathcal{E}_{xv}^{(2)}$, $\mathcal{H}_{zv}^{(2)}$ and $\mathcal{H}_{xv}^{(2)}$ at $y = \pm t$ leads to the eigenvalue equation

$$\tan \sigma_{yv} t = \frac{\rho_v}{\sigma_{yv}} \cot [\rho_v(d - t)]. \tag{18}$$

Equation (18) determines the allowed values of σ_{yv} and ρ_v, since according to (16) and (17) we have

$$\rho_v^2 = \sigma_{yv}^2 - (n^2 - 1)k^2. \tag{19}$$

Although σ_{yv}^2 is always positive, ρ_v^2 can be positive as well as negative. Modes with negative values of ρ_v^2 correspond to the guided modes of an open slab. For negative values of ρ_v^2, the cotangent function on the right-hand side of (18) becomes a hyperbolic cotangent function that approaches unity for large values of its argument. The eigenvalue equation (18) is thus identical [for large values of $|\rho_v|(d - t)$] to the eigenvalue equation (8.3–16) on page 308 of Ref. 4 for even TE modes of the slab waveguide.

Modes with positive values of ρ_v^2 correspond to the radiation modes of the open slab. However, instead of the continuous spectrum of radiation modes,[4,5] we now have a discrete spectrum of modes that approach the modes of the metallic parallel plate waveguide in the limit of vanishing slab thickness $2t$. The guided as well as the radiation modes of the narrow support slabs are thus represented by the same analytical expressions (14) and (15). The presence of the perfectly conducting planes has the added advantage of causing the mode spectrum to be discrete.

The parameter σ_{xv}^2 can also be positive as well as negative. Positive values of σ_{xv}^2 correspond to real values of σ_{xv}, so that the mode fields (14) and (15) represent traveling waves that carry power away from the core region into the slab. Coupling the modes in the core region and the slab regions thus results in a leaky wave. It is clear that we obtain guided single-material fiber modes only for negative values of σ_{xv}^2. We see from (16) that all σ_{xv}^2 are negative if σ_{x1}^2 of the lowest-order mode is negative, because the values of σ_{yv} increase with increasing mode number. It is thus immediately apparent that lossless guided modes of the single-material fiber are indeed possible. Neither the guided modes

of the slab (those tightly confined to the slab region) nor the "unguided" modes, which correspond to the radiation modes of the open slab, carry away power; all decay exponentially in x direction. This argument is not changed when we let the metallic plates move to infinity so that we obtain a truly free single-material fiber. The single-material fiber is thus seen able to support guided modes whose fields are confined to the vicinity of the fiber core. The existence of these guided modes is contingent on sufficiently large values of β. Whether such solution with large β values really exist depends on the solutions that we must yet derive of the eigenvalue equation for β. However, even at this stage we can state that guided modes that do not suffer radiation losses are possible at least in principle.

Using (11), (14), and (15) we can again relate the amplitude coefficient to the power unit P:

$$C_\nu = \left\{ \frac{4\omega\mu_0 |\sigma_{x\nu}|^3 P}{\beta(\beta^2 + \sigma_{x\nu}^2)\,|t + \{\cos^2 \sigma_{y\nu} t/\sin^2 [\rho_\nu(d-t)]\}(d-t) - (n^2-1)k^2 \sin(2\sigma_{y\nu}t)/2\sigma_{y\nu}\rho_\nu^2|} \right\}^{\frac{1}{2}}. \quad (20)$$

The TM modes of the support slabs are labeled by even integers ν and follow from their z components:

$$\mathcal{E}_{z\nu}^{(2)} = \begin{cases} D_\nu e^{-i\sigma_{x\nu}(|x|-b)} \cos(\sigma_{y\nu} y) e^{-i\beta z} & |y| \leqq t \\ -D_\nu \dfrac{\cos \sigma_{y\nu} t}{\sin [\rho_\nu(d-t)]} e^{-i\sigma_{x\nu}(|x|-b)} \\ \qquad \times \sin [\rho_\nu(|y| - d)] e^{-i\beta z} & t \leqq |y| \leqq d, \end{cases} \quad (21)$$

$$\mathcal{H}_{z\nu}^{(2)} = \begin{cases} \dfrac{i\sigma_{x\nu} n^2 k^2}{\omega\mu_0 \beta \sigma_{y\nu}} D_\nu e^{-i\sigma_{x\nu}(|x|-b)} \sin(\sigma_{y\nu} y) e^{-i\beta z} & |y| \leqq t \\ \dfrac{i\sigma_{x\nu} n^2 k^2}{\omega\mu_0 \beta \sigma_{y\nu}} D_\nu \dfrac{\sin \sigma_{y\nu} t}{\cos [\rho_\nu(d-t)]} \dfrac{y}{|y|} e^{-i\sigma_{x\nu}(|x|-b)} \\ \qquad \times \cos [\rho_\nu(|y| - d)] e^{-i\beta z} & t \leqq |y| \leqq d. \end{cases} \quad (22)$$

For TM modes we have $\mathcal{H}_{y\nu}^{(2)} = 0$.

The parameters $\sigma_{x\nu}$, $\sigma_{y\nu}$, and ρ_ν are again related by (16), (17), and (19). The eigenvalue equation for TM modes is

$$\tan \sigma_{y\nu} t = \frac{1}{n^2} \frac{\sigma_{y\nu}}{\rho_\nu} \cot [\rho_\nu(d-t)]. \quad (23)$$

For large imaginary values of $\rho_\nu(d-t)$, (23) becomes the eigenvalue equation (8.3–45), page 313 of Ref. 4, for odd TM modes of the free slab.

Finally, we have

$$D_\nu = \left\{ \frac{4\omega\mu_0\beta\sigma_{y\nu}^2 \left| \sigma_{x\nu} \right|}{n^2k^2(\beta^2 + \sigma_{x\nu}^2)\left| t + n^2 \frac{\sin^2 \sigma_{y\nu}t}{\cos^2 [\rho_\nu(d-t)]}(d-t) + \frac{(n^2-1)k^2 \sin 2\sigma_{y\nu}t}{2\sigma_{y\nu}\rho_\nu^2} \right|} \right\}^{\frac{1}{2}} . \quad (24)$$

Now we have written down the field expressions for the mode fields that must be substituted into the series expansions (1) and (2) for the mode of the single-material fiber. It remains to match the field in the core of the single-material fiber to the field in the regions of the support slab. We need to require continuity of E_z, E_y, H_z, and H_y only along the line $x = b$ and $0 < y < d$, since the boundary conditions in the remaining three quadrants are satisfied for reasons of symmetry. Since the numerical analysis can handle only a finite number of equations, we require continuity of the tangential field components only at a finite number of points. Adjusting the size of the series expansion to the number of matching points, we obtain a finite, homogeneous equation system for the determination of the expansion coefficients $c_\nu^{(i)}$. This equation system can only have a solution if the determinant vanishes. The condition of vanishing system determinant provides the eigenvalue equation for the propagation constant β of the single-material fiber.

III. SPECIAL CASES AND APPROXIMATE SOLUTIONS

In the limit $t = 0$, an exact solution of the guided-mode problem is easily obtained. Since, in this case, the distributions of the fields in the two regions have the same y dependence, the boundary conditions along the plane $x = \pm b$ can be satisfied without resorting to the series expansions (1) and (2). Using the field expressions (3), (4), (7), (8), (14), (15), (21), and (22) and requiring continuity of E_z, E_y, H_z, and H_y at $x = b$ leads to the eigenvalue equations

$$\tan k_x b = -\frac{k_x}{i\sigma_x} \quad (25)$$

or

$$\tan k_x b = n^2 \frac{i\sigma_x}{k_x}. \quad (26)$$

For guided modes we have

$$i\sigma_x = \eta, \quad (27)$$

with real positive η.

Equation (25) is the eigenvalue equation for odd TE modes of a slab waveguide, while (26) is the eigenvalue equation for even TM modes of the slab.[4] When the amplitude coefficients of the superpositions of the TE and TM modes (3), (4), (7), and (8) (that are determined with the help of the boundary conditions) are substituted into the field expressions, we obtain for the field in the fiber core belonging to (25)

$$E_z = -\frac{k_y}{\beta k_x} F \sin(k_x x) \cos(k_y y) e^{-i\beta z} \qquad |x| \leqq b \tag{28}$$

and

$$H_z = \frac{F}{\omega \mu_0} \cos(k_x x) \sin(k_y y) e^{-i\beta z} \qquad |x| \leqq b. \tag{29}$$

For this mode we have $E_x = 0$. Viewing this field from the boundary of the slab, $x = b$, we see that the normal electric field component vanishes. This is typical for TE modes of the slab waveguide so that it is not surprising that the propagation constant of this mode is determined by an eigenvalue equation of the TE type.

The mode belonging to (26) has the following z components:

$$E_z = \frac{k_x \beta}{n^2 k_y k^2} G \sin(k_x x) \cos(k_y y) e^{-i\beta z} \qquad |x| \leqq b \tag{30}$$

and

$$H_z = \frac{G}{\omega \mu_0} \cos(k_x x) \sin(k_y y) e^{-i\beta z} \qquad |x| \leqq b. \tag{31}$$

This mode has $H_x = 0$. With respect to the surface $x = b$, it is indeed a TM mode.

For simplicity, the fields outside the core are not stated. However, a good approximation to these field expressions is obtained by using (21) and (22) to extend the field (28) and (29) outside the core and similarly by using (14) and (15) with the core fields (30) and (31).

For $t \neq 0$, the mode field of the single-material fiber can only be described by an infinite series of modes. However, we find a crude approximation by using only the first two terms in this series expansion and obtain an eigenvalue equation by requiring that a certain wave impedance be matched at the interface $x = b$. We stated earlier that only those modes of our structure with vanishing normal field components at the metallic planes resemble modes of the true single-material fiber. The mode field (28) and (29) with $E_x = 0$ has a strong normal component of the electric field. We thus limit ourselves to the horizontally polarized field and use (30) and (31) as a crude approxima-

tion. Note that the field (30) and (31) consists of a superposition of one TE mode [eqs. (3) and (4)] and one TM mode [eqs. (7) and (8)].

The wave impedance,

$$\frac{E_z}{H_y} = \frac{i\omega\mu_0 k_x}{n^2k^2} \tan k_x b, \tag{32}$$

obtained from (30) and (31) does not depend on the y coordinate. Similarly, we use (14) and (15) to form

$$\frac{E_z}{H_y} = -\frac{\omega\mu_0\sigma_x}{n^2k^2 - \sigma_y^2}, \tag{33}$$

which is also independent of y. Since the tangential field components must be continuous at the boundary $x = b$ between the two regions, we require that (32) be equal to (33), obtaining the following approximate eigenvalue equation for the horizontally polarized modes (of only a certain special symmetry) of the single-material fiber:

$$\tan k_x b = \frac{i\sigma_x}{k_x} \frac{n^2k^2}{n^2k^2 - \sigma_y^2}. \tag{34}$$

The parameter σ_y must be obtained as the solution of the eigenvalue equation (18). In the limit $t = 0$, (34) should reduce to (26). To see that the correct limit is obtained, we use (19) to write

$$n^2k^2 - \sigma_y^2 = k^2 - \rho^2. \tag{35}$$

For $t = 0$, we obtain from (18)

$$\rho = (2\mu - 1)\frac{\pi}{2d}. \tag{36}$$

For small values of the integer μ and $kd \gg 1$, we have $\rho \ll k$ so that (26) and (34) become indeed approximately the same. We do not get exact agreement, since we approximated the field outside the core by (14) and (15) instead of using the exact field expressions. We see that our eigenvalue equation (34) is a good approximation in the two limiting cases, $t \to 0$ and $t \to d$. Once σ_y has been determined from (18) we find $\eta = i\sigma_x$ from

$$\eta = \sqrt{\sigma_y^2 - k_x^2 - k_y^2}, \tag{37}$$

and (34) with the help of (9). The propagation constant β can then be obtained from (5) or (16).

IV. DISCUSSION AND NUMERICAL EXAMPLES

Marcatili has shown by an approximate analysis[1] that the single-material fiber can be made to support only a single guided mode, even if its dimensions are large compared to the wavelength, if the ratio $(\pi/4)(bd/t^2)$ approaches unity. However, for large values of kd and large values of bd/t^2, the single-material fiber supports a large number of guided modes.

We are limiting our discussion to the lowest-order guided mode. Since the properties of the single-material fiber can be obtained adequately from the approximate solutions, it is our principal purpose to show how well the approximate solution (34) and Marcatili's approximate theory work, and to study the field distributions of the exact solution that cannot be obtained from the approximate analysis. As indicated earlier, we limit the discussion to the modes with horizontal polarizations ($E_y = 0$), since the vertically polarized modes ($E_x = 0$) are very strongly influenced by the presence of the perfectly conducting planes that were used only to simplify the analysis.* The analysis is further restricted to modes whose E_x component is a symmetric function in both x and y. The modes with other symmetries can be obtained similarly by using slab waveguide modes of the appropriate symmetries.

All the numerical examples shown here were computed for the following choice of parameters:

$$\begin{aligned} d/\lambda &= b/\lambda = 5 \\ n &= 1.5. \end{aligned} \tag{38}$$

The boundary conditions at the plane $x = b$ were satisfied by matching the fields at 10 points evenly distributed between $y/d = 0.05$ and $y/d = 0.95$. As a consequence, the field expansion uses 20 modes in each region, 10 TE modes and 10 TM modes. Adequate accuracy was obtained this way. However, an expansion using only 6 points to match the fields did not appear sufficiently accurate.

The computer program was written to solve, first, the eigenvalue equations (18) and (23) by an iterative search procedure. Next, the computer was instructed to use a large trial value for β and compute the normalized field amplitudes (12), (13), (20), and (24) as well as the matrix elements of the equations system resulting from the boundary conditions at the N matching points. Next, the system determinant was examined and β was decreased until the determinant changed its

* The case of vertically polarized modes can be treated by replacing the electrical short-circuit planes with magnetic short circuits.

sign. By narrowing the increments for β successively and oscillating around the point where the sign change of the determinant occurred, an approximate solution for β was determined. Since the order of magnitude of the determinant was not known *a priori*, no attempt was made to reduce the value of the determinant below a certain limit. Once an approximate eigenvalue had been found, the coefficient $c_1^{(1)}$ was set equal to unity, and the first equation of the system was omitted. The remaining equation system was solved by inverting the reduced coefficient matrix. The values of the expansion coefficients were finally used to calculate the magnitude and direction of the electric field in a grid of preselected points in the x–y plane.

Figures 3 to 6 compare the magnitude of the electric field vector of the lowest-order mode of the single-material fiber with the magnitude of the field of the rectangular waveguide if $t/d = 0$. Figure 3 applies to a single-material fiber with the dimensions given by (38) and with $t/d = 0.32$. The magnitude of the field intensity is plotted as a function of x/b for different values of y/d. It is apparent that the field intensity decreases with increasing values of y. The field is strongest on axis and vanishes at $y = d$. In the absence of metallic planes, the field would not be zero at $y = d$, but would decrease to a very small value. The solid curves indicate the field of the single-material fiber, while the broken curves apply to the rectangular waveguide ($t = 0$). In the region of the guide where the support slab is present, $y/d < 0.32$, the field of the single-material fiber reaches out much further than the field of the corresponding rectangular waveguide, since it penetrates into the slab. For $y/d > 0.32$, the field shape of the single-material fiber has become identical with the field distribution of the rectangular waveguide.

