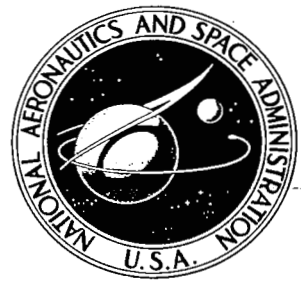
NASA CONTRACTOR REPORT



NASA CR-1755

AUTOMATIC TEST EQUIPMENT FOR ELECTRONIC COMPONENTS

Volume II - Final Technical Report

by William A. Heffner

Prepared by
MARTIN MARIETTA CORPORATION
Denver, Colo. 80201
for Electronics Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1971

1. Report No. NASA CR-1755	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AUTOMATIC TEST EQUIPMENT FOR ELECTRONIC COMPONENTS VOLUME II - FINAL TECHNICAL REPORT		5. Report Date March 1971 6. Performing Organization Code
7. Author(s) William A. Heffner		8. Performing Organization Report No. MCR-69-520 (Vol. II)
9. Performing Organization Name and Address Martin Marietta Corporation P. O. Box 179 Denver, Colorado 80201		10. Work Unit No. 11. Contract or Grant No. NAS12-2045
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, D. C. 20546		13. Type of Report and Period Covered Contractor Report 14. Sponsoring Agency Code

15. Supplementary Notes

Volume I is a summary of the report and contains the significant results of the study.

16. Abstract

This report defines an automatic test facility for testing discrete and integrated electronic components. The report covers the analysis of requirements, candidate configurations, specification of the selected configurations, a cost-effectiveness study and acquisition plan. The primary focus has been on technology changes and future trends.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Component Test; Computerized Testing; Automatic Test; Test Equipment	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	145	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

FOREWORD

This report is the product of a six-month study by Martin Marietta Corporation, Denver Division, under Contract NAS12-2045. Mr. Edward Sarkisian, NASA/ERC Automated Techniques Branch, served as Technical Monitor.

The purpose of the study was to define the requirements for an advanced test system for discrete, integrated, and experimental electronic components, and to perform a cost-effectiveness analysis of the system application.

This report contains the significicant results of the study. The primary focus has been on technology changes and future trends.

Contributors

Charles Brauch	University of Colorado
J. C. DeFellippie	Martin Marietta Corporation
J. Michael Meuer	Martin Marietta Corporation
John E. Cervi	Martin Marietta Corporation

CONTENTS

	Page
SUMMARY	1
INTRODUCTION	1
Report Description	1
Study Results	3
REQUIREMENTS DEFINITION	5
Data	5
Techniques Used in Acquiring and Developing Data	6
Survey Results	7
REQUIREMENTS ANALYSIS	18
General System Characteristics	18
Selection of Characteristics	20
Major Configurations	23
General Configurations	28
CONFIGURATION STUDY	31
Approach	31
Configuration A1 - Minimum Dedicated System for Functional Test of Digital Large-Scale Arrays	37
Configuration A2 - Configuration A1 with Addition of Parametric Test Capability for Digital Large-Scale Arrays	40
Configuration B1 - Minimum System for Time-Sharing of Functional and Parametric Test Stations	41
Configuration B2 - Addition of Reliability Testing	44
Configuration B3 - Dual Configuration for Addition of Reliability Testing	48
Configuration B4 - Time-Shared LSA and Parametric Testing with Supporting Installation for Software	50
Configuration C1 - Medium Scale Expandible CPU in Dedicated System for Functional Test of Digital LSA	52
Configuration C1′ - Background/Foreground System for Dedicated Test of Digital LSA	54
Configuration C2 - Configuration C1 with Addition of Parametric Test Capability for Digital LSA	57
Configuration C3 - Background/Foreground System for Time-Sharing of Reliability and LSA Test Stations	58

Configuration C4 - Expansion of C3 Time-Sharing System to Additional Users and Addition of Modem Channels 60
Configuration D1 - Remote User Time-Sharing System 64
Configuration Summary and Tradeoff 68
Configuration Study, Test Language Translator . . . 78
SPECIFICATIONS 91
General Recommendations 91
Test Station 93
Input/Output Control 99
Central Processor Specifications 100
Test Language and Translator 102
Supervisory Systems 106
COST EFFECTIVENESS STUDY 113
General 113
Organization 114
APPENDIX A -- PROBLEM AREAS 131
APPENDIX B -- RELATED STUDIES AND GRANTS . . . . . 135 thru 139

Figure

1 Dedicated Computerized Testing 24
2 Time-Shared Computerized Testing 25
3 Master/Slave Computerized Testing 26
4 Test Station Configuration T6 34
5 Test Station Configuration T7 35
6 Test Station Configuration T8 36
7 Configuration A1 Hardware 39
8 Configuration B1 Block Diagram 43
9 Configuration B2 Block Diagram 45
10 Configuration B3 Block Diagram 48
11 Configuration C1 Block Diagram 52
12 Configuration C3 Block Diagram 58
13 Configuration C4 Block Diagram 61
14 Configuration D1 Block Diagram 65
15 Configuration D1 Memory Allocation 66
16 Alternative Growth Path Cost Comparison 72
17 System Configuration Graphic Summary 75
18 Typical Stimulus Channel 98
19 Typical Result Channel 98
20 Cost Effectiveness Study Organization 116
21 Total Cumulative Costs 117
22 Cumulative Personnel Costs 117
23 Cumulative Equipment Maintenance Costs 118

24	Cumulative Equipment Costs	118
25	Equipment Layout, Specified System	119
26	Equipment Layout, Conventional System	120
27	Test Requirements and Capabilities	121
B-1	Fault Isolation and Diagnosis in Multiterminal Devices (NAS12-689)	136
B-2	Impulse Techniques for Testing Digital Microcircuits (NGR-31-001-132)	137
B-3	Combinatorial Analysis for Checkout Techniques (NGR-31-003-066)	138
B-4	Utilization of Topological Techniques for Modeling of Microcircuits (NGR-05-017-017)	139

Table

I	Job Function of Contact	5
II	Types of Testing Performance	7
III	Most Effective Tests	7
IV	Unique or Automatic Test Equipment Used	12
V	Commercially Available Test Sets	13
VI	Required Characteristics	20
VII	Requirements by Type of Testing	30
VIII	Typical Software Systems	32
IX	Configuration A1 Equipment List	38
X	Configuration A2 Equipment List	40
XI	Configuration B1 Equipment List (Delta to A1)	42
XII	Configuration B2 Equipment List (Delta to B1)	46
XIII	Configuration B2 Equipment List (Delta to A2)	47
XIV	Configuration B3 Equipment List	49
XV	Configuration B4 Supporting Installation for Software	51
XVI	Configuration C1 Equipment List	53
XVII	Configuration C1′ Equipment List	55
XVIII	Configuration C1′ Equipment List (Delta to C1)	56
XIX	Configuration C2 Equipment List	57
XX	Configuration C3 Equipment List (Delta to C1′)	59
XXI	Configuration C4 Equipment List	62
XXII	Configuration D1 Equipment List (Delta to C4)	67
XXIII	System Configuration Summary	69
XXIV	System Configuration Summary Code Explanation	71

XXV	Translator Characteristics	86
XXVI	Cost Effectiveness Summary	122
XXVII	Cost Effectiveness Study Parameters	123
XXVIII	Total Cumulative Costs	124
XXIX	Personnel Costs, Specified System	124
XXX	Personnel Costs, Conventional System	125
XXXI	Personnel Work Time	125
XXXII	Personnel Pay Rates	126
XXXIII	Equipment Maintenance, Specified System	127
XXXIV	Equipment Maintenance, Conventional System . .	127
XXXV	Equipment List, Specified System	128
XXXVI	Equipment Cost	129
XXXVII	Current and Projected Test Load	129
A-I	Items Considered Present Problem Areas	132
A-II	Items Considered Future Problem Areas	134

AUTOMATIC TEST EQUIPMENT FOR ELECTRONIC COMPONENTS

VOLUME II - FINAL TECHNICAL REPORT

By William A. Heffner
Martin Marietta Corporation

SUMMARY

This report defines an automatic test facility for NASA/ERC for testing discrete and integrated electronic components. In addition, the facility is to serve as a test bed for algorithms and diagnostic techniques under development by NASA/ERC. Capability for extension of the facility into other testing categories, and use by other NASA centers and industry, has been considered.

The report covers the analysis of requirements, candidate configurations, specification of the selected configurations, a cost-effectiveness study, and acquisition plan. Assessments of current industry test practices, languages, potential test equipment standards, and recommendations for additional development activity are made.

INTRODUCTION

A detailed description of the report structure is given below.

Report Description

Surveys have been conducted of a cross section of NASA and aerospace industry installations involved with component testing. Test practices, equipment, problems, and future trends were examined. The survey results are given in the section entitled "Requirements Definition."

The "Requirements Analysis" section establishes requirements for an automatic test system from the survey results. A series of candidate configurations is defined in the discussion under "Configuration Study," and optimum configurations are selected. In the "Specifications" section, more detailed specifications for the selected configurations are given.

The section, "Cost-Effectiveness," presents the results of a study using the most promising of the configurations. This study develops a typical component testing installation and then compares the cost-effectiveness of conventional equipment with the selected system configuration over a five-year period. A relationship between component testing costs and annual dollar volume is established for a typical aerospace contractor or NASA organization. From this, cost-effectiveness projections of the two testing approaches can be readily implemented for a variety of installations.

A plan providing for incremental acquisition of the facility by ERC in response to test needs and funding constraints is defined. The acquisition plan is carried to the extent of plotting extra-ERC uses of the facility.

Primary emphasis in this report has been placed on the following aspects of the problem:

1) Computer;
2) Man/machine interface;
3) Test station architecture.

Lesser emphasis has been placed on:

1) Specific stimuli or measurements;
2) Testing criteria or methods.

There is a considerable body of literature on component testing criteria and methods. The actual stimuli and measurement parameters for discrete and small-scale integrated circuits represent no serious technical problems for today's devices; test equipment is generally available to deal with them. What does seem to be the problem is the control and programing of this test equipment, the expense of acquiring new equipment for new parameters or devices, and the management of data -- both at the local installation level and from an industry standpoint. Proper use of the computer provides an answer to these problems. Control functions may shift from specially designed equipment to the computer; increased power in the computer permits higher level language communication, interaction during test, and sophisticated data analysis and control. In the future, large-scale integration and increasingly complex systems will be difficult to analyze and diagnose for faults by manual methods; the computer will play an increasingly large role in defining tests as well as controlling them and managing the data.

As more and more functions are turned over to the computer -- because they are too complex, or involve too much bookkeeping, or must be performed too fast -- the level of communication between man and system becomes itself limiting. There is no question that computer systems are today limited more by our ability to tell them what to do than by their capability to perform.

For these reasons we have chosen to emphasize those aspects of an automatic test system that we consider crucial. We assume that equipment (hardware) will be available to provide the physical interface to the devices when it is needed and have concentrated on devising a system to integrate these individual hardware items, rapidly and economically, into an integrated system for control and mangement.

Study Results

From a technical viewpoint, present component testing is reasonably adequately performed by existing test equipment to existing criteria. However, serious problems exist from an economic and procurement standpoint (22 weeks is a typical procurement span time for spaceborne components), and some considerable dissatisfaction exists with present test criteria.

The trend in components and component usage is such that the economic and procurement problem is accentuated, and serious new technical problems are created. The rapidly growing integrated circuits category of components has only recently achieved a government standard on test methods (MIL-STD-883). There is an unmistakable trend toward the use of medium- and large-scale integrated circuits, hybrids, thick films, and other types of miniaturized large-scale arrays. Components will shortly acquire the complexity of present subsystems, without the accessibility afforded by today's packaging techniques. Present test criteria and practices will rapidly become inadequate to cope with the escalating complexity and miniaturization -- not only in degree, but also in kind.

A computerized test system in response to the objectives of this study must be capable of change to meet the dynamic requirements described above and must also be compatible with present economic and practical constraints. A three-phase incrementally acquired system has been developed from numerous detailed configurations and tradeoffs.

In the initial configuration, a small-scale computer is used in a dedicated configuration with a single test station. The station is general-purpose for a class of test devices. A minimum test language and test monitor is implemented within the installation.

Phase II upgrades the central processor portion of the system to provide long-term expansion capability and greatly increased software power. The test station is retained; additional stations may be added. A language compiler or a meta-compiler is resident within the system. Interfaces are possible with advanced test algorithms and processors now under development. Operator interaction during tests is provided.

Phase III will provide for multiple remote operations, with an expanded central processor handling local test needs and those of other, remotely located, systems over low-cost telephone lines. All operations, including testing, translation, and data analysis, may be time-shared from the remote sites through the test center. The test station retains the same basic architecture except that a "minicomputer" is incorporated into the remote terminals for high-speed test operations. Very powerful software and between-installation data correlation is provided.

REQUIREMENTS DEFINITION

This section summarizes the present and projected test needs of NASA agencies and contractors in general and of NASA/ERC in particular for testing of electronic components.

Data

Data for this subsection were derived from the following sources (see table I):

1) Personnel directly engaged in testing, test definition, or reliability analysis for electronic components in various NASA centers and agencies;

2) Discussions with responsible individuals in the Automated Techniques Branch and other branches at ERC;

3) A two-day round table conference held at Martin Marietta Corporation, Denver, Colorado, in January 1969 with eight ERC Contractors and grantees engaged in development of test and diagnostic techniques;

4) Selected industry representatives including semiconductor manufacturers, test equipment manufacturers, aerospace suppliers, and commercial testing laboratories;

5) Personnel at Martin Marietta in Denver engaged in components, reliability, and test analysis for AAP, Viking, and other long-range NASA programs.

TABLE I

JOB FUNCTION OF CONTACT

Contact	Job function
1	Component specialist
2	Component specialist
3	Lead engineer - physics of failure group
4	Engineer - management consultant group (components)
5	Manager - unique programs (semiconductor manufacturer)
6	Manager - parts and reliability group
7	President - component testing and consulting laboratory
8	Manager - hi-rel product marketing group
9	Engineer - advanced checkout systems group
10	Engineer - incoming inspection and test group

We have also drawn freely on extensive corporation experience with computerized testing for onboard checkout (OCS), preflight systems checkout, and production and developmental testing.

Techniques Used in Acquiring and Developing Data

Extensive, checklist-guided telephone interviews were conducted with NASA and industry testing personnel. These interviews were tape recorded and transcribed. The checklist was designed to cover the following major areas of interest:

1) Types of testing performed;
2) Test devices;
3) Present and planned test equipment;
4) Test practices and criteria;
5) Present and anticipated problem areas.

An eight-page detailed questionnaire was developed for ERC testing and usage requirements. Most of the questionnaire items were answered as a result of extensive discussions between Martin Marietta personnel and ERC representatives. Some items will remain unanswered until further experience is gained and until techniques presently under development are clarified.

Input/output charts and gross interface parameters were developed, where appropriate, for related test and diagnostic studies funded by ERC. None of these were far enough along to define detailed interfaces and requirements. However, enough information is available to place them within the general framework and growth plan.

Flow diagrams, test installations, equipment lists, and costings were developed from information supplied by several of the sources. Some of these data appear in the cost-effectiveness study section.

At least as significant as the specific data detailed above is the large amount of subjective material, embodying judgments and opinion (and some strongly held beliefs), gleaned from many of those interviewed. Because these represent a cross section of responsible and knowledgeable individuals in component testing, we have not hesitated to use this material in assessing trends and interpreting data.

Survey Results

Test activity.- Components representing a significant volume of current test activity (tables II and III) may be categorized as follows:

1) Discrete semiconductors - Transistors, diodes, FETS;

2) Integrated circuits (IC) - Linear and digital monolithic circuits, 14 pins or less;

3) Medium-scale integration (MSI) - Monolithic circuits of less than 100 interface pins or 100 logic nodes, primarily standard, commercially available, digital circuits;

4) Resistors;

5) Capacitors;

6) Relays;

7) Miscellaneous electromechanical devices.

TABLE II

TYPES OF TESTING PERFORMED

User	1	2	3	4	5	6	7	8	9	10
1. Qualification		x						x		
2. Screening				x		x	x	x		
3. Reliability								x		x
4. Failure analysis			x							
5. Evaluation						x				
6. System									x	

TABLE III

MOST EFFECTIVE TESTS

1. Burn-in
2. High-temperature stress
3. Temperature cycling
4. Hermeticity
5. Centrifuge
6. Infrared inspection
7. X-ray inspection
8. Component dissection

Most of the test activity appears concentrated in the semiconductor area. This is primarily because of the glamor of the devices and the glamor of the required test techniques.

Criteria and practices.- The most universally accepted testing criteria today are the military standards and specifications. The most pertinent are:

1) MIL-STD-202C;

2) MIL-STD-883;

3) MIL-STD-781;

4) MIL-S-19500;

5) MIL-C-45662.

Other sources of criteria are:

1) Component manufacturer's programs (i.e., Motorola "MEGALIFE," Fairchild "FACT");

2) Aerospace contractors, via in-house specifications and procurement drawings;

3) Procuring agencies, often in contracts or specifications for a system.

For assemblies, as distinct from "piece-part" components, no standards exist that define specific tests and test methods similar to those for discrete and integrated components. It would be obviously impractical to provide detailed requirements for the extremely large number of assemblies that are generated each year. However, it does not seem impractical to provide general test methods and criteria for a class of assemblies. As a matter of fact, at the rate that integrated circuit components are diversifying, one might expect that in a few years requirements would be impossible to define at a detailed level for these also.

Types of testing.- The military specifications use a convenient grouping of component test requirements as follows:

1) Category A, 100%, prescreening - Includes electrical parameters, bake or burn-in, and selected impact shock and seal tests;

2) Category B, sample, extensive tests - Includes stress, leakage, seal, X-ray, life (nondestructive);

3) Category C, sample, destructive tests - Includes salt-spray, lead strength, and weldability/solderability.

This grouping is not particularly useful for this study. A broader categorization of testing requirements and practices is used in this report, based on operational rather than specification requirements. This categorization is described in the following paragraphs.

Receiving inspection (sample functional testing with occasional parametric or ac testing): The purpose is to provide some assurance by use of statistical techniques that a lot of parts conforms to specification.

Screening (a combination of stress and electrical parameter tests, usually burn-in and/or life in combination with dc parameters, performed 100% on a lot of parts): Occasionally impact shock, seal leakage, or other individual tests are included. The purpose is to screen out all active or latent defects before assembly.

Qualification: This is an extensive series of electrical, temperature, and environmental tests, both 100% and sample, designed to verify that a specific part type will meet criteria and properly operate in the usage environment.

Reliability testing (a set of electrical and/or stress tests designed to establish part operating history and failure rates): Normally, this is long-term testing, sometimes with considerable data acquisition and reduction.

Component manufacturing (test performed by the component manufacturer to MIL-Spec levels): Normally this includes Categories A, B, and C. Some tests such as burn-in (bake) or impact shock may be performed as part of the process. Manufacturers have developed a more or less standard set of tests that they perform on military and (sometimes) industrial grade components; the test levels and sequence frequently vary from manufacturer to manufacturer. Laboratories are also maintained to perform special testing requested by customers. Some tests, such as 100% life, are very unpopular with the larger manufacturers and are difficult to obtain on a timely basis. A number of independent testing laboratories have arisen to fulfill this function.

Component subassembly testing: This is testing performed on a group of individual components that have been assembled to perform some larger function. This category includes testing of printed circuit boards, modules, black boxes, and chassis. This class of testing is normally performed by equipment manufacturers as in-process testing.

Test devices may have from four terminals to 500; internal logic nodes may number in the thousands. A particular device may, but often does not, represent a functional entity. For example, part of a circuit may be packaged on one printed card and the mating portion on another. There is often an enormous variety of devices, functions and physical interface connections even within a small program. Quantities will range from the higher production rates associated with tactical weapons systems, to one of a kind, R&D quantities associated with many NASA projects. Special test equipment is usually expensive and voluminous; much of it is specially designed for a specific test. A large variety of commercially available measurement equipment is used; there are proportionally many general-purpose test systems available for this field. In many cases, future testing of LSI will reflect the problems and techniques now applicable to assembly testing.

Laboratory automation: While this field includes applications that cannot be strictly classified as testing, many techniques from the previous categories are applicable -- as is a properly designed computer-controlled test system. Some examples of typical tasks within this category are:

1) Real-time data acquisition and analysis for mass spectrometers, radiation detectors, and similar instruments;
2) Automatic mask generation;
3) Control and monitoring of deposition for thin film development;
4) Metrology and calibration standards applications.

These are normally relatively low data rate operations, but with relatively large amounts of data. Operators are usually skilled. Once established, an application remains relatively stable.

Experimental: This category is directed toward either the development of techniques for testing and diagnostics or the use of an automatic facility as a tool in developing new devices for technology. It is characterized by almost constant change in

what the automatic test system is asked to do, a broad spectrum of technical requirements, and a relatively low rate of actual testing. All other categories of testing are concerned with a specific through-put of test devices or data. In experimental testing, any through-put of actual devices is purely secondary; the output may be the description of a new technology, a program, a criterion or standard, or the development of a new device.

References to "type of testing" throughout the remainder of this report assume the definitions given in the preceding paragraphs.

Two distinct types of component "test" activity have been ommitted from the table, which deserve specific mention:

1) Evaluation - Normally implies dissection, X-ray and similar physical operations, to arrive at a judgment of the validity of a manufacturer's processes.

2) Characterization - Usually refers to intensive, in-depth measurement of parameters - - some of them exotic - - to define the behavior of a part family, a geometry, or a process - - rather than measurement of conformance of a production group of parts to a predefined standard.

Neither of the above "test" activities fall strictly within the definition of testing as used herein nor are they amenable to performance by an automatic test system. Both are laboratory activities, requiring considerable human judgment, and are performed relatively infrequently compared to the other types of testing activity.

Test equipment - Equipment actively in use at various installations is defined in table IV. Analysis of a number of commercially available test sets is given in table V. The low level of computer utilization in this equipment is striking in view of the considerable interest and literature currently being generated.

TABLE IV

UNIQUE OR AUTOMATIC TEST EQUIPMENT USED

Equipment	User									
	1	2	3	4	5	6	7	8	9	10
Semiautomatic commercial IC tester		x	x			x	x	x		x
Semiautomatic commercial transistor tester		x	x			x		x		x
Automatic commercial relay tester										x
Automatic facility-built IC tester			x							
Automatic facility-built discrete component tester			x							
Automatic commercial capacitance bridge						x	x			
Automatic commercial multimeter							x			
Automatic commercial resistance-capacitance tester										x
Electron microscope		x	x							
Tap tester			x							
Bond puller			x							

TABLE V

COMMERCIALLY AVAILABLE TEST SETS

	LSA	ICs	Transistor	Diode	Resistor	Capacitors	Relays	Maximum pin capability	Tests/sec	Stimulus accuracy, %	Measure accuracy, %	Sequence	Data Recording	Program input rate char/sec	Data acces	Program storage	Programing method	Computational	Language
Fairchild																			
5000	X	X	1	1	1	1	1	100	200	0.1→0.2	0.1→0.2	F	Opt		Transient or nonres opt	Disc	Keyboard or opt tape	No	Machine
8000	X	X	1	1	1	1	1	144	10K	1	1	I	Opt		Nonres	Disc or comp	TTY or punched tape	Yes	Machine
E H 4002	2	2	2	2				120	200	5	1	P	Incr tape		Nonres	Mag core	Keyboard opt tape	No	Machine
Tektronix S-3130	2	2	2	2					100	3	3	P			Transient or nonres	Paper tape and disc	Punched tape	No	Machine
AAI 1000		X	1	1	1	1	1	60	180	0.1	0.22 I 0.11 V	P	Opt	10	Transient or nonres	Disc	TTY and punched tape	No	Machine (mnemonic)
Teradyne J259	3	3	3	3	1	1	1	63 Spec arr opt	200	0.15→0.5	0.15-V 0.25-1 I	I	Mag or paper tape		Nonres	Mag core		Yes	Machine
Texas Instruments 553	X	X	1	1	1	1	1	148	50	0.05→2	0.2→ 3.5	P	Opt	1000	Nonres	Mag core	TTY and punched tape	Yes	Machine

General comments and problems.- Test data management is a present day problem and is very likely to grow exponentially. If there were a single major problem to be extracted from the entire requirements survey, it would be in the area of data management. This was mentioned explicitly by more contacts than any other.

Many problems, although not stated in terms of data management, nevertheless had this as the root. For example, lack of correlation between qualification test results and acceptance test results can be partly ascribed to the problems of data handling and analysis. A suspicion about the validity of tests that are performed as a matter of course today was often heard; again, one of the limiting factors in verifying the usefulness of present tests, or in developing new tests that might better predict the health of a part, is the sheer practical aspect of acquiring, analyzing, and correlating the vast amounts of data.

Establishment of standardized testing practices, parameters, and stresses for increasingly complex and specialized circuits, such as LSI, promises to be a difficult and drawn-out proposition.

The lack of standards for testing in the correlated field of component assemblies today is illustrative of the problem to be expected. This particular area is presently characterized by a wide divergence of test requirements at the various assembly levels, both from company to company and within a company. Assembly level requirements vary from extremely detailed testing directed toward reverifying individual component parameters after assembly, to go-no/go, "bang-bang" testing designed to verify only that the unit is operating. The considerable and growing emphasis on the former type of testing (that of reverifying parameters) indicates a general distrust, by design engineers, of component level tests. It is interesting to note that 100% screening or other tests that would establish confidence in the mind of the design engineer are often not implemented for economic reasons; yet, assembly level tests substituted to establish such confidence are usually considerably more expensive. This contradictory position does exist on some scale within the aerospace industry. A justification advanced for piece-part testing after assembly is the possibility of component degradation due to the assembly process. Analysis of device failure mechanisms, taking into account the physics of the device and the assembly processes, is seldom undertaken except on a superficial and random basis; consequently, such justification is often not technically sound, and most often is simply filling the vacuum created by the lack of adequate standards.

Component testing today appears to be a relatively stable activity technically, except for MSI and the so-called high-reliability testing. Component manufacturers themselves do most component testing, both for their own in-process needs and to satisfy military specifications. Most users rely heavily on the component manufacturers for testing. Components are either purchased to the manufacturer's part number, to some standardized testing program of the component manufacturer's, or to a procurement specification released by the user. User testing is frequently limited to "receiving inspection" supplemented, in some cases, by a screening program on selected high-risk parts.

Component manufacturers use automatic test equipment almost unversally.

High reliability testing is, as a general rule, unpopular with component manufacturers because of:

1) The wide variation in requirements for different users;
2) The sometimes unrealistic test levels and exotic parameters;
3) Economics.

Life testing, extended burn-in requirements, and other tests that are imposed 100% on high-reliability parts create the economic problem. For a given investment in test facilities, the manufacturer can move considerably fewer parts on this basis than by sample testing. The difficulties created by the variance in requirements for high reliability testing is indicated by the comment of one manufacturer: "It is a very difficult matter to get people to treat 500 parts out of a million in a special way."

Many of the comments and problems noted above point up rather strongly the potential usefulness of a general-purpose, automatic test facility that allows the computer its proper role. That such a facility be general-purpose in design is an economic necessity. The cost of such an installation is not trivial. To be useful in solving the above problems, the computer will require a reasonable degree of power, certainly greater than the "mini" size and, for many applications, larger than the small-scale class. Reasonable input/output capability will be a necessity. For the more complex devices, the control circuitry in the test station becomes a significant cost element. Few installations will have funding available to duplicate such an installation for

each new kind of part as it reaches common use. (To a large extent, this is what is happening today; installations that have invested several hundred thousand dollars in an automatic integrated circuit test system may discover that they must now invest half again as much in a test system for LSI.)

New part types demand new test equipment. The most prevalent new types are MSI and LSI. These have a large number of interface points and the digital circuits will require a greatly increased size of test pattern. Along with increasing speed, these characteristics will force development of an entirely new generation of automatic test equipment beyond that being marketed today for integrated circuits.

A common problem today in assembly level testing that will move to the component level when LSI appears is diagnostic and malfunction-isolation capability. Techniques for detecting faulty junctions or "components" within the LSI chip must be developed for LSI to become economically practical. This is not in reference to user repair but to the ability to fabricate a working chip. This appears to be the most difficult problem associated with LSI testing. Almost certainly, sophisticated use of the computer will be necessary.

What might be called "sampled junction" testing is likely to be increasingly employed to indicate the integrity of a complete LSI device. This technique provides accessibility to several representative junctions on the chip so that complete parametric testing of those junctions can be extrapolated for the entire chip. This alleviates the problem of verifying process parameters common to the entire device (such as doping), but does not remove the necessity for complete functional testing or diagnostics for in-process repair.

A development that is often overlooked today is linear LSI devices. In all but two cases, the survey contacts assumed that LSI defined purely digital circuitry. The recent introduction of complex analog circuitry in single monolithic devices does not support this assumption. Such devices will make extreme demands on the flexibility of test equipment.

Concentration on LSI, however, is likely to leave one unprepared for the introduction of other new test devices. No crystal ball is needed to predict that there will be at least as many new kinds of devices developed in the next five years as were developed in the past five. Predicting exactly what these will be and when they will be introduced is extremely difficult. Reasonably likely or at least typical new members of the component family might include those described in the following paragraphs.

Coherent light devices: Optical memories are under development today; holography may become an entirely new means of computer input. Laser measurement devices are already finding their way into commercial use.

Active electromechanical components: Inclusion of the miniaturized electronic circuitry within what was a purely mechanical or passive electromechanical device is beginning. Switches with built-in ICs, tranducers with built-in miniaturized amplifiers, and relays with integral driving circuits are simple, present-day examples. If this trend continues, a relatively severe impact on component test equipment can be expected in marrying the mechanical and electrical stimulus and monitoring functions.

Hybrid devices: Particularly in the area of sensors, it is reasonable to assume that the future will show an incrase in types of devices that have inputs or outputs that are both electronic and nonelectronic in nature (i.e., alpha radiation as an input and an electronic signal as an output).

Long-duration missions will ultimately require a much more sophisticated approach to testing and data management than exist today. The criteria may very well change from, "Do the parts meet their specifications?" to "Which are the best parts out of a lot?" There is some feeling that present methods are simply not adequate for very expensive, long life spacecraft.

It is likely that the component test effort in general will remain divided between that performed by the component manufacturer to maintain process integrity and that done by the user to assure the quality of his purchased component.

As an indication of the actual problems and activities (present and future) occupying component test personnel, abbreviated comments are included in Appendix A.

It is interesting to note that a listing of the tests considered most effective by the survey contacts (table I) are heavily oriented to environmental (stress) testing. Electrical testing is, of course, necessary to verify that the device performs during or after stress application; nevertheless, today's electrical tests are obviously not considered sufficient to verify device integrity by themselves.

REQUIREMENTS ANALYSIS

General System Characteristics

For purposes of this study, characteristics are properties that describe the functional operation of a system. They form a basis for classifying real systems and are, in fact, derived from an analysis of existing test systems. The characteristics of most significance to this study are defined in this subsection, followed by the selection of characteristics required for the ERC system.

Computational/noncomputational.- The computational system is one that is capable of performing general arithmetic manipulations on stored data; the key word is "general." Hardware limit checking and similar fixed logic operations are noncomputational.

Data access.- Four classifications of data access exist:

1) Global - All data from previous or unrelated test operations are accessible in real time to an existing test operation;

2) Local - Only data obtained during the present test are available;

3) Nonresident - Provides no access to data other than the immediate values, but data are permanently available from the system (punched tape, cards) for off-line processing;

4) Transient - Provides no permanent data record in machine form. (Data logging systems that sequentially print data are in this category. The data are not accessible to a machine for processing.)

Program method.- Two types are recognized:

1) Stored-program - The system operates by sequentially accessing an internal memory into which instructions were previously loaded. Program modification is possible;

2) External program - The system operates by sequentially accessing instructions from an externally provided media or device such as punched tape, cards, or patch boards. The program cannot be modified by the machine during execution. Loading and subroutine programing techniques are not practical.

Conditional execution.- The major possibilities for conditional branching during program execution are:

1) Predefined,

   a) Fixed - The program proceeds in a completely predetermined sequence with no branching capability,

   b) Data-branch - Alternative program paths may be taken based on a branch decision made by the program on data obtained during execution,

   c) Operator branch - Alternative program paths may be taken based on an operator decision made during execution time. The operator decision is conveyed to the program by means of breakpoints, mode switches, or other simple "yes or no" means. All possible decisions and paths are preplanned and may be defined by a flow chart or tree;

2) Interactive - In a system providing interaction with the operator, program execution proceeds in a sequence that cannot be preplanned. The operator may spontaneously choose actions, branches, modifications, or new tests that are not predictable at program generation time.

Free-standing/network.- The test system may be directly interconnected with other computing systems such as a data bank. Such a network can be used to transmit test data from a small computer-controlled system to a large data processor for complex correlations or automatic updating of a data bank. The network, however, is not necessary to the normal operation of the test system, but may be considered a superset of the system. A free-standing system has no direct on-line interface with other installations, although magnetic tape or other media may furnish off-line cummunication.

Independent/dependent.- All processing and supporting activities such as program translations and data processing are contained in an independent system. Systems that require other installations to perform necessary tasks (such as program translation) are "dependent." The supporting installation must be considered in the system definition, together with the implications of priority of support, cost-effectiveness, and future conversions of the supporting installation.

Test hardware configuration.- All test equipment is permanently installed (hardware) in a fixed-configuration system. A variable configuration permits plug-in changing of the test equipment for a particular test.

Static/dynamic operation.- This characteristic is the most difficult to define, but is intuitively apparent when comparing systems. In systems permitting dynamic operation, the computer performs an active and essential role in the test measurement to the extent that the test equipment operation is not understandable without it. Synthesis of a wave shape by a digital-to-analog converter under computer control is a simple example of a dynamic function. Setup of a self-contained item of test equipment, such as a function generator, is static. A dynamic system attempts to minimize and generalize the hardware, and looks to software for gross sequencing and setup.

Selection of Characteristics

The characteristics required for the automatic test facility that is the subject of this study are discussed below and summarized in table VI.

TABLE VI

REQUIRED CHARACTERISTICS

Type of testing	Analysis method	Data access	Program method	Conditional execution	Relation to other systems	Test hardware configuration	Test control	Language
Receiving inspection	Either	Transient nonresident	Stored	Data and operator	Free-standing, dependent	Fixed	Static	Macro
Screening	Computational	Global	Stored	Data and operator	Dependent	Fixed	Either	Macro
Reliability	Computational	Global	Stored	Interactive	Network, dependent	Fixed	Dynamic	Compiler
In-process assembly	Computational	Resident	Stored	Interactive	Free-standing, independent	Either	Dynamic	Compiler
Depot	Computational	Resident	Stored	Interactive	Free-standing, dependent	Either	Dynamic	Macro, compiler
Experimental	Computational	Global	Stored	Interactive	Independent	Either	Dynamic	Compiler

Computational/noncomputational.- The system must be computational. Data analysis, logical operations for complex large-scale arrays, and minimization of special-purpose station hardware dictate this requirement.

Data access.- The method of test data handling needed will vary with the type of test. In the initial system uses, nonresident data may be sufficient; as diagnostic capabilities are developed, local data access will be needed. Some degree of global data access will be required for reliability usage.

It is recommended that initial implementation provide local data access. The very slow output provided by punched paper tape will be an unnecessary burden in many test runs when the decision on whether the data are significant and should be retained may not be made until after test completion.

Providing local data access permits this decision (and consequent investment in time) to be made on test completion. Later phases of the system should provide global data access to the extent necessary to detect trends or correlations necessary for real-time operation.

Program method.- Externally programed devices, while suitable for rigid sequence operations such as numerical control, some types of production testing, and other operations requiring a very low level of operator participation, are unsuited for the majority of test categories in present applications and totally unsuitable as solutions to future problems and requirements. Stored programed systems providing looping, branching, and program modification capability, with a potential for a high degree of operator interaction, are required.

Conditional execution.- Fixed sequence systems may be disposed of with the comments above. For the ERC application, some degree of interaction is a fundamental requirement from the beginning. In fact, interaction is probably more significant in the beginning stages than in the final configurations; the otherwise limited central processor and software will impose very severe penalties in time when every change necessitates a reprocessing of a predefined program or experiment.

Free-standing/network.- No firm requirement can be established for on-line interconnection with other systems, especially in the earlier phases. The choice is not clear cut; high data output operations such as continuous monitoring during reliability testing post data management problems regardless of the acquisition technique. Useful reduction methods are not really established, and it is expected that considerable developmental activity will take place along these lines in the next five years. Because of economic considerations, it is assumed that a free-standing system will serve the purpose initially, with local reduction, correlation, or trend detection to some level. Suitable output media for residual data will be provided as required.

Independent/dependent.- This characteristic is probably the most difficult of the list to establish. The decision is primarily economic and operational. The independent system offers numerous operational advantages if an adequate processing capability is available. On the other hand, the apparent initial cost advantage of a dependent system can rapidly disappear in conversion costs when either the supporting or the test installation is changed.

The recommendation is that all operations directly affecting the test itself (such as program translation) should be included within the system as independent functions; operations that do not directly support the test, such as posttest data analysis, formatting, or filing can be deferred to a supporting installation on the basis of individual economic tradeoffs.

Fixed/variable.- There is little advantage to a variable configuration system where only a single or several test stations are used. The primary advantage to a variable configuration is attained when a large number of special-purpose test channels may be circulated among a number of test stations (to avoid providing the total number of test channels in each test station). This is normally characteristic of production or depot test operations rather than the classes of testing addressed by this study. A fixed configuration system is recommended.

Static/dynamic.- This characteristic affects the test station, the input/output control, the method of command and result access, and the conceptual structure of the test monitor and test language. The primary hardware implication is in terms of speed, i.e., the bandwidth of the test unit/test program communication loop. Careful design and sound conceptual structure is most important; there is little if any penalty in terms of hardware cost. Therefore, full dynamic operation has been assumed as a characteristic of the system even though initially the full use of this capability may not be required. The economic advantage lies with building the capability into the system architecture from the initial stages, even at the price of potentially somewhat higher software and system integration cost. Conversion later to provide such capability can be prohibitive. The long-term economic advantages in minimization of special-purpose hardware cost and technical obsolescence are overriding.

Major Configurations

From the standpoint of general system architecture, three major configurations are presently possible:

1) Dedicated;

2) Time-sharing;

3) Master/slave.

These configurations are illustrated in figures 1 thru 3. Theoretically, these variations in architecture do not effect or constrain any characteristics of the system. However, in practical terms, a very significant interrelationship comes into play when cost is introduced. The dedicated configuration can certainly be given central processing capability as powerful as a time-sharing configuration. It will then be inefficient in terms of computer use and, in many applications, not cost-effective. The processing capability will either be scaled down, which will change the characteristics, or additional test interfaces will be added and time shared.

In a practical sense, each configuration implies certain operating considerations normal to that configuration. Some of these are briefly discussed in the following paragraphs.

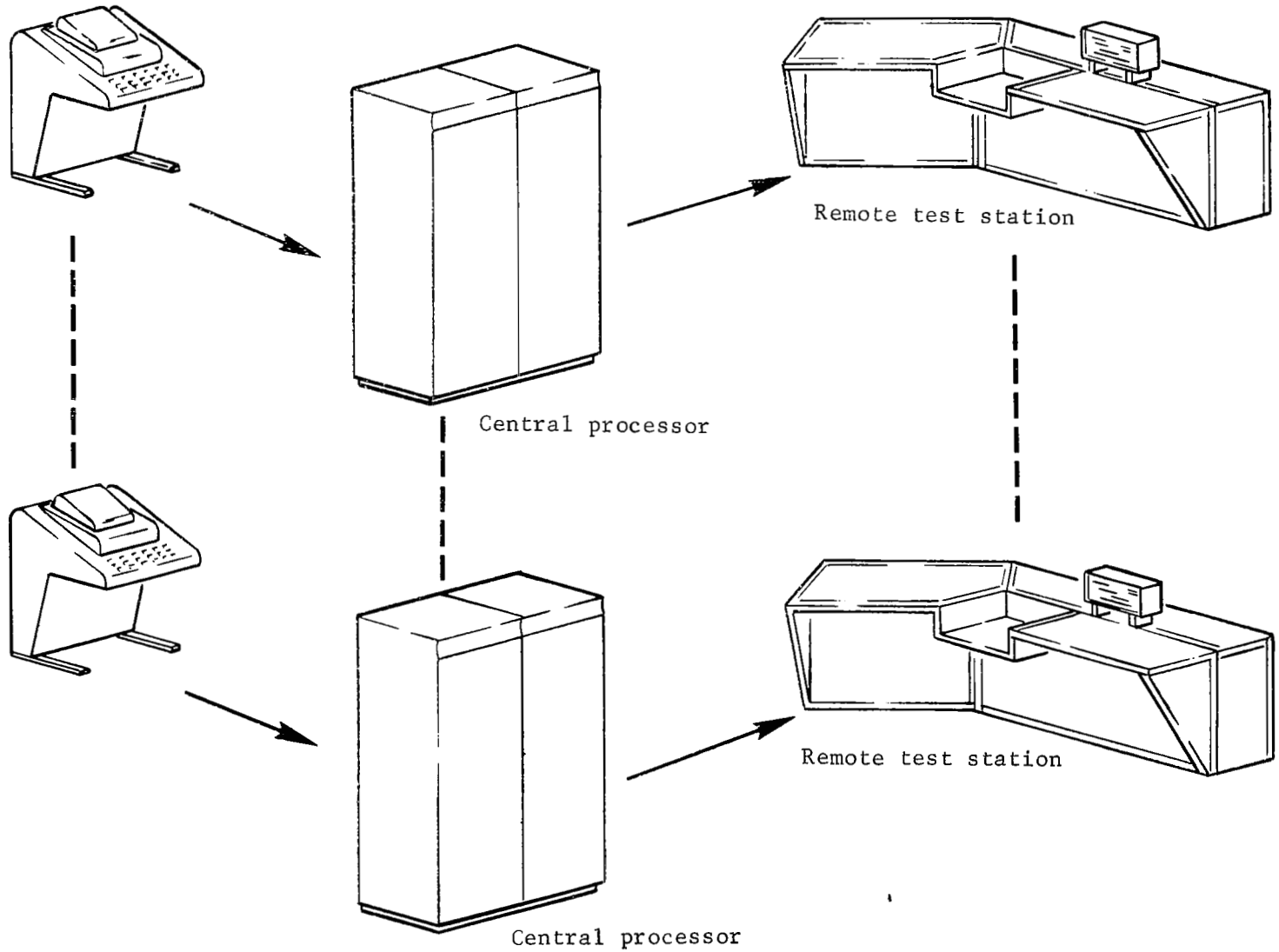


Figure 1.- Dedicated Computerized Testing

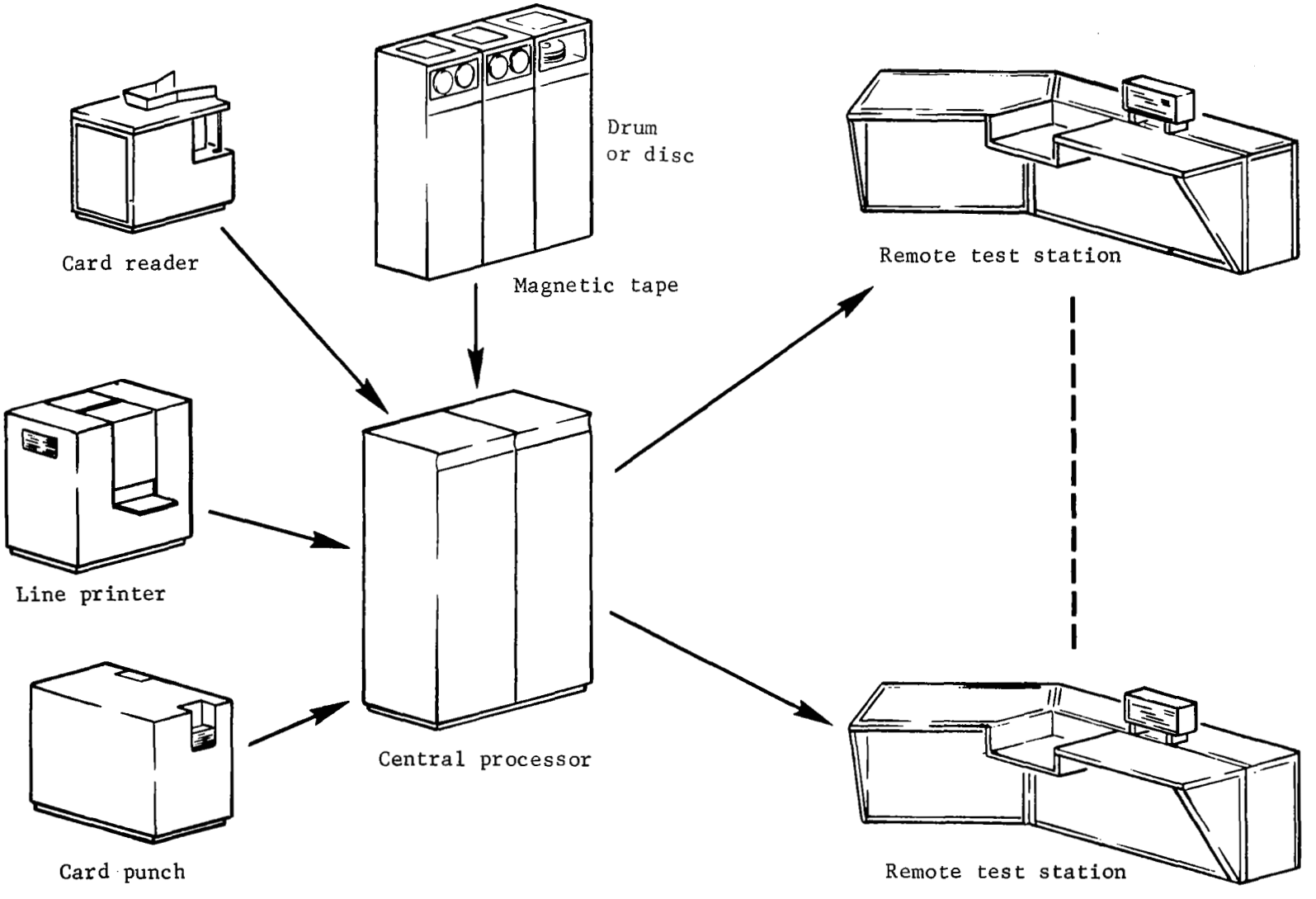


Figure 2.- Time-Shared Computerized Testing

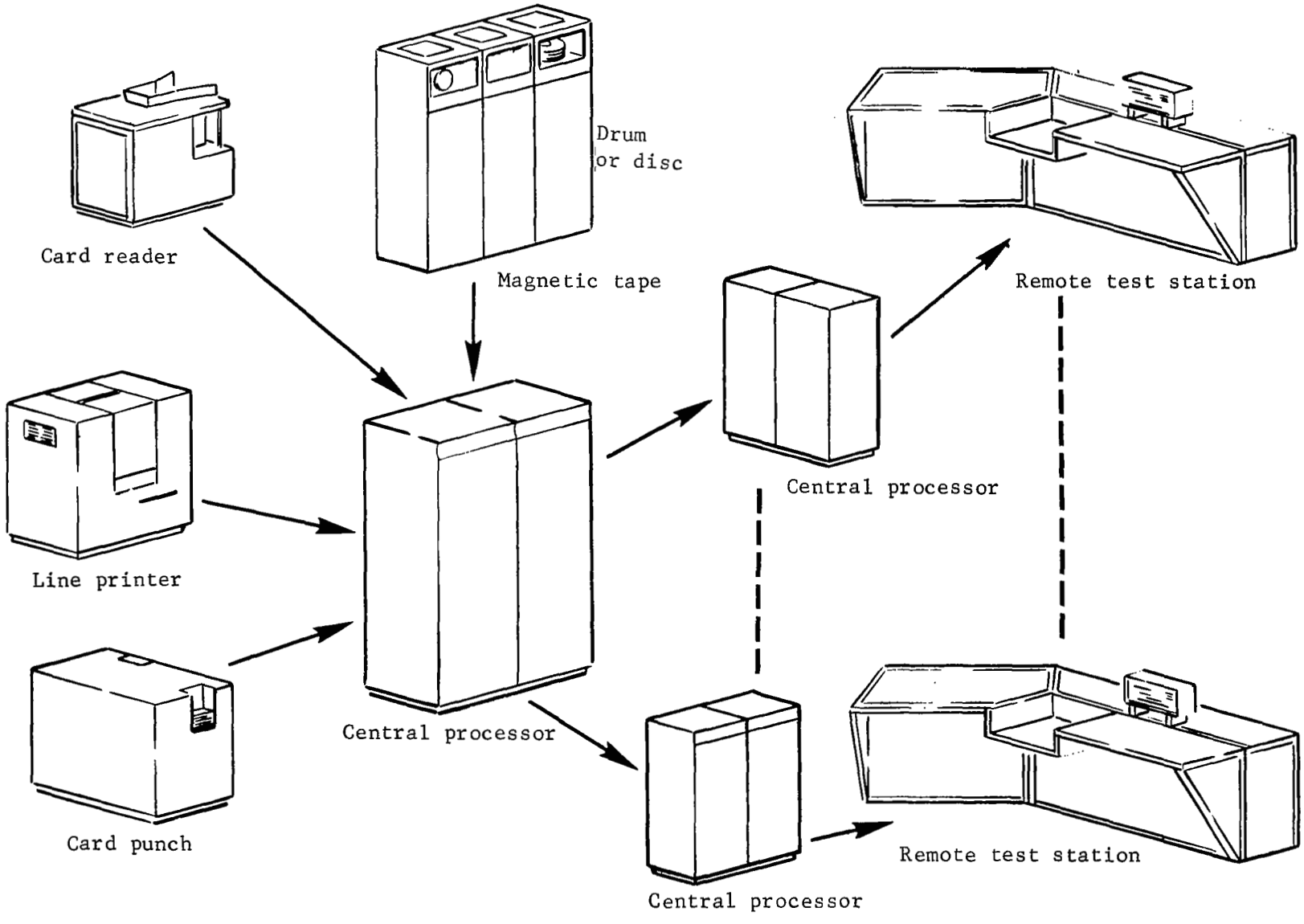


Figure 3.- Master/Slave Computerized Testing

Dedicated.- A free-standing computer for each test terminal implies the following operating considerations:

1) Locations may be widely scattered;
2) Dependent on other installations for processing;
3) Limited data access;
4) Less initial implementation, increased application cost;
5) Probably minimum interaction;
6) Unlimited expansion of terminal quantity.

Time sharing.- This configuration consists of multiple test terminals controlled by a single central processor so that a terminal or a test proceeds as if it had the full capability and attention of the central processor. The configuration implies the following operating considerations:

1) Remote terminal location distance limited by speed of test progression;
2) Independent of other installations for support processing;
3) Offers scientific/data processing auxiliary usage;
4) Can provide global data access;
5) Higher initial implementation cost, efficient application;
6) High-level interaction;
7) Practical limitations on terminal quantity.

Master/slave.- Multiple dedicated systems controlled by a single central processor imply the following operating characteristics:

1) Bandpass wider than time-sharing. However, decrease in data rates for the central computer is sometimes offset by decreased power of dedicated computer;
2) Can be implemented incrementally, but total cost is higher;
3) Less efficient than time shared system in use of total computing capability;
4) Other considerations same as time sharing.

General Configuration

If certain precautions are taken in the initial design, there is a relatively small penalty to pay in converting from a dedicated system to a time-sharing system. These precautions may be summarized as follows:

1) Direct memory access is used;

2) External registers and associated logic are provided for accessing test station commands and storing results without program attention once initialized;

3) Test programs do not address test stations directly, but request a monitor to perform station operations;

4) The monitor does not address test stations directly, but addresses the external registers and logic for that station;

5) Test programs do not engage in input/output operations, but request all such functions from a test monitor;

6) The test monitor provides at least the skeleton architecture for time-sharing functions even though these are not implemented initially. These include such functions as dynamic memory assignment, device assignment and sequencing, time-critical operations assignment, relocatable load and linking, and input/output handlers that use buffers and terminal interrupts so that program attention is not required during input/output.

In short, the dedicated system should be a subset of a time-shared system. If this design philosophy is followed it is feasible and probably advantageous to begin with a dedicated system in the initial development stages. Beginning with a dedicated system that is designed around a single application, on the other hand, will almost certainly result in severe economic penalties in the usage required by ERC.

The addition of more test terminals to the system can, of course, be handled by alternative physical connection of the additional stations; and in a batch-process limited system such as Configuration A1, this may be practical for at least an interim period if test rates are low. Sooner or later the conflicts between uses become intolerable, and concurrent testing at all stations must be used or additional central processors must be provided. Both approaches are explored in the detailed configurations; the obvious economic advantage is with time sharing.

Data rates or analysis complexities for all requirements within the scope of this study can be handled by an efficient time-shared system. If overhead is properly minimized and buffered IO is used exclusively, the system does not become computer limited until almost the maximum expansion capability has been achieved. Therefore, there appears little need for a master/slave configuration (i.e., an auxiliary computer located at, and dedicated to, an individual test station) unless a low-bandpass modem interface to a remote user is provided. In this case, if the test requires a high bandpass, a very small (mini) computer in the remote station provides local reduction of data and time-critical control. This is illustrated in Configuration D1. The very small computer that is integral with the test station is capable of controlling nearly all test station functions presently conceivable; but it is extremely limited in provisions for language translation or other "batch" functions, especially when compared with the master central processor. Where significant batch processes are required at the remote installation, the most efficient power vs cost approach is to install a remote batch terminal that also communicates with the master processor via a modem interface. This will make available the full power of the central installation to the remote user at a moderate through-put rate.

Recommendations for the major system configuration are as follows (table VII):

1) The initial implementation may be dedicated, but designed as a subset of a time-shared configuration;

2) Additional test stations may be handled on a manually switched basis until usage conflicts become intolerable. At that point, the system should be converted to time sharing;

3) The system should be capable of eventually driving multiple remote installations through a remote interface for both test and batch activities.

TABLE VII

REQUIREMENTS BY TYPE OF TESTING

Type of testing	Language (a)			Inter-action (b)	Data (c)				Malfunction isolation (d)	Batch operation (e)	Interface (f)
	New tests	Flex	Power		Quan	Rate	Dur	Comp			
Receiving inspection	3	1	1	1	1	3	S	1-2	0	1	0
Screening	2	1	1	2	2	1	L	1-2	0	1	0
Qualification	2	2	2	2	2	1	L	2	0	2	0
Reliability	1	2	2	2	4	2-4	L	4	0	2	1
Component in-process	2	2	2	1	1-3	4	S	1	0	1	2
Assembly in-process	4	3	3	2	2	2	S	2	3	4	1
Laboratory automation	2	3	3	3	2	1-3	S-L	1-4	1	3	0
Experimental	4	4	4	4	2	1-4	S-L	2-4	2	4	3

[a]Language:
New tests - Frequency of writing new programs;
Flex - Variety of test functions;
Power - Need for simple expression of complex functions.

[b]Interaction - Degree of on-line operator control, changes, and decisions required.

[c]Data:
Quan - Relative daily output of data;
Rate - Required speed of data output;
Dur - Period of duration of continuous data output -- S = short duration (less than ½ hr)
L = long duration (more than ½ hr);
Comp- Complexity of data analysis required on-line.

[d]Malfunction isolation - Level of malfunction isolation capability required.

[e]Batch operation - Percentage of use for batch processing (program translation, etc) -- 0 = 5%, 4 = 80%.

[f]Interface - Level of requirement for interfacing with other computer programs, such as computer-aided design.

CONFIGURATION STUDY

The requirements survey established the general parameters of electronic component testing as practiced in the aerospace industry today. The requirements analysis developed the characteristics of an automatic test system from the survey parameters. This section will define a series of candidate configurations, evaluate each, and select the configurations that best meet cost and technical guidelines.

Approach

Minor variations are possible in each configuration; in some cases, one configuration can grade into another without a sharp dividing line. Furthermore, it becomes rapidly apparent that no single configuration will meet all requirements and constraints. Tradeoffs between additional stations, language capabilities, core size, input/output (I/O), and operational factors become very complex if all possible combinations are considered. To reduce the tradeoff study to reasonable proportions, the following ground rules will be followed:

1) Many potential configurations are illogical or grossly inefficient (e.g., a large time-sharing installation with extremely limited I/O capability); each configuration treated in this section will be reasonably balanced between I/O, test terminals, computing power, and software.

2) Each configuration will represent a specific design solution to a set of requirements; variations within configurations will not be shown. This does not imply that expansion is limited to the configurations depicted. Certainly many expansion steps are possible within configurations, and, in practice, such intermediate steps are frequently desirable.

3) An important influence on any computer configuration is the standard software available from the computer manufacturer. This is frequently available in various levels of capability, each with fixed requirements for hardware. This tends to predefine plateaus in hardware configuration. A typical series of operating systems available from computer manufacturers is shown in table VIII.

4) The equipment and software will not be restricted to a specific supplier or to known off-the-shelf items.

TABLE VIII

TYPICAL SOFTWARE SYSTEMS

Level	Description and functions	Use	Recommended hardware
S1	Basic monitor: Fortran, assembler, simple link and relocating loader, library routines. Many standard programs are free-standing (i.e., binary load from input). Simple control message servicing.	Relatively simple processing and assemblies. Small Fortran jobs, data acquisition, testing.	Usually minimum usable for computation. 8K, I/O device (may be paper tape); drum/disc or two tape units; listing device.
	Test monitor: Free-standing, not foreground; must provide most service, utility, handling functions. May be limited time-sharing.	May time-share several test stations if hardware-assisted.	
S2	Real-time monitor (RTM): Background/foreground; translation or other batch concurrent with real-time test. Compilers, overlay loaders, swapping (checkpoint), file management, more powerful control.	Complex processing-scientific, compilations, and data analysis, concurrent with data acquisition, test, or control.	16 to 24K minimum; high-speed drum/disc; I/O device; listing device; interrupts; real-time clock.
	Test monitor: Operates under RTM, simpler than S1 because relies on RTM for I/O, service, handling functions.	Capable of full time-sharing of multiple test stations, and sophisticated data analysis and control.	
S3	Augmented real-time monitor: Provides all of S2 plus complete drum management, remote batch, "symbiont" concurrent I/O. Considerably more efficient and powerful.	Remote multiplexed batch operations, concurrent with time-shared test.	24 to 32K minimum; 6 Mbyte drum; card reader, line printer, magnetic tape; memory protect; interrupts; RT clock; multiple ports, register block, and I/O processor.
	Test monitor: Same comments as S2. Will require revision from S2.		
S4	Time-sharing monitor: Provides all of S3 plus on-line, interactive terminals for user data processing.	May be used to provide test stations and most software processing to many remote users in time-sharing.	48K; others same as S3.
	Test monitor: More complex than S3 because of dual time-sharing		

The study is separated into test station, language, and system configurations. Operating system tradeoffs were not deemed necessary because these are inherent in the system configuration.

Test station.- The intent here is to develop an optimum architecture, rather than specific designs. The primary consideration is to make the test station as independent as possible of test interface parameters and the computer and software system driving it. It is assumed that a test station that is uniquely designed for a specific application offers no useful advantage over existing equipment.