Figure 4 shows the field distribution as a function of y/d for four different values of x/d. The solid curves describe again the field of the single-material fiber, while the broken curves belong to the rectangular waveguide. In the y direction, both fields vanish at $y = d$ but, near the edge of the single-material fiber, its field intensity is quite different from the rectangular waveguide field. We have plotted the ratio of the field intensity to the maximum value (the value that the field assumes for each value of x/d) at $y/d = 0$. Far from the edge of the core, the single-material fiber field is identical to the field of the rectangular waveguide. However, near the edge, at $x/b = 1$, the field is strong in the region $0 < y/d < 0.32$, since it is allowed to penetrate into the support slab. But in the range $0.32 < y/d < 1$, where it encounters the dielectric interface, it is relatively much weaker. The field of the rectangular waveguide is likewise weak near the dielectric

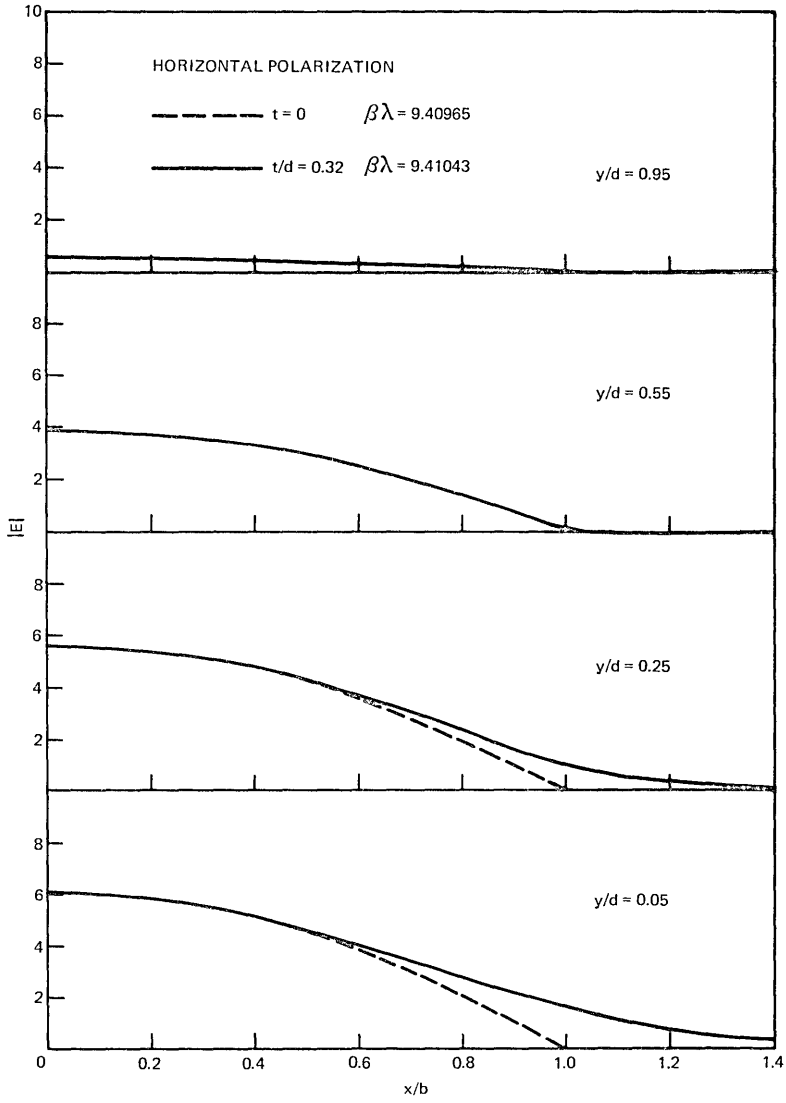


Fig. 3—Magnitude of the electric field vector shown as a function of the normalized horizontal dimension x/b. Solid curves describe the single-material fiber with $t/d = 0.32$, and broken curves apply to the fiber with $t/d = 0$.

interface; it appears strong only because of our normalization with respect to the maximum field intensity at $y/d = 0$. For $x/b > 1$, the rectangular waveguide field is no longer plotted since it decays rapidly to insignificant values outside the waveguide core. The single-material

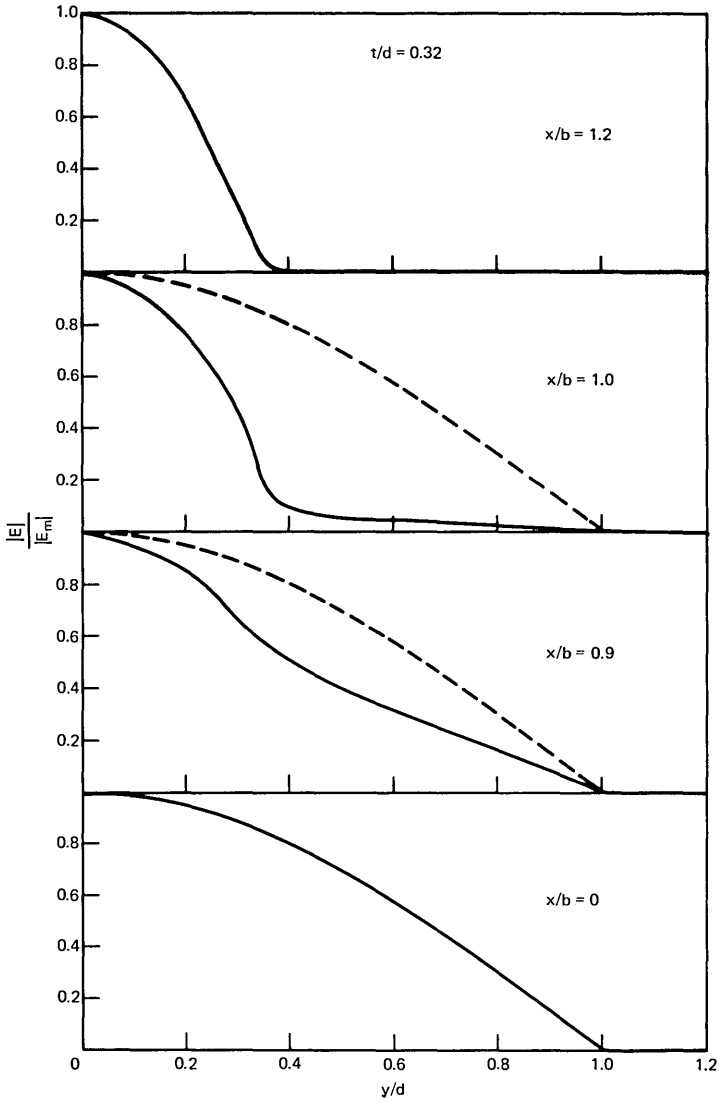


Fig. 4—Magnitude of the electric field vector relative to its maximum value at $y = 0$ as a function of the normalized vertical dimension y/d. Solid and broken curves describe the single-material fiber with $t/d = 0.32$ and $t/d = 0$.

fiber field shows the distribution typical of the lowest-order mode in the support slab.

Figures 5 and 6 show the same behavior for a single-material fiber with a much wider slab, $t/d = 0.8$. The field penetrates even further into the support slab, as can be seen from Fig. 5. However, the field

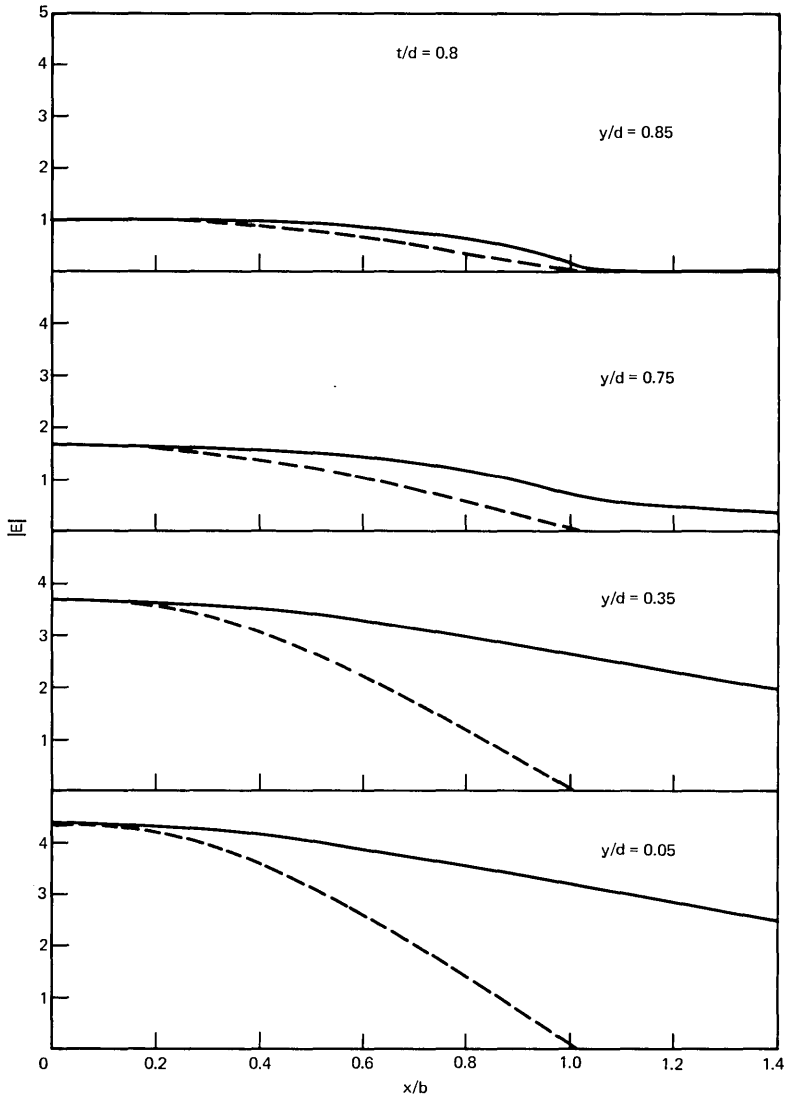


Fig. 5—Magnitude of the electric field vector shown as a function of the normalized horizontal dimension x/b. Solid curves describe the single-material fiber with $t/d = 0.8$, and the broken curves apply to the fiber with $t/d = 0$.

distribution in the vertical plane, shown in Fig. 6, is now much closer to the field distribution in the core of the rectangular fiber.

Figures 7 and 8 show the mode spectra for the single-material fiber with $t/d = 0.32$. Figure 7 presents the mode content of the field in the core. Because of our normalization, the square of the mode amplitudes

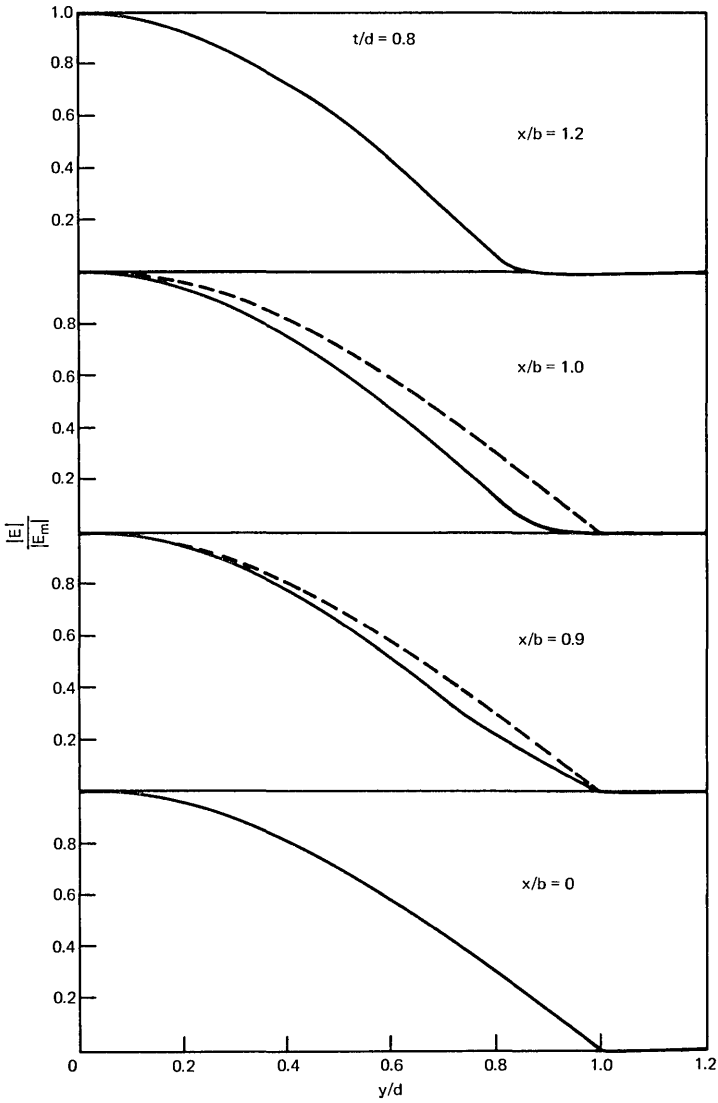


Fig. 6—Magnitude of the electric field vector relative to its maximum value at $y = 0$ as a function of the normalized vertical dimension y/d. Solid and broken curves describe the single-material fiber with $t/d = 0.8$ and $t/d = 0$.

$c_\nu^{(1)}$ represents the relative power carried by each mode of the series expansion (1) and (2). The broken vertical lines give the mode content of the corresponding mode of the rectangular waveguide. It is remarkable how nearly identical the mode amplitudes of the lowest-order TE and TM modes are in either case. Note that the mode amplitudes of

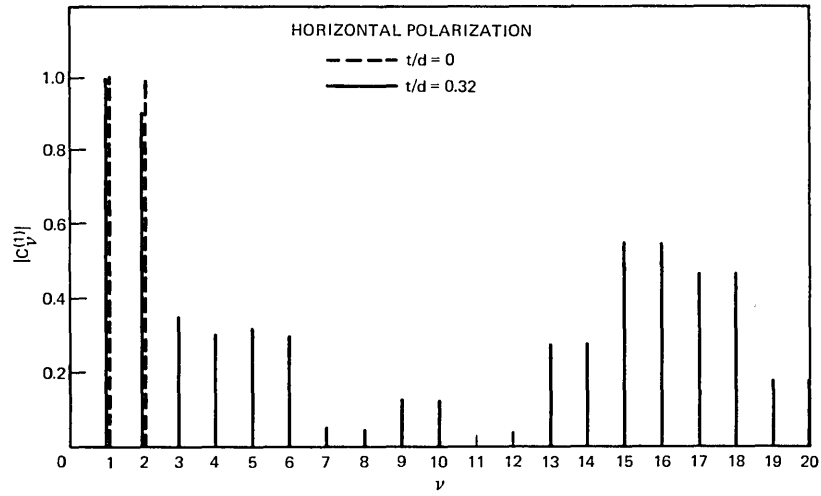


Fig. 7—Mode spectrum of the lowest-order single-material fiber mode inside its core. The solid vertical lines describe a fiber with $t/d = 0.32$, the broken vertical lines belong to the case $t/d = 0$.

the higher-order modes vanish because of the presence of the perfectly conducting planes; without them, the rectangular waveguide modes would also have to be represented by infinite-series expansions with a very slight mixture of higher-order modes. The mode of the single-material fiber consists of a mixture of the higher-order modes required

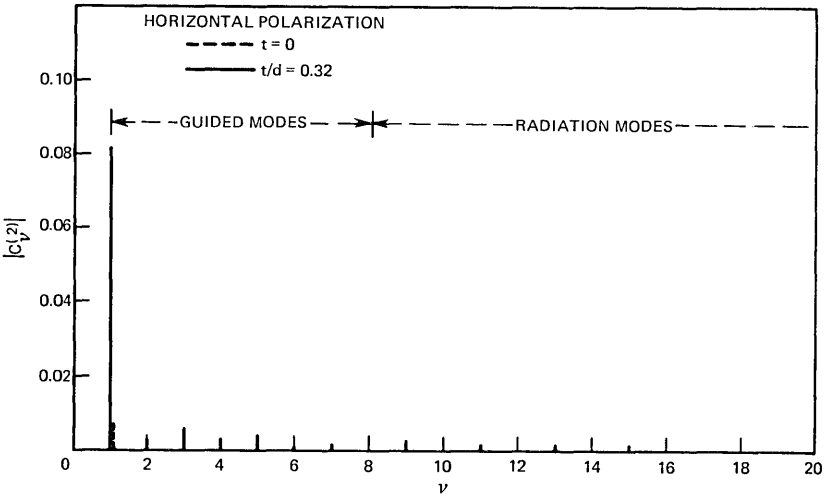


Fig. 8—Mode spectrum of the lowest-order single-material fiber mode in the region of its support slab with $t/d = 0.32$. The short vertical broken line represents the mode content of the fiber with $t = 0$.

to produce the field distortions in Figs. 4 and 6. The rise of the mode amplitudes for modes with $\nu > 12$ is not truly representative of the actual mode content. If the mode number is varied in the numerical approximation, it is found that, near the last mode, $\nu = N$, the mode amplitudes always tend to assume increased values. The appearance of the mode spectrum is thus somewhat dependent on the total number N of modes used in the series expansion. However, the distribution of lower-order modes was found to be very similar for $N = 16$ compared to the spectrum shown in Fig. 7 for $N = 20$. Only the highest-order modes appear with different amplitudes. When $N = 12$ was used, a different mode spectrum and an implausible field distribution was obtained, indicating insufficient accuracy.

Figure 8 shows the mode content of the field in the support slab. The mode amplitudes are much smaller, since much less power is carried outside the fiber core. The lowest-order TE mode is most prominent. The short broken line at $\nu = 1$ represents the much weaker contribution of the rectangular waveguide ($t = 0$). For our model, the lowest-order TM mode outside the core, $\nu = 2$, contributes slightly to the mode field of the rectangular dielectric waveguide, but its amplitude is too small to be visible on the scale of this figure. It is interesting that the field of the single-material fiber in the region of the support slab is represented to a very good approximation by the lowest-order TE mode of the support slab. The modes $\nu \leqq 8$ are guided slab modes with imaginary values of ρ_ν; modes with $\nu > 8$ have real valued parameters ρ_ν corresponding to the radiation modes of open slabs.

Figure 9 is a plot of the direction of the electric field vector in the vicinity of the corner of the dielectric material at $x/b = 1$, $y/d = 0.32$. Far from this corner, the field is horizontally polarized. It is remarkable how little distortion is evident near the dielectric discontinuity. There is no peak in the field intensity at the sharp dielectric corner, and the field direction is likewise almost unperturbed.

Finally, we present solutions of the eigenvalue equation (system determinant = 0). Instead of plotting values for the propagation constant β, we present values for the relative effective width of the fiber core. If the core boundary at $x = b$ were a metal wall we would have

$$k_x = \frac{\pi}{2b} \tag{39}$$

for the lowest-order mode. The actual values of k_x deviate from the value (39) partly because the dielectric discontinuity at $x = b$ is not an electrical short circuit, and also because the field penetrates some

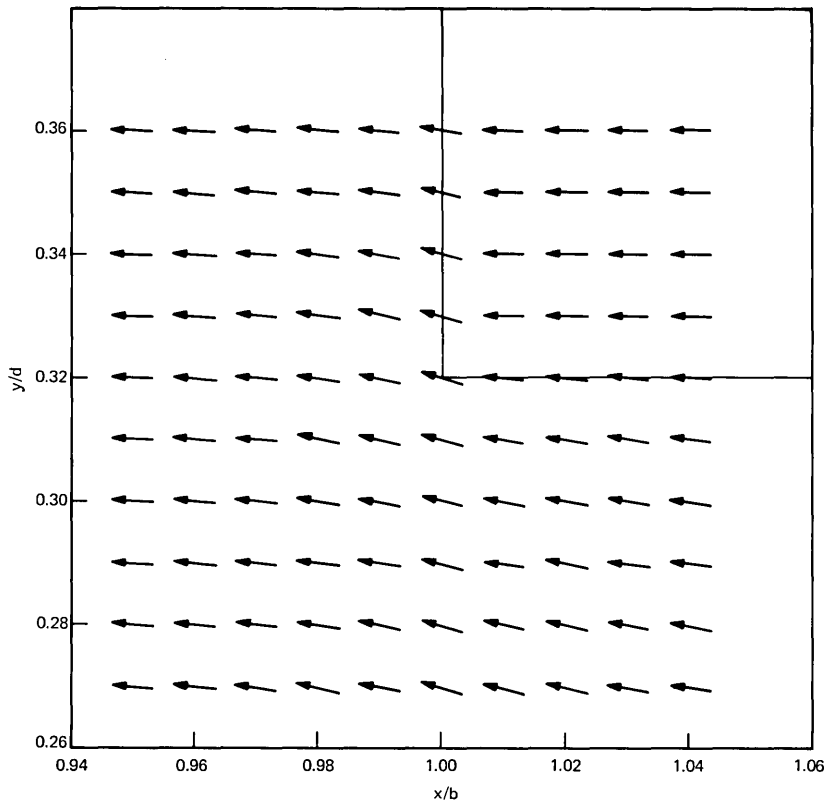


Fig. 9—Short arrows indicate the direction of the electric field vector of the lowest-order mode of the single-material fiber, with $t/d = 0.32$ near the corner of the dielectric material where the support slab is attached to the core.

distance into the support slab. We use the actual value of k_x to define an effective core width

$$b' = \frac{\pi}{2k_x}. \tag{40}$$

The value of k_x is obtained from the solution β of the eigenvalue equation with the help of (5) and (9)

$$k_x = \left[n^2k^2 - \beta^2 - \left(\frac{\pi}{2d}\right)^2\right]^{\frac{1}{2}}. \tag{41}$$

The solid line in Fig. 10 represents the relative effective width of the core for the lowest-order mode of the single-material fiber with the

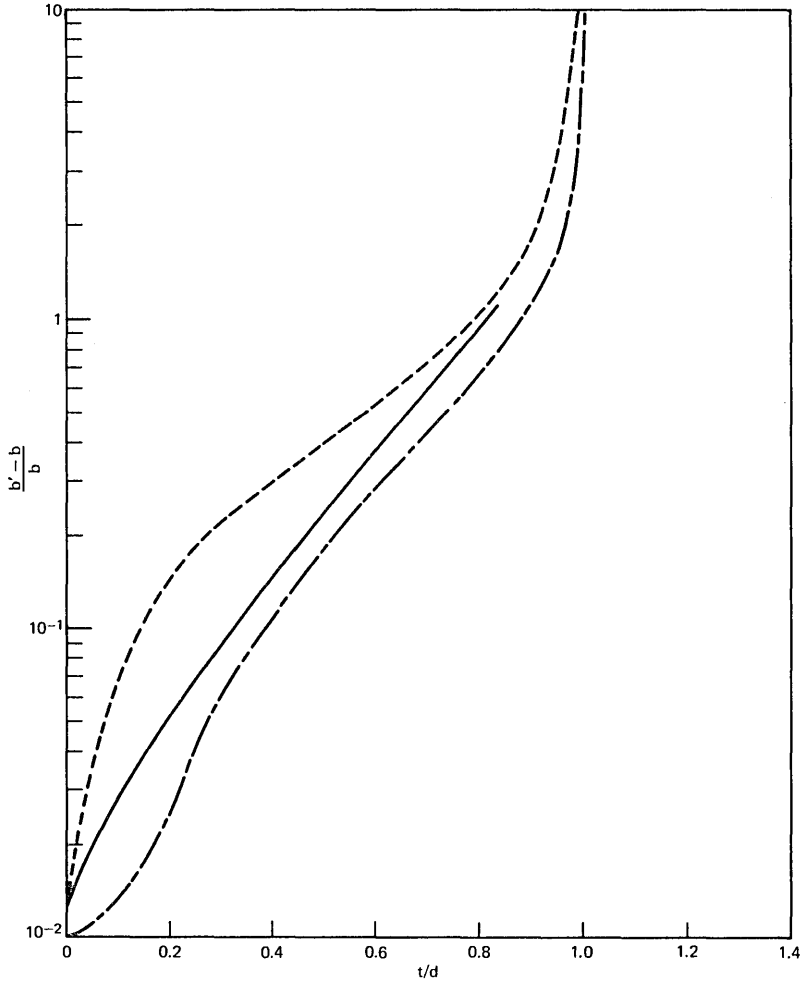


Fig. 10—Relative effective width of the core of the single-material fiber as a function of the relative thickness of its support slab t/d for the lowest-order mode. The solid line is the result of the numerical solution of the complete theory; the broken line was calculated from the solution of the approximate eigenvalue equation (34); and the dash-dot line is the result of Marcatili's theory.

dimensions stated in (38). This mode does not suffer a cutoff. It can propagate without power outflow into the support slab for arbitrarily small values of $1 - (t/d)$. The computer program had difficulties solving the eigenvalue problem for $t/d > 0.8$; thus, the solid curve is not continued beyond this point. The nonzero value of $(b' - b)/b$ at

$t/d = 0$ represents the field penetration of the rectangular waveguide mode outside the dielectric core.

The upper dotted line of Fig. 10 is the result of solving the approximate eigenvalue equation (34). For t/d near zero and near unity, the approximation is excellent. The departure of the approximation in the middle of the range is not surprising when we look at Fig. 4. The approximate solution uses the field distribution represented by the dotted line of Fig. 4, which is clearly a poor approximation of the actual field distribution. In fact, it is surprising how good the approximate solution for $(b' - b)/b$ is, even in this case. Even though the solid curve does not extend past $t/d = 0.8$, we can trust the dotted curve in this region, since the actual field distribution becomes very close to the approximate distribution. This is evident from a comparison of the solid and broken curves of Fig. 6.

The large error in the approximation in Fig. 10 causes only a very slight error for β. For $t/d = 0.32$, we obtain from the solid curve of Fig. 10 $(b' - b)/b = 0.1$ corresponding to $k_x\lambda = 0.2856$ or $\beta\lambda = 9.41043$. From the broken curve we obtain $(b' - b)/b = 0.24$, $k_x\lambda = 0.2534$ or $\beta\lambda = 9.41135$. The relative error in the β value is thus only $\Delta\beta/\beta = 0.01$ percent.

The dash-dot curve shown in Fig. 10 is a plot of eq. (15) of Ref. 1. This curve was plotted by using the following identification of the symbols in Ref. 1 with our symbols: $T = 2t$, $W = 2b$, and $H = 2d$. The dash-dot curve of Fig. 10 shows clearly how remarkably accurate Marcatili's approximate theory describes the effective width and hence the propagation constant of the single-material fiber mode. His approximation deviates more from the "exact" solution (given by the solid curve) near $t/d = 0$ and $t/d = 1$ than does the dotted curve. The disagreement near $t/d = 0$ is caused by assuming that the field must vanish at the boundary $x = b$ of the rectangular waveguide.

V. CONCLUSIONS

We have studied the properties of the lowest-order mode of the single-material fiber using a model that departs from the actual fiber by the presence of two perfectly conducting planes shown in Fig. 2. Horizontally polarized modes are not appreciably distorted by the presence of these planes. In particular, we are confident that the influence of the support slab on the field distribution and propagation constant of the single-material fiber mode is represented very accurately by this model. The agreement of the model with metallic planes and

the free single-material fiber becomes better for fibers with large values of d/λ.

By representing the field of the single-material fiber as a superposition of the modes of the dielectric slabs in the core region and in the region of the supports, we find solutions by matching the boundary conditions in the plane $x = b$ at a finite number of points. We find that matching along 10 points in the range $0 < y/d < 1$ (requiring 20 modes in each region of the guide) provides satisfactory accuracy.

This study shows that the field of the single-material fiber in the vicinity of the edge, at $x = b$, departs considerably from the field distribution that would result for $t = 0$. However, for very narrow as well as very wide support slabs, a simple approximation using only the two lowest-order modes of the series expansion yields satisfactory results. Our theory thus serves the purpose of clarifying the range of applicability of approximate descriptions[1,2] of the single-material fiber and of inspiring confidence in the validity of such approximations.

In particular, it is our aim to show that Marcatili's approximate theory of the single-material fiber is indeed justified and yields very good results compared to our more precise treatment.

VI. ACKNOWLEDGMENT

The progress of this work was stimulated and influenced by a number of illuminating discussions with E. A. J. Marcatili.

APPENDIX

It is claimed that the analysis presented in this paper is an almost exact description of the single-material fiber, and yet the structure that is analyzed differs from the actual single-material fiber by the presence of the perfect metallic conductors attached to the fiber core, as shown in Fig. 2. In defense of this procedure, two remarks may be made here.

The performance of the single-material fiber is dominated not by the dielectric-air interface on the two sides at $y = \pm d$ of the fiber core but by the presence of the attached support slabs. The electromagnetic fields of the single-material fiber modes extend much further into the support slabs than they do into the air space outside the core, as shown in Figs. 3 through 6. The dielectric-air boundary acts almost like an electrical short circuit, so that the presence of actual short circuits at the dielectric-air interface at $y = \pm d$ has a very slight effect. In particular, it is the radiation of power into the support slabs rather than into the air space outside the core that signals the cutoff of the guided modes. This behavior is described correctly by our analysis.

It is easy to estimate the field penetration into the air space above and below the core in the absence of the perfect conductors. The field outside the fiber is described by the functional dependence $\exp(-\gamma y)$. The decay parameter γ is defined as [see eqs. (1.2–14) and (1.3–44), Ref. 5]:

$$\gamma = \left[(n^2 - 1)k^2 - \left(\frac{\pi}{d}\right)^2 \right]^{\frac{1}{2}}. \tag{42}$$

With the numbers used in the numerical example, we obtain $\gamma\lambda = 7$. This means that, at a distance of $\lambda/7$ from the air-dielectric interface, the field has decayed to $1/e^2$ (or 14 percent) of its power density at the interface. Instead of having an effective electric short circuit at this distance, the presence of the metallic planes moves the short circuit a relative distance of 1.4 percent (in terms of the fiber diameter) nearer to the fiber core. This small change of the electrical width of the core has only a very slight effect on the field penetration into the slabs, which is the most interesting feature of the single-material fiber. Furthermore, this change can be taken into account by allowing the value $2d$ of the modified fiber to be 3 percent larger than that of the actual fiber.

The cutoff condition of the modes follows from the eigenvalue equation, which is the condition for the vanishing determinant of the equation system resulting from the continuity requirements for the tangential field components. At cutoff, the propagation constant β ceases to have real solutions, but becomes complex. No analytical expression can be given for the cutoff point. Its determination from the numerical analysis is difficult. In this respect, the approximate theory proves to be more powerful, since it is able to estimate the cutoff point.

REFERENCES

1. E. A. J. Marcatili, "Slab-Coupled Waveguides," B.S.T.J., *53*, No. 4 (April 1974), pp. 645–674.
2. P. Kaiser, E. A. J. Marcatili, and S. E. Miller, "A New Optical Fiber," B.S.T.J., *52*, No. 2 (February 1973), pp. 265–269.
3. P. Kaiser and H. W. Astle, "Low-Loss Single Material Fibers Made from Pure Fused Silica," B.S.T.J., *53*, No. 6 (July-August 1974), pp. 1021–1039.
4. D. Marcuse, *Light Transmission Optics*, New York: Van Nostrand Reinhold Co., 1972.
5. D. Marcuse, *Theory of Dielectric Optical Waveguides*, New York: Academic Press, 1974.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

Theory of the Single-Material, Helicoidal Fiber

By J. A. ARNAUD

(Manuscript received March 29, 1974)

The theory of propagation in a new single-material, single-mode, optical fiber is given. The modes are of the whispering-gallery type, with the propagation taking place along helicoidal paths close to the boundary of a cylindrical dielectric rod. The beams are confined in the azimuthal direction in helicoidal ridges. It is shown that single-mode, low-loss operation is possible if the helix period is of the order of the rod cross-section area divided by the wavelength and the ridge area is of the order of 1 percent of the rod cross-section area for two channels. The rod is supported by helicoidal wings that play a role in the mode-selection mechanism.

I. INTRODUCTION

The best-known single-mode optical fiber is the clad fiber. If the difference in refractive index between core and cladding is small, single-mode propagation can be achieved for core diameters that are large compared with the wavelength. It is, however, desirable to use just one material, such as quartz, that exhibits low impurity and scattering losses. In a previous work,[1] we indicated that single-mode propagation could be achieved in a single-material configuration that we called a "helicoidal fiber." Figure 1 represents a more recent version of this type of fiber.

To explain the mechanism of operation, let us consider first a cylindrical dielectric rod with radius $a = \rho$. The refractive index of the rod is perhaps $n = 1.45$ (quartz), and the surrounding medium is air. Waves are guided along the rod boundary as shown in Fig. 2a. These so-called "whispering-gallery modes" can be represented by rays repeatedly reflected from the boundary because of total reflection. In the interior of the rod, the modes are described by Bessel functions $J_\nu(kr) \times \exp(i\nu\phi)$, where ν is a large integer and kr is a large number of the order of ν. Because kr and ν are both large and comparable to one

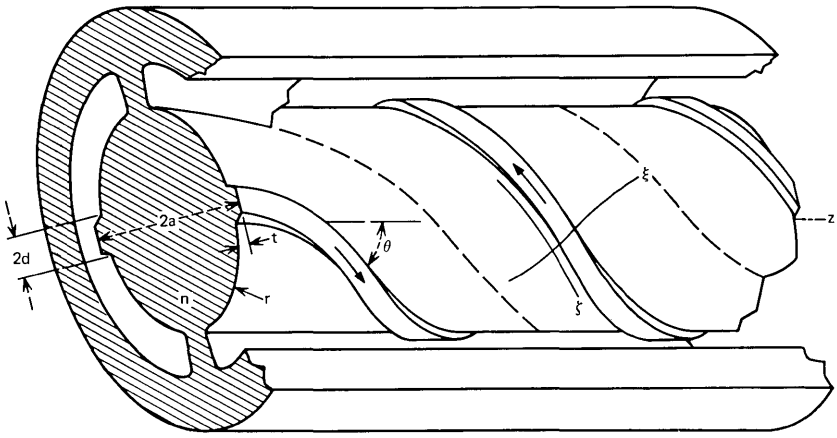


Fig. 1—Open view of the single-material, helicoidal fiber for two optical channels. The two optical beams propagate in the two ridges shown, with areas $t \times 2d$. The helicoidal motion is essential to maintain confinement. High-order modes are not confined. They radiate away to the envelope through the wings (only part of one is shown). The ratio period/diameter is much larger than that shown in the figure.

another, the Bessel functions can be approximated by Airy functions. The field is oscillatory from the rod boundary down to a slightly smaller radius r_c called the caustic (or turning point) radius. For radii smaller than r_c, the field decays exponentially. Thus, the field of whispering-gallery modes clings tightly to the rod boundary. The distance between the caustic and the boundary, which defines in some sense the "thickness" of the mode, is for the fundamental mode of the order of $(\lambda^2 a)^{\frac{1}{3}}$, with λ the wavelength in the medium and a the rod radius. For example, if $\lambda = 1$ μm and a is equal to 8 mm, the fundamental mode thickness is of the order of 20 μm. It increases with the mode number m, approximating as $m^{\frac{1}{3}}$. As m increases, the phase velocity increases too.