For purposes of the cost-effectiveness survey, typical costs of adapting a general-purpose station to a category of testing have been derived and are presented in the appropriate section; for system configurations, it is assumed that the test station cost is the same regardless of the system used.

Language.- The language used by the test engineer or researcher to define tests in a step-by-step fashion to the machine is heavily dependent on the computing power available. Six language configurations are defined, and the characteristics of each are described. Ideally, one should select the language level and operating system necessary for the job and acquire the hardware necessary to support it. In practical life, the reverse is often true. More compromises to economics have been made in this area than in any other of the study.

System.- In this section, the test station and language are treated as black boxes. The language is costed as part of the system, as is all other necessary basic system software. The test station is costed for reference but is not included in the system tradeoff analysis because its specific configuration is independent of the system.

The basic assumption is that system usage and requirements will grow; the primary focus of the study is to select the optimum path between the immediate economic practicality and the ultimate need.

System configuration presentation.- The system configurations are presented consecutively, each with a verbal description and remarks, a block diagram, and an equipment list. A summary is provided at the end of the detailed configurations.

Test station configurations are depicted and explained in figures 4 thru 6. These are referenced throughout the system configuration equipment lists.

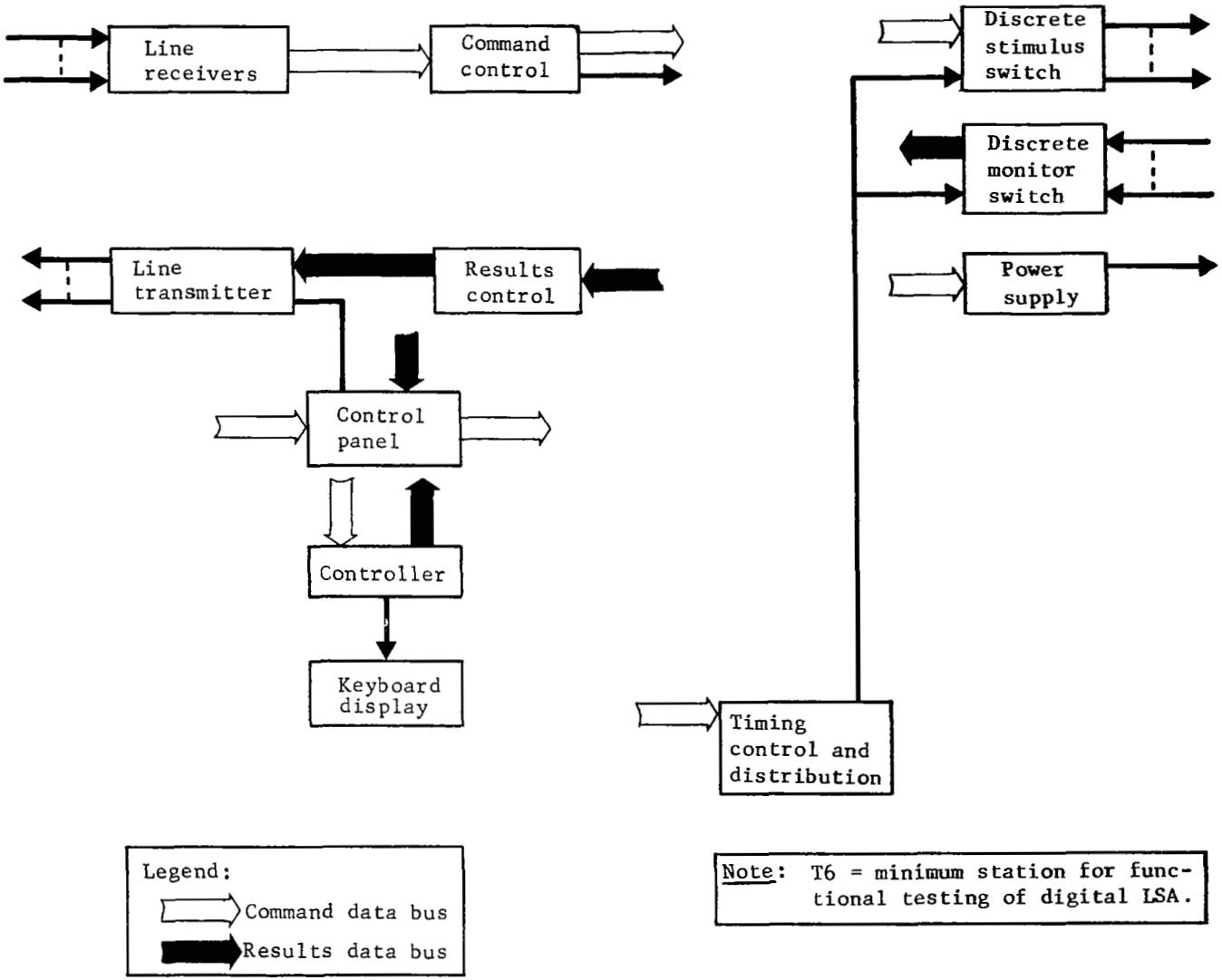


Figure 4.- Test Station Configuration T6

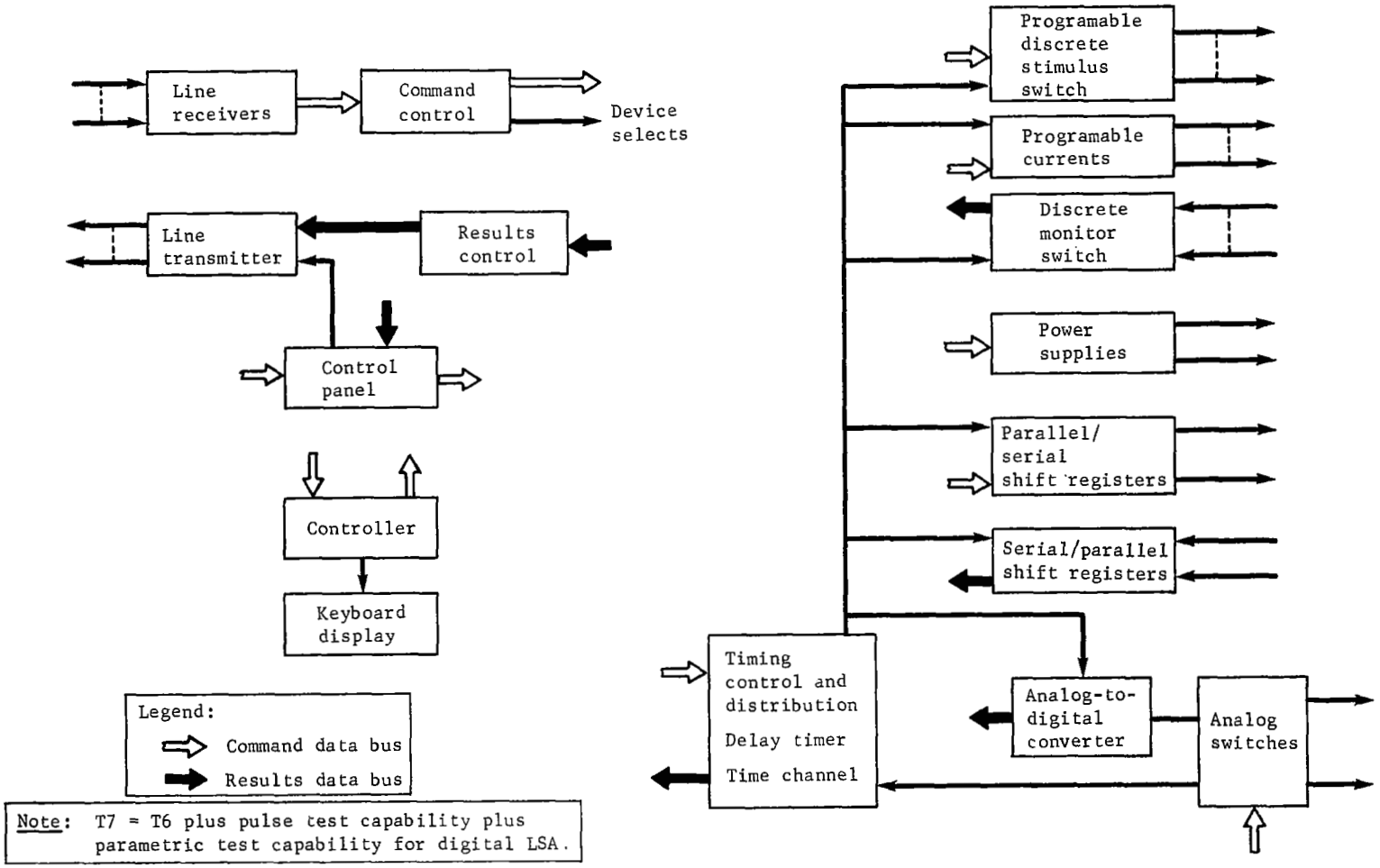


Figure 5.- Test Station Configuration T7

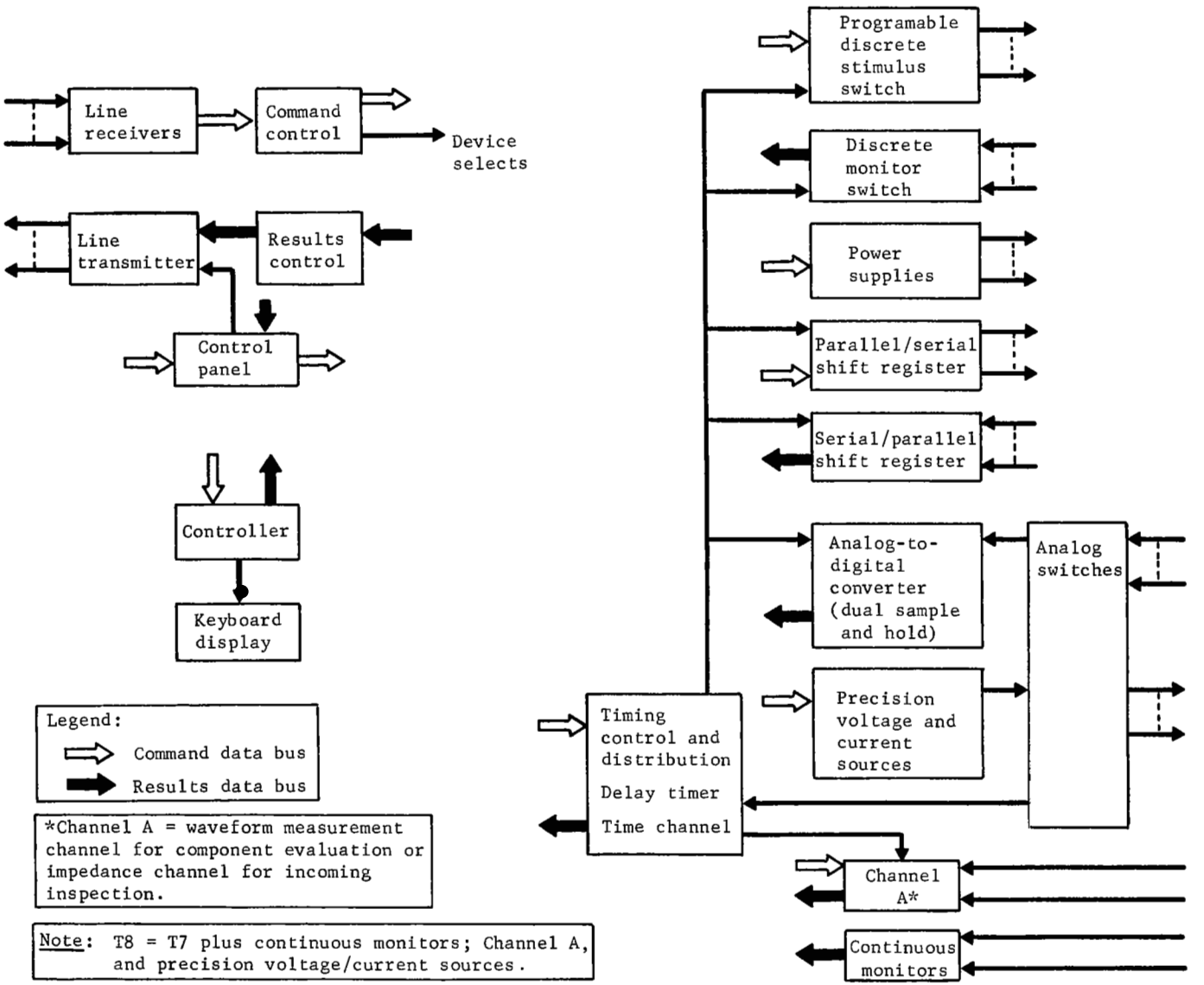


Figure 6.- Test Station Configuration T8

Configuration A1 - Minimum Dedicated System for Functional Test of Digital Large-Scale Arrays

Description (see Table IX and fig. 7).- This is a minimum configuration using available CPU equipment in a dedicated configuration to drive a single test station. A minimum test language configuration and test monitor operate within the system. The usage is experimental functional testing of digital large-scale arrays up to 100 logic nodes and 150 interface points.

Remarks.- The test station configuration is T6 (functional test only). A separate input/output (I/O) control provides compatibility with the computer. This architecture is necessary for the test station to be compatible with later phases.

The small-scale computer environment will support only a quite simple test language, either of the macroprocessor or interpretive type, and limited data analysis.

An overlay loader or additional core will be needed for anything more than simple test programs.

Interaction will be limited, but probably will be initially adequate.

As configured, the system will be inconvenient and time-consuming to use because of the limited peripherals, computing capability, and supporting software. It is estimated that 70% of the total operational time will be devoted to batch processing, translations, listings, and other make-ready or supporting functions, rather than test. The language level is not adequate for experimental use over the long range. However, these limitations may be tolerable for a short period in the initial development stages. There is some advantage in starting with a simple, bare-skeleton system.

A listing output of a 2000-line program will take from $1\frac{1}{2}$ to $3\frac{1}{2}$ hr at the standard TTY speed of 10 characters/sec.

TABLE IX

CONFIGURATION A1 EQUIPMENT LIST

	Cost, $
CPU hardware	
Honeywell 516 computer with 8000-word memory	31.8×10^3
ASR-35	1.2
Drum, 2 Mbyte	28.5
Interrupts (4 additional)	1.6
Real-time clock	1.6
Direct memory access, 1 channel	7.0
Additional DMA channel	1.5
SKS/OCP (16)	1.6
Parallel and output/input	3.2
Arithmetic package	3.0
Additional 4000-word memory and memory parity	11.0
Subtotal	92.0
Less available	77.6
A1 total	14.4
Station hardware	
Input/output control	15.0
Test station (T6)	65.0
Subtotal	80.0
Software	
Test language (Configuration L1)	27.0
Test monitor and service routines	25.0
Subtotal	52.0
Other items	
Documentation and training	6.0
System integration	12.0
Subtotal	18.0
Total	164.4

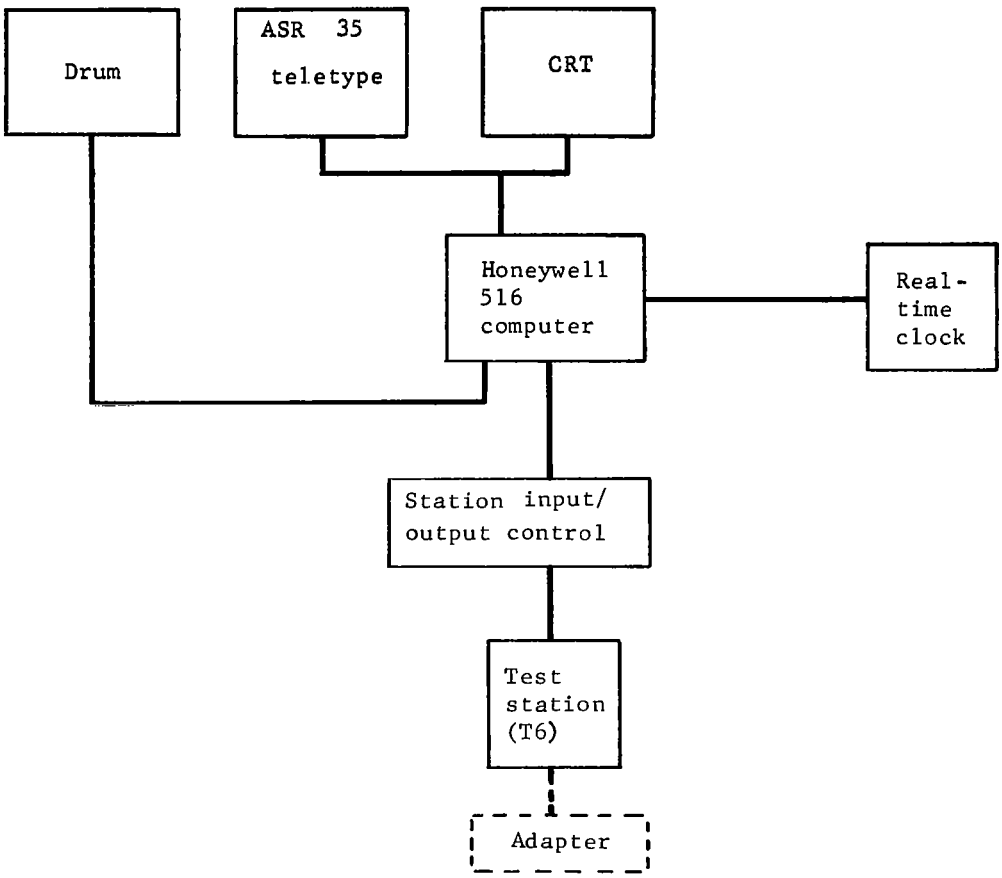


Figure 7.- Configuration A1 Hardware

Configuration A2 - Configuration A1 with Addition of Parametric Test Capability for Digital Large-Scale Arrays

Description.- The minimum CPU, language, and monitor of Configuration A1 is retained. Parametric test capability requires only the addition of analog stimulus and measurement test channels to the test station, and the incorporation of the corresponding statements in the test language.

The addition of parametric test capability for digital large-scale arrays to Configuration A1 will require changes to the test station and test language only (table X). The system block diagram is identical to A1 (fig. 7).

TABLE X

CONFIGURATION A2 EQUIPMENT LIST

	Cost, $
Configuration A1	164.4×10^3
Addition of parametric capability to test station (T7 configuration)	
Basic cost	45
Field modification	5
Addition of new functions to test language	18
Delta total	68

Remarks.- This modification of the test station can vary from an efficient, modular expansion to a complete redesign and rebuild, depending on how well the basic station design is implemented. The figures used herein are based on proper implementation of the test station architecture specified in the section on "Specifications."

Configuration B1 - Minimum System for Time-Sharing of Functional and Parametric Test Stations

Description (see table XI and fig. 8).- This configuration adds the capability for time-sharing two stations to the dedicated system of Configuration A1. The addition of time-sharing has no effect on the test station or computer hardware itself. The changes involve the following:

1) Input/output control - Expanded to interface with two stations and resolve conflicts due to simultaneous access requests;

2) Test language (interpreter) - Provide housekeeping and restructuring for time-sharing, and additional functions for parametric capability;

3) Test monitor - Major effect. The monitor must maintain status of each concurrent operation (including interpretation), resolve conflicts between simultaneous requests for the same device or routine, provide for time-critical operations, and maintain service queues and similar functions unique to time-sharing.

A paper-tape reader, additional core, and a high-speed teletype are provided to support the increased usage. Teletypes are provided at the stations so that every user can be on-line simultaneously.

Remarks.- Definite constraints are placed on the language interpreter by time-sharing within the limited environment of this configuration. The most important are that each program segment must be self-sufficient (i.e., no references between segments); time-critical operations must be contained within one segment; and only one user's segment may be interpreted at a time. These items are discussed more fully in the language section.

Use of a macroprocessor (offline translation) in a time-shared configuration rapidly gets out of hand unless relatively powerful background/foreground processing is provided. Conflicts between test use and batch processing become so severe that a macroprocessor was not considered.

TABLE XI

CONFIGURATION B1 EQUIPMENT LIST (DELTA TO A1)

	Cost, $
CPU hardware	
Configuration A1 CPU list (total, reference)	$(92.0) \times 10^3$
High-speed teletype and coupler	11.0
Additional four priority interrupts	.4
Paper-tape reader	3.0
Additional memory (8000 words)	16.0
Parity for additional 8000 words	3.5
Total CPU	33.9
Station hardware	
Addition to I/O control for time-sharing	6.0
Addition of teletype and coupler (ASR 35) to TS 2	8.0
Modification of existing control ASR 33 for TS 1 use (includes coupler addition)	4.5
Second Test Station (Parametric Capability)	110.0
Subtotal	128.5
Software	
Test monitor addition for time-sharing, interaction, and interface with interpreter	16.0
Interpreter modification for time-sharing	18.0
Addition of new functions to interpreter (parametric)	10.0
Subtotal	44.0
Other items	
Documentation and training	4.0
System integration	12.0
Subtotal	16.0
Total	222.4

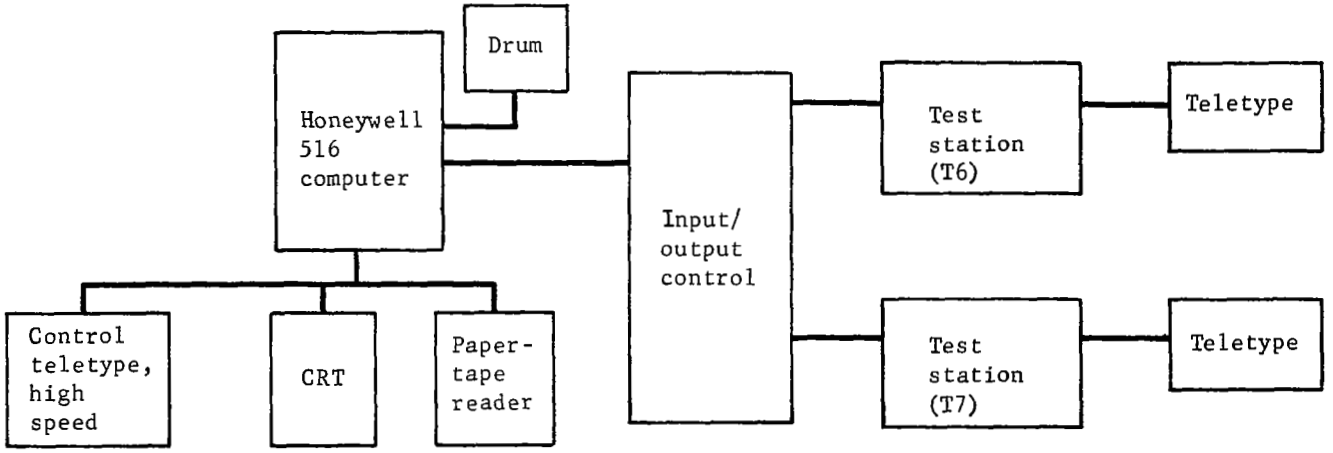


Figure 8.- Configuration B1 Block Diagram

Configuration B2 - Addition of Reliability Testing

Description (see fig. 9).- This configuration adds a test station and incorporates a typical new requirement. The addition is costed as a delta to Configuration B1 (table XII) for a three-station configuration, and as a delta to Configuration A2 (table XIII) for a two-station configuration.

The new requirement selected was a reliability testing program similar to that contemplated by the Device Research Branch at ERC. A group of components (transistors, diodes, ICs) are continuously monitored during the long-term application of stress (temperature, etc) to detect parameter variations that may correlate with device reliability.

Modular increments to memory and I/O control are implemented to service the third station. The data analysis and output requirements implicit in the reliability usage make necessary the addition of magnetic tape, the addition of functions to the monitor and language, and a second high-speed teletype.

Remarks.- The major additional technical problems are:

1) Test station interface requirements;
2) Data analysis and reduction in real time;
3) Time demands on the system due to continuous monitoring.

Additions to the CPU are required to provide:

1) Increased core size;
2) Data reduction and analysis;
3) A higher-speed data output device. For the quantity of data expected, paper tape is not feasible and, therefore, a magnetic tape is added.

The I/O control is modified to control three stations in a time-sharing mode.

No change occurs to the existing test stations. The new test station incorporates those test channels required for reliability testing.

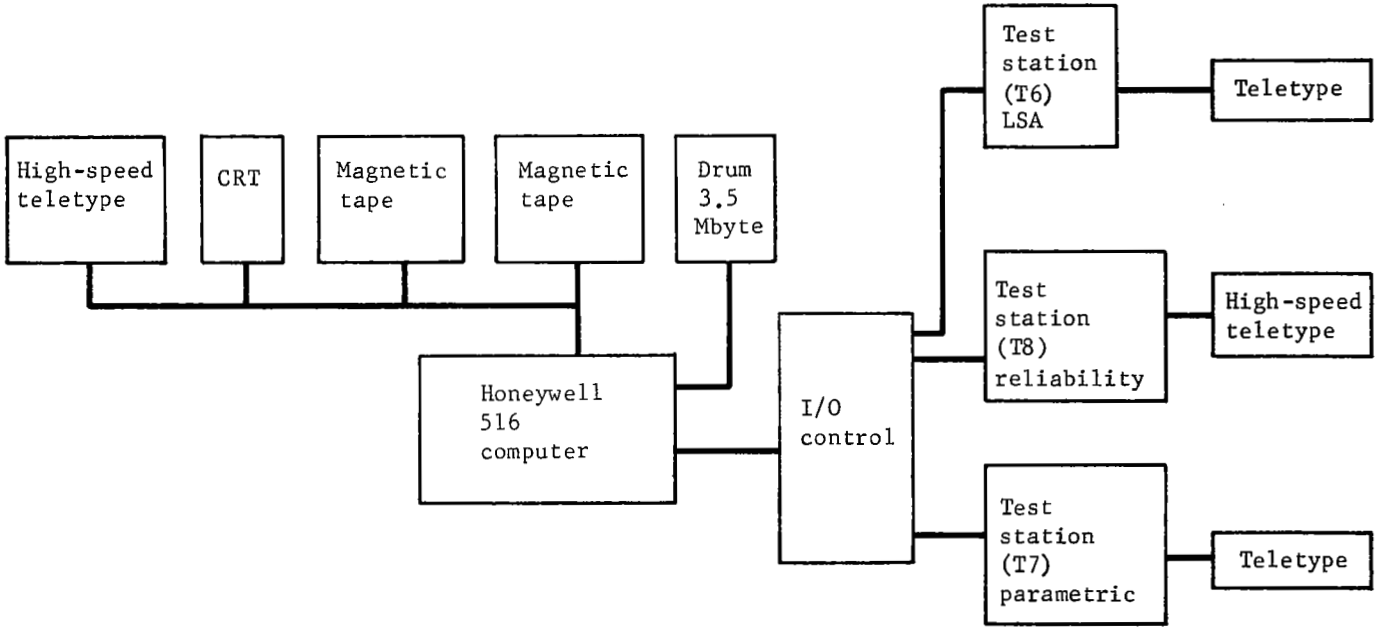


Figure 9.- Configuration B2 Block Diagram

TABLE XII

CONFIGURATION B2 EQUIPMENT LIST (DELTA TO B1)

	Cost, $
CPU hardware	
Additional 8000 words of core for total of 24 000	16.0×10^3
Additional 1.8 M of mass storage for total of 3.6 M	17.4
Magnetic tape, 36 ips, 9 track, 2 deck, 270/556	36.6
Parity for additional 8000 words of core memory	3.5
Additional priority interrupts (4)	.4
	73.9
Station hardware	
Addition of high-speed teletype and coupler at remote station	11.0
Modification to I/O control for three stations	3.5
T8 station	110.0
	124.5
Software	
Addition to time-shared interpreter for new functions, data analysis, recording	20.0
Addition to test monitor for tape handling, data analysis, recording, and interaction	15.0
Data analysis drivers	12.0
	47.0
Other items	
Addition for documentation and training	7.0
System integration	6.0
	13.0
Total	258.4

TABLE XIII

CONFIGURATION B2 EQUIPMENT LIST (DELTA TO A2)

	Cost, $
CPU hardware	
8000 word core increment (quantity 2)	32.0×10^3
Additional 1.8 M of mass storage for total of 3.6 M	17.4
Magnetic tape	36.6
Parity for 16 000 words	7.0
High-speed teletype (1)	11.6
	104.1
Station hardware	
Same as table XI	128.5
Software	
Test monitor addition for timesharing, tape, reliability test	24.0
Interpreter for timesharing + reliability	28.0
Data analysis drivers	12.0
	64.0
Other items	
Same as table XI	16.0
Total	312.6

Configuration B3 - Dual Configuration for Addition of Reliability Testing

Description (see fig. 10 and table XIV).- This configuration examines the cost and practicality of an independent system, similar to Configuration A1, for reliability testing. Because the hardware is essentially duplicated, the variable of interest here is savings in development time.

A tape unit, larger memory and drum, and additions to the test language and test monitor are included. The test station configuration is T8.