These whispering-gallery modes can be generalized to take into account a motion along the rod axis z. The combined rotation and axial motion results in a helicoidal path that can be understood from simple ray-optics considerations. The only significant difference from the previous case is that the radius a of the rod should be replaced, in the expression for the mode thickness, by the helix radius of curvature, $\rho = a/\sin^2\theta$, where θ denotes the angle that the helix makes with the rod axis. For example, if a is 80 μm and $\theta = 0.1$ radian, the mode thickness is the same as in the previous example, where a was assumed to be 8000 μm. If θ is equal to zero, there is of course no confinement

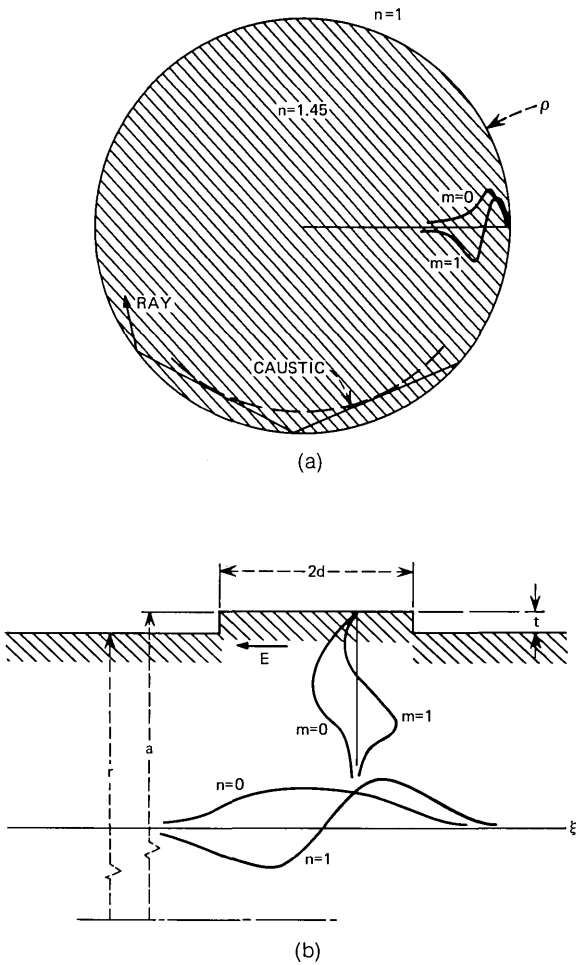


Fig. 2—(a) Whispering-gallery modes clinging to a circular boundary with effective radius $\rho\,(=a/\sin^2\theta)$. The field ($m = 0, 1, \cdots$) is described exactly by Bessel functions and approximately by Airy functions. (b) Cross section in a local r, ξ plane.

near the rod boundary. Observation of helicoidal rays in optics has been reported.[2]

Let us now assume that we have selected a convenient value for θ, perhaps $\theta = 2.5°$, and that we wish to define one or more channels in the azimuthal direction. This can be achieved with helicoidal separators,[1] or ridges, as shown in Fig. 1, that follow the path of the desired

whispering-gallery modes. It is clear, intuitively, that the optical power will tend to remain in the ridges. As the mode number increases, either in the azimuthal or radial direction, the modes occupy larger and larger volumes and eventually "spill out" of the ridges. Because there is maximum confinement in the ridge when $\theta = \pi/2$ and no confinement at all when $\theta = 0$, it is plausible that only a single mode remains confined in the ridge for a proper choice of θ. The higher-order modes radiate away from the ridge along the boundary. They can be absorbed easily without degrading the fundamental mode. In this paper, we justify the above intuitive arguments and show that strong discrimination against unwanted modes can indeed be obtained.

The first step of the calculation is to obtain the propagation constants of whispering-gallery modes in circular cylinders in a convenient form. This is done in Section II. In Section III, we investigate the case of helicoidal boundaries in the local mode approximation and obtain the design parameters. In Section IV, the case of small ridges is investigated with the help of a new perturbation method.

The single-material helicoidal fiber discussed in this paper can be compared to the single-material ridge guide recently demonstrated[3] and analyzed.[4,5] These two single-material fibers have features in common. The mode-selection mechanism rests on similar general principles. It can be ascribed to a coupling between ridges carrying trapped modes and two-dimensional substrates carrying radiation modes.[6] In the case of the ridge guide, the slab constitutes the two-dimensional substrate needed to ensure single-mode propagation. In the case of the helicoidal fiber, the dielectric rod itself can be considered a two-dimensional mode sink because the whispering-gallery modes that it guides have a restricted thickness in the radial direction, as we have discussed before. In both cases, a good discrimination against high-order modes should in principle be obtained by increasing the distance between the absorbing elements and the ridges, because these elements are coupled through the radiation field rather than through evanescent waves.

The theory given in this paper is applicable to purely metallic helicoidal waveguides as well as to dielectric waveguides. The metallic helicoidal waveguide is attractive as a low-loss, multichannel, single-mode system for long-distance microwave communication. It can be compared to the groove guide[7] shown in Fig. 4c. The metallic helicoidal waveguide has the advantage that TEM modes are absent. In the groove guide, any lack of symmetry between the two plates introduces a large loss through coupling to the (slower) TEM modes. This is, in

fact, also the reason for the superiority of the dielectric ridge guide[4] over the metallic groove guide.[7]

II. PROPAGATION OF WHISPERING-GALLERY MODES IN CYLINDRICAL SURFACES

Let us consider a circular dielectric cylinder with radius a. For E modes, the ϕ and z components of the electric field have the form

$$E(r, \phi, z) = J_\nu(ur/a) \exp [i(\nu\phi + k_z z]], \tag{1}$$

where

$$u^2 \equiv (k^2 - k_z^2)a^2. \tag{2}$$

Assuming that the discontinuity in refractive index is sufficiently large, a condition well satisfied for quartz rods in air, the boundary condition at $r = a$ is

$$J_\nu(u) = 0, \tag{3}$$

because, for the type of mode considered, the field tends to vanish at the boundary. The zeros of J_ν are denoted $u_m(\nu)$, $m = 0, 1, 2, \cdots$.

We introduce new coordinates ξ, ζ in place of y, z (see Fig. 1)

$$\begin{aligned} \xi &= r \cos\theta\,\phi - \sin\theta\, z, \\ \zeta &\equiv \cos\theta\, z + r \sin\theta\,\phi, \end{aligned} \tag{4}$$

where

$$\theta \equiv \tan^{-1}(2\pi r/p), \tag{5}$$

the quantity p, for "period," being for the moment an arbitrary constant. The wave numbers Γ_ξ, Γ_ζ in the new coordinate system are related to ν, k_z by

$$\nu = r \cos\theta\,\Gamma_\xi + r \sin\theta\,\Gamma_\zeta, \tag{6a}$$

$$k_z = -\sin\theta\,\Gamma_\xi + \cos\theta\,\Gamma_\zeta. \tag{6b}$$

They are such that

$$\Gamma_\xi \xi + \Gamma_\zeta \zeta \equiv \nu\phi + k_z z. \tag{6c}$$

The characteristic equations, (2) and (3), are now written, using eqs. (6),

$$(-\sin\theta\,\Gamma_\xi + \cos\theta\,\Gamma_\zeta)^2 a^2 + u_m^2(r \cos\theta\,\Gamma_\xi + r \sin\theta\,\Gamma_\zeta) = k^2 a^2. \tag{7}$$

Equation (7) provides us with the desired relation between Γ_ζ and Γ_ξ. We wish to simplify this relation. Because we are interested in whispering-gallery modes corresponding to large values of ν, we can use the

Table I — Values of b parameter in eq. (10)

m	b_m	$[3\pi(2m + 3/4)/2\sqrt{4}]^{\frac{1}{3}}$ (J.W.K.B.)
0	1.85575	1.841
1	3.24461	3.239
2	4.38167	4.379
3	5.387	5.385

following approximation for $u_m(\nu)$[8]

$$u_m(\nu) = \nu + b_m\nu^{\frac{1}{3}}, \tag{8}$$

where b_m is given in Table I. In the second column in Table I, the J.W.K.B. approximation for b_m obtained from simple ray optics considerations is given. As we can see, the error does not exceed 1 percent even for small m.

We note further that a is not very different from r. Thus, we set $a = r + t$, $t \ll r$. Because we are considering waves that do not depart very much from the reference helicoidal path, Γ_ζ is very close to k, and the transverse wave number Γ_ξ is small compared with Γ_ζ. Neglecting products of small quantities, (7) becomes

$$\Gamma^2 \equiv \Gamma_\zeta^2 + \Gamma_\xi^2 = k^2[1 + 2t/\rho - 2b_m(k\rho)^{-\frac{2}{3}}], \tag{9}$$

where we have introduced the reference helix curvature $\rho = a/\sin^2\theta$. The term $2t/\rho$ expresses the fact that, at the reference radius r, the phase velocity is smaller than at the boundary with radius a. The term $2b_m(k\rho)^{-\frac{2}{3}}$ results from the radial variation of the field. The larger the radial mode number m, the smaller the tangential wave number Γ. Note that the system is approximately isotropic.

III. HELICOIDAL BOUNDARY

In the previous section we have assumed that the boundary is a circular cylinder with radius a. We now assume that a is a function of ξ, but that it remains independent of ζ. By letting a vary with ξ, we generate a helicoidal surface. Azimuthal confinement of the whispering-gallery beams can be expected for various well-shaped profiles $a(\xi)$. For simplicity, we assume here that $a(\xi) = a$, a constant, for $-d < \xi < d$, and $a = r$, where r denotes the reference radius, anywhere else in the period. A slightly tapered transition region is assumed. Mode mixing can therefore be neglected in the evaluation of the propagation constants of the modes $m = 0, 1, 2, \cdots$. A small amount of mode mix-

ing is nevertheless needed for this mode-selection mechanism to operate.[4–7]

From (9), the wave numbers for the fundamental mode ($m = 0$) and first-order mode ($m = 1$) in the ridge (unprimed number) and outside the ridge (primed number) are, respectively,

$$\Gamma_0^2 = k^2[1 + 2t/\rho - 2b_0(k\rho)^{-\frac{2}{3}}], \tag{10a}$$

$$\Gamma_{0'}^2 = k^2[1 - 2b_0(k\rho)^{-\frac{2}{3}}], \tag{10b}$$

$$\Gamma_1^2 = k^2[1 + 2t/\rho - 2b_1(k\rho)^{-\frac{2}{3}}], \tag{10c}$$

$$\Gamma_{1'}^2 = k^2[1 - 2b_1(k\rho)^{-\frac{2}{3}}]. \tag{10d}$$

The axial wave numbers Γ_{z0} and Γ_{z1} for the two modes 0 and 1 are now obtained using the standard dielectric slab theory. If we normalize the axial wave number Γ_{z0} by defining

$$K^2 \equiv (\Gamma_{z0}^2 - \Gamma_{0'}^2)/(\Gamma_0^2 - \Gamma_{0'}^2) \tag{11}$$

and introduce the V parameter

$$V^2 \equiv (\Gamma_0^2 - \Gamma_{0'}^2)d^2, \tag{12}$$

we have, for the modes $m = 0$, an explicit relation between V and K,

$$V = \{\tan^{-1}[K(1 - K^2)^{-\frac{1}{2}}] + n\pi/2\}(1 - K^2)^{-\frac{1}{2}}, \tag{13}$$

where $n = 0, 2, 4, \cdots$ correspond to even modes and $n = 1, 3, \cdots$ to odd modes.* A similar relation holds for the modes $m = 1$, $n = 0, 1, 2, \cdots$.

The axial wave numbers Γ_{zmn} of these various modes m,n are plotted in Fig. 3 as functions of the ridge width $2d$, for $\lambda = 1$ μm, $n = 1.45$ (quartz), a rod radius $a = 50$ μm, and a ridge height $t = 3.5$ μm. We have chosen $\theta = 2.5°$, corresponding to a helix radius of curvature $\rho = 25$ mm. Modes whose axial wave number is less than $\Gamma_{0'}$ (the wave number of the fundamental mode outside the ridge) suffer radiation losses.

This figure clearly shows that only one mode ($m = 0$, $n = 0$) is free of radiation loss if $2d$ is less than 14 μm. For $2d = 14$ μm, the field of the fundamental mode decays in azimuth by a factor of $1/e$ at a distance

$$\xi_0 = (\Gamma_{z0}^2 - \Gamma_{0'}^2)^{-\frac{1}{2}} = 6.3\ \mu\text{m} \tag{14}$$

on either side of the ridge. For two channels, the "wings" holding the

* The mode number n should not be confused with the refractive index n.

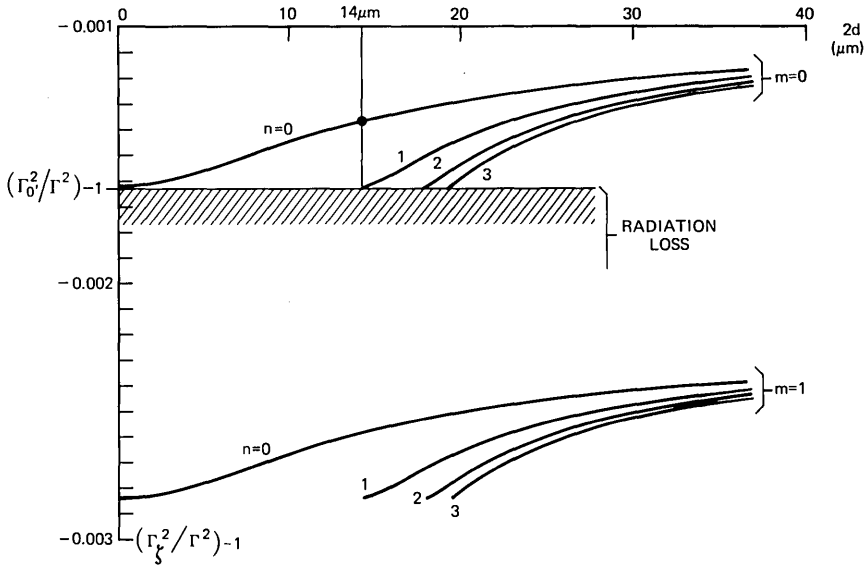


Fig. 3—Axial wave numbers $\Gamma_{\xi mn}$ for various m and n modes (r and ξ coordinates). Radiation loss is suffered whenever $\Gamma_\xi < \Gamma_{0'}$. The radiation zone is shaded. This figure shows that single-mode propagation is possible if the ridge width $2d$ is less than 14 μm (for $\lambda_0 = 1$ μm, $n = 1.45$, $t = 3.5$ μm, $a = 50$ μm, and $\theta = 2.5°$).

rod in Fig. 1 are located at a distance $\pi a/2 = 80$ μm from the corrugation. At that distance, the field has decayed by a factor of more than 10^5. The fundamental mode therefore suffers negligible radiation loss (bending losses are not considered here). If we set the condition that ξ_0 be $\frac{1}{10}$ of $\pi a/4$, we obtain the approximate condition $\theta \approx \lambda/a$ for single-mode low-loss operation. More detailed relations are given at the end of Section IV.

The local mode theory used in this section is expected to be applicable when the ridge width $2d$ is large compared to the ridge height, t, and large compared to the wavelength, λ. In the next section, we find that a perturbation method applicable to small ridge areas ($2\ td$) leads to almost identical conclusions.

IV. LINE PERTURBATION OF SURFACE WAVES

We give a general theory of the trapping of surface waves by rods of small cross section. This theory is then applied to helicoidal ridges of small cross section.

Let us consider an isotropic surface, perhaps an inductive surface, supporting a plane wave with wave number k. Let us introduce a di-

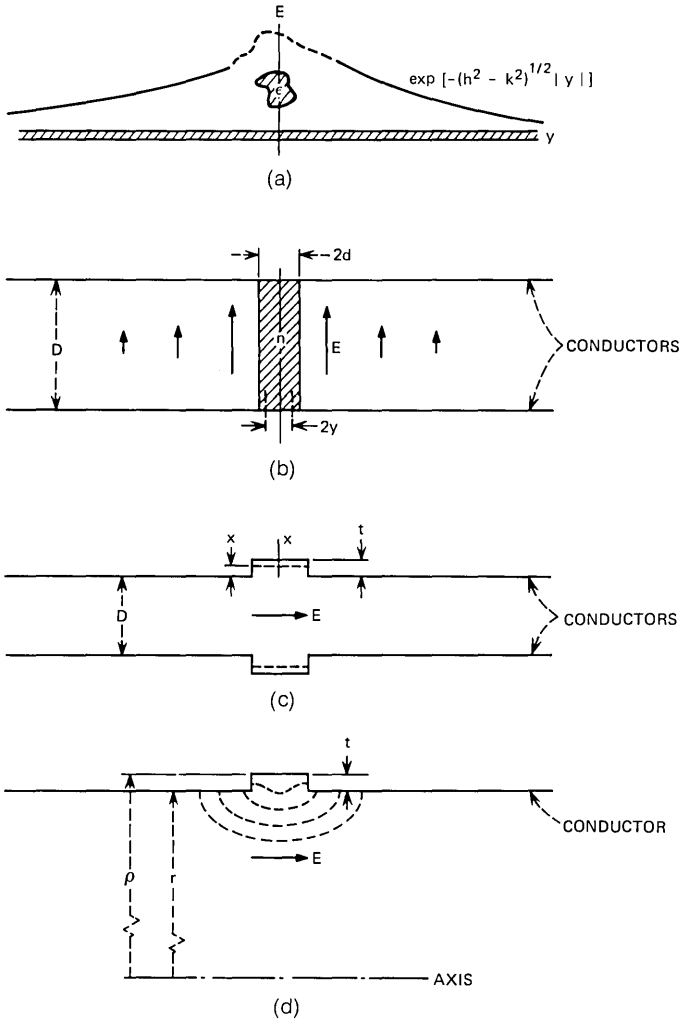


Fig. 4—Various methods of introducing a line perturbation on a surface wave. (a) Reactive (e.g., corrugated) surface perturbed by a dielectric rod. (b) TEM waves perturbed by a dielectric slab. (c) "Groove guide."[7] (d) Ridge guide considered in this paper. The torsion of the helicoidal motion is not essential. The radius of curvature ρ of the helix is the important parameter that determines mode selection.

electric rod of very small cross section parallel to this surface, some distance away from it (Fig. 4a). Because the power carried by the plane wave is infinite, a straightforward application of the conventional perturbation method does not give any meaningful result. Therefore,

we shall proceed the other way around. We start from the perturbed state and assume that we know the propagation constant $h > k$. The wave is in that case confined transversely with a decay rate $(h^2 - k^2)^{\frac{1}{2}}$. We now "peel off" the rod and evaluate the successive perturbations until the perturbation resulting from the rod vanishes. By specifying that $h \to k$ in that limit, the initial value of h is obtained.

For simplicity, this method is first explained for the case where $\epsilon \approx \epsilon_0$, the perturbed field being of the order of the unperturbed field. Let σ be a parameter such that $\sigma = 0$ corresponds to the absence of the rod and $\sigma = 1$ corresponds to the presence of the rod. Furthermore, σ is so chosen that, in the perturbation formula

$$dh/d\sigma = \alpha(h^2 - k^2)^{\frac{1}{2}}, \tag{15}$$

α is a constant. This can be done because we have factored out a term $(h^2 - k^2)^{\frac{1}{2}}$ inversely proportional to the power carried by the mode. (The distortion of the field in the close neighborhood of the rod does not contribute significantly to the total power, because of the large transverse extent of the field.) The ratio σ is essentially the ratio of the present cross-section area of the perturbating rod to its original cross-section area. Integrating eq. (15) from $\sigma = 1$ to $\sigma = 0$, we obtain

$$\int_k^h (h^2 - k^2)^{-\frac{1}{2}} dh = \alpha \tag{16a}$$

or

$$h = k \cos \alpha \approx k(1 - \alpha^2/2). \tag{16b}$$

To clarify the significance of this result, let it be applied to a configuration where the exact solution is known. Consider two parallel perfectly conductive plates with spacing D carrying TEM modes, as shown in Fig. 4b. If we introduce a dielectric slab with $\epsilon \approx \epsilon_0$ and width $2d$, we obtain the so-called "H-guide" configuration proposed by Tischer.[9] (Note, however, that we consider here the H modes rather than the low-loss modes.) The parameter $\sigma = y/d$, where y is shown in Fig. 4b, clearly satisfies the requirements set up above. The conventional perturbation formula (see, for example, Ref. 5, Part II, eq. (21), with $\mathbf{E}_p \approx \mathbf{E}$, $\mathbf{H}_p \approx \mathbf{H}$, $\mathbf{E}^\dagger = \mathbf{E}$, $\mathbf{H}^\dagger = -\mathbf{H}$) is

$$\Delta h = \tfrac{1}{2}\omega \int (\epsilon - \epsilon_0) E^2 dS \Big/ \int \mathbf{E} \times \mathbf{H} \cdot \mathbf{d}S. \tag{17}$$

For our case, we obtain, taking into account the $\exp[-(h^2 - k^2)^{\frac{1}{2}}|y|]$

dependence of the field on y,

$$dh/d\sigma = (\mu_0/\epsilon_0)^{\frac{1}{2}}(h^2 - k^2)^{\frac{1}{2}}(\epsilon - \epsilon_0)\omega d \equiv \alpha(h^2 - k^2)^{\frac{1}{2}}. \tag{18}$$

Thus the constant α is, from (18),

$$\alpha = (n^2 - 1)kd, \tag{19}$$

where $\epsilon/\epsilon_0 \equiv n^2$. Application of (16b) now gives the perturbed wave number h

$$h = k[1 + \tfrac{1}{2}(n^2 - 1)^2k^2d^2]. \tag{20}$$

The exact solution to the problem, for small $(n^2 - 1)kd$, is well known [see, for example, Ref. 5, Part II, footnote after eq. (11)]. We have, with the approximation $\tan[(n^2 - 1)^{\frac{1}{2}}kd)] \approx (n^2 - 1)^{\frac{1}{2}}kd$,

$$h^2 - k^2 = (n^2 - 1)^2k^4d^2. \tag{21}$$

Equation (21) coincides with our perturbation result, (20), because $h \approx k$. Having satisfied ourselves with the validity of our perturbation technique, we apply it to a small wall perturbation. We assume that the case of quartz in air is the same as the case of a metallic boundary, except for the wavelength λ/n replacing λ.

For a wall perturbation with cross-section area s, the perturbation formula is

$$\Delta h = \tfrac{1}{2}\omega\mu_0 H^2 s \Big/ \int \mathbf{E} \times \mathbf{H} \cdot \mathbf{d}S, \tag{22}$$

if the electric field is equal to zero. Note that h increases if the volume is increased, e.g., if we introduce corrugations in the wall.

For H waves uniform along the y-axis (see Figs. 4c and 4d) ($E_y \equiv E$), we have, from Maxwell's equations,

$$H_x = -(k/\omega\mu_0)E, \tag{23a}$$

$$H_z = (i\omega\mu_0)^{-1}\partial E/\partial x. \tag{23b}$$

Substituting in (22) we obtain the perturbation

$$\Delta h = (s/2k)(\partial E/\partial x)^2(h^2 - k^2)^{\frac{1}{2}} \Big/ \int E^2 dx. \tag{24}$$

Defining σ as x/t (see Figs. 4c or 4d), the constant α defined in (15) is found to be

$$\alpha = (s/2k)(\partial E/\partial x)^2 \Big/ \int E^2 dx, \tag{25}$$

the derivative being evaluated at the first zero of $E(x)$ for the mode $m = 0$, the second for the mode $m = 1$, and so on.

For whispering-gallery modes, we have

$$E = \mathrm{Ai}(t), \tag{26a}$$

where Ai( ) denotes the Airy function and

$$t \equiv (2k^2/\rho)^{\frac{1}{3}}x - (2k^2/\rho)^{-\frac{2}{3}}(k^2 - h^2), \tag{26b}$$

ρ being the boundary radius. Substituting in eq. (25), we obtain

$$\alpha = 2f_m ktd/\rho, \tag{27}$$

where f_m is a numerical factor

$$f_m = (d\mathrm{Ai}/dt)^2_{t=t_m} \Big/ \int_{-\infty}^{t_m} \mathrm{Ai}^2(t)dt, \tag{28}$$

$t_0, t_1, \cdots, t_m, \cdots$ being the zeros of $\mathrm{Ai}(t)$. By numerical integration, we find

$$f_0 = 0.981\cdots \tag{29a}$$

$$f_1 = 0.955\cdots. \tag{29b}$$

Thus, the change in propagation constant resulting from a wall deformation of area $s \equiv 2td$ is, from (27),

$$h_m - k = \tfrac{1}{2}f_m^2 k^3 s^2/\rho^2. \tag{30}$$

As long as the perturbation is small, h_1 remains smaller than $\Gamma_{0'}$ and only the mode $m = 0$ is free of radiation loss. For a sufficiently large perturbation, however, h_1 may exceed $\Gamma_{0'}$. Then the modes $m = 0$ and $m = 1$ are both free of radiation loss, that is, the system is no longer single-mode. The condition for the system to be single-mode is therefore

$$h_m - \Gamma_{1'} < \Gamma_{0'} - \Gamma_{1'},$$

or

$$f_m^2 k^3 s^2/\rho^2 < 2k(b_0 - b_1)(k\rho)^{-\frac{2}{3}}. \tag{31}$$

$\Gamma_{0'}$, $\Gamma_{1'}$, and the constants b_0, b_1 were defined in Section III. The above condition can be written, using the values in Table I for b_0, b_1,

$$k^2 s < 1.74(k\rho)^{\frac{2}{3}}. \tag{32}$$

If we take the limit $d \to 0$ in the expressions given in Section III, we obtain instead

$$k^2 s < 1.68(k\rho)^{\frac{2}{3}}, \tag{33a}$$

which is very close to the perturbation result, (32). Thus, the local mode approach and the perturbation approach agree closely, not only in form, but also numerically. Because $\rho \approx a/\theta^2$, θ being the angle that the helix makes with the z axis, and $\theta \approx 2\pi a/p$, p being the helix period, condition (33) for single-mode operation can be rewritten

$$s < 0.012(\lambda^2 p^2/a)^{\frac{1}{3}}. \tag{33b}$$

The extent ξ_0 in azimuth of the fundamental mode, defined as the $1/e$ point of the field, $\xi_0 = (h_0^2 - k^2)^{-\frac{1}{2}} \approx (2k)^{-\frac{1}{2}}(h_0 - k)^{-\frac{1}{2}}$, is, from (30) with $m = 0$, given by

$$k\xi_0 \approx \rho/ks. \tag{34a}$$

If we specify that the fundamental mode field has decayed by a factor of 10^5 at the "wings," located, for two channels (see Fig. 1), a distance $\pi a/2$ away from the ridge, we must have $\xi_0 = (\pi a/2)/11.5$. Introducing the helix period p, this condition for the fundamental mode to have small radiation loss can be written

$$s > 0.005(\lambda p/a)^2. \tag{34b}$$

The condition for a single mode to propagate, (33b), and for the fundamental mode to have small radiation losses, (34b), are consistent if

$$p < 5a^2/\lambda. \tag{35}$$

For example, if $a = 50\ \mu$m, $\lambda = (1/1.45)\ \mu$m, according to eq. (35), the helix period, p, must be smaller than 18 mm. A period of 10 mm, for instance, would be quite adequate. Note that for such rather long periods the optical path is not significantly increased by the circular motion. Radiation into free space is negligible, as long as the medium surrounding the ridge is air. For mechanical reasons, however, we may want to use a material with lower index. In that case, radiation into the surrounding medium may be a limitation.

V. CONCLUSION

The single-material helicoidal fiber proposed earlier by the author has been shown to support only one mode, with low radiation loss, provided the following two conditions are satisfied:

Helix period $\approx$ rod cross-section area/wavelength.
Ridge area $\approx$ rod cross-section area/70.

More detailed calculations, similar in spirit to the ones given in Ref. 5, would be necessary to specify the magnitude of the radiation

losses and the exact value of the mode discrimination. The helicoidal fiber, like any single-mode fiber with large mode cross section, may be sensitive to bending losses. The bending loss is therefore another key point that needs to be investigated.

REFERENCES

1. J. A. Arnaud, "Note on the Use of Whispering-Gallery Modes in Communication," unpublished work, September 1971.
2. F. G. Reick, "The Optical Whispering Mode of Polished Cylinders and its Implications in Laser Technology," Appl. Opt., 1965, 4, pp. 1395–1399.
3. P. Kaiser, E. A. J. Marcatili, and S. E. Miller, "A New Optical Fiber," B.S.T.J., *52*, No. 2 (February 1973), pp. 265–269.
4. E. A. J. Marcatili, "Slab-Coupled Waveguides," B.S.T.J., *53*, No. 4 (April 1974), pp. 645–674.
5. J. A. Arnaud, "Transverse Coupling in Fiber Optics—Part II: Coupling to Mode Sinks," B.S.T.J., *53*, No. 4 (April 1974), pp. 675–696; "Part III: Bending Losses," B.S.T.J., *53*, No. 7 (September 1974), pp. 1379–1394.
6. J. A. Arnaud, "Selection of Waveguide Modes by Two-Dimensional Mode Sinks," Topical Meeting on Integrated Optics, New Orleans, Louisiana, January 21–24, 1974, Digest of Tech. Papers, p. WB12.1.
7. F. J. Tischer, "The Groove Guide, a Low-Loss Waveguide for Millimeter Waves," IEEE J. of Microwave Theory and Techniques, *11*, 1963, p. 291. The equivalent resonator problem ($k^2 - k_z^2 \to k^2$) is treated in: L. A. Weinstein, *Open Resonators and Open Waveguides*, Boulder, Colorado: The Golem Press, 1969, p. 87 and Fig. 20b; the mode selection mechanism is discussed by T. Nakahara and N. Kurauchi in *Advances in Microwaves*, L. Young, ed., New York: Academic Press, 1969.
8. *Royal Society Mathematical Tables*, Bessel Functions, Vol. 7, Part 3, Cambridge, England: Cambridge University Press, 1960, p. 18.
9. F. J. Tisher, "The H Guide, a Waveguide for Microwaves," I.R.E. Conv. Record, 1956, Microwaves Inst., Part 5, p. 44.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

A Proposed Multiple-Beam Microwave Antenna for Earth Stations and Satellites

By E. A. OHM

(Manuscript received October 3, 1973)

An offset Cassegrainian antenna with essentially zero aperture blockage is expected to support closely spaced well-isolated beams suitable for earth stations and satellites. Each beam is fed with a separate small-flare-angle corrugated horn and has good area efficiency over a 1.75:1 bandwidth. Each beam also has good cross-polarization properties. The antenna is compact, and the design appears practical for a 4- and 6-GHz earth station, a 20- and 30-GHz earth station, and a 20- and 30-GHz satellite.

I. INTRODUCTION

Satellite communication systems with large capacities can be achieved if the satellites and earth stations are provided with multiple-narrow-beam antennas.[1] The capacity is proportional to the number of satellites, and thus it is important to use as many as practical in the limited orbital space. A moderate number of the resulting closely spaced satellites can be served by a single antenna at each earth-station site if the antenna is patterned after the offset Cassegrainian antenna shown in Fig. 1. This design allows an orderly expansion in communication capacity by the addition of feed horns. Since only one antenna is needed at each site, the design also permits a large saving in earth-station costs. Good multiple-beam performance can be achieved across all up/down pairs of satellite frequency bands, including those well below 10 GHz. At 20 and 30 GHz, a large earth-station antenna with acceptable thermal and wind distortion is hard to achieve. However, with the design outlined here, these problems can be largely overcome because the main reflector and subreflector can be fixed in position, thus allowing a stiffer structure. The steering of each beam is achieved by moving one of the feed horns, resulting in a steerable angle sufficient for tracking near-synchronous satellites.

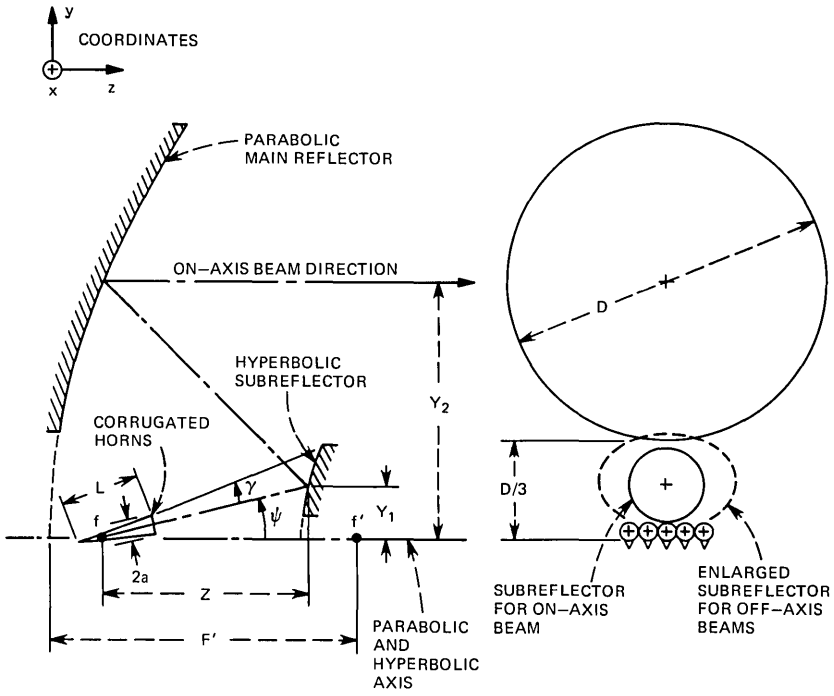


Fig. 1—Geometry of antenna and feed system. The feed horns are scaled for a 3m 20- and 30-GHz satellite antenna. For a 30m 4- and 6-GHz earth-station antenna, L and $2a$ are half as large as shown.

The offset Cassegrain is also appropriate for use aboard a satellite because all beams, including those moderately far off-axis, have high area efficiencies and low side-lobe levels. However, good results on a satellite are restricted to bands well above 10 GHz because the antenna size is limited by the launch vehicle.