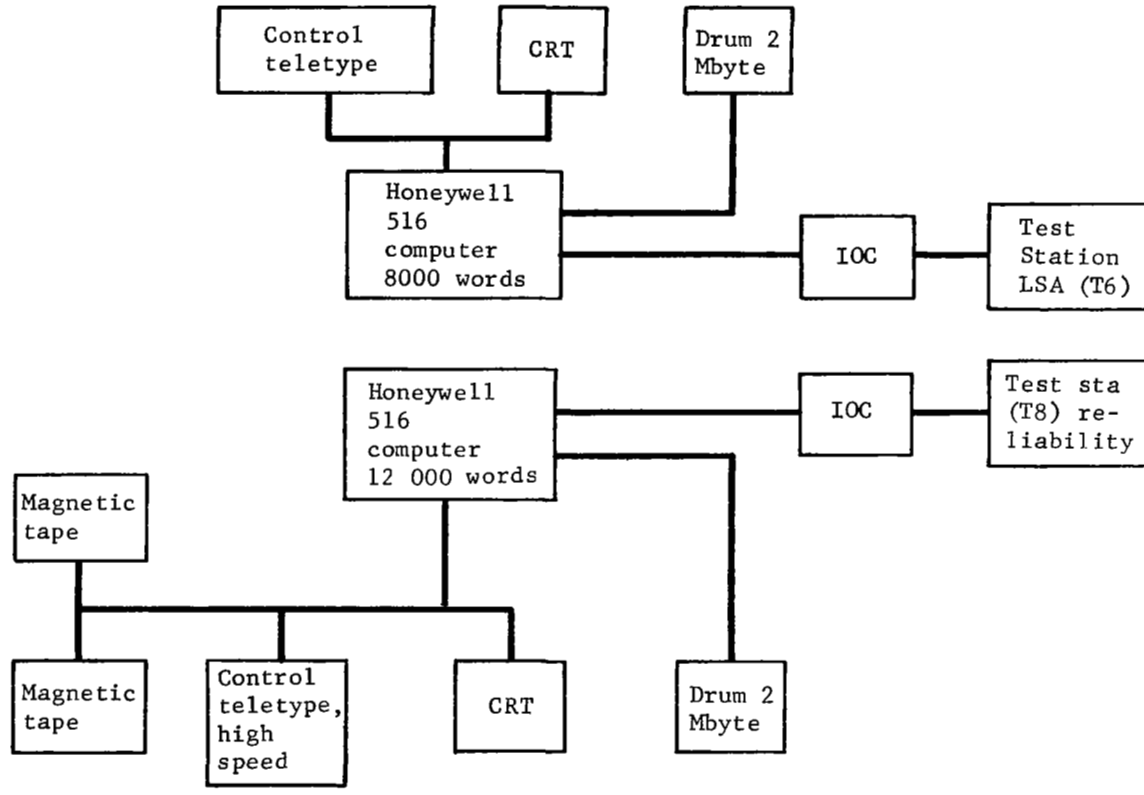


Figure 10.- Configuration B3 Block Diagram

TABLE XIV

CONFIGURATION B3 EQUIPMENT LIST

	Cost, $
CPU hardware	
Configuration A1 hardware (less drum)	54.8×10^3
Additional 4000 words of core memory	8.0
Parity for additional 4000 words of core memory	2.0
Magnetic tape, 36 ips, 4 track, 2 deck, 200K 56 for data record	36.6
High-speed teletype	11.0
Fast access disc, 300 000 words	39.0
	151.4
Station hardware	
T8 station	110.0
Additional IOC	10.0
	120.0
Software	
Addition to Configuration A2 interpreter for new functions and analysis	12.0
Data analysis drivers	12.0
Addition to monitor for tape handler and additional (limited) interaction	15.0
	39.0
Other items	
Addition for documentation and training	4.0
System integration	5.0
	9.0
Total	319.4

Configuration B4 - Time-Shared LSA and Parametric Testing with Supporting Installation for Software

Description (see table XV).- This configuration attempts to overcome limitations in translation of programs and batch processing by implementing the language translator and other software on a separate existing installation. The test system is used only for test. Data analysis requirements, time sharing software and hardware, and other functions remain similar to those for Configuration B1.

Remarks.- Implementation of test software in an outside computer installation changes the basic system characteristic from "independent" to "dependent." The implications of this are described under "Characteristics" in the Requirements Analysis Section.

Given a larger supporting installation, a more powerful translator is possible. However, the same inherent compatibility requirements apply when the test system must be expanded beyond the capabilities of its computer, i.e., a major revision or complete throwaway of the translator is necessary unless a meta-compiler technique is used.

Three translator implementations are costed as alternatives:

1) A set of language procedures, assuming the basic procedure processor and hardware is available;

2) A specially written compiler;

3) A set of language specifications, assuming a meta-compiler is available.

TABLE XV

CONFIGURATION B4 SUPPORTING INSTALLATION FOR SOFTWARE

	Cost, $
CPU hardware -- Same as Configuration B1 (table XI) with the following changes:	
Delete "high-speed teletype and coupler"	
Delete "paper tape reader"	
Station hardware - Same as Configuration B1 (table XI) with the following changes:	
Substitute "high-speed teletype and coupler" for "modification of existing control ASR 33 for TS 1 use"	
Software	
Test monitor addition for time-sharing and interaction (same as Configuration B1 less interpreter)	14.0×10^3
Loader and binary compatibility programs	5.0
Procedure-processor, SDS 930 environment (Alternative 1)	11.0
Compiler (Alternative 2)	50.0
Meta-compiler language specification (Alternative 3)	18.0
Documentation and training	8.0
Systems integration	14.0

Configuration C1 - Medium-Scale Expandible CPU in Dedicated System for Functional Test of Digital Data

Description (see fig. 11 and table XVI).- This configuration is similar to Configuration A1, except that a "stripped-down" version of a computer capable of expansion to a medium- or large-scale application is used for the central processor. The computer is a 32-bit 0.8 μsec cycle-time machine, with 16 general registers and a large instruction set. Perhaps the most important feature is the relative independence of the computing unit, memory, and input/output. This type of architecture is common to several major lines, among them the SDS Sigma series and the IBM 360 series. Minimum peripherals, a basic operating system, and a simple interpretive language identical to Configuration A1 are included. The differences between Configuration A1 and this configuration lie in the additional convenience of the operating system, the lower cost of implementing language and monitor programs, and I/O control.

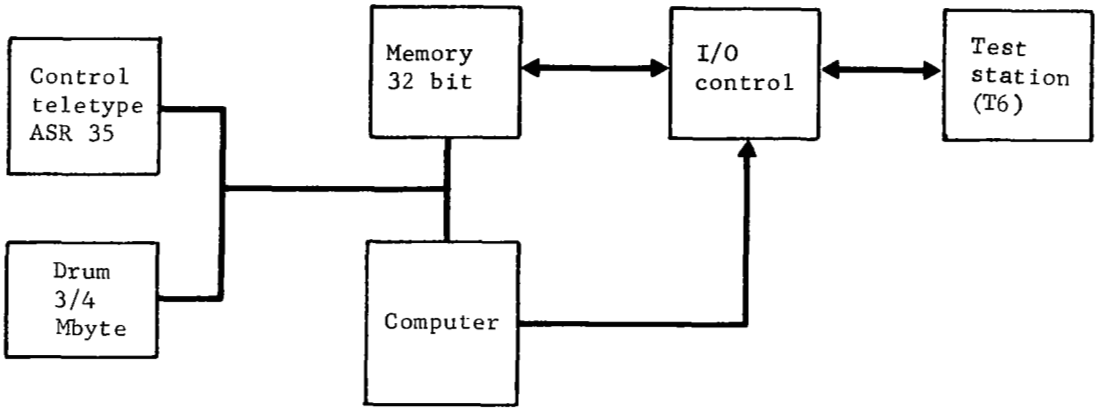


Figure 11.- Configuration C1 Block Diagram

TABLE XVI

CONFIGURATION C1 EQUIPMENT LIST

	Cost, $	
	Original buy	Delta to A1
CPU hardware		
Computer, 32-bit, 0.8 μsec, integral I/O processor	50.0×10^3	50.0×10^3
Memory, 32-bit, 0.8 μsec, 8000 words	50.0	50.0
Two-way access to memory	7.0	7.0
Interrupt control	2.2	2.2
Interrupts, two pair	.7	.7
Real-time clock	STD	----
External interface	2.0	2.0
Teletype, PT punch and read	7.0	7.0
Rapid access drum, 3/4 Mbit	26.0	26.0
	144.9	144.9
Station hardware		
I/O control	10.0	4.0
Test station (T6)	65.0	----
	75.0	4.0
Software		
Operating system	6.0	6.0
Test language, Configuration L1	23.0	20.0
Test monitor (reduced version)	15.0	15.0
	44.0	41.0
Other items		
Documentation and training	6.0	4.0
System integration	8.0	3.0
	14.0	7.0
Total	277.9	196.9

Configuration C1′ - Background/Foreground System for Dedicated Test of Digital LSA

Description (see tables XVII and XVIII).- This system is identical to Configuration C1 except that a 4000-word increase in core size permits a stripped-down background/foreground operating system and a relatively powerful language processor (L3). The rest of the configuration is essentially the same.

Remarks.- The background/foreground operating system permits the computer to process a job stream (translation, media conversion, data analysis, etc) and execute testing concurrently. Background tasks will be automatically "swapped out" to disc when additional core is required by foreground (test) programs, and automatically reloaded and continued when test core requirements decrease.

The primary advantage of this configuration is upward compatibility. From this point upward, each expansion becomes a compatible superset of the preceding. This includes the language, test monitor, and operating systems.

TABLE XVII

CONFIGURATION C1′ EQUIPMENT LIST

	Cost, $	
	Original buy	Delta to A1
CPU hardware		
Computer, 32 bit, 0.8 μsec, integral I/O processor	50.0×10^3	50.0×10^3
Memory, 32 bit, 0.8 μsec, 12 000 words	68.0	68.0
Two-way access to memory	7.0	7.0
Interrupt control	2.2	2.2
Interrupts, two pair	.7	.7
Real-time clock	Standard	Standard
External interface	2.0	2.0
High-speed teletype, PT punch and read	12.0	12.0
Rapid-access drum, 3/4 Mbit	26.0	26.0
	167.9	167.9
Station hardware		
I/O control	10.0	4.0
Test station (T6)	65.0	----
	75.0	4.0
Software		
Operating system	6.0	6.0
Test language, Configuration L3	12.0	12.0
Test monitor	17.0	17.0
	35.0	35.0
Other items		
Documentation and training	6.0	4.0
System integration	8.0	4.0
	14.0	8.0
Total	291.9	214.9

TABLE XVIII

CONFIGURATION C1′ EQUIPMENT LIST (DELTA TO C1)

	Cost, $
CPU hardware	
Memory increment, 4000 words	17.5×10^3
High-speed teletype	11.0
	28.5
Software	
Operating system	6.0
Test language	12.0
Test monitor	13.0
	31.0
Other	
Documentation and training	6.0
System integration	5.0
	11.0
Total	70.5

Configuration C2 - Configuration C1 with Addition of Parametric Test Capability for Digital LSA

Description.- This configuration is identical to C1 with the exception that the test station configuration is T7, for parametric test capability.

The addition of parametric test capability for digital large-scale arrays, to Configuration C1 will require changes to the test station and test language only (table XIX). The block diagram is identical to Configuration A1 (fig. 7).

TABLE XVII

CONFIGURATION C2 EQUIPMENT LIST

	Cost, $
Addition of parametric capability to test station (T6)	
Basic cost	45×10^3
Field modification	5
	50
Addition of new functions to test language	8
Delta total	58

Configuration C3 - Background/Foreground System for Time-Sharing of Reliability and LSA Test Stations

Description (see fig. 12 and table XX).- This configuration converts C1′ to time-sharing and adds a test station and software for reliability testing. Time-sharing was defined for Configuration B1, and reliability testing was described for Configuration B2.

The CPU hardware increase is in core size and the addition of magnetic tape. The I/O control is modified for time-sharing. Software changes include modification of the test monitor for time-sharing and the addition of new test and data analysis functions to the test monitor and language. Additional capabilities are added to the operating system.

Remarks.- The advantages of software compatibility, and a machine with adequate power, are demonstrated by this configuration. Both time-sharing and reliability functions are added to the basic configuration for a total delta of $121 000; whereas the corresponding delta A1-B2 was costed at $250 000, with less resultant capability.

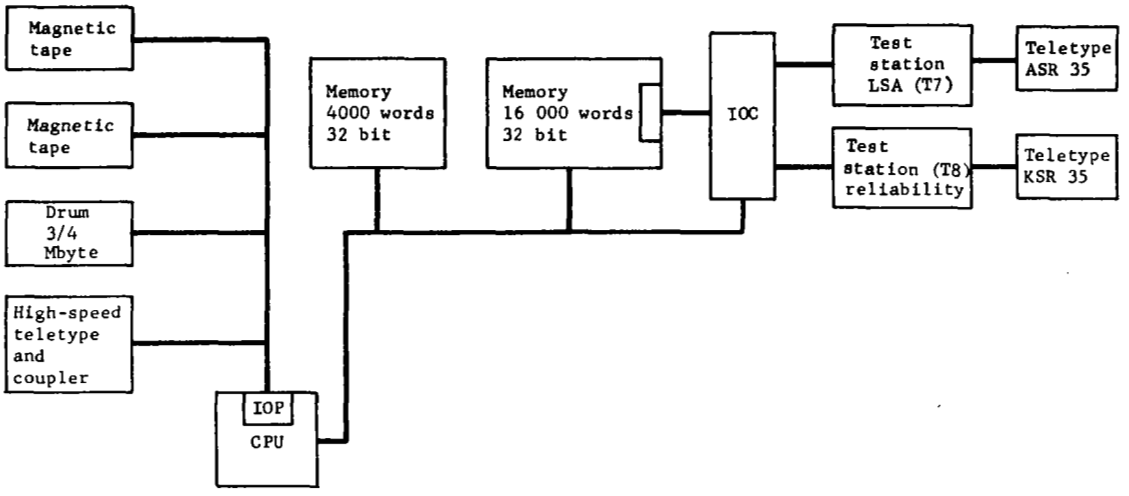


Figure 12.- Configuration C3 Block Diagram

TABLE XXI

CONFIGURATION C3 EQUIPMENT LIST (DELTA TO C1′)

	Cost, $
CPU hardware (delta only)	
Memory module, 4000 words (total 16 000 words)	17.5×10^3
Memory module, 4000 words (second bank)	33.0
Interrupts, two pair	.7
Tape controller and two decks, 37.5 ips	44.0
Memory protect option	4.0
	99.2
Test station hardware	
I/O control, delta for second station	4.0
Test station, T8 configuration (reliability)	110.0
High-speed teletype and coupler for test station	11.0
Modification to ASR 35 and coupler for Station 1 (ASR from Configuration C1)	3.5
	217.7
Software	
Modification to test monitor for time-sharing	7.0
Test language additional functions	3.0
Operating system modification for time-sharing	2.0
Data analysis drivers	3.0
Documentation and training	5.0
	20.0
Other	
Documentation and training	5.0
System integration	12.0
	17.0
Total	249.7

Configuration C4 - Expansion of C3 Time-Sharing System to Additional Users and Addition of Modem Channels

Description (see fig. 13 and table XXI).- The capability of the Configuration C3 time-sharing system to accept additional test stations, and the consequent increase in batch processing load, is tested in this configuration. Two diverse applications were selected:

1) A typical laboratory automation application (data acquisition and preliminary analysis for a mass spectrometer);

2) A typical developmental application where the system is included in a control and simulation loop (flight dynamics).

The only hardware expansion is a modular increase in drum size for data and program storage, and software extensions for the new usages.

Remarks.- No major technical problems were encountered in absorbing either application. The bandpass of the modem may occasionally restrict sweep speed of the mass spectrometer. The system expansion is almost entirely a function of the added load.

The primary advantage of this configuration is that it provides to additional installations all the capability of a powerful dedicated system at a fraction of the investment and operating cost. Thus, the primary installations all the capability of a powerful investment over multiple external users.

No limit is placed on distance of the remote installations, because local I/O and control is provided at each site.

One remote user is shown in this configuration. Maximum expansion capability is estimated as follows:

1) 8 parallel or integral-computer test stations;

2) 4 modem test stations;

3) 4 remote batch terminals.

The hardware expansion steps to achieve the above service consist of expansion of core to 40 000 words, addition of an I/O processor to increase available computer cycle time by approximately 50%, upgrading of peripheral speeds, and addition of tertiary mass storage (disc).

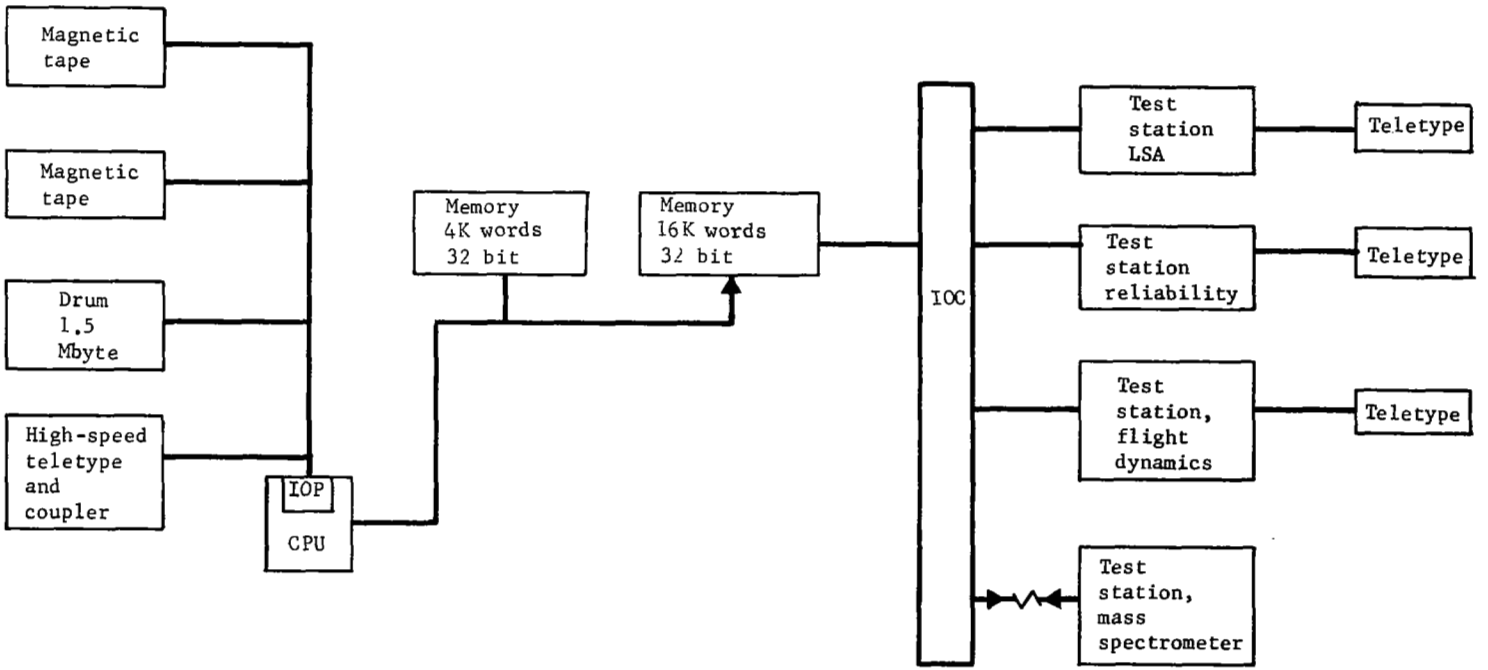


Figure 13.- Configuration C4 Block Diagram

TABLE XXI

CONFIGURATION C4 EQUIPMENT LIST

	Cost, $
CPU hardware	
Additional rapid-access storage unit, 3/4 Mbyte	18.0×10^3
Priority interrupt, two pair	1.4
	19.4
Station hardware	
I/O control extension to four stations	.4
Test station, mass spectrometer	29.0
Test station, flight dynamics	117.0
Modem unit (telephone transmission)	4.0
(2400 baud line rate = $55/mo)	
	150.4
Software	
Test language additions	6.0
Test monitor additions	6.0
Basic analysis and driver programs	10.0
(spectrometer and flight dynamics)	
	22.0
Other items	
Documentation, training, miscellaneous	6.0
System development and integration	14.0
	20.0
Total	211.8

A remote batch terminal is provided in the CPU area initially. As additional remote batch users are added, this terminal may be reinstalled at the remote location and more effective peripherals added to the central installation.

This installation will support several levels of test language, experimental diagnostic processors and algorithms, and very advanced data analysis and correlations, including between-installation correlations.

Configuration D1 - Remote User Time-Sharing System

Description (see figs. 14 and 15 and table XXII).- True remote user capability is provided in this configuration, including remote batch I/O. A dedicated "minicomputer" is added to the test terminal for local control and data compression. The CPU is expanded and the next higher level of operating system is installed to provide control of remote batch terminals.

Remarks.- The modem station interface introduced in Configuration C4 is based on 2400 baud telephone lines. While nearly all static or functional testing can operate successfully within this bandwidth, dynamic, closed-loop, or extremely time-critical operations are not permitted. Use of 50-kHz lines will permit some, but not all, of these operations; but the line lease rates are expensive. The use of an integral "minicomputer" in the remote station appears to be a more effective solution. This unit would operate as a slave to the central installation, providing local data storage or reduction, and time-critical control functions. Thus, the amount of raw data and communication necessary during an actual test cycle would be minimized.

A remote batch terminal, consisting of a card reader, card punch, and line printer, permits the remote user to input test translations or analysis programs and receive listings or card decks of data, translated programs, or other items. The full capabilities of the central processor are available. The user can create and maintain his own system-resident files; translate, load, and execute test programs; and access any system routines or processors by control card command. The remote job streams are interleaved with the central processor job stream and may proceed concurrently with test, even at the remote location.

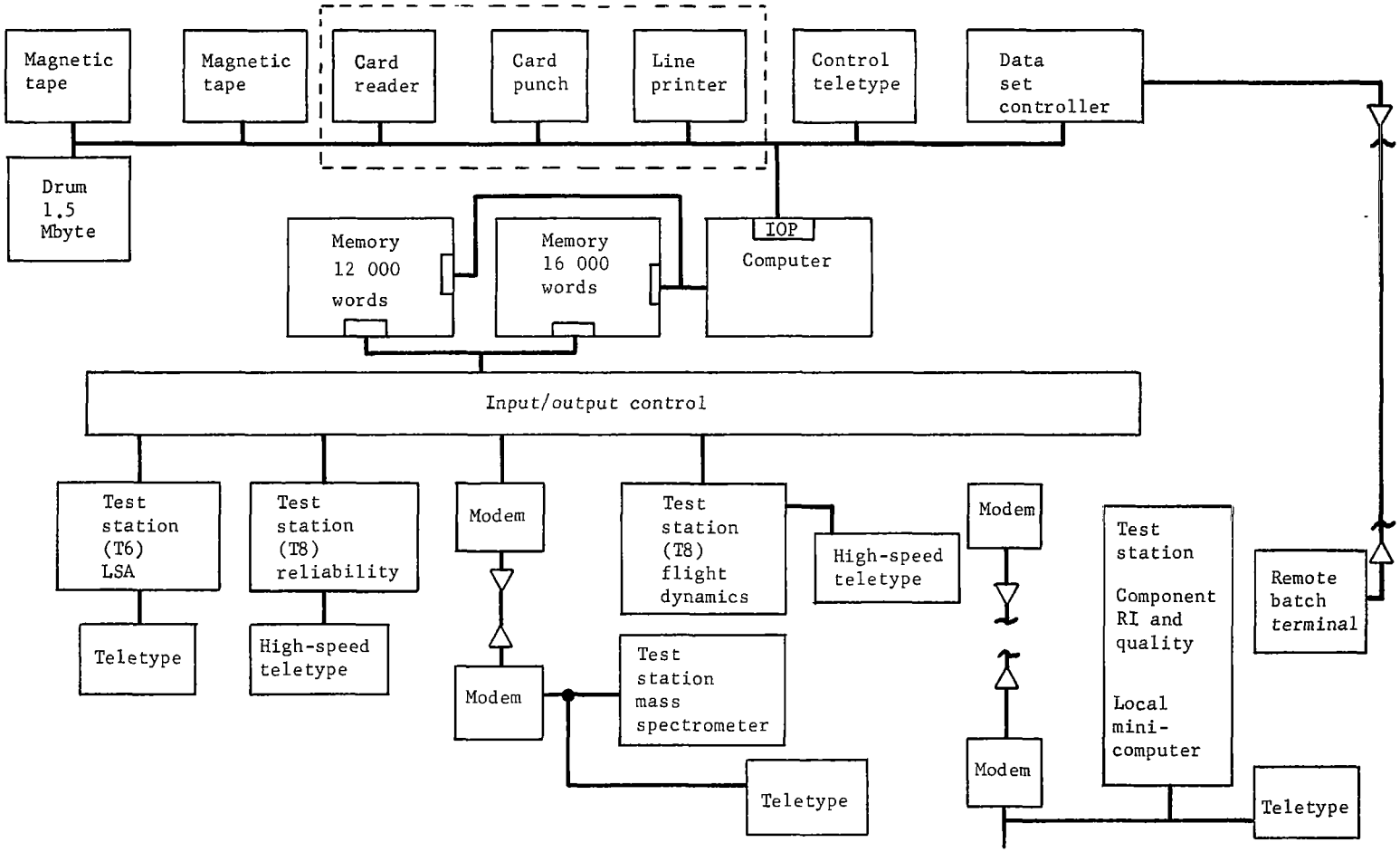


Figure 14.- Configuration D1 Block Diagram

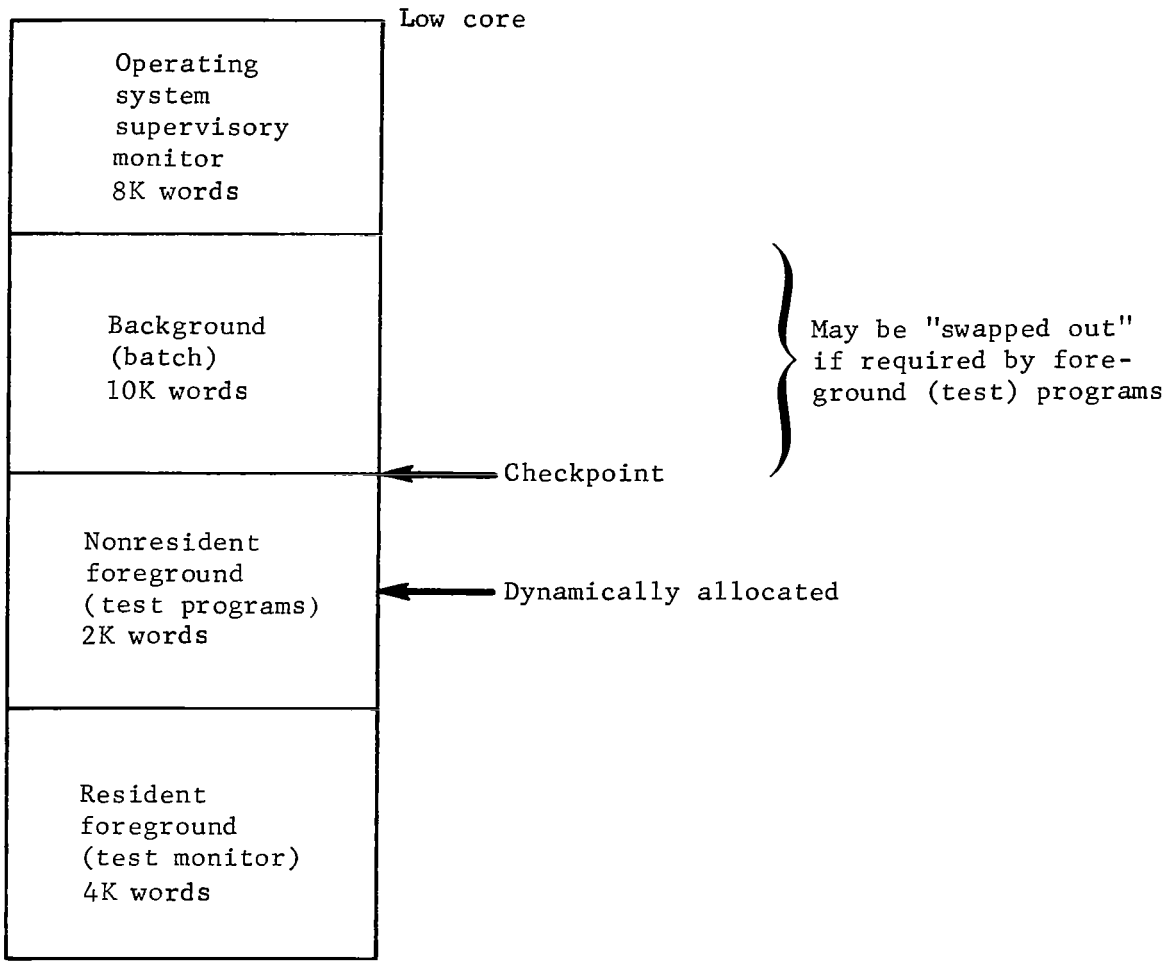


Figure 15.- Configuration D1 Memory Allocation

TABLE XXII

CONFIGURATION D1 EQUIPMENT LIST (DELTA TO C4)

	Cost, $
CPU hardware	
Remote batch terminal	36.0×10^3
Card reader 200 cpm Card punch 100 cpm Line printer 250 lpm	
Data set controller	7.0
Additional register block	2.5
Priority interrupt, one pair	.35
Memory increment, 8000 words	35.0
Expansion of second memory module to two-way access	7.0
	87.9
Station hardware	
Expansion of IOC to five terminals	7.0
Remote test terminal with integral computer	150.0
Additional modem interface	4.0
Teletype (KSR 35) and coupler	5.5
Remote batch terminal	36.0
	202.5
Software	
S3 operating system	8.0
Test language additions, modifications for new OS	4.0
	12.0
Other	
Documentation, training, miscellaneous	8.0
Integral computer test station development	25.0
System development and integration	25.0
	58.0
Total	368.4

Configuration Summary and Tradeoff

Table XXII summarizes the detailed system Configurations A1 thru D1. Table XXIV explains the codes used in table XXIII.