It has been previously shown that a multiple-beam antenna can be achieved in a variety of ways,[2–5] where each approach has emphasized one feature desired in a practical antenna. By combining several of these with a corrugated feed horn[6] and an enlarged subreflector, it is possible to achieve a compact antenna with exceptionally good multiple-beam characteristics. In particular, in the offset Cassegrainian antenna shown in Fig. 1:

(*i*) An offset design essentially eliminates beam blockage, thus allowing a significant reduction in side-lobe level.[7] This, in turn, results in higher isolation between beams and a lower antenna noise temperature.

(*ii*) The Cassegrainian feed system is compact and has a large focal-length-to-diameter (F/D) ratio.[8] The large F/D ratio reduces aberrations to an acceptable level, even when a beam is moderately far off-axis.

(*iii*) A corrugated feed horn is essentially a Gaussian-beam launcher[9] and, as such, it can be used to achieve beams with low side-lobe levels. The corresponding feed-horn aperture[6] is small enough to allow the beams to be closely spaced.

(*iv*) An enlarged subreflector, as indicated by the dashed line in Fig. 1, allows the main reflector to be properly illuminated, even when a beam is moderately far off-axis.

These features can be achieved over a wide range of antenna parameters. Using the results developed here, the sample calculations summarized in Table I show that (*i*) the off-axis beam angles are practical, (*ii*) the coma aberration is small, (*iii*) the feed-horn dimensions are reasonable, and (*iv*) the isolations between beams are large.

II. OFF-AXIS DESIGN CRITERIA

Consider a parabolic reflector that is circularly symmetric and illuminated with a feed at its prime focus. If the aperture is large in wavelengths and the prime focal-length-to-diameter, F'/D, ratio is 2 or more, it is well-known that a beam can be scanned over tens of beamwidths by lateral displacement of the feed.[2] A Cassegrainian antenna normally has a secondary focal length F larger than F', and thus a larger F/D ratio.[8] Consequently, a scanned beam can also be obtained by displacing a feed at the secondary focus.[5] For the small off-axis angle reported in Ref. 5 (4 beamwidths $\equiv 0.9°$), the on-axis and off-axis beam characteristics are nearly identical, and the residual differences can be readily explained in terms of an equivalent parabola.[5,8] The equivalent parabola, in turn, has characteristics identical to those of a prime-focus parabola. Consequently, the prime-focus theory[2] can be used to predict the off-axis equivalent-parabola results, and thus the Cassegrainian results. This chain of reasoning assumes that the equivalent-parabola concept is valid for the antenna parameters (F'/D and F/D ratios and off-axis angles) considered here. In support of this assumption, it is of interest to note that the chief off-axis beam parameter of a prime-focus parabola, namely,[2]

$$X' = \frac{N\left(\frac{D}{F'}\right)^2}{1 + 0.02\left(\frac{D}{F'}\right)^2}, \tag{1}$$

where N is the off-axis angle in half-power beamwidths, has a value in Ref. 5 of about 30. Thus, the equivalent-parabola concept is valid for X' values at least through 30. Furthermore, the known results indicate that the region of validity can be extrapolated to X' values well beyond 30. In particular, Ref. 5 shows that the coma lobe, which is the first side lobe aimed toward the on-axis direction, increases very slowly as a function of off-axis beam angle. From Ref. 2, it is also known that an increase in coma-lobe level is a sensitive leading indicator of serious aberration problems, and that X' increases rapidly with coma-lobe level. It follows that X' in Ref. 5 can be much larger than 30 before a larger increase in coma-lobe level signals the onset of serious aberrations. The upper limit of X' should and can be calculated but, in the meantime, some of the results in Table I include an engineering judgment that the equivalent-parabola concept is valid for X' values through 45. Even if the upper limit turns out to be somewhat less, the offset Cassegrain can still support a respectable number of multiple beams, i.e., for $X' = 30$, the number of 1°-spaced beams from the earth-station antenna of Table I is 7 rather than 11.

An important parameter of an off-axis beam is the third-order phase error across the beam at the antenna aperture. This error, $\Delta\phi$, increases the level of the coma lobe.[2] For a symmetrical parabola illuminated with a feed displaced laterally from the prime focus, the peak value of $\Delta\phi$ at the edge of the aperture can be calculated from eq. (12) of Ref. 2. Similarly, when an offset parabola (as in Fig. 1) is illuminated with a feed displaced laterally from the prime focus (in the x direction in Fig. 1), the maximum third-order phase error, $\Delta\phi'$, which occurs at the side edge of the aperture, can be calculated from[10]

$$\Delta\phi' = \frac{2\pi}{32}\frac{F'}{\lambda}\frac{\sin\theta}{(F'/D)^3}\frac{1}{1+(Y_2/2F')^2}, \tag{2}$$

where F' is the prime focal length, θ is the off-axis angle of the beam, D is the diameter of the offset aperture, and Y_2 (see Fig. 1) is the offset height of the aperture. Equation (2) assumes that the feed is also displaced slightly in the longitudinal direction (the $-z$ direction in Fig. 1) to cancel field curvature.

Comparison of eqs. (12) and (13) of Ref. 2 shows that $\Delta\phi'$ is proportional to X'. Noting that $\Delta\phi'$ in (2) is defined in terms of the aperture diameter, D, independently of whether the aperture is centered or offset, it follows that D in eq. (1) should be interpreted in the same way, i.e., it is the diameter of the offset aperture, D, and not the diameter of the aperture of the full parabola (8/3 D in Fig. 1).

If the prime-focus feed illuminating the offset parabola is replaced with a Cassegrainian feed system, as in Fig. 1, and the equivalent-parabola concept is valid, F' in (1) and (2) can be replaced with the Cassegrainian focal length F. In Fig. 1, F is the distance Z times the ratio of centerline-ray heights where they intercept the main and subreflector heights, i.e., $F = Z(Y_2/Y_1)$. For the antenna parameters listed in Table I, the values of $\Delta\phi$ calculated from (2) are substantially less than 90°. For these values, the first side lobe, or coma lobe, is increased in amplitude, but the side lobes which are positioned further out, i.e., those that determine the minimum spacing of well-isolated beams, are virtually unchanged. Accordingly, in the remainder of this paper, it is assumed that $\Delta\phi$ is zero. The corresponding values of X, which are calculated from (1) after replacing F' by F, are found to be 4.5 or less. From the plots given in Ref. 2, the off-axis and on-axis beam characteristics are essentially identical for these values of X.

III. BEAM SPACING

Suppose the amplitude distribution across an unblocked aperture is that of a dominant-mode Gaussian beam, that the amplitude at the edge is truncated at the −15-dB point, and that the phase front is uniform. The envelope of the resulting radiation pattern is shown in Fig. 2. For the offset Cassegrain shown in Fig. 1, the above amplitude and phase distribution can be achieved by placing a corrugated feed horn[6] at the secondary focal point, f. Comparison of Dragone's results[9] with the standard Gaussian-beam equations[11] shows that the radius of the beam, ω, at the −8.686-dB (or $1/e$ amplitude) point, is related to the feed-aperture radius, a, by

$$\omega = 0.647\, a. \tag{3}$$

The comparison also shows that the phase-front radius is equal to the slant length of the feed-horn, L. Using Gaussian-beam equations,[11] the beam parameters in any other region in the feed system can be calculated. One result is that the required feed-horn length, L, can be found from the half-angle, γ, subtended at the focus f by the subreflector, and the illumination taper, T, in dB, at the edge of the subreflector.

$$L = 0.076 \frac{\lambda}{\gamma^2} T_{\text{dB}}. \tag{4}$$

Equation (4) includes the feed-horn design criterion[6]

$$a^2/\lambda L = 1, \tag{5}$$

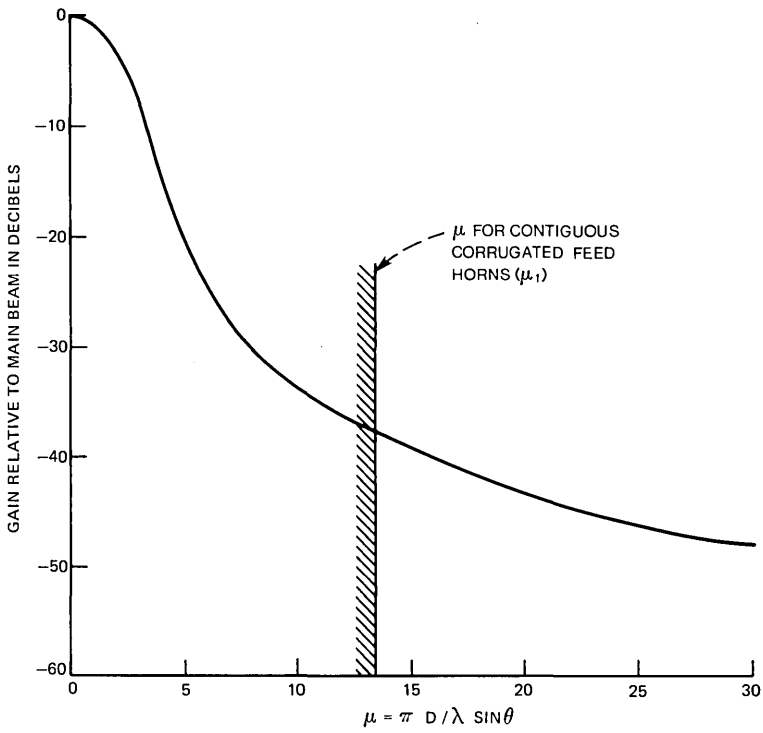


Fig. 2—Estimate of the side-lobe envelope resulting from a Gaussian illumination taper truncated at the −15 dB point, courtesy of T. S. Chu.

where, for a 1.75:1 bandwidth, λ is specified at the low end of the frequency range. Equation (4) is strictly valid only when $\gamma \gg \lambda/D_{\text{sub}}$, where D_{sub} is the diameter of the subreflector. For an equivalent parabola with focal length F,[8] it can be shown that the γ criterion is automatically satisfied when the F/D ratio is less than 5.

The corresponding feed-horn aperture radius, a, is found by solving $a^2/\lambda L = 1$ for a, and substituting L from (4):

$$a = 0.275 \frac{\lambda}{\gamma} \sqrt{T_{\text{dB}}}. \tag{6}$$

Suppose the antenna shown in Fig. 1 has a diameter-to-wavelength ratio, D/λ, in the hundreds, an equivalent focal length, F, and an F/D ratio larger than 2. Then if a second feed-horn is placed adjacent to the on-axis feed, the second beam will be aimed in an off-axis direction,

$\theta_1 = 2a/F$. Inserting a from Eq. (6) and noting[8] that $\gamma = D/2F$,

$$\theta_1 = 1.1 \frac{\lambda}{D} \sqrt{T_{\text{dB}}}. \tag{7}$$

Inserting (7) into the parameter on the abscissa of Fig. 2, the value of u for contiguous corrugated feed horns is

$$u_1 = 3.46\sqrt{T_{\text{dB}}}. \tag{8}$$

For $T = 15$ dB, $u_1 = 13.4$. From Fig. 2, the -3-dB beamwidth is 3.62; thus, u_1 corresponds to $13.4/3.62 = 3.7$ beamwidths. For $u_1 = 13.4$, Fig. 2 shows that the side-lobe envelope level is -37 dB; this is approximately equal to the isolation of two beams spaced θ_1 degrees apart. The isolations for typical beam spacings are included in Table I. In the earth-station example, the minimum beam spacing is 0.6°, but the corresponding isolations, 37 and 43 dB at 4 and 6 GHz, respectively, are too small for allowable adjacent-satellite interference.[12] These isolations can be increased to 45 and 49 dB, respectively, by increasing the beam (and satellite) spacing to 1°. The increased beam spacing also allows room between feed horns, so they can be moved individually to track small errors in satellite positioning.

IV. AREA EFFICIENCY

Suppose an off-axis plane wave is incident on the main-reflector aperture shown in Fig. 1. The rays intercepted and reflected by the main reflector are displaced laterally with respect to those from an on-axis beam. But if the subreflector surface is sufficiently broadened, each of these rays will be intercepted and focused to a new point that is displaced laterally with respect to focal point f. To accommodate off-axis beams in the horizontal plane, the subreflector width is increased; similarly, for beams in the vertical plane, the height is increased, as indicated by the dashed line in Fig. 1.

The lateral displacement of the focus, corresponding to an off-axis beam at an angle θ, is equal to θ times the equivalent focal length, F. It is assumed that a separate corrugated feed horn is optimally positioned about the focus of each off-axis beam, i.e., each feed is pointed such that the original on-axis amplitude distribution is maintained across the main-reflector aperture, and each feed is longitudinally positioned to minimize aberrations.

The phase center of a corrugated horn can be calculated as a function of frequency.[9] This in turn allows the longitudinal position of the feed to be optimized for broadband performance.

Assuming the foregoing precautions are observed, each beam of the antenna in Fig. 1 has a computed gain about 1 dB less than that obtainable from an aperture with a uniform amplitude distribution. The underlying reasons for the good area efficiency, 80 percent, are (*i*) the main reflector does not have to be enlarged to accommodate off-axis beams, and (*ii*) the F/D ratio of a Cassegrainian antenna is fairly large.

V. POLARIZATION CROSS-COUPLING

T. S. Chu and R. H. Turrin have shown that the cross-coupling of an offset reflector is a function of (*i*) the angle between the feed axis and the reflector axis and (*ii*) the half-angle subtended at the focus by the reflector.[13] In an offset Cassegrainian antenna with a moderate F/D ratio, these angles are fairly small; thus, the cross-coupling is very small. In particular, in Fig. 1, $\psi = 14°$ and $\gamma = 8.5°$. For linearly polarized excitation, the cross-polarized lobes have a peak value of -45 dB. It is anticipated that, in beams with small off-axis angles, as in Table I, the cross-coupling will be about the same.

VI. MULTIPLE-BEAM ANTENNA PARAMETERS

The off-axis beam parameters and corresponding feed-horn dimensions of an offset Cassegrainian antenna fed with corrugated horns can be calculated once the main-aperture diameter and operating

Table I — Multiple-beam antenna parameters

	Earth Station at 4/6 GHz	Satellite at 20/30 GHz
Aperture diameter, D	30 meters	3 meters
Wavelength, λ	7.5 cm/5 cm	1.5 cm/1 cm
Beamwidth, β	0.165°/0.11°	0.33°/0.22°
Primary focal length, F'	30 meters	3 meters
Off-axis beam angle, θ	5°	4°
No. of beamwidths, $N = \theta/\beta$	30/45	12/18
Off-axis parameter, X'	30/45	12/18
Main-reflector offset, Y_2	25 meters	2.5 meters
Subreflector offset, Y_1	5 meters	0.5 meter
Equivalent focal length, F	100 meters	10 meters
F/D ratio	3.33	3.33
Coma aberration, $\Delta\phi$	18°/27°	14°/21°
Off-axis parameter, X	3.0/4.5	1.2/1.8
Feed-horn length, L	3.8 meters	76 cm
Feed-horn diameter, $2a$	1.03 meters	20.5 cm
Beam spacing θ_1	0.6°	1.2°
Isolation at θ_1 spacing	37 dB/43 dB	37 dB/43 dB
Isolation at 1° spacing	45 dB/49 dB	—
No. of available beams	16 (in a row)	18 (within U. S.)

wavelengths are specified. Typical results for an earth-station antenna at 4 and 6 GHz and a satellite antenna at 20 and 30 GHz are given in Table I. Similar results for other diameters and wavelengths can be found by following the text and performing the calculations in the order listed in Table I.

VII. CONCLUSIONS

An offset Cassegrainian antenna fed with corrugated horns is expected to have well-isolated multiple beams that are broadband and dual-polarized. The antenna has good area efficiency and is relatively compact. This combination of properties makes the antenna well-suited for earth stations and satellites.

VIII. ACKNOWLEDGMENT

The author wishes to thank R. F. Trambarulo and T. S. Chu for stimulating discussions, and T. S. Chu for the envelope profile shown in Fig. 2. The author also wishes to thank M. J. Gans for his calculation of area efficiency and his off-axis radiation patterns, which show that the further-out side lobes are not affected by coma aberration.

REFERENCES

1. L. C. Tillotson, "A Model of a Domestic Satellite Communication System," B.S.T.J., *47*, No. 10 (December 1968), pp. 2111–2137.
2. John Ruze, "Lateral Feed Displacement in a Paraboloid," IEEE Trans. on Antennas and Propagation, September 1965, pp. 660–665.
3. T. S. Chu, "A Multibeam Spherical Reflector Antenna," IEEE Antennas and Propagation Int. Symp., Program and Digest, December 9, 1969, pp. 94–101.
4. Henry Zucker, "Offset Parabolic Reflector Antenna," U. S. Patent 3,696,435, filed Nov. 24, 1972.
5. William C. Wong, "On the Equivalent Parabola Technique to Predict the Performance Characteristics of a Cassegrain System with an Offset Feed," IEEE Trans. on Antennas and Propagation, *AP-21*, No. 3 (May 1973).
6. S. K. Buchmeyer, "Corrugations Lock Horns with Poor Beamshapes," Microwaves, January 1973, pp. 44–49.
7. C. Dragone and D. C. Hogg, "The Radiation Pattern and Impedance of Offset and Symmetrical Near-Field Cassegrainian and Gregorian Antennas," IEEE Trans. on Antennas and Propagation, *AP-22*, No. 3 (May 1974), pp. 472–475.
8. Peter W. Hannan, "Microwave Antennas Derived from the Cassegrain Telescope," IRE Trans. on Antennas and Propagation, March 1961, pp. 140–153.
9. C. Dragone, unpublished work, June 1972.
10. H. Zucker, unpublished work, December 1969.
11. H. Kogelnik and Tingye Li, "Laser Beams and Resonators," Appl. Opt., *5*, No. 10 (October 1966), pp. 1550–1567.
12. AT&T Application for a Domestic Communications Satellite System, before the FCC, March 3, 1971, Table IX.
13. Ta-Shing Chu and R. H. Turrin, "Depolarization Properties of Offset Reflector Antennas," IEEE Trans. on Antennas and Propagation, *AP-21*, No. 3 (May 1973), pp. 339–345.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 8, October 1974
Printed in U.S.A.