The following premises are established for this tradeoff:

1) The analysis may be usefully partitioned into test stations, central processor hardware, and basic software;

2) Test performance is primarily influenced by variations in the test station. Central processor and software variations have little direct effect beyond the minimum support level;

3) The interface with the user (test engineer, etc) is primarily influenced by the central processor and software, given an adequate technical capability in the test station.

Figure 16 depicts a number of probable growth paths through incrementally larger configurations, for the central processor and basic software.

The horizontal axis is a relative measure of system capability. The rating for each configuration is a sum of the individual ratings in table XXIII. It is assumed that the numeric sum of the above factors is a measure of relative system performance.

The vertical axis is the cumulated expenditure for hardware and basic software.

Each configuration is plotted against its performance rating, and cumulated expenditure in arriving at that configuration.

Analysis of table XXIII and figure 16 establishes certain conclusions. These are discussed in the following paragraphs.

The most economic path is A1 thru C4. The least economic path is A1, A2, B2, C1′.

Certain paths can be determined as uneconomic by examination, without additional analysis. For example, any further extension of the B2-C1′ path can never result in a cost advantage.

TABLE XXIII

SYSTEM CONFIGURATION SUMMARY

Configuration	Description	Detailed characteristics	Test language power	Operating system power	STR	O_I	T_S
A1	Minimum dedicated system for functional test of large-scale arrays (LSA)	Minimum configuration using small-scale CPU equipment in dedicated configuration to drive single test station Minimum test language configuration and test monitor operate within system Usage is experimental functional testing of digital LSA up to 100 logic nodes and 150 interface points	1	0	0	0	2
A2	Configuration A1 with addition of parametric test capability for digital LSA	The minimum CPU, language, and monitor of A1 retained; parametric test capability requires only addition of analog stimulus and measurement test channels to test station and incorporation of corresponding statements in test language	1	0	0	0	2
B1	Minimum system for time-sharing test stations	This configuration adds capability for time-sharing two stations to dedicated system of A1; addition of time-sharing has no effect on test station or computer hardware itself	1	0	1	0	4
B2	Addition of reliability testing	This configuration adds one typical new requirement (reliability testing program similar to that contemplated by ERC's Device Research Branch). Modular increments to memory and I/O control implemented to service additional station; addition of magnetic tape, functions to monitor and language, and second high-speed teletype	1	0	1	1	6
B3	Dual configuration for addition of reliability testing	This configuration examines cost and practicality of independent system (similar to A1) for reliability testing; because hardware essentially duplicated, variable of interest here is savings in development time	1	0	5	1	4
B4	Time-shared LSA and parametric testing with supporting installation for software	This configuration implements language translator and other software on separate existing installation; test system used only for test; other functions remain similar to B1					

TABLE XXIII.- Concluded

SYSTEM CONFIGURATION SUMMARY

Configuration	Description	Detailed characteristics	Test language power	Operating system power	STR	O_I	T_S
C1	Third-generation CPU in dedicated system for functional test of digital LSA	This configuration similar to A1, except that "stripped-down" version of computer capable of expansion to medium- or large-scale application used for central processor; minimum peripherals, basic operating system, and simple interpretive language identical to A1	1	1	1	1	
C1	Background/foreground system for dedicated test of digital LSA	This system identical to C1 except 4000-word increase in core size permits translation concurrent with testing and a relatively powerful language processor (L3); rest of configuration essentially same	3	2	2	2	
C3	Background/foreground system for time-sharing of reliability and LSA test stations	This configuration converts C1 to time-sharing and adds test station and software for reliability testing. CPU hardware in core size and addition of magnetic tape; I/O control modified for time-sharing; software changes include modification of test monitor for time-sharing and addition of new test and data analysis functions to test monitor and language; additional capabilities added to the operating system	3	2	3	3	
C4	Expansion of C3 time-sharing system to additional users and addition of modern channels	The capability of C3 time-sharing system to accept additional test stations, and consequent increase in batch processing load, is tested here; two diverse applications selected: (1) typical laboratory automation application (mass spectrometer), and (2) typical developmental application (flight dynamics) Only hardware expansion is modular increase in drum size for data, and program storage and software extensions for new uses					
D1	Remote user time-sharing system	Full remote user capability provided, including remote batch I/O; dedicated "minicomputer" added to test terminal for local control and data compression; CPU expanded and next higher level operating system installed to provide control of remote batch terminals	3	3	4	3	

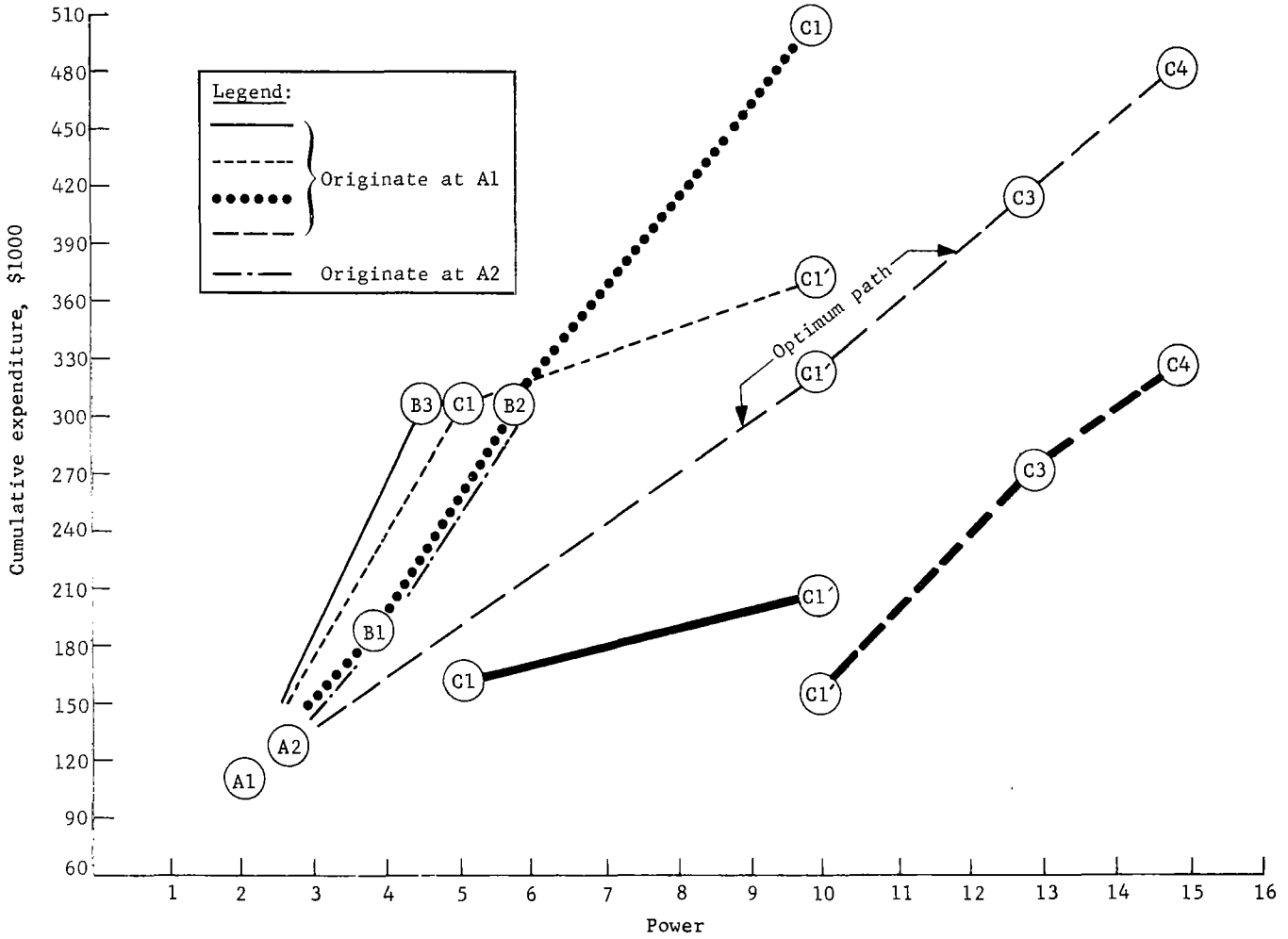
TABLE XXIV

SYSTEM CONFIGURATION SUMMARY CODE EXPLANATION

Code	Configuration reference
Test language power	Rating of the power of the language on a scale of 0 to 4; also used for operating system
STR	Rating of the support-to-test ratio achieved by the system on a scale of 0 thru 4 (see Note 2)
O_I	Rating of the operator interaction provided by the configuration on a scale of 0 thru 4
T_S	Total "power" of the system. The sum of the individual ratings for test language, operating system, support-to-test ratio, and operator interaction; and the number of test terminals supported by the configuration

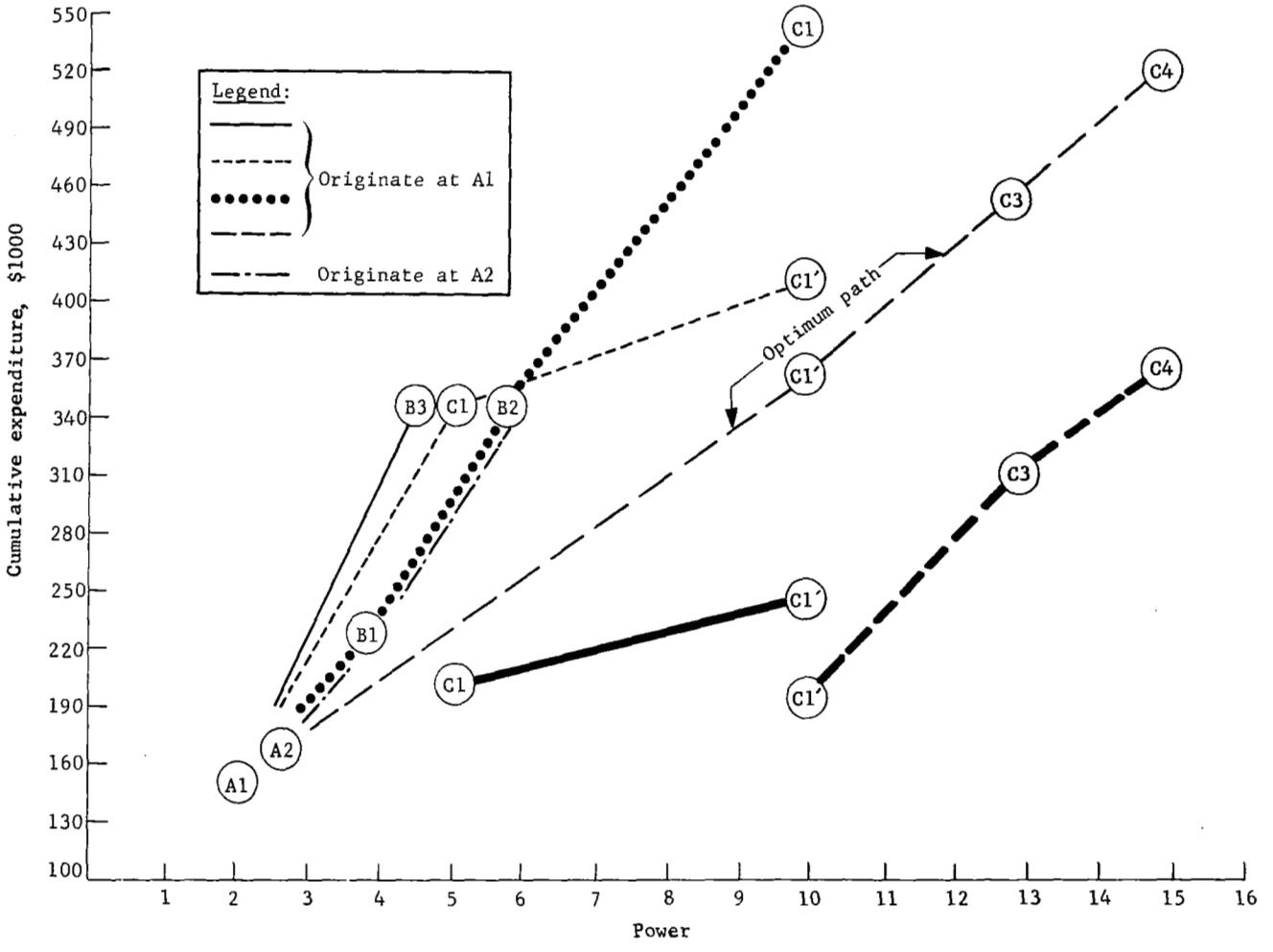
Note: 1. The various ratios used in this summary are, for the most part, subjective. However, the differences in capability of the lowest rating vs the highest rating are normally substantial.

2. The support-to-test ratio refers to the amount of time the system is available for actual test execution as compared to the amount of time the system is occupied with supporting or "get-ready" operations. These include test program translation, loading of programs, etc.



(a) CPU available

Figure 17.- Alternative Growth Path Cost Comparison



(b) CPU not available

Figure 17.- Concluded

The C1′-C3-C4 path is quite efficient. Power vs expenditure is nearly linear and independent of previous configurations. Examination of the detailed costing shows little throwaway activity. The conclusion is forced that little can be gained by alternative paths once C1′ is reached. Thus, eliminating C3 and proceeding directly to C4 results in little change in the cumulative expenditure.

One can also assume from this that the expenditure is proportional no matter where the path is entered. Thus, entering at C3 from B1 can never result in a cost advantage over entering at C1′. The cost, C1′ to C3, must still be paid.

There is a very obvious step in performance/cost ratio for the C1′ configuration.

From the cumulative cost standpoint alone the obvious best performance is C1′-C4. However, other considerations intervene. The most obvious is the ability to support and economically use the more expensive unit in the initial stages of an overall development program. It takes time to develop skills, applications, and demand; and while demand and use are historically underestimated for computer installations, a case can be made for starting at a lower level even if the cumulative cost is ultimately higher.

It is, however, interesting that lower early expenditures often lead to very significantly higher total program cost. One might follow the A1-B1-B2 path, assured by the initial small investment and acceptable later deltas, until it was too late to economically recover. From such instances come the familiar horror stories of "conversion costs." At least some of these situations are made, not born.

The best compromise in this study seems to lie in the A1-A2-C1′ axis (see fig. 17). Total cumulative expenditure for this path is $90 000 greater than entering at C1′ directly.

The recommendation of this study is that the growth and acquisition plan be based on this path (i.e., initial implementation of Configuration A1 with tight control of software expenditures), then proceeding to Configuration C1′ for the next expansion. The "B" configurations (basically, time-sharing with a small-scale computer) are not recommended.

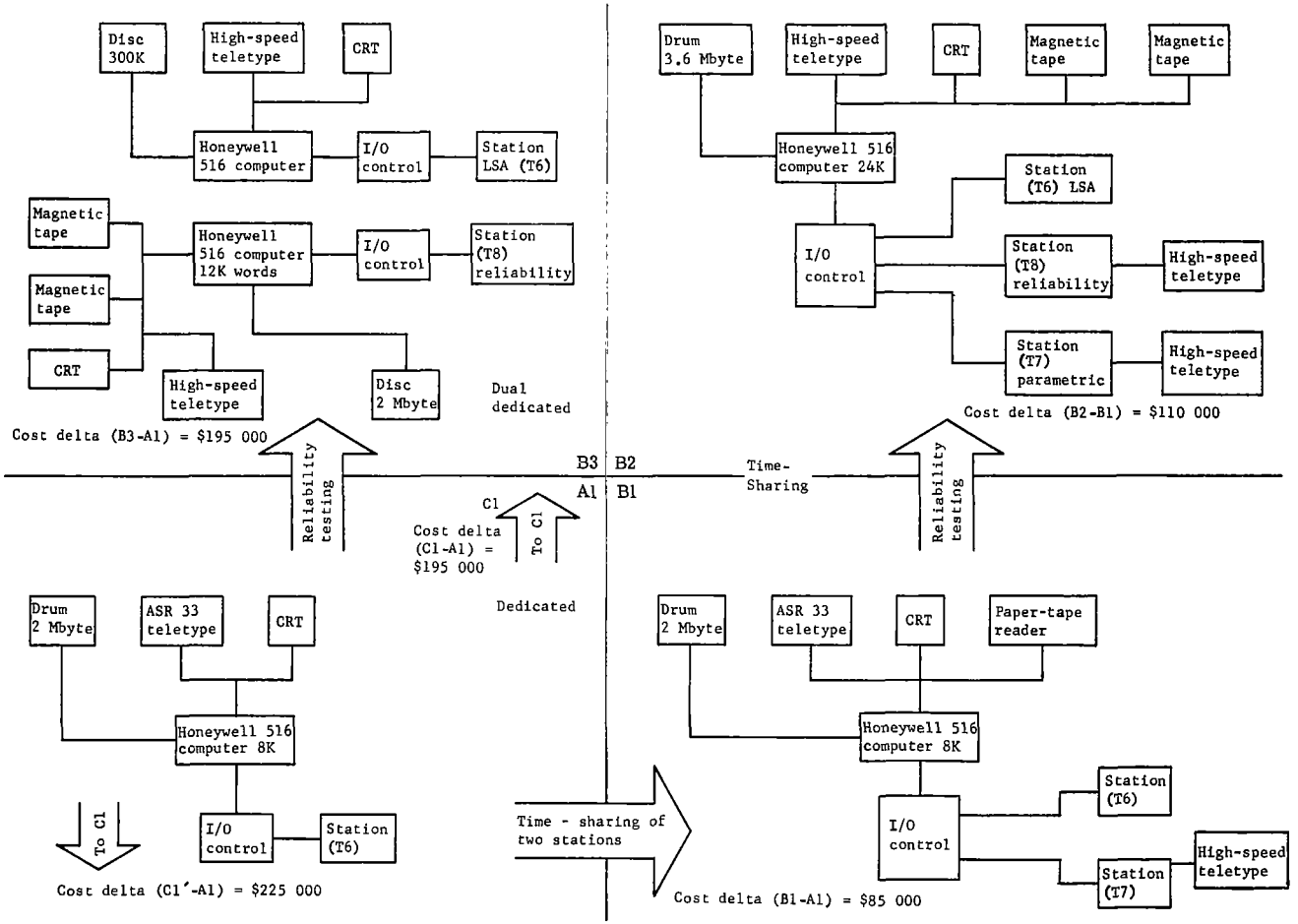


Figure 17.- System Configuration Graphic Summary

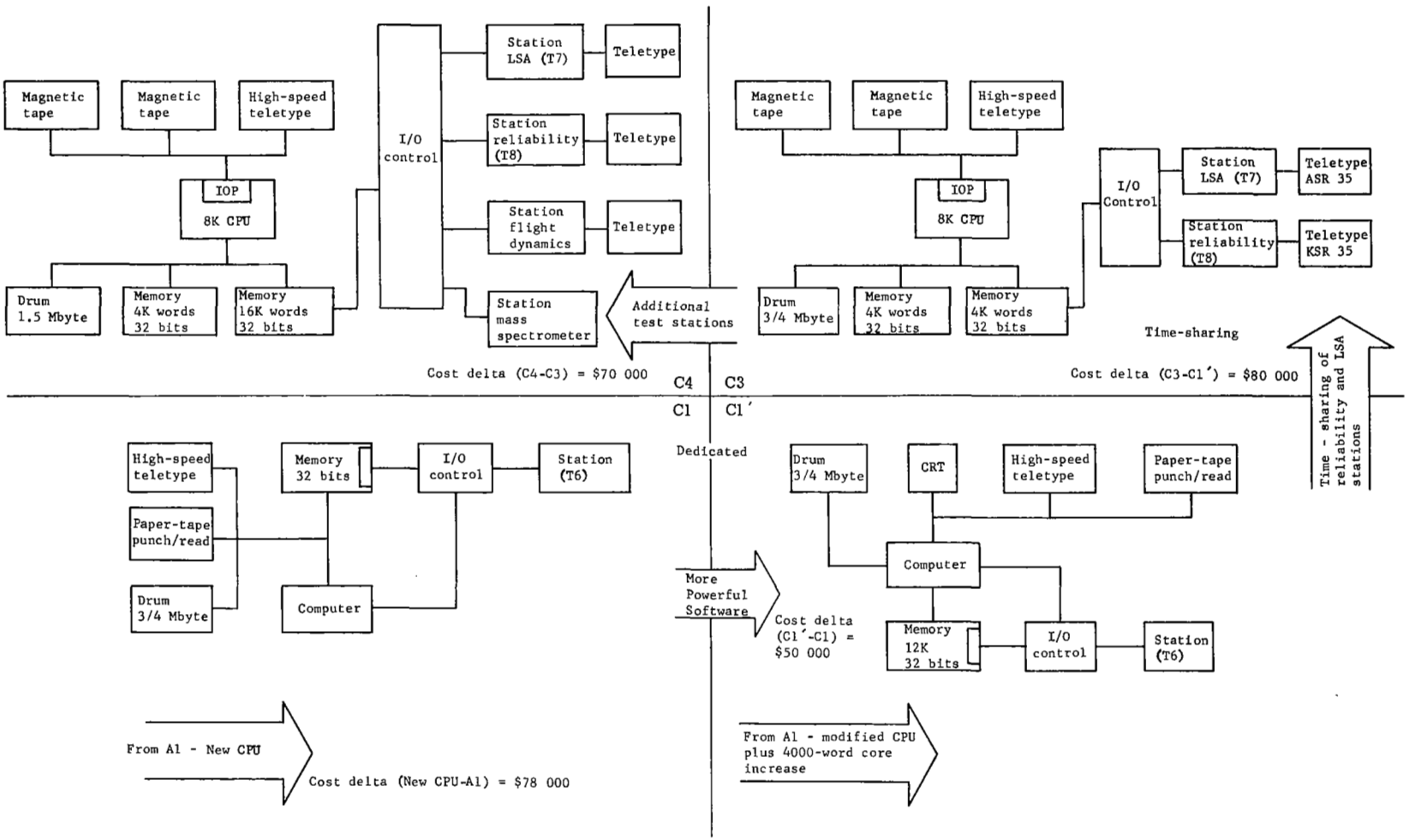


Figure 17.- Concluded

Direct initial implementation of Configuration C1′ would, of course, be the most economic and desirable in terms of long-range economics and capability if economic and technical justification can be made. The combined requirements of reliability and large-scale array testing appear to offer such justification.

Conclusions.- Conclusions are as follows:

1) An initial development phase using existing computer hardware can be economically implemented;

2) Maximum expansion of this initial configuration provides only limited potential because of the inherent limitations of a small-scale computer;

3) Conversion to a suitable medium-scale computer used in a time-sharing mode can provide considerable growth capacity;

4) A significant cost penalty can ultimately result if the initial configuration and the conversion are not properly controlled;

5) Adherence to the optimum growth path in figure 16 can minimize the expansion loss to less than 20% of final cost;

6) Growth patterns projected in figure 16 are, for a five-year period, projected at a lower rate than typical test installation growth projected from the requirements survey. This is in cognizance of the ERC orientation toward technology development rather than actual test production;

7) For purposes of this study, the initial configuration using a small-scale computer is defined as Phase I and conversion to a medium-scale computer capable of additional expansion is defined as Phase II;

8) Expansion beyond the needs of ERC alone appears technically and economically feasible. This is defined as Phase III and provides for servicing of complete remote installations by a centrally located, expanded, Phase II central processor. This possibility and the application of the Phase II developed system in other testing areas is further analyzed in the "Cost Effectiveness Study" section;

9) Summary specifications for the central processor, software, and test station for each of the three phases are given in the "Specifications" section.

Configuration Study, Test Language Translator

For purposes of this report, language translators may be divided into two general categories -- interpreters, and assemblers and compilers.

An interpreter performs the translation and execution of each statement of the test language at test time. Assemblers and compilers perform the translation of the entire test program before test time. The output of the assemblers and compilers is normally a binary (object) code that must be loaded into the computer before execution.

Combinations of both of the above are also possible. For example, a pretranslator may be used to reduce a higher level language to a simpler language that is then processed by an interpreter. The pretranslation occurs off-line, as for assemblers and compilers. However, the output is in symbolic form, not in binary object code. At test time, the simplified symbolic program is then fed into the interpreter. This configuration offers the power of an assembler or compiler and the test-time flexibility of an interpreter.

The distinction between an assembler and a compiler is in terms of the power of language that is possible for each. It is difficult to define a sharp dividing line separating the two. In general, assemblers operate on a one-for-one basis, i.e., one statement results in one line of machine code. Compilers will often generate many lines of machine code for a single source statement. The assembly language is normally in terms of the operations of the machine; the compiler is normally in terms of the problem itself, without concern for the mechanization or "how" of solving it.

There are also several ways of implementing a compiler or assembler. The first might be termed "special purpose"; the processor is specially written for a specific target machine and source language. A second approach is termed "procedure processor" in this report. A set of procedures is written defining the input language and the output code. The translator operates by processing the input language in accordance with the set of procedures. Writing a set of procedures defining a new language or a new target machine is much less expensive than writing an entire special-purpose translator. The third approach is termed "meta-compiler." Here, a source language and target machine are defined to the meta-compiler by a set of specifications. The meta-compiler then generates a special-purpose compiler or assembler that in turn translates in the normal manner.

The last two approaches provide a high degree of machine independence. The language, the test stations, the test computer, and even the computer in which the compiler is to operate may be changed without incurring all of the expense associated with rewriting a special-purpose compiler or assembler. The distinction between the second and third approach lies in the translation method. The entire procedure processor itself is brought into play for each translation. The meta-compiler is used only once -- to generate a special-purpose compiler. From that point, the meta-compiler is not used again.

Procedure processors are normally slower in execution and may be more demanding in terms of core requirements than special-purpose translators. They are, however, more flexible in that the procedure deck may be changed at any time without requiring regeneration of an entire translator, as would be the case with the meta-compiler approach.

Each of the above approaches to translation are analyzed in more detail in the following paragraphs.

In general, cost of development varies directly with sophistication of the test language and the translation process. A factor that is difficult to evaluate is the extent to which the software is developed and supplied by the computer manufacturer; this will be taken into account in the system specifications.

Test language translation can be a complicated process and may exceed the capacity of the test computer. Because test translation (except interpretation) may be performed via standard batch processing, installations other than the test computer may be used. These are considered under "Test Language Translators."

Configurations L1 thru L5 are such that computer requirements are satisfied by the test computer itself, and sufficient time is available for test translation and other off-line operations. Computer time can be made available either by dedicating the test computer for such processing (off-line) or by providing the capability to process simultaneously with automatic testing (background/foreground). The latter approach places additional demands on the supervisory system.

L1 - interpreter description: The interpreter performs the translation and execution of each statement of the test language at test time. No pretranslation or off-line processing is included in this configuration. The following steps are performed in the course of "interpreting" each statement:

1) Decode each statement and isolate the function identification and associated parameters;

2) Convert the parameters and symbols into binary bit patterns according to a prescribed format;

3) Execute each statement, i.e., perform the specified function with the given parameters.

The test language acceptable to the interpreter would permit specification of all basic functions of the test station. Additional functions would be provided for timing specifications, conditional tests, etc. Conditional or direct transfers in the test language would be limited to backward references in the current sequence (those statements just interpreted) or the beginning of some other sequence whose location is known to the system by a resident name and disc address list.

One of the main advantages of the interpreter is that entries can be made on-line from the operation console using the test language. The statement is simply entered at the keyboard and passed to the interpreter under control of the supervisory system. The statement is then processed via the same three steps described above for test execution.

To simplify the interpreter, the format for the test language would be rigid fixed field.

Remarks: Although the interpreter is easily implemented and quite flexible, it is not readily adaptable to time-sharing applications. The inefficiency and core memory requirements at test execution time are inherent drawbacks that are not present in other translators.

Advantages: Advantages are as follows:

1) Relatively inexpensive to develop;

2) Modifications and additions are relatively simple to implement;

3) Man/machine communication at execution time easily accommodated;

4) A wide range of complexity and sophistication may be considered.

Disadvantages: Disadvantages are as follows:

1) Inherent inefficiency of execution;
2) More core memory requirements at execution time;
3) The associated test language is limited;
4) Forward transfers of control or references to other statements are not permitted;
5) Inclusion of assembly instructions in the test language is difficult;
6) Adaptability of the interpreter to time-sharing is inherently difficult.

L2 - simple macro-processor description: The macro-processor accepts statements (macros) that have previously been defined and expands them at the assembly level. Each statement then generates a station instruction, an open routine in line, or a call to a subroutine. Arguments (parameters) may be specified with the statement.

Each statement of the test language would correspond to an assembly level macro. The syntax of the test language would be restricted by the permissible syntax of the macro assembler.

Remarks: The availability of a powerful assembler with macro capability would make this configuration quite reasonable for a minimal translator. The test language itself is inherently limited but would permit inclusion of assembly level computer instructions.

Advantages: Advantages are as follows:

1) Relatively inexpensive to develop;
2) Modifications and additions are relatively simple to implement;
3) Efficient translation and execution.

Disadvantages: Disadvantages are as follows:

1) The test language is rigid and very limited;
2) On-line modifications at execution time are difficult.

L3 - procedure processor description: The procedure processor consists of two distinct parts -- a set of procedures and the processor itself. The procedure set in effect describes the source language and the code that is to be generated. The test language consists of procedure references and parameters (operands). (That is, for the statement POWER 28V, POWER is a reference to a procedure called "Power"; "28V" is the parameter.) The processor follows the assembly procedure defined for each statement.

The procedures are normally written once for any specific language and/or target machine. The processor translates entirely by the procedures and a limited number of translator directives. The user need not be aware of the procedure method of translation; the source language and the translation process itself is identical to a special-purpose assembler or compiler. There is little inherent limitation on the power of the test language that can be provided, except that procedures become very complex as the language power approaches true compiler level.

Because the processor itself is written and assembled by its own set of procedures, it is possible to change the operating computer of the processor by supplying a new procedure set and reassemblying the processor. Therefore, both the target machine, the language, and the operating machine may all be changed without rewriting the processor itself. There are several practical limitations on this theoretical capability:

1) I/O handlers, file management, operating system linkages, and similar functions are installation-dependent. These portions of the processor will almost certainly require revision when changing the host computer.

2) The binary output of the translator is normally in a standard relocatable format. This format is not normally definable by the procedure deck or by translator directives. When moving the procedure processor to another operating environment, or when generating on one machine a program to be executed on another machine, some provision for compatibility with the target machine's loader must be made. This can best be achieved by a binary processor that converts the binary output format of the procedure processor to a form suitable for loading by the target machine's loader.

Note that although much freedom is provided in defining an input language, the syntax of that language is still limited by the syntax acceptable to the processor. Functions such as the allowable length of symbols, operator precedence, expression delimiters, and the basic character set itself may not usually be changed.

Remarks: Procedure processors have been implemented and proven in use. They are expensive to implement initially. Some manufacturers provide a procedure processor as part of the standard software for their computers; or one may have been implemented by a user; or one may be converted from a similar machine. For purposes of this study, it is assumed that a procedure processor is available for the medium-scale computer; language costing is based on implementation of a set of procedures only. It is estimated that implementation of the entire processor would be approximately double the cost given for a special-purpose test compiler.

Advantages: Advantages are as follows:

1) A powerful language, readily expanded or modified;
2) Complete flexibility with regard to assembly level instructions;
3) Readily adaptable to different test computers;
4) Inexpensive implementation (if basic processor is available).

Disadvantages: Disadvantages are as follows:

1) Relatively expensive to develop initially;
2) Sophistication of the procedure processor implies a more complex system for the test engineer;
3) On-line interaction and modification is relatively difficult;
4) Slow.

Test compiler description: The test compiler accepts the source test language and translates it directly into binary machine instructions for the test computer. The compiler has the capability of generating open, in-line subroutines as well as linkage for closed subroutines. Complete conditional and logical statements can be included (e.g., generation and evaluation of truth tables).

The test compiler can provide the ultimate in terms of power of the test language. The statements of the language can be near English, and it can be oriented to the test engineers. The cost of development varies directly with the sophistication of the language.

Remarks: Note that a true compiler level source language, comparable to Fortran or Algol, does not exist for testing use. There are some inherent and very difficult problems associated with a general-purpose, problem-oriented language for testing. The most significant is that the test situation is very sensitive to the machine-level implementation of its program (critical timing, sequence of commands, and similar functions). With a problem-oriented language, by definition, one cannot control the machine-level implementation.

Advantages: Advantages are as follows:

1) Power of the test language, ease of use;
2) Extensive, meaningful diagnostics;
3) Debugging aids;
4) Full subprocedure capability.

Disadvantages: Disadvantages are as follows:

1) Expensive to develop;
2) Oriented completely to a single computer for execution of tests;
3) On-line interaction and modification is difficult.

L5 - meta-complier description: The meta-compiler provides full capability with regard to the test language, and provides machine independence with regard to the test computer. The meta-compiler accepts directives that define the test computer and source language. The output is an object compiler that has the capability to translate the source tests and to generage binary object code for the test computer. In short, the meta-compiler is a program that "writes" a compiler. The meta-compiler couples the advantages of the procedure processor and compiler.

Remarks: The technique described should lend itself reasonably well to automatic testing. The requirements for the test compiler are straightforward, and the class of computers used for automatic test fall within certain bounds. The technique has not been proven for general use, however, and is still in an experimental stage. The literature on meta-compilers is relatively scant.

Configuration summary: Table XXV summarizes significant characteristics of the five configurations of translators. Each configuration is rated for a number of characteristics on a scale of 0 thru 4, with the higher rating signifying superiority. Some ratings (e.g., each of use for the interpreter) are high because of the simple nature and limited capability rather than because of some positive attribute.

Dependent system (auxiliary computer required).- Where limited capability or limited availability of the test computer exists, an alternative is for the test translation and other associated off-line processing to be performed on an auxiliary computer. The requirements for the auxiliary computer include sufficient capability and compatibility with the test computer for transferral of the object programs from the translation process.

Disadvantages: There are inherent disadvantages to dependent systems. It must be remembered that every addition or alteration to a test will require at least one pass through the auxiliary computer. Also, when writing new test specifications, many runs may be required on the auxiliary computer. At best, the test development, modification, and debugging processes would be difficult. More important, these processes are completely dependent on the reliability, turnaround, and satisfactory performance of the auxiliary computer. Unless significant commitments and operational arrangements are made, the use of an auxiliary computer for translation must be ruled out in most test situations.

TABLE XXV

TRANSLATOR CHARACTERISTICS

Category	Type of translator				
	L1 interpreter	L2 macro-processor	L3 procedure processor	L4 compiler	L5 meta-compiler
General					
Capability of language	0	0	1	1-3	1-3
Expandability	2	2	2	0	1
Conditional tests	0	0	1	3	3
Convertability (to other digital comp)	1	2	3	0	3
Ease of use	3	1	1	2-3	2
Ease of debugging	3	2	1	1	1
Reporting/documentation possibilities	0	0	1	2	1
"In-line" assembly level instructions	0	3	3	2	3
"Leveling" and "nesting" of procedures	0	0	3	3	2
Development/characteristics - translation					
Core requirements (translation)	4K	6K to 8K	10K to 14K	16K	16K to 32K
Efficiency (translation)	3	3	2	1	1
Implementation on Honeywell 516	Yes	Yes	No	No	No
Development (translator only)	4 MM	4 MM	20 MM	40 MM +	48 MM
Development (basic subroutines) (drivers)	15 MM	15 MM	17 MM		
Total development	19 MM	19 MM	37 MM	40 MM +	48 MM
Characteristics - test execution					
Core requirements (overhead for first execution only)	2K to 6K	0	0	0	0
Basic subroutines (linkage)	Closed	Open	Both	Both	Both
Efficiency	1	2	2	1	1
On-line interaction	3	2	2	1	1
On-line modification	3	1	1	0	0
Adaptability to time-sharing	0	2	2	1	1
Error diagnostics and debugging aids	1	0	2	3	3

Test language translators - dependent systems: Most of the translator configurations may be implemented on an auxiliary computer for off-line processing. Some comments for each configuration are as follows:

1) L1 - interpreter - Not applicable.

2) L2 - simple marco-processor - One of the advantages of the macro-processor is ease of implementation, using a basic assembler supplied by the computer manufacturer. This assumes an assembler that operates in the test machine. If translation is performed on another computer, the entire translator must be specially written, including the basic assembler. In this case, the test and auxiliary computers define the characteristics of the translator and a change in either computer will require that changes be made in the translator.

3) L3 - procedure processor - By virtue of the fact that the object code from the procedure processor can be changed with relative ease, the procedure processor achieves a certain amount of independence from the test computer. It is, however, somewhat dependent on the auxiliary computer.

4) L4 - test compiler - The test compiler is highly dependent on both the test and auxiliary computers and, because of its complexity, would be difficult to convert. Initial development would not be affected significantly by the inclusion of an auxiliary computer into the system.

5) L5 - meta-compiler - The meta-compiler can be adapted more readily to the dependent system than any of the other translators because of its high degree of machine independence.

Translator selection.- The interpreter was selected as the configuration offering the best all-around compromise for the initial, minimum systems. The limited core size and power of these initial systems prohibit the implementation of a procedure processor or compiler within the system; and while an auxiliary installation might be used for translation, the limited interaction and on-line change capability available makes such an operation too unwieldy to be recommended. Furthermore, because special development for the small initial computer would be required, these larger software systems represent a significant investment very early in the development program that, in our opinion, would be significantly out of balance with the rest of the system.

The choice between an interpreter and a macro-processor is somewhat difficult. The macro-processor can be considerably less limited in language than the interpreter; it is less demanding of core during test execution and, once assembled, can be executed more efficiently. On the other hand, the interpreter permits on-line change in symbolic language and does not require test shut-down for translation. This last point becomes very significant if additional test stations are to be time-shared by the same limited computer. The interpreter will permit testing to proceed at a degraded rate concurrently with interpretation of a new program.

The small-scale computer projected for use in the initial installation does not provide a macro-processor as part of the standard software. If a translator is to be written, the flexibility and time-sharing advantages of the interpreter make it the preferred choice.

Conversion to a medium-scale CPU is the second decision point. With 8000 words or less of memory available, the compiler or procedure processor approach is again ruled out. A major revision to the existing interpreter will be necessary for it to operate on the new computer. It is possible that a macro-processor that will operate in 8000 words will be available. The choice then is between a major modification of the existing interpreter or a modification of the macro-processor. The decision is heavily influenced by the next few expansion steps. With 12 000 to 16 000 words of core available, foreground/background processing becomes a possibility and a compiler or procedure processor may be used. If either a procedure processor operating in the test computer or a meta-assembler operating in other installations is available, the conversion to higher-level language may be made with reasonable economy. Extending backward from this, it is then recommended that the interpreter be modified to operate in the minimum configuration of the new computer. This offers several advantages as follows:

1) Continuity - The language and operating methods familiar to the operators and test engineers are continued until the conversion to a higher level language. Use of the macro-processor, on the other hand, would introduce new language operating methods for an interim period.

2) The interpreter can be maintained within the system even after a higher level language is implemented. It is quite plausible that future users would sometimes prefer the simple flexible approach that the interpreter offers over the additional power of a compiler. In addition, even though the main program is written in the compiler language, it is quite feasible that supplementary or experimental tests may be desired and, for this, the interpreter could be used.

3) Continuation of the interpreter would offer more compatibility with existing test programs, minimizing conversion costs.

Choice of a higher level language, when the machine size permits, is heavily influenced by availability and economics. Objective differences exist between the procedure processor approach and the compiler approach, even at the same power of language. The procedure processor operating in the test computer offers more flexibility than a meta-assembler operating in an auxiliary installation. Statements may be rapidly added and changes to the test station simply implemented in the language. Multiple sets of procedures may be maintained within the system with much less storage requirements than multiple compilers; therefore, it becomes practical to have several languages, each oriented around specific problems for testing activities.

This possibility should not be minimized. The laboratory researcher dealing with a mass spectrometer uses a different natural language in talking to his experiment than the component test engineer. Even within component testing, different natural languages exist (e.g., functional testing of large-scale arrays are most easily described by truth tables or Boolean equations, while parametric testing of transistors is best described in terms of the standard parameter symbols, such as H_{FE} and I_{CBO}). All of these languages may or may not generate similar commands to the test station. With a procedure processor, it is possible, economical, and operationally simple to provide procedure decks for each language, selectable by control command at translation time; no change to the basic processor itself is required, and no interaction between multiple languages can occur. With the meta-compiler approach, either a complete unique compiler for each language or a more complex single compiler, servicing all the languages, would be necessary. The former approach is quite expensive in terms of storage requirements, and the latter risks interaction and the reflection of new problems into existing translations when a new language is added.

The procedure processor, however, is slower in translation, often significantly so. The availability of an interpreter within the system would alleviate, to some extent, time demands on the system for translation because difficult sequences can be worked out and developed interpretively without the necessity of full translation. The question really revolves around the ultimate level of language power possible or suitable for testing purposes. The true compiler, whether generated by another (meta) compiler or specially written, is probably capable of inherently higher power than a procedure processor. There is considerable doubt as to whether the generality possible with the true compiler can be used in the test situation. Very frequently, strict control of test station hardware, or even computer analysis segments, is necessary. One has considerably better opportunity for control of the generated code with a procedure processor than with a compiler.

An ultimate system might well be one that combines a powerful and flexible translator with an interpreter. In this system, the translator would generate, not binary machine code, but a simplified symbolic source language at the macro level. This symbolic code then becomes an input to the interpreter. This approach would offer the power that an interpreter alone cannot achieve and the on-line change capability that a pretranslator cannot offer. In addition, the intermediate (simplified symbolic) output of the translator would be accessible to the test engineer, so that critical time sequences or other problems demanding tight control of the generated code could be examined and rearranged before test execution.

Recommendations.- The selection of the most suitable translator and language based on the considerations given above, is made in each detailed system configuration earlier in this section. Specifications for the recommended translator for Phases I, II, and III are given in the section following.

SPECIFICATIONS

This section will specify a system and each major element in sufficient detail to provide a basis for procurement activity, either as a complete system or as separate elements.

Approach

General recommendations.- The following guidelines are established as a result of the previous sections:

1) Initiate a working test system at minimum cost;
2) As requirements and usages grow, incrementally increase capacity of test system;
3) Each augmentation of the system should, ideally, be a superset of the previous system. Therefore the later growth paths should be determined early in the acquisition cycle;
4) Subsystems should be as independent of each other as possible, so that expansion or change in one element (e.g., the computer) reflects minimally into other subsystems (e.g., the test station);
5) The basic system design should be independent of technical test parameters (stimulus and monitor capability) as far as possible;
6) The system should exhibit the characteristics defined in the requirements analysis section;
7) Benefits not directly related to testing should be taken advantage of where feasible.

Phasing plan.- To implement the above recommendations on a logical basis, three phases of acquisition are defined.

1) Phase I - start-up,
   a) The objective of this phase is to "get in business,"
   b) Available equipment is used wherever possible to produce a working minimum system for functional testing of a single class of devices (large-scale arrays),

c) Some compromise is made in the area of software and the test language because the Phase I software investment will be largely lost when progressing to Phase II. The restrictions imposed by a limited central processor also force software compromises;

2) Phase II - growth,

a) The objective of Phase II is to establish a system that is expandible to any foreseeable ERC need of the next five years,

b) The first implementation provides a minimum-configuration, third-generation central processor. The test stations are carried over from Phase I. An operating system is introduced, and a more powerful language is provided.

c) Conversion from Phase I can be accomplished without excessive loss of investment if specifications defined herein are followed,

d) Later expansion within Phase II provides,

(1) Timesharing of multiple test stations,

(2) Powerful test language,

(3) Sophisticated data analysis and management,

(4) Foreground/background operation (minimizes conflict of testing and language translation),

(5) System capability for any test category defined in the "Requirements Definition" section;

3) Phase III - exploitation,

a) The objective of this phase is to exploit, both technically and economically, the base developed during Phase II,

b) The central processor is modularly expanded to provide time-sharing test and batch service to other users (who may be widely scattered),

c) Users may share in the amortization of common equipment and software at a very significant cost savings,

d) Very advanced test and data management techniques are possible, including,

(1) A network of common test programs and algorithms,

(2) A common test data base,

(3) Powerful language and computation capability,

(4) User communication via the central location, for real-time data correlations and similar cross-use purposes.

Organization of specifications.- The system is divided into the following subsystems, which are separately specified:

1) Test station;

2) Input/output control;

3) Central processor;

4) Operating system;

5) Test language and translator.

The test station and input/output control are common to all phases. The central processor, operating system, and test language translator are each specified for Phases I, II, and III.

Test Station

Design concepts.- Analysis of the test situation indicates that many functions associated with control, sequencing, timing, and "housekeeping" are common to almost any visualizable test. Other functions associated with stimulus and measurement values, levels, and dynamics can be standardized to some extent, but are, for the most part, highly subject to change.

The basic concept developed for the test station is:

1) To provide a common station core for standard functions;

2) To provide a series of modular test channels, each serving a particular stimulus or monitor function, which are controlled and serviced by the common core and provide a standard interface to it;

3) To use a simple unique adapter between the test channels and the test unit itself -- to provide special loads, test unit connection, and other functions unique to a specific test article;

4) To make the test station computer-independent by providing a separate, small I/O control unit that will adapt the computer to the standard test station control interface, thus limiting the impact of different computers on the test stations to a single, small device.

Recommended word size for the test station is 32 bits. This size has been selected over several alternatives for the following reasons:

1) Eight bits should be reserved for device address. This permits future expansion to 256 test equipment devices. This 8-bit reserve is also compatible with most third-generation byte sizes.

2) Present state-of-the-art resolution in test equipment is equivalent to 15 bits or higher. While address and data value could be contained in 24 bits (8-bit device address and 15-bit data), only one control function could also be contained in the same word (such as single/continuous mode command or range).

3) For discrete, "bit map" control, only eight lines or functions can be contained in a 16-bit word in addition to address, versus 24 in a 32-bit word. This is an increase of three times in the contained information, for a doubling of word size.

4) Many medium-scale third-generation computers are structured around a 32-bit word.

For a smaller word-size computer, the I/O control unit may provide packing of multiple smaller words to form a complete 32-bit command.

The test station specification presented in this section is directed toward implementing the above design concept. Interface functions have been generalized and reduced to the minimum required conceptually. Signal levels, durations, and other quantitive specifications are given only where necessary to ensure compatibility of separately procured items. Obviously, with the wide variety of nonstandard control interfaces offered today

by test equipment manufacturers, something has to give. By standardizing the basic station control interface, the burden of signal conditioning and conversion is placed on the test equipment test channels (not necessarily within the test equipment itself), which is usually a unique or special device to begin with.

No actual test station is known to the study team that will meet all of the specifications defined in this section. But one must start somewhere; accepting a presently designed unit as a starting point simply because it exists only postpones the problems to the first attempt to modify or expand. It is doubtful that any off-the-shelf unit would meet the functional test requirements. If modification or some design activity is to take place, the standard module approach is best implemented at once. It is believed that the specifications are reasonable, in the mainstream of present digital techniques, and should offer no major problem to system houses conversant with modern integrated circuits.

The specifications are divided into the following categories:

1) Control interface;
2) Standardized test channel interface;
3) Basic station;
4) Operator's panel;
5) Test interface.

CPU control interface.- All input signals to the test station shall be transformer-coupled to provide a cumulative isolation greater than 1 megohm between test station and computer equipment.

Output signals from the test station shall be capable of driving 50-ohm coaxial line at an amplitude of 8 ± 2 V positive going, for an ON or true condition, with a minimum duration of 0.2 μsec and rise - fall times less than 20 nsec. An OFF or false condition shall be 0.5 ± 1 V, with a minimum duration of 0.5 μsec. Noise and ripple shall not exceed 0.5 V for either state.

ON or true input signals to the test station shall be greater than 4 V positive going, including all noise and ripple, with a minimum duration of 0.2 μsec, and rise or fall times less than 50 nsec. OFF or false signal levels shall be the same as specified for output signals.

Control input signals to the test station are defined as follows:

1) COMMAND DATA PRESENT - A signal indicating the presence of valid data on the data lines. The fall of this signal shall be the strobe for command data;

2) DATA LINE n - Thirty-two lines carrying parallel command data to the test station;

3) COMMAND PARITY - A signal from the central processor that provides even parity for the 32 data lines;

4) COMPUTER PARITY ERROR - A signal indicating that parity did not check when result data were received;

5) RESULT DATA RECEIVED - Indicates that the previous transmission of the result data word has been stored in computer memory and the CPU is ready to receive the next data word.

Control output signals from the test station are defined as follows:

1) REQUEST FOR ACCESS - When true, indicates that valid result data are present on the result data lines. The fall of this signal is used by the CPU as a strobe for result data;

2) RESULT DATA LINE n - Thirty-two lines carrying a parallel result data to the CPU;

3) RESULT PARITY - A signal generated by the test station to produce even parity for the 32 result data lines;

4) INTERRUPT 1 - Generated by the test station when all requested results have been transmitted to the CPU and the current station operations are complete;

5) INTERRUPT 2 - Generated by the test station by operator activation of a control.

Standardized test channel interface.- Control interface signals to and from all test channels shall be compatible with DTL integrated circuits. Signal conditioning shall be provided, as necessary, by the test channel. The interface for stimulus test channels is defined in figure 18. The interface for measurement test channels is defined in figure 19.

Basic station control.- The functions of this subsystem of the station are as follows:

1) Receive commands from the CPU;
2) Verify correct parity;
3) Decode operation portion of command;
4) Store command in input buffer during processing;
5) Drive command data bus;
6) Provide individual device selection signals required for test channels;
7) As required, provide separate device buffers for test channels;
8) Provide logic and sequencing necessary to activate measurement test channels;
9) Sequence measurement data, in order of hierarchy, to CPU result data lines;
10) Drive result data lines to CPU;
11) Generate parity for transmission of words to CPU;
12) Provide drive and control for operator's panel;
13) Detect END-OF-COMMANDS operation code or signal and initiate measurement operations.

Operator's Panel.- The operator's panel will provide, as a minimum, the following functions:

1) Display of current command data word;
2) Display of current result data work;
3) A means of locally programing command data words with the computer locked out;

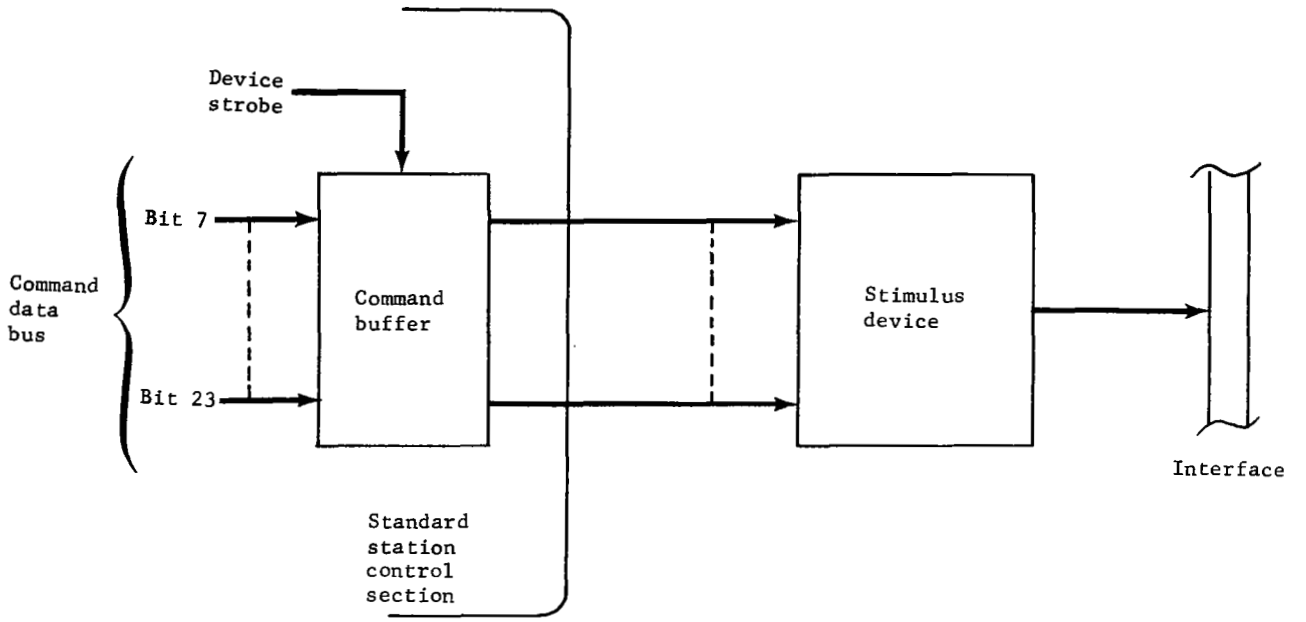


Figure 18.- Typical Stimulus Channel

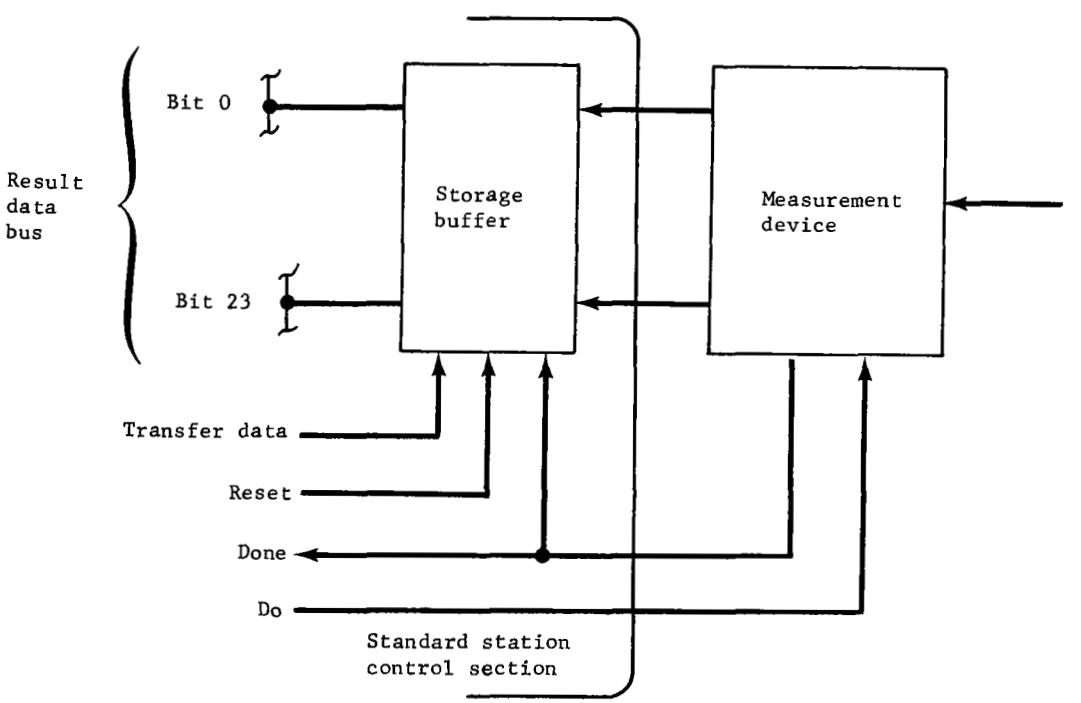


Figure 19.- Typical Result Channel

4) Mode switching to provide,

   a) Automatic (computer) operation,

   b) Step operation (accessing of one command at a time under operator control),

5) A minimum of four breakpoint switches;

6) A manual station reset;

7) Two interrupt switches;

8) Proper error indicators for parity errors, etc;

9) Power switches and indicators.

Test interface.- The purpose of the test interface is to provide a standard, external set of connectors through which the signals and power to the various test channels are routed.

The test interface will interconnect to an adapter that will, in turn, mate the test unit. The adapter, which is a separate deliverable unit and not part of the remote station, provides the necessary patching, special loads, filters, and other functions unique to a specific test unit.

The connector interface (to the adapter) shall use self-aligning panel-mounted connectors, for easy and rapid insertion and removal of adapters.

Provision shall be made for up to 400 interface points through the test interface connectors.

Power, high-level ac, and switched transient loads above 1 A shall be routed through a separate connector.

Input/Output Control

The function of the input/output control (IOC) is to provide signal level, buffering, and logic interface between computer memory and the test station so that the test station itself may remain independent of the computer.

The IOC will be designed to provide initial control of a single test station; however, the design will provide for expansion capability for up to eight remote test stations driven from a single computer.

The IOC shall be located as close as practical to the computer interface and preferably be installed in the computer itself.

The interface shall be with the direct memory port of the computer. This interface is variously called second path, direct memory access, and other terminology by the different computer manufacturers. It is distinguished by the requirement to externally provide a memory address; by the fact that program attention is not required for a memory access; and that accesses may be "read" or "write" interleaved in any sequence.

The test station interface is defined under "CPU Control Interface"; the IOC will supply or accept all signals described therein as "CPU."

The computer interface is unique to a specific computer and will not be defined herein.

The IOC will be capable of accessing memory every computer cycle, if so driven.

The IOC will provide packing for computer word sizes less than the 32 bits required by the test station.

Central Processor Specifications

Phase I.- The H-516, properly configured, will satisfy the testing requirements for Phase I. The central processor configuration is defined in the "Configuration Study" section, Configurations A1 thru B4.

Phase II.- The primary requirements for the Phase II central processor are as follows:

1) Processor,
   a) Comprehensive instruction repertoire, with at least 6-bit Op code,
   b) Maximum of 1.75 μsec cycle time,
   c) Indirect addressing with up to "n" levels of nesting,
   d) Real-time clock with a minimum of 100 μsec resolution,
   e) Either momory write protection or memory lockout so that programs below the "master" level cannot alter memory outside their allocated core storage,

f) A true hardware priority interrupt system, expandible to 64 interrupts,

g) Privileged instructions to permit control of memory protection or lockout, enable/disable of the priority interrupts, and control of the real-time clock,

h) Direct addressing of entire (maximum) memory,

i) Hardware-assisted environment switching,

j) Automatic traps for detection of error conditions (e.g., memory faults, division by zero, register overflow, etc),

k) Sixteen general registers, expandible to eight groups,

l) Floating point arithmetic;

2) Memory,

a) Minimum word length of 32 bits,

b) Simultaneous, overlapped access through multiple ports,

c) Memory expandable to 131 072 words in 4000 word increments,

d) Parity checking on memory accesses;

3) Input/output,

a) Buffered input and output for communication with external devices; once initiated, such I/O may continue without further action required by the processor,

b) Completion of buffered I/O will optionally trap or interrupt the processor,

c) N I/O channels will be available for communication with peripheral devices. Transmission will include a parity bit,

d) A direct-access for the second-path I/O channel will be provided, completely independent of buffered and peripheral I/O. This channel will be capable of transfers at maximum memory cycle time, and will provide for interleaved "read" or "write";

4) Peripheral equipment,

a) Operation terminal with keyboard and either printed hardcopy output or cathode ray tube character display;

b) Card reader (binary and Hollerith read) - Minimum of 250 cards per minute,

c) Card punch (binary and Hollerith punch) - Minimum of 200 cards per minute,

d) Line printer - Minimum of 500 lines per minute and 132 print positions per line,

e) Magnetic tape - One controller and two subsidary magnetic tape transports. Minimum specification for the magnetic tape transports is 30 in./sec, with recording densities to include 556 and 800 bits per inch,

f) Random access storage - One device controller and storage device with minimum capacity of 300 000 characters (8 bits/character), and worst-case access time of 50 msec. Transfer rate shall be at least 250 000 characters/sec. Random access storage must be expandible to 3-million characters.