Limiting the Propagation of Errors in One-Bit Differential CODECs

By J. C. CANDY

(Manuscript received March 27, 1974)

An improved delta modulator is described that communicates to the receiver changes in the magnitude of the signal instead of changes in the amplitude. It is shown that propagation of errors in this system is limited, even when digital accumulators without leakage are used for integration.

I. INTRODUCTION

A major factor in the design of differential CODECs is achieving rapid recovery from transmission errors. Traditionally,[1] slow leakage in analog integrators has allowed error signals to decay with a defined time constant. We describe here another method for curtailing the propagation of errors, one that is well suited for use with digital integration. Digital integration[2–4] is attractive for differential CODECs constructed of integrated circuits. It allows the conversion to analog format to be left to a late stage of signal processing, thereby avoiding the need for high-grade amplifiers and carefully matched pulses, and enables signal amplitudes to be companded by appropriate design of the digital-to-analog (D/A) conversion network.

Introducing slow linear leakage into digital integrators is inconvenient. Indeed, leakage is often undesirable because perfect integration has an advantage of its own, once the effect of circuit and transmission faults has been contained.

Reference 2 demonstrates how a periodic clamp or an overload of the integrator corrects errors, but neither of these methods is suitable for use with speech signals. The proposed method is a simple one; instead of signaling changes of *amplitude*, the coder merely signals changes of *magnitude*. This small modification causes errors to fall quickly to zero. It has application to a variety of CODECs, being especially well suited for delta modulators and related 1-bit CODECs used for

transmitting speech. Application to multibit differential CODECS is somewhat restricted.

II. DIFFERENTIAL CODECS INCORPORATING DIGITAL INTEGRATION

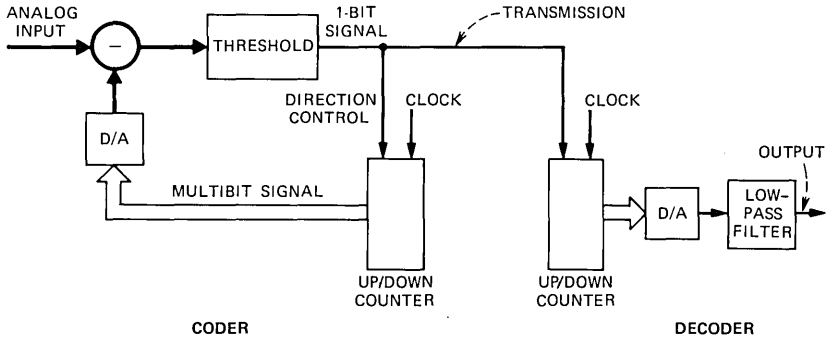
There are several ways of providing digital integration for differential CODECS; Fig. 1 shows three methods. The first is a delta modulator that uses an up-down counter.[3–4] The second is a multilevel differential CODEC that uses a conventional accumulator,[2] comprising an adder and a register. The third is an interpolative CODEC[5] that uses a bidirectional shift register. If this register is fed 1's at the lower input and 0's at the upper, the entire contents shift up or down during each cycle, in response to the output of the threshold decision circuit. Details of this third circuit will be discussed in a later paper.

It is clear that a transmission error or inaccurate start-up procedure in any of these circuits can result in permanent mistracking of the transmitting and receiving integrators. Such mistracking may not be very serious for uniformly quantized signals, but when the D/A conversion levels are companded, mistracking would be catastrophic. Logarithmically companded magnitudes are useful for transmitting speech, and they are easily obtained by means of the circuit in Fig. 1c.

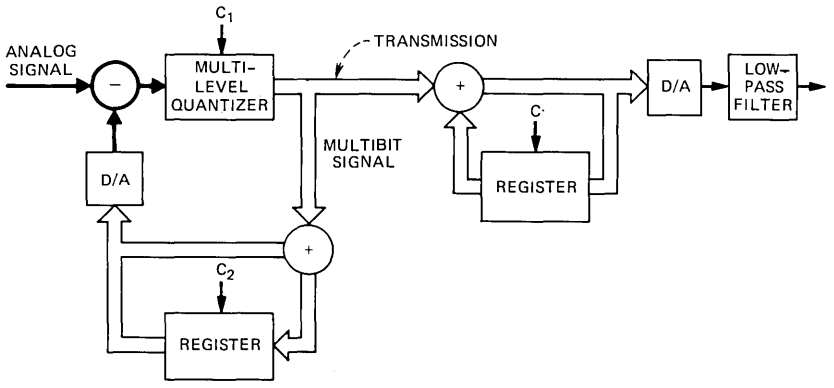
Figure 2 shows a CODEC modified to communicate changes of magnitude: Whenever the most significant bit in the counter is 0, the code is inverted for transmission. The code is reinverted at the receiver under control of the most significant bit of the receiving counter. Notice that for symmetric signals, such as speech, the most significant bit indicates the polarity of the signal, and the remaining bits describe its magnitude, negative magnitudes being in 2's complement format.

The circuits in Fig. 1 do not explicitly show the means for protecting the digital integrators from overflowing, but such protection is needed to prevent serious distortion of very large signals. Figure 2 incorporates two gates, A and B, that detect when the counter is full or empty. Their outputs inhibit threshold decision that would cause over- or underflow.

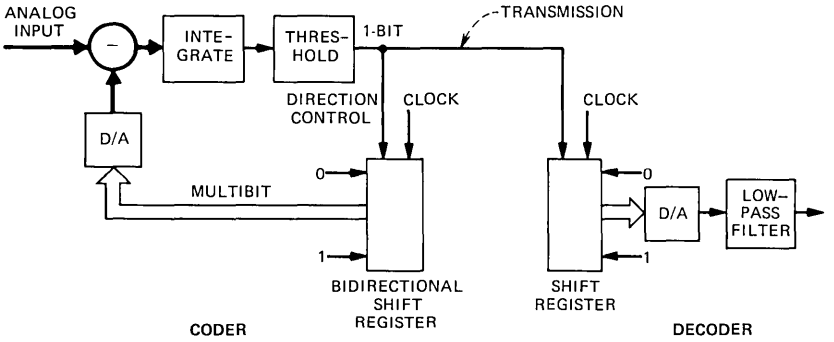
Figure 3 contrasts the responses of the circuits in Figs. 1a and 2 when transmission errors occur in the eighth and the nineteenth cycle. Permanent mistracking can occur when amplitude changes are signaled, but the errors quickly disappear when magnitude changes are signaled. The speed of recovery depends on the frequency of zero crossings of the input signal: A single positive error is wiped out when the signal would have crossed through zero going positively, or when the erroneous signal crosses zero going negatively. Zero crossings in the other direction correct negative errors.



(a)



(b)



(c)

Fig. 1—Differential CODECS employing digital integration. (a) Steps are counted in a delta modulator. (b) Steps are accumulated in a differential CODEC. (c) A shift register stores 1's in an interpolative CODEC.

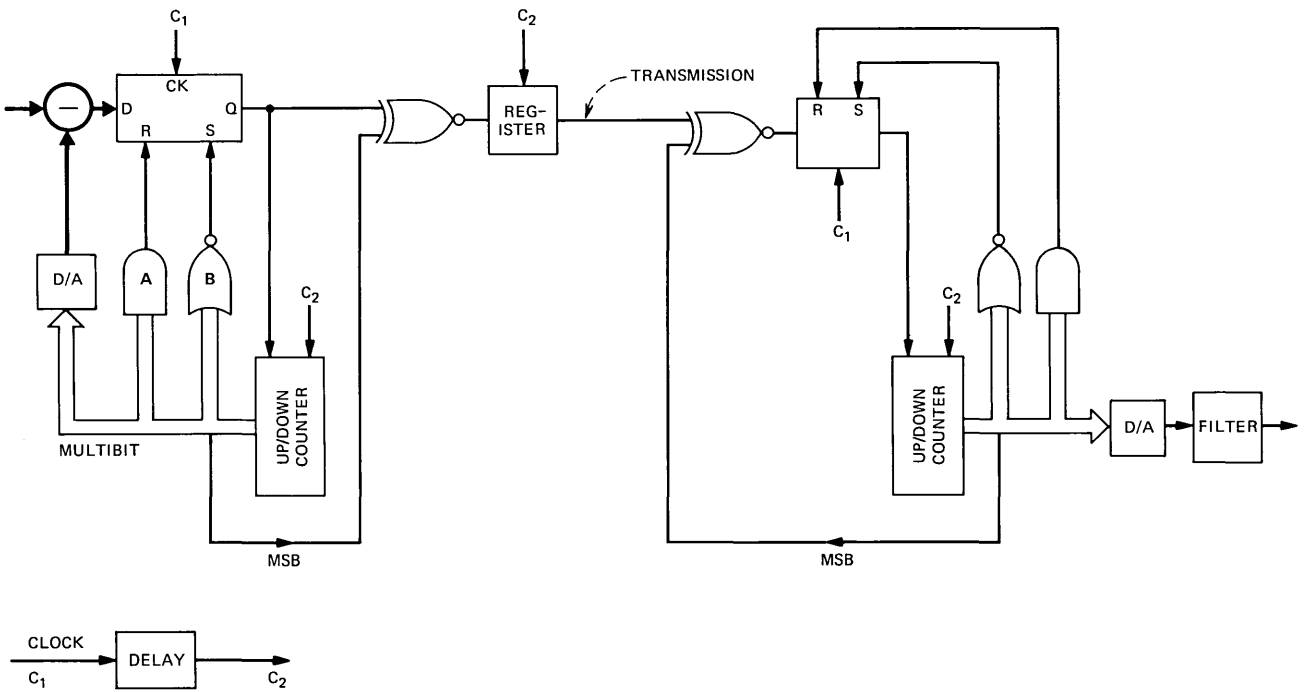


Fig. 2—Circuit that signals changes of magnitude.

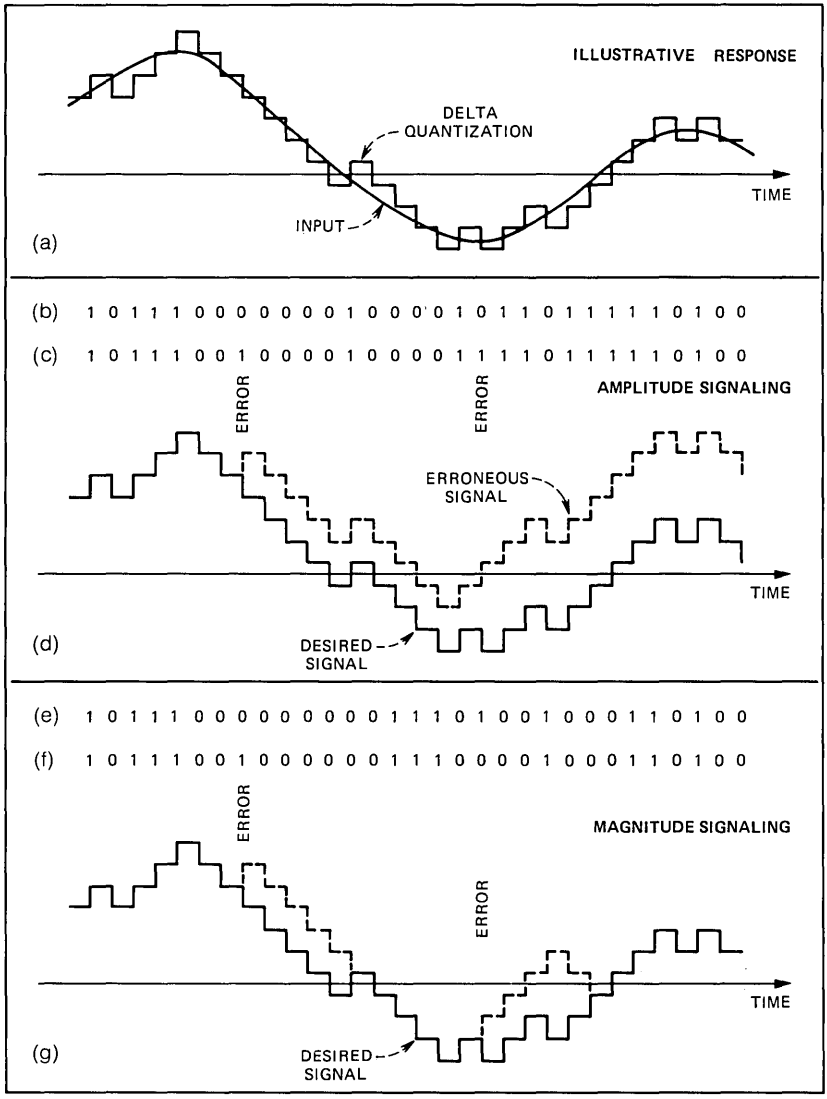


FIG. 3—CODEC responses. (a) Representative input and output. (b) Delta modulator code. (c) Delta modulator code with two errors. (d) Delta modulator responses. (e) Code that signals magnitude change. (f) Code with two errors. (g) Output waveforms.

An alternative circuit arrangement for signaling magnitude is shown in Fig. 4. Here the exclusive NOR gate that inverts the code is placed in the feedback loop of the coder, and the counter holds code that describes only the magnitude of the signal. Polarity is defined by the

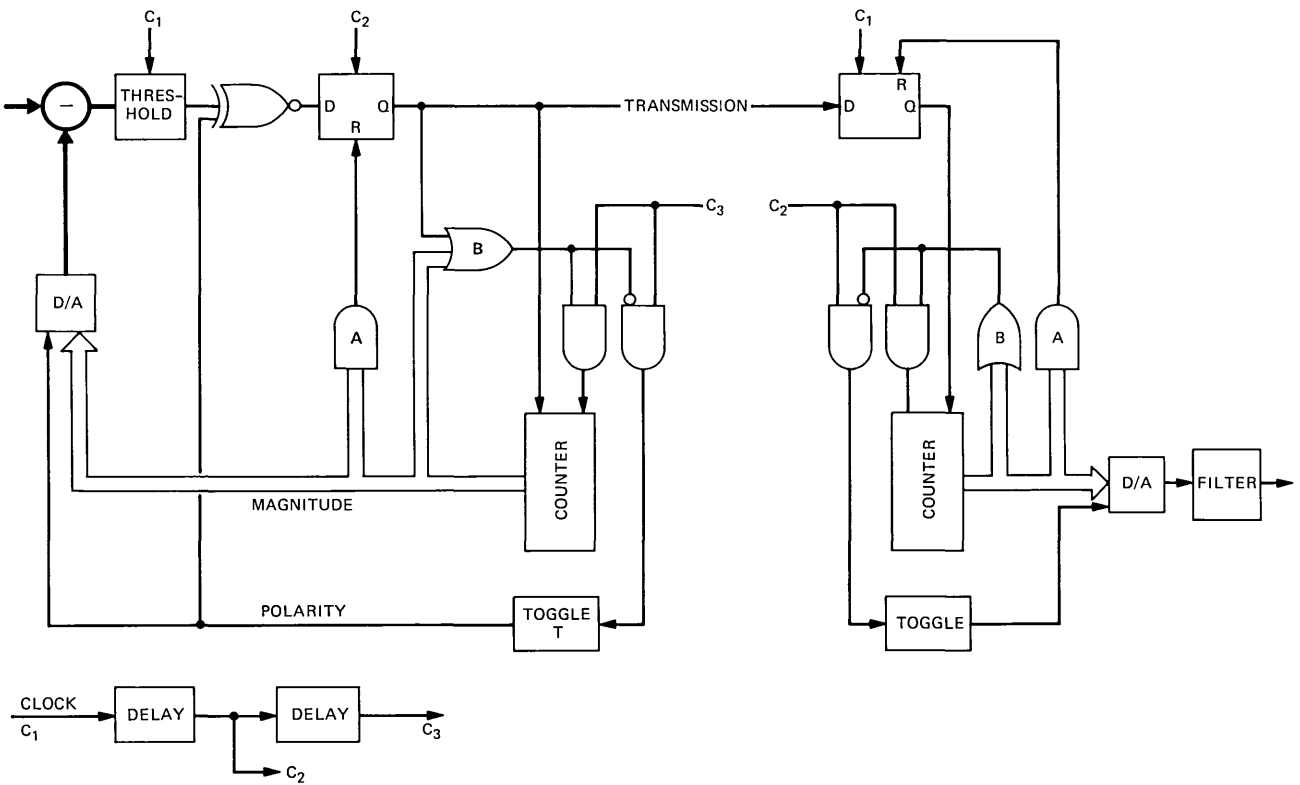


Fig. 4—Another circuit that signals magnitude.

state of a toggle circuit, T. The AND gate A prevents overflow of the counter. The OR gate B detects when the counter is empty, and when a further decrease in magnitude is demanded, it diverts the clock from the counter to changing the state of the toggle. When the toggle indicates a negative polarity, its output inverts both the code and the output of the D/A network, thereby preserving negative feedback. This circuit arrangement is often easier to implement for companded codes than is the arrangement in Fig. 2. The methods illustrated in Figs. 2 and 4 can be used to modify any CODEC in Fig. 1. For the application to multilevel coding, the exclusive NOR gate should be used to invert only the polarity bit, the magnitude in the transmitted bit remaining unchanged.

III. PRESERVING SIGNAL POLARITY

A liability of signaling magnitudes is the inability to continuously inform the receiver of signal polarity. We now demonstrate that transmission errors cannot cause an inversion of the signal.

The output signal at any time has one of a set of discrete values. In Fig. 5a, these values have been numbered in order, as have the cycle times. The graph starts at cycle 1 on 5, an odd-numbered level; thereafter, the signal always has an odd-numbered value on an odd-numbered cycle, even after a transmission error has occurred. An inversion of polarity, illustrated in Fig. 5b, requires that the signal assume even-numbered values at odd-numbered cycles. This can occur only after incorrect start-up or loss of synchronization; once polarity is established, it will be preserved as long as the system remains synchronized, transmission errors having only a transitory effect.

Notice that the method used for avoiding overflow of the counters in Figs. 2 and 4 preserves the timing of the system in a way that prevents an inversion of polarity when error causes an overload. It also helps to eliminate errors in a way that is analogous to the method described in Ref. 2.

The above discussion applies whether or not the output levels are companded, but it does assume that the signal steps up or down by one level at a time. Multilevel differential CODECS permit the signal to step through several levels at a time; they can lose their hold on polarity after an error unless step sizes are chosen with care. Specifically, the sum of any odd number of steps should never be equal to the sum of an even number of steps. Regardless of the manner in which the steps are chosen, multilevel steps tend to increase the time taken to wipe out errors, as is illustrated in Fig. 6. These observations indi-

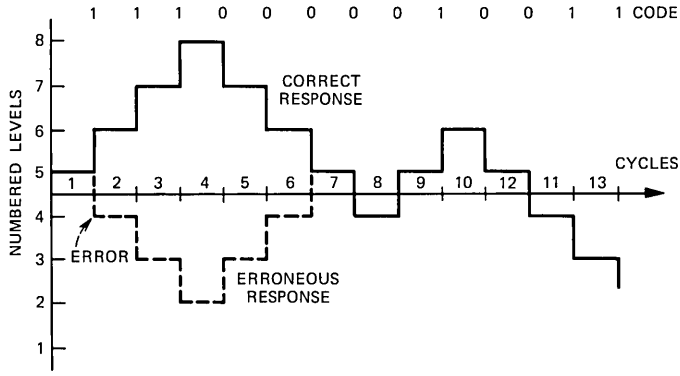


Fig. 5a—CODEC responses showing the correspondence between odd-numbered cycles and odd-numbered levels.

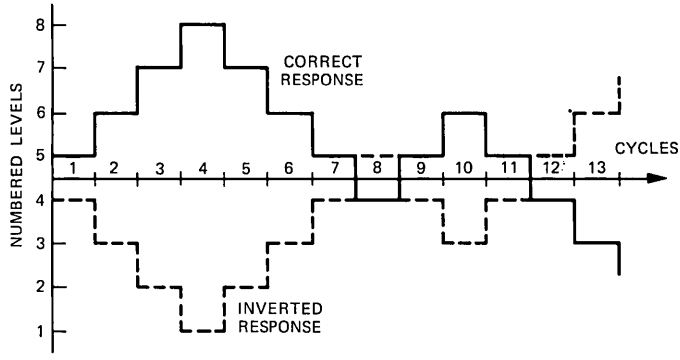


Fig. 5b—An inversion of polarity caused by incorrect start-up.

cate that signaling changes of magnitude is best suited for 1-bit coding, stepping one level at a time, but the levels themselves may be companded.

IV. CONCLUSION

Codes that signal changes in magnitude direct the output to step either toward or away from zero amplitude. We have seen that such coding wipes out the effect of a transmission error when next the signal passes zero in an appropriate direction.

Use of zero as the reference is appropriate for coding speech signals, because they frequently pass through zero amplitude. In general, the

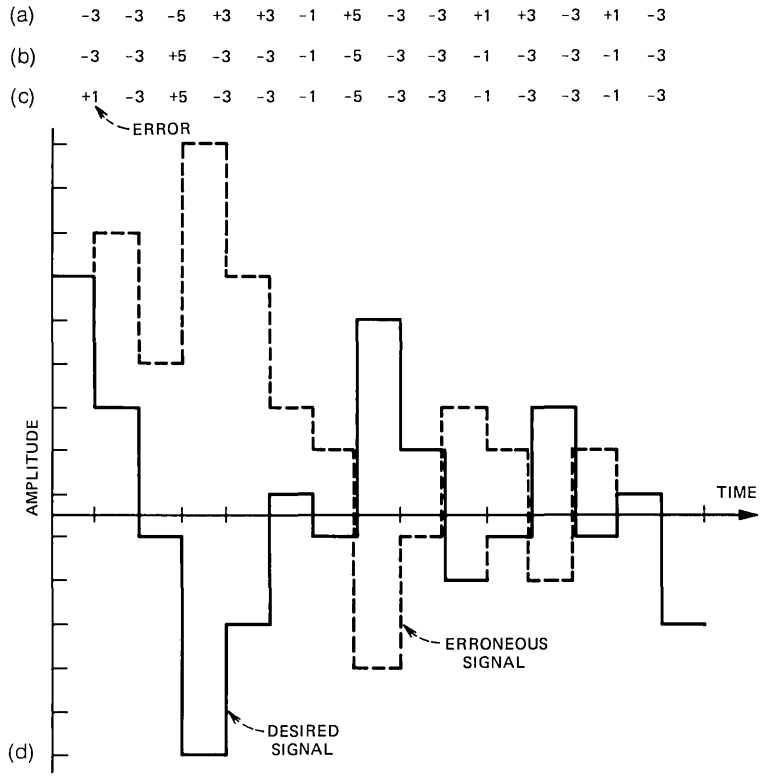


Fig. 6—Response of a multilevel differential CODEC having step sizes ± 1, ± 3, ± 5. (a) Ordinary differential code. (b) The code that signals changes of magnitude. (c) The code with an error. (d) The correct and erroneous responses.

reference can be any value that does not correspond exactly to a possible output level. For coding video, the reference is best set at an amplitude corresponding to medium brightness so that errors can be wiped out quickly.

Conventional differential CODECS signal changes in amplitude; they may be regarded as having a reference set outside the signal range and do not use the error correcting properties associated with an internal reference.

V. ACKNOWLEDGMENT

The author thanks G. D. Boyd, D. J. Connor, C. S. Phelan, and B. Prasada for helpful advice.

REFERENCES

1. J. F. Schouten, F. deJager, and J. A. Greefkes, "Delta Modulation, A New Modulation System for Telecommunication," Philips Tech. Rev., *13*, No. 9 (March 1952), pp. 237–268.
2. J. O. Limb and F. W. Mounts, "Digital Differential Quantizer for Television," B.S.T.J., *48*, No. 7 (September 1969), pp. 2583–2599.
3. C. B. Forrest and S. S. Green, "Analog-to-Digital Converter," U. S. Patent No. 2,836,356 applied for February 1952, issued May 1958.
4. David A. Harms, "PCM Coder," U. S. Patent No. 2,638,219 applied for May 1969, issued January 1972.
5. J. C. Candy, "A Use of Limit Cycle Oscillations to Obtain Robust Analog-to-Digital Converters," I.E.E.E. Trans. Comm., *22*, No. 3 (March 1974), pp. 298–305.

Contributors to This Issue

Jacques A. Arnaud, Dipl. Ing., 1953, Ecole Supérieure d'Electricité, Paris, France; Docteur Ing., 1963, University of Paris; Docteur es Science, 1972, University of Paris; Assistant at E.S.E., 1953–1955; CSF, Centre de Recherche de Corbeville, Orsay, France, 1955–1966; Warnecke Elec. Tubes, Des Plaines, Illinois, 1966–1967; Bell Laboratories, 1967—. At CSF, Mr. Arnaud was engaged in research on high-power traveling-wave tubes and supervised a group working on noise generators. He is a supervisor of a group currently studying microwave quasi-optical devices and the theory of optical wave propagation. Senior Member, IEEE; Member, Optical Society of America.

Thomas T. Butler, B.S. (E.E.), 1958, Auburn University; M.E.E., 1962, New York University; Bell Laboratories, 1958—. Mr. Butler has been concerned with the design and development of electronic switching systems. Member, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi.

J. C. Candy, B.Sc., 1951, Ph.D., 1954, University of Wales, Bangor; British Atomic Energy Authority, 1956–1959; Research Associate, University of Minnesota, 1959–1960; Bell Laboratories, 1960—. Mr. Candy has worked on digital circuits and pulse transmission systems. He is studying methods for processing video signals and designing digital coders. Member, IEEE.

Herbert Y. Chang, B.S.E.E., 1960, M.S.E.E., 1962, and Ph.D., E.E., 1964, University of Illinois; Bell Laboratories, 1964—. Mr. Chang has worked primarily on maintenance techniques for electronic switching systems, design techniques for self-checking processors, digital-fault-simulation methods, and computer-aided-design techniques. He currently supervises a group responsible for the development of advanced design automation systems. Member, Eta Kappa Nu, Tau Beta Pi, Pi Mu Epsilon, Sigma Xi; Senior Member, IEEE.

Stephen G. Chappell, B.E.E., 1969, Georgia Institute of Technology; M.S. (Electrical Engineering), 1971, and Ph.D. (Computer Science), 1973, Northwestern University; Bell Laboratories, 1969—. Since joining Bell Laboratories, Mr. Chappell has been concerned with

problems of logic-design automation, particularly logic simulation and automatic test generation. Member, IEEE, Eta Kappa Nu, Tau Beta Pi.

Ta-Shing Chu, B.S., 1955, The National Taiwan University; M.S., 1957, and Ph.D., 1960, Ohio State University; Bell Laboratories, 1963—. Mr. Chu has been engaged in research on tropospheric wave propagation and microwave antennas. He is currently concerned with electromagnetic problems in the area of satellite communications. Member, IEEE, Commission II of URSI, Sigma Xi, Pi Mu Epsilon.

Charles H. Elmendorf, B.S. (Engineering Physics), 1969, Cornell University; M.S. (Electrical Engineering), 1971, Northwestern University; Bell Laboratories, 1969—. Since 1971 Mr. Elmendorf has been involved in different aspects of machine-aided design including logic simulation. Member, Tau Beta Pi.

Thomas G. Hallin, B.S., 1968, Oregon State University; M.S., 1970, Northwestern University; Bell Laboratories, 1968—. Mr. Hallin has been working on fault diagnosis for large digital systems. Member, IEEE, Tau Beta Pi, Eta Kappa Nu, Sigma Xi.

Gary W. Heimbigner, B.S.E.E., 1969, University of Washington; M.S.E.E., 1971, Northwestern University; Bell Laboratories, 1969–1974. Mr. Heimbigner has worked on maintenance techniques for a small exploratory processor; he was more recently involved with the development of advanced design automation techniques. Member, IEEE, Tau Beta Pi.

Keith W. Johnson, B.S.E.E., 1966, and M.S.E.E., 1971, Northwestern University; Bell Laboratories, 1966—. Mr. Johnson has worked in circuit design and diagnostic programming areas of No. 1 ESS and No. 4 ESS developments. His main interest is in the design and testing of easily diagnosable logic circuits. Member, Tau Beta Pi, Eta Kappa Nu, Pi Mu Epsilon, Phi Eta Sigma.

John J. Kulzer, B.S.E.E., 1964, M.S.E.E., 1965, and Ph.D., 1969, Illinois Institute of Technology; Bell Laboratories, 1969—. Mr. Kulzer has worked on diagnostic design for large digital circuits and has car-

ried out fault simulation studies using the LAMP simulator. His other work has included the modification of scheduling routines for the IBM TSS 360/67 to tailor the system to LAMP simulation job mixes. Member, IEEE.

Dietrich Marcuse, Diplom Vorpruefung, 1952, Dipl. Phys., 1954, Berlin Free University; D.E.E., 1962, Technische Hochschule, Karlsruhe, Germany; Siemens and Halske (Germany), 1954–1957; Bell Laboratories, 1957—. At Siemens and Halske, Mr. Marcuse was engaged in transmission research, and studying coaxial cable and circular waveguide transmission. At Bell Laboratories, he has been engaged in studies of circular electric waveguides and work on gaseous masers. He spent one year (1966–1967) on leave of absence from Bell Laboratories at the University of Utah. He is presently working on the transmission aspect of a light communications system. Mr. Marcuse is the author of three books. Fellow, IEEE; member, Optical Society of America.

James McKenna, B.Sc. (Mathematics), 1951, Massachusetts Institute of Technology; Ph.D. (Mathematics), 1961, Princeton University; Bell Laboratories, 1960—. Mr. McKenna has done research in quantum mechanics, electromagnetic theory, and statistical mechanics. He has recently been engaged in the study of nonlinear partial differential equations that arise in solid-state device work and in the theory of stochastic differential equations.

Edward A. Ohm, B.S. (E.E.), 1950, M.S. (E.E.), 1951, and Ph.D. (E.E.), 1953, University of Wisconsin; Bell Laboratories, 1953—. Mr. Ohm has worked on low-loss waveguide components and two experiments demonstrating the feasibility of low-noise satellite communication systems: the measurement of background sky temperature and the reception of very small signals from a passive satellite (Echo I). He is currently working on a dual-polarized polarization-tracking feed for the earth-station antennas of the 4- and 6-GHz domestic satellite system.

Lonnie D. Schmidt, B.S. (Computer Science), 1969, University of Missouri at Rolla; M.S. (Computer Science), 1971, Northwestern University; Bell Laboratories, 1968–1973. While at Bell Laboratories, Mr. Schmidt worked on design automation and logic-circuit simulation. Member, ACM, Kappa Mu Epsilon, Tau Beta Pi.

N. L. Schryer, B.S., 1965, M.S., 1966, and Ph.D., 1969, University of Michigan; Bell Laboratories, 1969—. Mr. Schryer has worked on the numerical solution of parabolic and elliptic partial differential equations. He is currently studying problems of this type that arise in semiconductor device theory.

George W. Smith, Jr., B.E.E., 1952, North Carolina State College; M.S.E.E., 1958, Stevens Institute of Technology; M.A., 1961, Princeton University; Ph.D., 1963, Princeton University; Bell Laboratories, 1952—. Mr. Smith's early work included military missile projects and military telephone switching systems. He later worked on the development of Bell System message switching systems and automated telephone directory assistance service. Since 1971, he has been head of the Design Automation Department which is engaged in providing computer programs for aiding the design of electronic switching systems, automatic drafting, automatic logic verification, and automatic trouble location. Member, IEEE, Eta Kappa Nu, Sigma Xi, Tau Beta Pi.

R. H. Walden, B.E.S., 1962, M.E.E., 1963, and Ph.D., 1966, New York University; Bell Laboratories, 1966—. Mr. Walden's initial activities in the Semiconductor Device Laboratory were concerned with switching properties of VO_2, followed by work on the conduction properties of Al_2O_3 films. He was also involved in a study of the properties of charge-transfer devices and is presently working on the design of integrated circuits using CMOS technology.

Robert B. Walford, B.S.E.E., 1961, Illinois Institute of Technology; M.S.E.E., 1963, and Ph.D., E.E., 1967, University of Southern California; Bell Laboratories, 1967—. Mr. Walford has worked on a variety of projects including real-time information processing, computer-aided design, and ESS software support. He is currently supervisor of a group working on new methods of software development. Member, NSPE, IEEE, Eta Kappa Nu, Tau Beta Pi.

THE BELL SYSTEM TECHNICAL JOURNAL is abstracted or indexed by *Abstract Journal in Earthquake Engineering. Applied Mechanics Review, Applied Science & Technology Index, Chemical Abstracts, Computer Abstracts, Computer & Control Abstracts, Current Papers in Electrical & Electronic Engineering, Current Papers on Computers & Control, Electrical & Electronic Abstracts, Electronics & Communications Abstracts Journal, The Engineering Index, International Aerospace Abstracts, Language and Language Behavior Abstracts, Mathematical Reviews, Metals Abstracts, Science Abstracts,* and *Solid State Abstracts Journal*. Reproductions of the Journal by years are available in microform from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

Bell System