Phase III.- The Phase III central processor is identical to the expanded Phase II central processor with the addition of 3-million characters of random access storage and 2400 bps data sets for communication with remote terminals.

Test Language and Translator

Phase I.- The recommended language processor for Phase I is an interpreter (Configuration L1). The relatively limited language possible with an interpreter is a significant disadvantage; however, other considerations outweigh this limitation as follows:

1) Minimum investment in system software is desirable for Phase I because such software will generally not be compatible with Phase II;

2) The interpreter is relatively low in cost;

3) Experience with a low-cost language permits development of criteria for more expensive versions in Phase II;

4) The simplicity of the language is of benefit in the early implementation phases;

5) The usage flexibility of an on-line interpreter is of high value, particularly with a limited CPU;

6) If necessary, the interpreter can be converted to Phase II at relatively low cost.

Language characteristics are as follows:

1) One-to-one correspondence between statements of the test language and test functions;

2) Statement formats may be fixed field. However, ease of use is a prime requirement;

3) Use of meaningful "key words" is required in the language wherever feasible;

4) Specification of quantitative parameters shall be possible in decimal number base and in convenient units;

5) The language shall be oriented to the test station rather than to the unit under test.

Operational characteristics are as follows:

1) A complete set of functions for testing shall be possible by use of the test language. In particular, all capabilities of the test station shall be included in the repertoire of functions;

2) Functions shall be provided to permit specification of time limits, time delays, and other time-critical operations as required for testing;

3) The interpreter shall be capable of processing at least 100 prespecified, consecutive functions as a group before execution;

4) Conditional transfer between statements within a prespecified group of at least 100 statements shall be possible with a maximum time of 100 μsec allowed to effect the transfer;

5) A conditional transfer may refer to a group of statements currently residing in bulk storage. Statement labels will be provided to permit such a reference;

6) The interpreter will perform extensive function and parameter checking. In the event of an illegal function or parameter, the execution of the test will be halted, and adequate diagnostic information will be provided;

7) The test operator shall have the capability, via the operator's console, to define or alter test functions and parameters. This capability shall be facilitated by entry of test language statements at the keyboard.

Restrictions imposed by system configuration: Resident core storage requirements for the interpreter shall be held to a minimum. A prime requirement is that the resident memory requirements of the interpreter, test monitor, and all other resident software is such that Item 3), above, may be satisfied. This requires that the interpreter operate within a maximum memory size of 6000 words in the small-scale computer.

Phases II and III.- One of the primary advantages of the Phase II system is the incremental expansion possible with the system. It is necessary that the test language be expanded to accommodate new functions, but a change during Phase II of the test language basic configuration is undesirable. The procedure processor (L3) provides the most flexibility together with a powerful language capability. The relatively slow translation speed is not expected to be a serious disadvantage in the ERC usage environment. This configuration is therefore recommended as the next language processor and is specified below.

Other alternatives exist, however. Conversion of the Phase I interpreter to the Phase II system will maintain continuity in the test language, although still restricting the language power. An optional path is to do both, i.e., a combination of procedure processor and interpreter. This would provide an easily changed and expandible compiler-level language as a pretranslator, and an on-line, flexible interpreter.

The specifications that follow assume either a dependent system or background/foreground capability in the test computer.

Test language characteristics.- Characteristics are:

1) Statements written in the test language shall provide the capability to define tests and associated operations to a level corresponding to individual functions of the test station. All functions of the test station shall be definable in the test language.

2) Test language statements will permit use of "near-English" expressions and will be oriented to the various test requirements.

3) The Phase II test language will permit "free-form" formats.

4) The processor shall provide for specification of subprocedures that may be referenced in higher level procedures. Subprocedures shall have the capability to accept variable arguments from the referencing procedures. Nesting of subprocedures shall be possible to five levels.

5) The test language shall provide for specification of parameters in accepted engineering units; quantitative values may be specified in decimal number base or scientific notation.

6) The translator shall, in a single operation on the computer, process source language statements and generate object binary code executable by the computer or test station.

7) Outputs from the translation process shall include,

   a) Printed listing of source statements;

   b) Adequate error diagnostics that relate directly to a) above;

   c) Expanded list (optional) - This shall display the object code generated by each source statement.

8) The object code resulting from the translation process may be optionally directed to any file or media (e.g., punched cards, magnetic tape, auxiliary storage).

9) The object code shall be relocatable.

10) External references or definitions shall be possible. The translator will generate the proper information to enable the operating system to process the references or definitions at load time.

11) A complete set of arithmetic operations shall be provided to include addition, subtraction, division, multiplication, and basic mathematical functions. Arithmetic expressions will use established mathematical and scientific notation.

12) Conditional transfers or symbolic references may be to any statement within the procedure. Alphanumeric symbols, up to eight characters long, may be attached to any statement to facilitate the conditional transfer. The logical expression will accept variables generated in the program or generated as the result of a testing function.

13) It shall be possible to define, by system procedures only, the object code to be generated by any language statement. New language functions shall be implemented by definition of a new procedure. Alternative sets of system procedures may be resident, and the specific procedure set to be used shall be definable via operator control statement at the time of translation. Source language statements shall consist entirely of procedure reference lines. Reference parameters shall be variable in number; and literals, arithmetic or logical operators, symbolic expressions, and single or double-precision, fixed or floating point quantities, shall be permitted.

14) The test language processor shall be designed to conform with system specifications (e.g., core storage requirements).

15) Test object code shall conform to the specification of the supervisory system. Functions performed at execution time, which require interfacing with the supervisory system, shall conform to procedures and requirements of the supervisory system.

16) Object code generated by the test language translator shall be efficient and conform to accepted software concepts.

Supervisory Systems

Phase I.- Because of the minimum Phase I computer, the Phase I supervisory system will be limited to the test monitor. Assemblies and other off-line operations cannot be performed concurrently with testing; during testing the computer will be dedicated to the testing operation.

Off-line processing: In the minimal Phase I system, the specification that follows may be satisfied by manual loading of the proper routine or module to perform the specified function.

The following capabilities and program modules shall be included:

1) Standard two-pass assembler capable of generating relocatable object code;*

2) ASA standard Fortran IV compiler capable of generating relocatable object code;*

3) Relocatable loader capable of accepting object code from either 1) or 2) above. Object routines from both 1) and 2) above may be submitted and proper linkage will be made by the loader;*

4) ASA subroutines and basic closed mathematical subroutines, shall be provided;*

5) Separate identifiable programs for driving individual peripheral equipments shall be provided;*

6) A symbolic editor that will allow editing on a record-by-record basis of a single file in auxiliary storage shall be provided.

Test monitor.- To minimize the size of test programs, the test monitor is in direct control of all test operations in the Phase I system. All utility routines, etc, required for test shall be integrated into the test monitor as submodules. The specifications for the test monitor are categorized as follows:

1) Execution of tests;

2) Time sharing of test stations.

Execution of tests: Tests are executed as follows:

1) The prime job of the test monitor is the proper execution of test functions specified in the test language. The test monitor shall control the interpretation and provide for the execution of test functions as defined in the test language specifications;

2) Functions for which timing constraints have been specified shall be executed accordingly with the aid of the real-time clock;

3) The test monitor shall communicate with the IOC for initiation of all test station operations;

---

*Items 1) thru 5) are normally supplied by the manufacturer of the central processor at no cost.

4) Error checking will be provided, as far as possible, with regard to test functions and operation of the test station(s). Recovery procedures shall be provided to minimize catastrophic failure of the monitor in case of error;

5) Test programs will not access peripheral devices. The test monitor will provide all peripheral device input/output (e.g., data acquisition in bulk storage files, messages);

6) The status of the test, test stations, etc will be communicated to the operator via the operation console;

7) The capability to generate or alter a test from the operator's console shall be provided.

Time sharing of test station: The initial test monitor for Phase I will be capable of servicing a single test station. Expansion of the system to service up to three test stations shall be possible with appropriate additions and modifications to the system. The test monitor shall be designed so that these additions or modifications are made at the module level without major redesign.

The time-sharing capability of the expanded test monitor shall be such that the device under test cannot detect that it does not have the full attention of the system.

Phase II.- Phase II is discussed in the following paragraphs.

General considerations: The primary advantage of the Phase II system is the potential for incremental expansion. The Phase II supervisory system, then, must be designed so that a more powerful test language, additional time-shared test stations, etc, can be accommodated with minimum cost and modification of the supervisory system.

The specification that follows assumes the implementation of background/foreground processing.

The main attributes of the Phase II supervisory system which distinguish it from the Phase I system are:

1) Background/foreground processing;

2) Management of auxiliary storage and basic file manipulation;

3) Overlay or segment loader;

4) Checkpoint dump capability;

5) The test monitor operates under control of the supervisor.

The following program modules shall be included with the Phase II supervisory system:

1) Supervisor;

2) Test monitor;

3) Operator interface;

4) Real-time controller;

5) Input/output scheduler;

6) Background processing.

Background processing: The following capabilities and program modules shall be available in the background mode:*

1) Standard two-pass assembler with macro processing, capable of generating relocatable object code;

2) ASA Fortran IV compiler capable of generating relocatable object code;

3) Relocatable loader capable of accepting object code from either 1) or 2) above. Object routines from both 1) and 2) above may be submitted and proper linkages will be made by the loader;

4) ASA subroutines, basic closed mathematical subroutines, and normal utility programs;

5) An overlay or segment loader capable of accepting specified "overlays" or "segments";

6) Provisions for communication between background programs and peripheral equipment;

7) Basic file manipulation to include copying, editing, and positioning on a record basis;

8) The supervisor will perform any of the operations described in 1) thru 7) above as directed by control information supplied with each job;

---

*Normally supplied by the manufacturer of the central processor at no cost.

9) The supervisor will have the capability to accept and sequentially process jobs stacked in a specified file in the background mode.

Supervisor: The supervisor shall:

1) Allocate the central processor to the background or foreground function as required to meet the requirements of the test station(s);

2) Generate records that contain adequate accounting information for each background and foreground job. The records may be directed to a storage file;

3) Provide a means of temporarily inhibiting background operations (such as I/O) or peripheral devices that will conflict with a time-critical foreground (test) operation upon request of the test monitor;

4) Introduce no more than a 30-μsec delay between occurence of a foreground interrupt and transfer of control to the test monitor;

5) Ensure that direct, dynamic linkage of an interrupt to a foreground program shall be possible upon request of the test monitor;

6) Service all requests (either directly or via internal routine) for access to bulk storage and other peripheral equipment.

Test monitor: The test monitor shall:

1) Provide for execution of tests presented in the form of object code from the test translation process. The test monitor will operate in the foreground mode;

2) Accomplish specified time-critical tasks by interfacing with the real-time controller.

Operator interface: The operator interface module shall provide the capability for limited operator interaction as follows:

1) Major test milestones, such as initiation of a test segment load and termination, shall be communicated to the operator via the operator's console;

2) Test data shall be stored, formatted, and displayed;

3) The operator may redirect, alter, add, or delete test sequences via the keyboard/display.

Real-time controller: Maintenance of timers and timing interrupts and requests shall be handled by an individual program module.

Input/output scheduler: Requests for access to peripheral devices shall be directed to the input/output scheduler. Requests shall be assigned priorities so that the requirements of testing are satisfied. This may mean the requests from the foreground operation are given priority over background requests.

Time sharing of test stations: The Phase II test monitor shall have the capability to timeshare multiple test stations. Although the initial system may service only a single test station, the supervisory system must be designed so that additional test stations, to a total of eight, can be accommodated with additions and modification at the module level only.

The time-sharing capability of the expanded test monitor shall be such that the device under test cannot detect that it does not have the full attention of the system.

Phase III.- The Phase III supervisory system is a superset of the Phase II supervisory system. The main additional requirements are the servicing of remote batch terminals; significantly increased file management capability; and concurrent input/output. All user programs, including test programs, capable of operation under the Phase II system shall operate under the Phase III supervisor without modification.

Remote batch terminal service: The Phase III supervisory system shall be file oriented. All inputs and outputs will be filed in appropriate files on bulk storage. The supervisor will then service the files as associated peripherals become available.

The remote batch terminals will be serviced by a special program module on an interrupt basis. Buffers of information will be transferred between bulk storage devices and the remote terminal on a file-by-file basis. Between each transmission, a synchronization message will ensure proper communication and operation. Priorities of service may be assigned.

Whenever the terminal is in a ready status, synchronization messages shall be exchanged periodically. Any detected anomalies will be reported at both the central and the remote site if possible.

The jobs transmitted from the remote site to the central site will be stacked in the job file and eventually processed in the background mode.

File management: The supervisor will provide for creation, reservation, and deletion of dynamic user files. Private and public designation for any files will be possible.

Access to system or public user files will be possible by any user.

COST EFFECTIVENESS STUDY

General

The cost effectiveness study compares the partial and total costs of conventional test equipment to those of the test system specified in this report. The primary partial costs evaluated for both systems are those of equipment, equipment maintenance, and personnel.

The term "conventional" as used in this section refers to the standard automatic test equipment available commercially. It is generally special-purpose in that an item of equipment, including the computer, will test only one class of components (such as integrated circuits). The term "specified" refers to the system defined in this report; it is characterized by general-purpose use of the computer and other common functions to control a variety of test equipment devices.

The receiving inspection laboratory of an aerospace manufacturer was selected as the baseline test installation. Screening and limited qualification testing are also performed in this laboratory.

Nonelectronic testing and equipment (i.e., vibration, seal, X-ray) are excluded from the study because they are common to both systems.

The result of the study is a factor, K, by which the total value of a development contract may be multiplied to obtain the approximate cost of component testing for that program. The K factor is calculated, for each of the two systems, as follows:

$$K = m\ n,$$

where

m = Total cumulative test cost determined by this study divided by number of components tested, as assumed by this study,

n = Number of components used in an average of three actual aerospace contracts divided by Contract overall value in millions of dollars.

The units of K are test cost per million dollar contract value. Therefore, multiplying K by any contract value in millions will give test cost.

Organization

The scheme of the cost effectiveness study is illustrated in figure 20. The study is summarized in table XXVI. Further details are provided by figures 21 thru 27 and tables XXVII thru XXXVII.

The baseline test installation is typical of several aerospace manufacturers' in-house testing facilities. Past growth is based on actual expenditures and equipment lists, as is test capability and staffing. Future growth is projected at the same percentage rate as past growth. "Growth" here is a composite of new technical requirements, test loading, and obsolescence.

Test requirements.- Test requirements consist of:

1) First electrical parameter measurement;
2) 168 hr burnin;
3) Second electrical parameter measurement;
4) 1000 hr life test (sample 2%);
5) 250 hr parameter test;
6) Environmental tests (sample 2%);
7) Last electronic parameter measurement.

The data reduction consists of:

1) Tolerance acceptance;
2) Delta limit calculation;
3) Percent of rejection;
4) Vendor data review;
5) Parameter distribution;
6) Correlation technique (automatic, cross, historical).

Quantity of components tested.- Table XXXVII shows the projected number of components to be tested (acceptance and qualification) during the period 1969 thru 1975. This test load is applied to both the conventional and automatic systems.

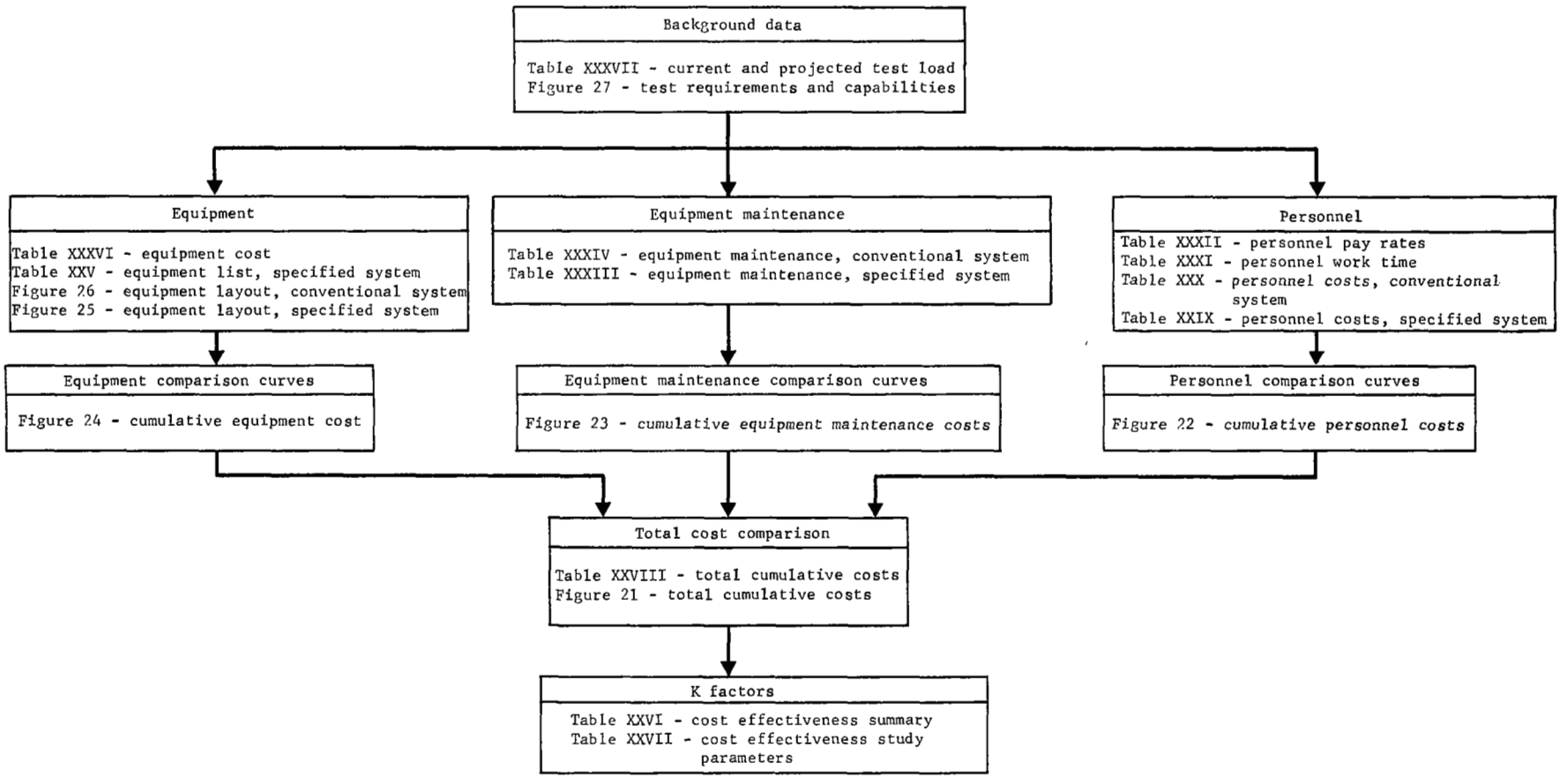


Figure 20.- Cost Effectiveness Study Organization

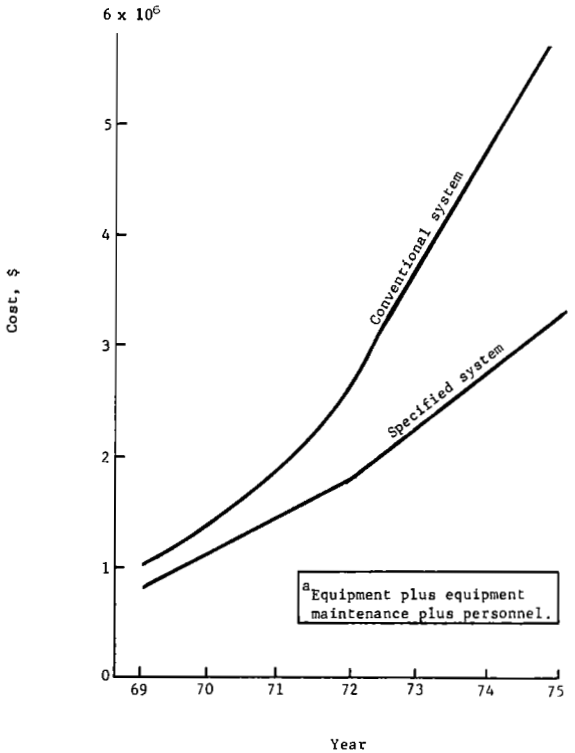


Figure 21.- Total Cumulative Costs[a]

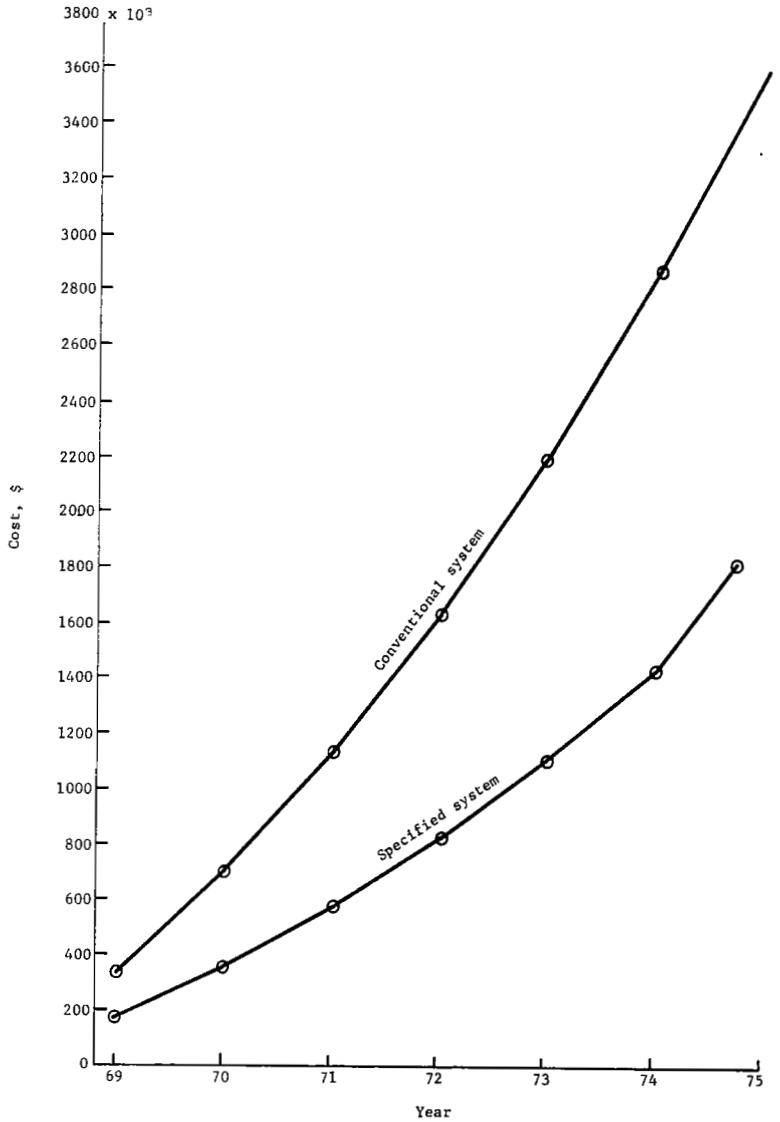


Figure 22.- Cumulative Personnel Costs

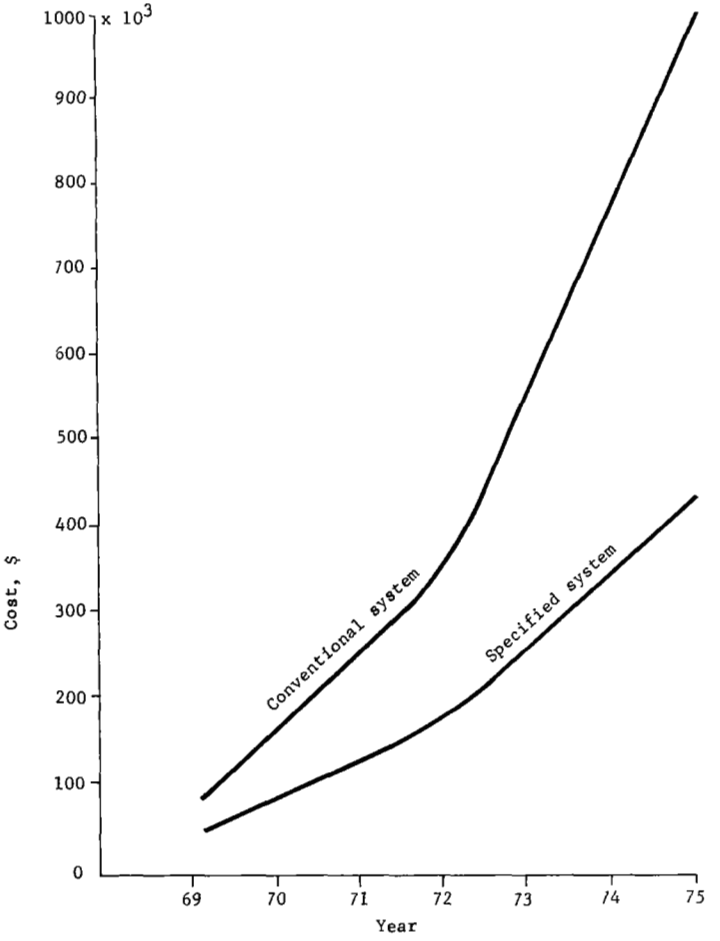


Figure 23.- Cumulative Equipment Maintenance Costs

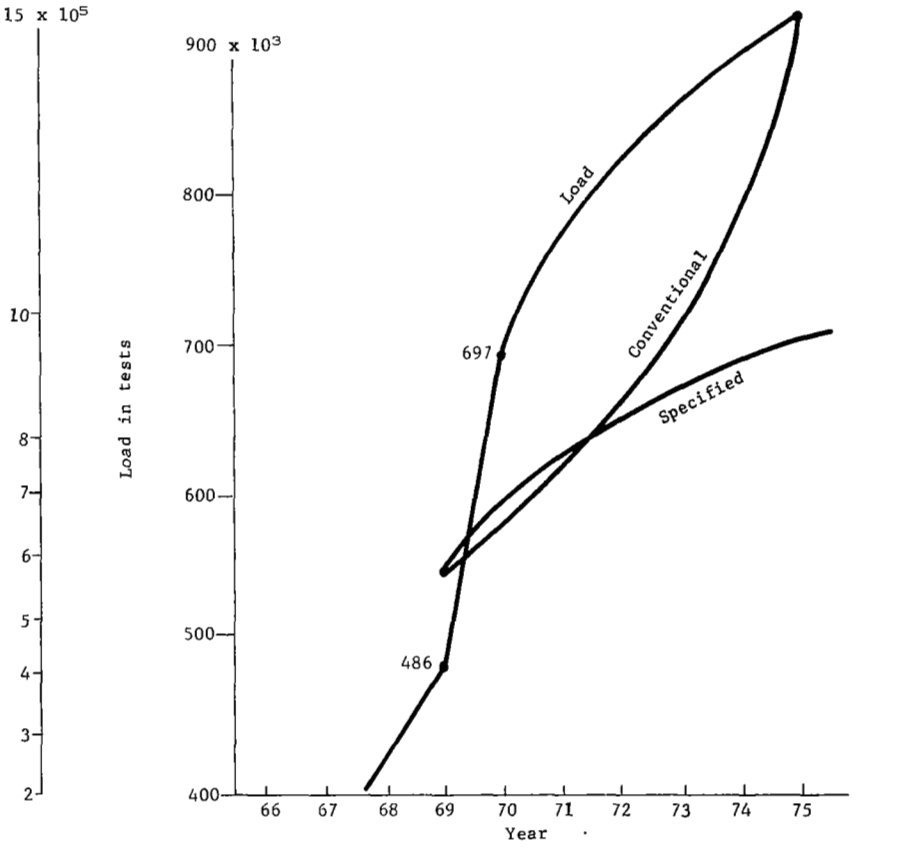
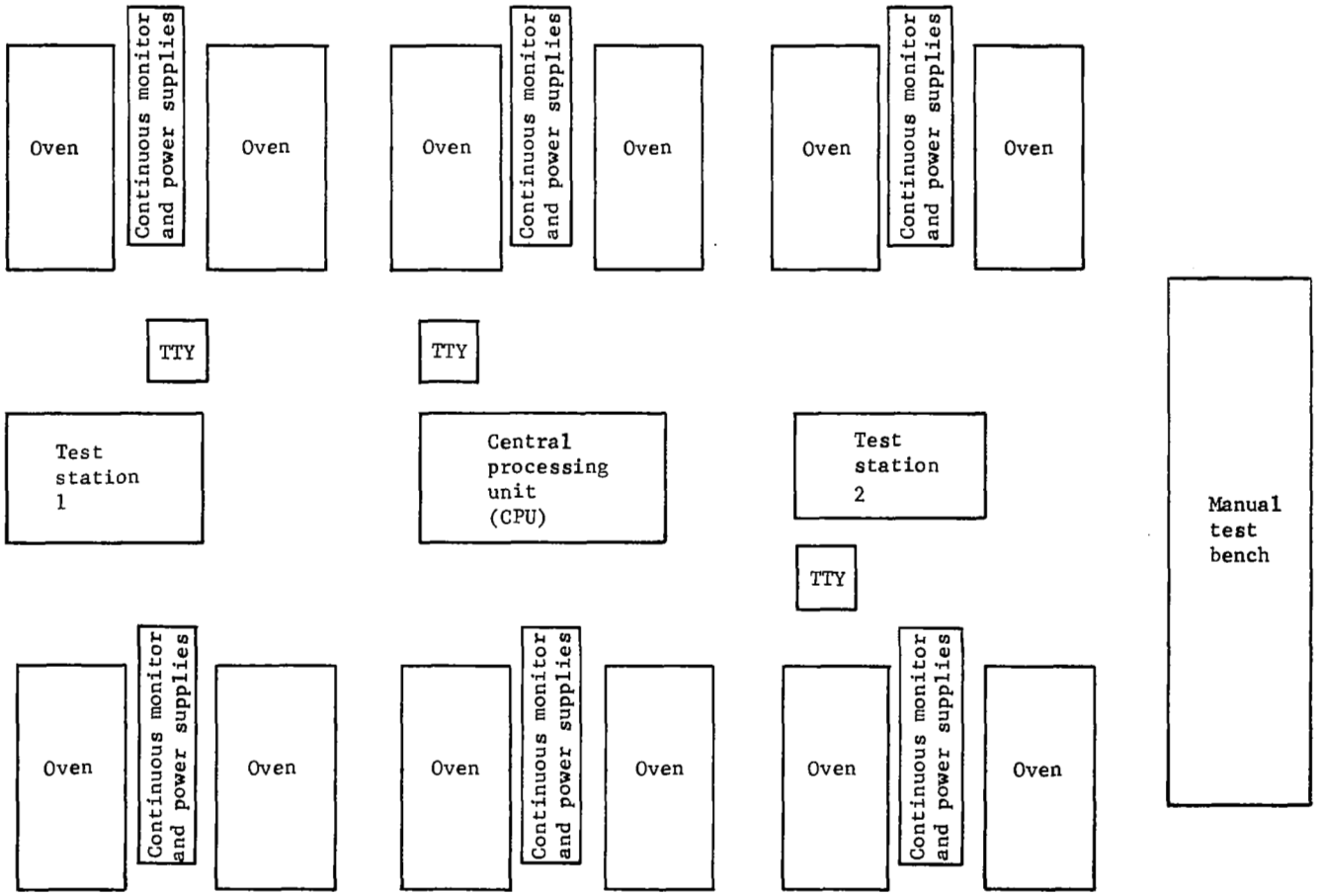


Figure 24.- Cumulative Equipment Costs



Note: 1. One test station 1969-1972; two test stations 1973-1975.
2. 48 equipment items.

Figure 25.- Equipment Layout, Specified System

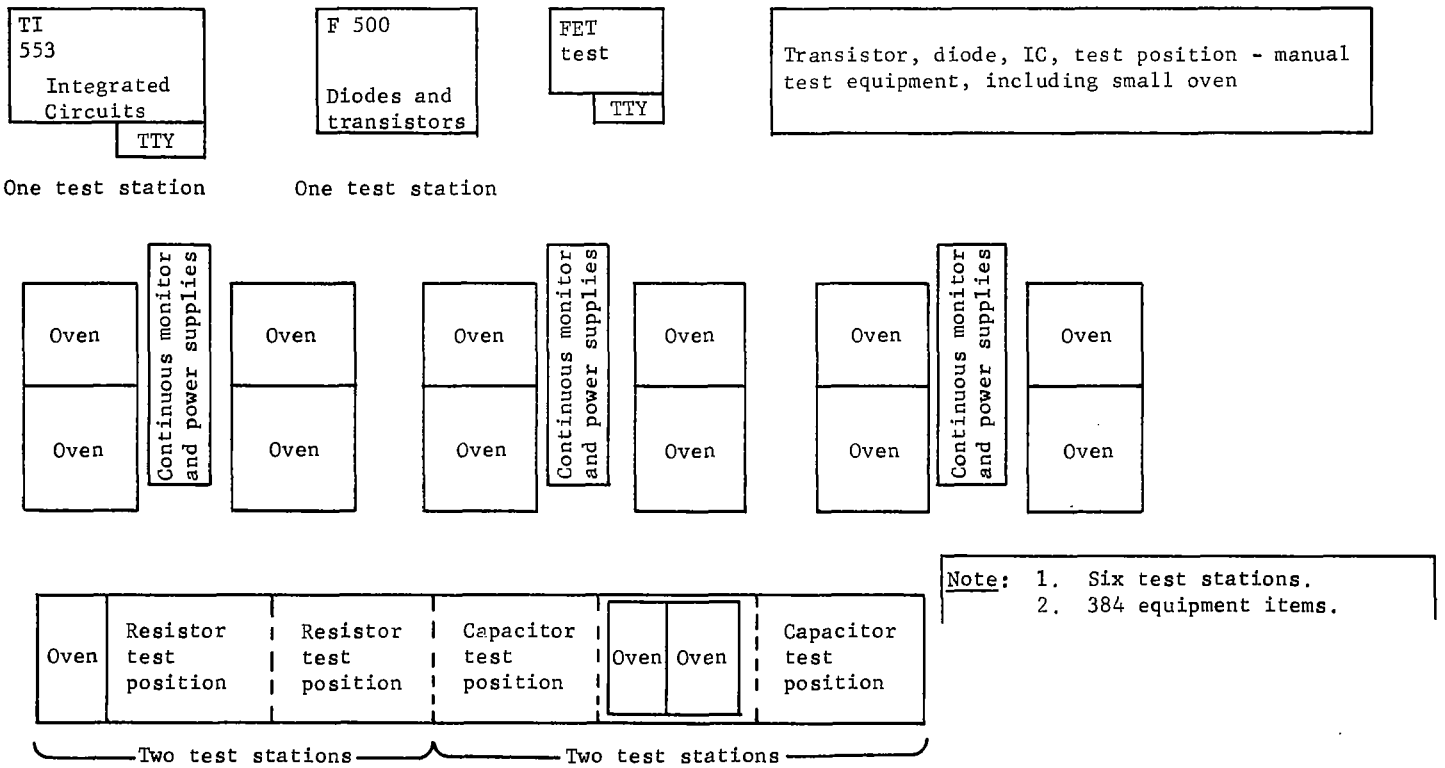
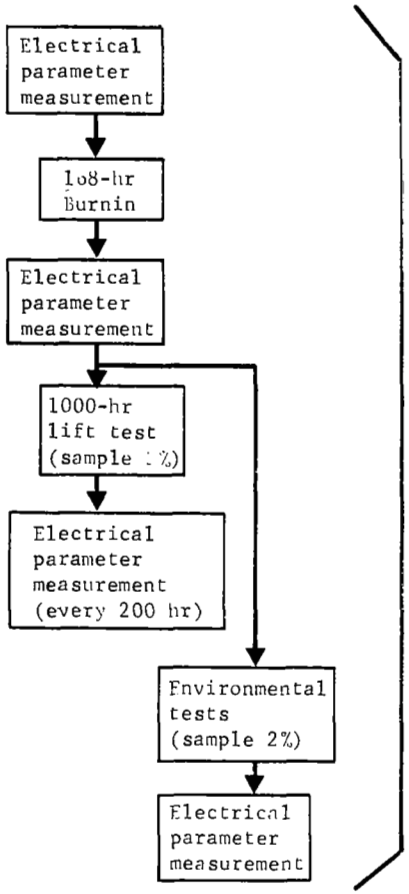


Figure 26.- Equipment Layout, Conventional System



Data reduction	Conventional system	Specified system
Tolerance	X	X
Delta limit calculation	X	X
Percent rejection	X	X
Vendor data review	Overview	Detailed
Parameter distribution	0	X
Correlation technique	0	X
Automatic	0	X
Cross	0	X
Historical	0	X

Legend:
X = performs
0 = does not perform

Figure 27.- Test Requirements and Capabilities

TABLE XXVI

COST EFFECTIVENESS SUMMARY

Item	Conventional system	Specified system	Difference
Equipment, $			
1. Original investment	272 031	272 031	0
2. 1969 investment (cum)	570 557	572 031	1 474
3. 1972 projected investment (cum)	869 083	831 031	-38 052
4. 1975 projected investment (cum)	1 466 135	970 000	-496 135
Equipment maintenance, $			
1. 1969 annual cost	82 900	35 270	-47 630
2. 1972 projected cost (cum)	429 250	190 950	-238 300
3. 1975 projected cost (cum)	1 048 250	484 495	-563 755
Personnel, $			
1. 1969 annual cost	327 000	163 946	-163 054
2. 1972 projected cost (cum)	1 650 692	829 046	-821 646
3. 1975 projected cost (cum)	3 640 980	1 851 533	-1 789 447
Total cumulative cost, $			
1. 1969 annual cost	980 457	771 247	-209 210
2. 1972 projected cost	2 949 025	1 851 027	-1 097 998
3. 1975 projected cost	6 155 365	3 426 059	2 729 306
Capability			
1. 1969 status			
People	19	8	-11
Days	3 971	1 672	-2 299
2. 1972 projected			
People	27.3	11.5	-15.8
Days	5 705.7	2 403.5	-3 302.2
3. 1975 projected			
People	35.6	15.1	-20.5
Days	6 387	3 155.9	-3 231.1
Test cost per component, $			
1. 1969 cost per unit	2.02	1.58	-.44
2. 1975 cost per unit	6.70	3.73	-2.54
Test cost factors			
1. Present (1969)	K = 3 020	K = 2 365	
2. Projected (1975)	K = 10 000	K = 5 580	

TABLE XXVII

COST EFFECTIVENESS STUDY PARAMETERS

Item	Test system	
	Conventional	Specified
Test facilities	Six stations Resistors - two stations Capacitors - two stations Transistors and diodes - 1 station Integrated circuits - 1 station	Two stations to test the same five component types
Equipment items	384	48
Equipment cost, $	570 557	572 031
Equipment maintenance cost, $/mo	3 200	1 900
Burnin labor, devices/day	8 000	8 000
Personnel required	Based on current rates (and projected 4% increase 1970 to 1975)	
Supervisor	One at $9.00/hr	One at $9.00/hr
Engineers	Four at $6.71/hr	Two at $6.71/hr
Technician (lead)	Two at $4.64/hr	----
Technician	Seven at $3.61/hr	Three at $3.61/hr
Clerks	Four at $2.97/hr	One at $2.97/hr
Secretary	One at $3.19/hr	One at $3.19/hr
Personnel growth	Based on projected test load 1969-1975	
Personnel lost time, days/yr	NASA-authorized:	NASA-authorized:
Holidays	10	10
Vacation	30	30
Sick leave	12	12

TABLE XXVIII

TOTAL CUMULATIVE COSTS

Item	Cost, $					
	Conventional system			Specified system		
	1969	1972	1975	1969	1972	1975
Equipment	570 547	869 973	1 466 125	572 031	821 031	970 000
Equipment maintenance	82 900	429 250	1 048 250	35 270	190 950	484 495
Personnel	327 000	1 650 692	3 640 982	163 946	829 046	1 851 533
Total	980 447	2 949 025	6 155 365	771 247	1 851 027	3 426 059

TABLE XXIX

PERSONNEL COSTS, SPECIFIED SYSTEM

Year	Components tested	Increase in components (sum)	Percentage increase (sum)	Annual personnel cost (sum), $	Cost increase above previous year, $	Personnel required	Personnel increase above previous year	Average cost per person per year, $
1969	486 000	----	----	163 946	----	8	----	20 493
1970	558 000	72 000 (72 000)	15 (15)	188 536 (352 482)	24 590	9.2	1.2	20 493
1971	630 000	72 000 (144 000)	13 (28)	221 655 (574 137)	33 019	10.4	1.2	21 313
1972	702 000	72 000 (216 000)	11 (39)	254 909 (829 046)	33 254	11.5	1.1	22 166
1973	774 000	72 000 (288 000)	10 (49)	292 760 (1 121 806)	37 751	12.7	1.2	23 052
1974	846 000	72 000 (360 000)	9.5 (58.5)	333 239 (1 455 045)	40 479	13.9	1.2	23 974
1975	919 000	73 000 (433 000)	8.9 (67.4)	396 488 (1 851 533)	63 249	15.1	1.2	26 257

TABLE XXX

PERSONNEL COSTS, CONVENTIONAL SYSTEM

Year	Components tested	Increase in components (sum)	Percentage increase (sum)	Annual personnel cost (sum), $	Cost increase above previous year, $	Personnel required	Personnel increase above previous year	Average cost per person per year, $
1969	486 000	----	----	327 000	----	19	----	17 210
1970	558 000	72 000 (72 000)	15 (15)	375 188 (702 188)	48 188	21.8	2.8	17 210
1971	630 000	72 000 (144 000)	13 (28)	440 315 (1 142 503)	65 127	24.6	2.8	17 899
1972	702 000	72 000 (216 000)	11 (39)	508 189 (1 650 692)	67 874	27.3	2.7	18 615
1973	774 000	72 000 (288 000)	10 (49)	580 770 (2 231 462)	72 581	30	2.7	19 360
1974	846 000	72 000 (360 000)	9.5 (58.5)	662 134 (2 893 596)	81 364	32.8	2.8	20 187
1975	919 000	73 000 (433 000)	8.9 (67.4)	747 386 (3 640 982)	85 252	35.6	2.8	20 994

TABLE XXXI

PERSONNEL WORK TIME

Year	Conventional system							Specified system						
	Personnel required	Holidays (a)	Vacation (b)	Sick leave (c)	Total (d)	Work days average (e)	Actual work days (f)	Personnel required	Holidays (a)	Vacation (b)	Sick leave (c)	Total (d)	Work days average (e)	Actual work days (f)
1969	19	190	570	228	988	4959	3971	8	80	240	96	416	2088	1672
1972	27.3	273	819	327.6	1419	7125.3	5706.3	11.5	115	345	138	598	3001.5	2403.5
1975	35.6	356	1068	427.2	1851.1	9291.6	7440.4	15.1	151	453	181.2	785.2	3941.1	3155.9

[a]NASA-authorized legal holidays (10) x the number of personnel required.

[b]NASA-authorized vacation days (30) x the number of personnel required.

[c]NASA-authorized sick leave days (12) x the number of personnel required.

[d]Sum of (a), (b), and (c).

[e]Work days available after excluding weekends (365 - 52 x 2) x the number of personnel required.

[f]Actual number of work days (e) - (d).

TABLE XXXII

PERSONNEL PAY RATES

Personnel type	Conventional system							Specified system						
	Pay rate, $			Total pay rate, $				Pay rate, $			Total pay rate, $			
	No.	Hour (a)	O/H (b)	Hour (c)	Day (d)	Week (e)	Annual (f)	No.	Hour (a)	O/H (b)	Hour (c)	Day (d)	Week (e)	Annual (f)
Supervisor	1	9.00	9.00 (18.00)	18.00	144.00	720.00	37 440.00	1	9.00	9.00 (18.00)	18.00	144.00	720.00	37 440.00
Engineers	4	6.71	6.71 (13.42)	53.68	429.44	2 147.20	107 854.40	2	6.71	6.71 (13.42)	26.84	214.72	1 073.60	55 827.00
Technician (lead)	2	4.64	4.64 (9.28)	18.56	148.40	742.40	38 604.80	0	----	----	----	----	----	----
Technician	7	3.61	3.61 (7.22)	50.54	404.32	2 021.60	105 123.20	3	3.61	3.61 (7.22)	21.66	173.28	866.40	45 052.80
Clerk	4	2.97	2.97 (5.94)	11.88	95.04	475.20	24 710.40	1	2.97	2.97 (5.94)	5.94	47.52	237.60	12 355.20
Secretary	1	3.19	3.19 (6.38)	6.38	51.04	255.20	13 270.40	1	3.19	3.19 (6.38)	6.38	51.04	255.20	13 270.40
Total	19	30.12	30.12 (60.24)		1 272.32	6 361.60	327 003.20	8	25.48	25.48 (50.96)	78.82	630.56	3 152.80	163 945.40

[a] Based on accepted labor rates as of 6-16-69.

[b] O/H = overhead costs based on assumed 100% of pay rate/hour (a).

[c] Pay rate (a) + (b) x the number of personnel required.

[d] Pay rate (c) x 8 hrs.

[e] Pay rate (d) x 5 days.

[f] Pay rate (e) x 52 weeks.

TABLE XXXIII

EQUIPMENT MAINTENANCE, SPECIFIED SYSTEM

Item	Cost, $		
	1969	1970 thru 1972	1973 thru 1975
Service contract[a]	22 800	90 000	134 800
Original equipment maintenance[b]	4 450	14 390	16 095
Unique equipment	8 020	51 290	142 650
Total	35 270	155 680	293 545
Cumulative total	35 270	190 950	484 495

[a]Service contract ($1 900.00/month) x 12 months = $22 800.00/year. This increases annually by 3.8%.

[b]The value of original-investment equipment not maintained by contract is $250 000.00. This is 90% obsoleted by the specified system, leaving $25 000.00. The $25 000.00 is multiplied by a 17.8% maintenance factor to give $4 450.00. This then increases annually by 3.8%.

TABLE XXXIV

EQUIPMENT MAINTENANCE, CONVENTIONAL SYSTEM

Item	Cost, $		
	1969	1970 thru 1972	1973 thru 1975
Service contract[a]	38 400	201 750	456 400
Original equipment maintenance[b]	44 500	144 600	162 600
Total	82 900	346 350	619 000
Cumulative total	82 900	429 250	1 048 250

[a]Service contract ($3 200.00/month) x 12 months = $38 400.00/year. This increases annually by 3.8%.

[b]The value of original-investment equipment not maintained by contract is $250 000.00. This is multiplied by a 17.8% maintenance factor to give $44 500.00. This then increases annually by 3.8%.

TABLE XXXV

EQUIPMENT LIST, SPECIFIED SYSTEM

	Cost, $
CPU hardware	
Identical to C1′	167.9×10^3
Station hardware	
Input/output control	10.0
Test station (T8)	110.0
Remote scanner	3.0
Adapters for IC, transistor	3.0
	126.0
Software	
C1′ software	----
Language extension for component test	----

Other items	
Documentation, training, miscellaneous	3.0
System integration	3.0
Development (scanner, adapters)	----
	6.0
Total	299.9

Note: Development costs of approximately $100 000 are not included. It is assumed that the system was developed by ERC contract.

TABLE XXXVI

EQUIPMENT COST

	Year	Cost, $	Cumulative investment, $
Conventional system			
Original investment	1966	272 031	272 031
Present	1969	298 526	570 547
26%/year on $298 526	1972	298 526	869 073
	1975	597 052	1 466 125
Specified system			
Original investment	1966	272 031	272 031
Present	1969	300 000	572 031
26%/year on 15% unique portion of $300 000	1972	46 000	618 031
Additional expansion for test loading	1972	213 000	831 031
	1975	138 969	970 000

TABLE XXXVII

CURRENT AND PROJECTED TEST LOAD

Test item	Components			
	Qualification		Acceptance	
	1969	1975	1969	1975
Resistors	20×10^3	20×10^3	160×10^3	170×10^3
Capacitors	15	18	26	40
Transistors	30	20	35.5	36
Integrated circuits	150	500	15	70
Diodes	15	22	20	23
Total	230	580	256.5	339

APPENDIX A

PROBLEM AREAS

APPENDIX A

TABLE A-I

ITEMS CONSIDERED PRESENT PROBLEM AREAS

Procedure - technical
No meaningful tests for long-duration missions. Need for mathematical models to extrapolate long-duration data from short-term test results.
Need for manufacturer-oriented test procedures.
Inadequate test techniques for thermal resistance.
Acceptance/rejection criteria too arbitrary.
Inadequate analysis of parameter-drift data.
Difficulty of failure analysis on MSI.
Inadequacy of go/no-go testing in environmental extremes.
Need for testing at assembly or subsystem rather than component level.
Inability of present equipment to test LSI.
Weaknesses caused by excessive stress levels during test.
Poor repeatability with changes in test instruments and sockets.
Insufficient long-term test data since the Minuteman program.
Procedure - operational
Failures caused by improper hermeticity testing.
Unnecessary testing producing excessive data.
Component defects
Defects in capacitors.
Defects in diodes.
Contamination in electromechanical devices.
Poor reliability of electromechanical devices, especially for long-duration missions.
Poor reliability because of low-volume production.
Bonding of IC packages.
Poor wire bonds within ICs.

APPENDIX A

TABLE A-I - Concluded

ITEMS CONSIDERED PRESENT PROBLEM AREAS

Workload
Need for screening assistance in identifying unacceptable parts.
Sheer volume of test data to be analyzed.
Lack of qualified test personnel.
General practices/situations
Difficulty in determining the most effective allocation of funds for testing components, assemblies, subsystems, and systems.
Qualification of one lot but use of another.
Wholesale transfer of test specifications from one part to another that appears similar.
Need for better engineer/technician ratio for greater assurance of test integrity.
Communication gap between radiation-effects people and those concerned with other environments.
Customer specification of test tolerances beyond the accuracies of any available equipment.
Too much testing with insufficient consideration of test goals.

APPENDIX A

TABLE A-II

ITEMS CONSIDERED FUTURE PROBLEM AREAS

Procedure - technical
No meaningful tests for long-duration missions.
Acceptance/rejection criteria too arbitrary.
Need for maintaining sterility from the component level through system assembly.
General practices/situations
Difficulty in obtaining industry-wide acceptance of sophisticated test systems.
Difficulty in determining the most effective allocation of funds for testing components, assemblies, subsystems, and systems.
Component defects
Heat bonding and metallization problems with LSI.

APPENDIX A

TABLE A-II

ITEMS CONSIDERED FUTURE PROBLEM AREAS

Procedure - technical
No meaningful tests for long-duration missions.
Acceptance/rejection criteria too arbitrary.
Need for maintaining sterility from the component level through system assembly.
General practices/situations
Difficulty in obtaining industry-wide acceptance of sophisticated test systems.
Difficulty in determining the most effective allocation of funds for testing components, assemblies, subsystems, and systems.
Component defects
Heat bonding and metallization problems with LSI.

APPENDIX B

RELATED STUDIES AND GRANTS

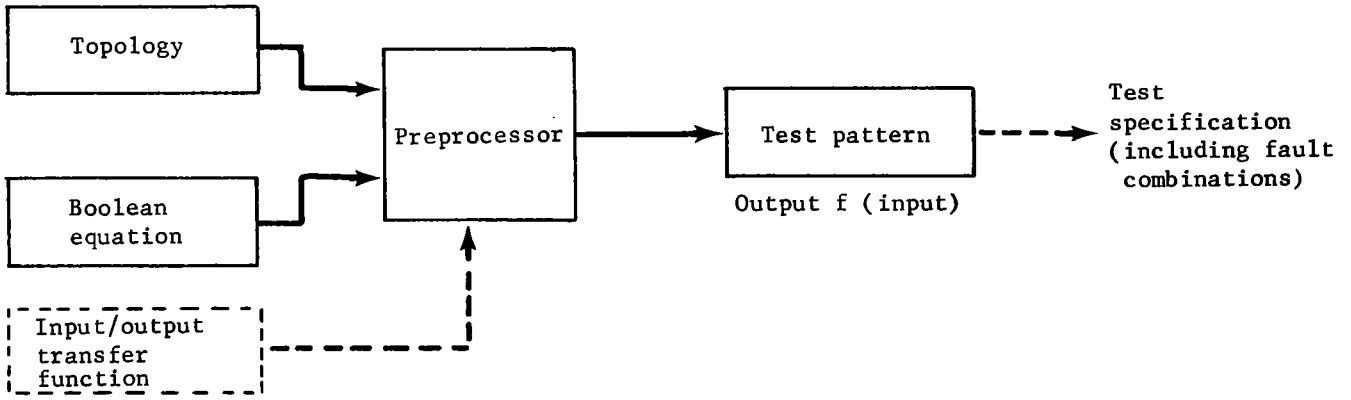
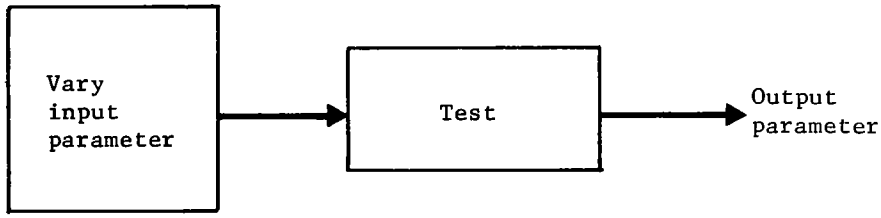


Figure B-1.- Fault Isolation and Diagnosis in Multiterminal Devices (NAS12-689)

APPENDIX B



Nonlinear (logic) devices

Note: Vary level and width of pulse; correlate with output.

Figure B-2.- Impulse Techniques for Testing Digital Microcircuits (NGR-31-001-132)

APPENDIX B

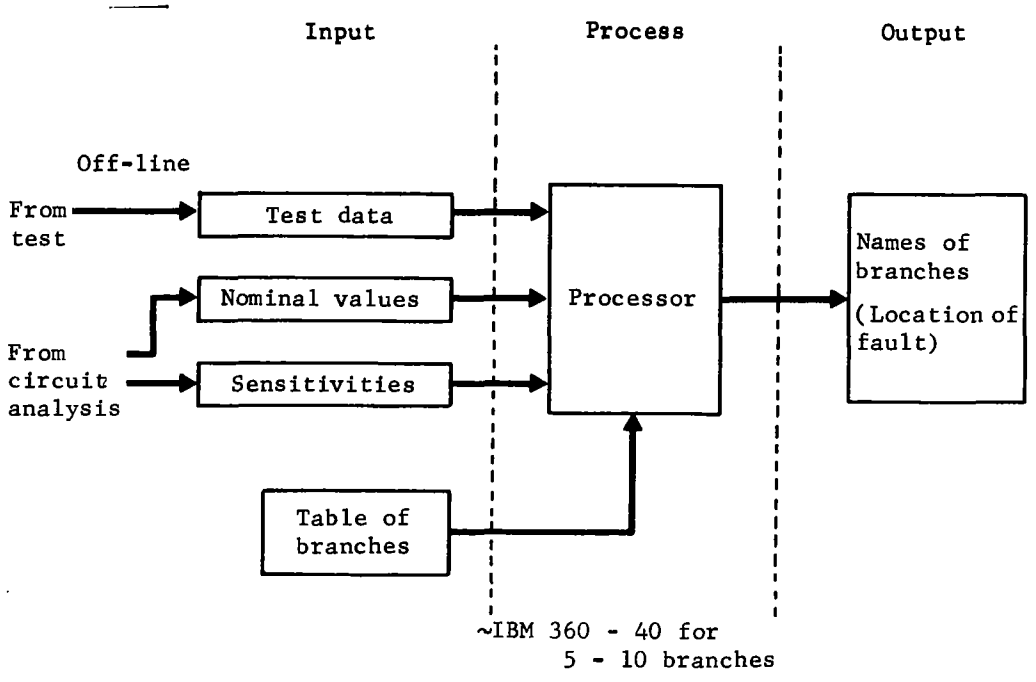
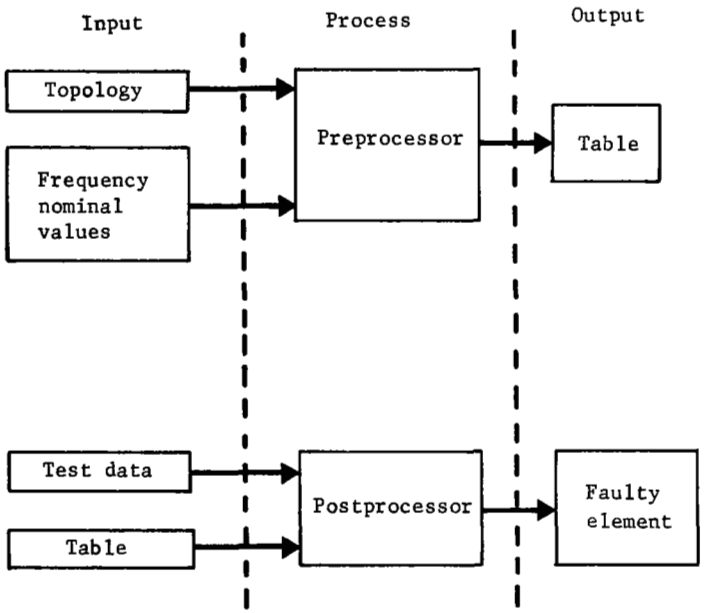
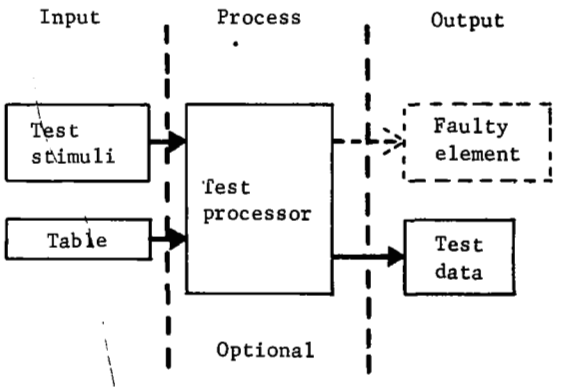
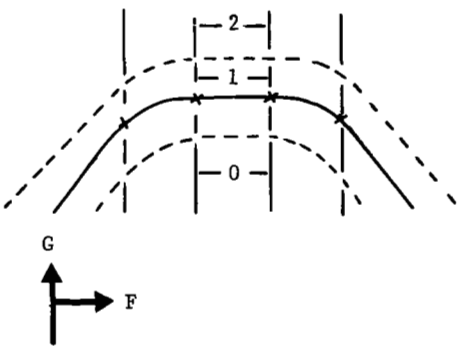


Figure B-3.- Combinatorial Analysis for Checkout Techniques (NGR-31-003-066)



Simplified curve fit of actual Bode plot against predicted Bode plots for each fault





Table

Fault	1	1	1	0	0
Fault	2	0	0	0	0
Fault	3	1	0	1	1

Simplified example

Figure B-4.- Utilization of Topological Techniques for Modeling of Microcircuits (NGR-05-017-017)