DOUGLAS PAPER NO. 4010

X1.17

THE ROLE OF SIMULATION IN THE DEVELOPMENT OF AN AUTOMATIC CHECKOUT SYSTEM

PREPARED BY:
GEORGE F. MEISTER, JR.
ADVANCE APPLICATIONS
SATURN ELECTRONICS

FOR PUBLICATION IN:
LUFTFAHRTTECHNIK RAUMFAHRTTECHNIK
AUGUST, 1966

DOUGLAS MISSILE & SPACE SYSTEMS DIVISION
SPACE SYSTEMS CENTER – HUNTINGTON BEACH, CALIFORNIA

THE ROLE OF SIMULATION IN THE DEVELOPMENT OF AN AUTOMATIC CHECKOUT SYSTEM

INTRODUCTION

Simulation may be defined as the representation of a device, phenomenum, or some combination of these, by another device, phenomenum, or combination thereof, in order to achieve some advantage over using the prime object for the purpose intended. The advantage may be economic, one of time utility, one of ease of observation or measurement, or some combination of these. Some of the uses of simulation are the following:

1. To study the behavior of large masses of people or objects where the cost and reliability of measuring and recording would be prohibitive if the masses themselves were used.
2. To study the life cycle of a device in a few hours where such life cycle might span months, years, or even centuries.
3. To train pilots on the ground at a considerable savings in cost and with less risk to personnel and equipment.
4. To study the performance of a system of devices which has not yet been designed and manufactured, allowing wide manipulation of the device characteristics without having to spend the time and money to change the devices themselves.
5. To solve theoretical problems in a short period of time that are either unsolvable or require intensive calculations when solved by analytic means.

The above list would have to continue for several pages in order to cover all the various uses of simulation. To illustrate the use of this method in one of the most modern fields of technical development this paper describes the application of simulation in the development of an automatic checkout system, and, in particular, the operating procedures used for the testing of a complex space vehicle with that system.

The following discussion is divided into four major parts. These are:

1. The various aspects of the development of an automatic checkout system.
2. The type of missile or stage* simulators used in previous, non-automatic checkout systems and their shortcomings for current automatic systems.
3. The S-IVB Stage Simulator and Systems Integration Laboratory and the role they played in the development of the S-IVB Automatic Checkout System.
4. Software simulators and their present and future role in connection with automatic checkout systems.

DEVELOPMENT OF AUTOMATIC CHECKOUT SYSTEMS

In order to discuss the role of simulation, it is first necessary to describe the various aspects and phases of the development of a checkout system. Using the S-IVB system as a model, Figure 1 shows the breakdown of a total system development. The delineation of hardware and software as separate development items is artificial. The interfaces between the two are many and complex. Their development is so intertwined that it is impossible to separate the two except on paper. However, this breakdown does provide a convenient starting point for the following discussion.

As can be seen in Figure 1, the hardware and software developments each consist of a basic and stage peculiar portion. For further clarification the basic hardware system is shown in Figure 2 and the stage peculiar hardware is shown in Figure 3.

The basic hardware/software system provides a given capability independent of stage configuration, much the same as a general purpose computer provides a certain capability "off the

---

* Stage as used here means one of the major sections of a multi-stage booster vehicle, e.g., the Saturn V/Apollo booster consists of the S-IC Stage, the S-II Stage, the S-IVB Stage and the Instrument Unit.

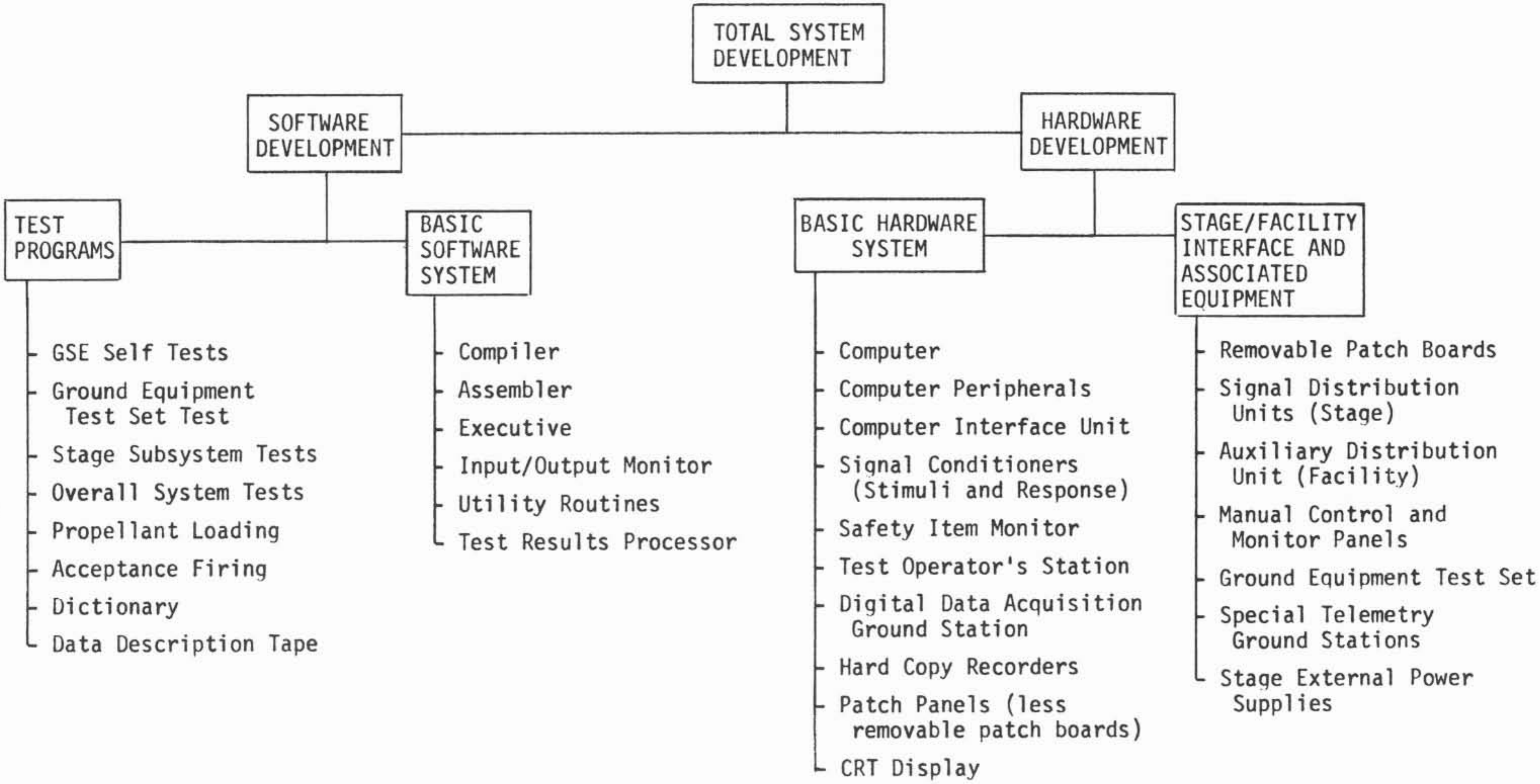
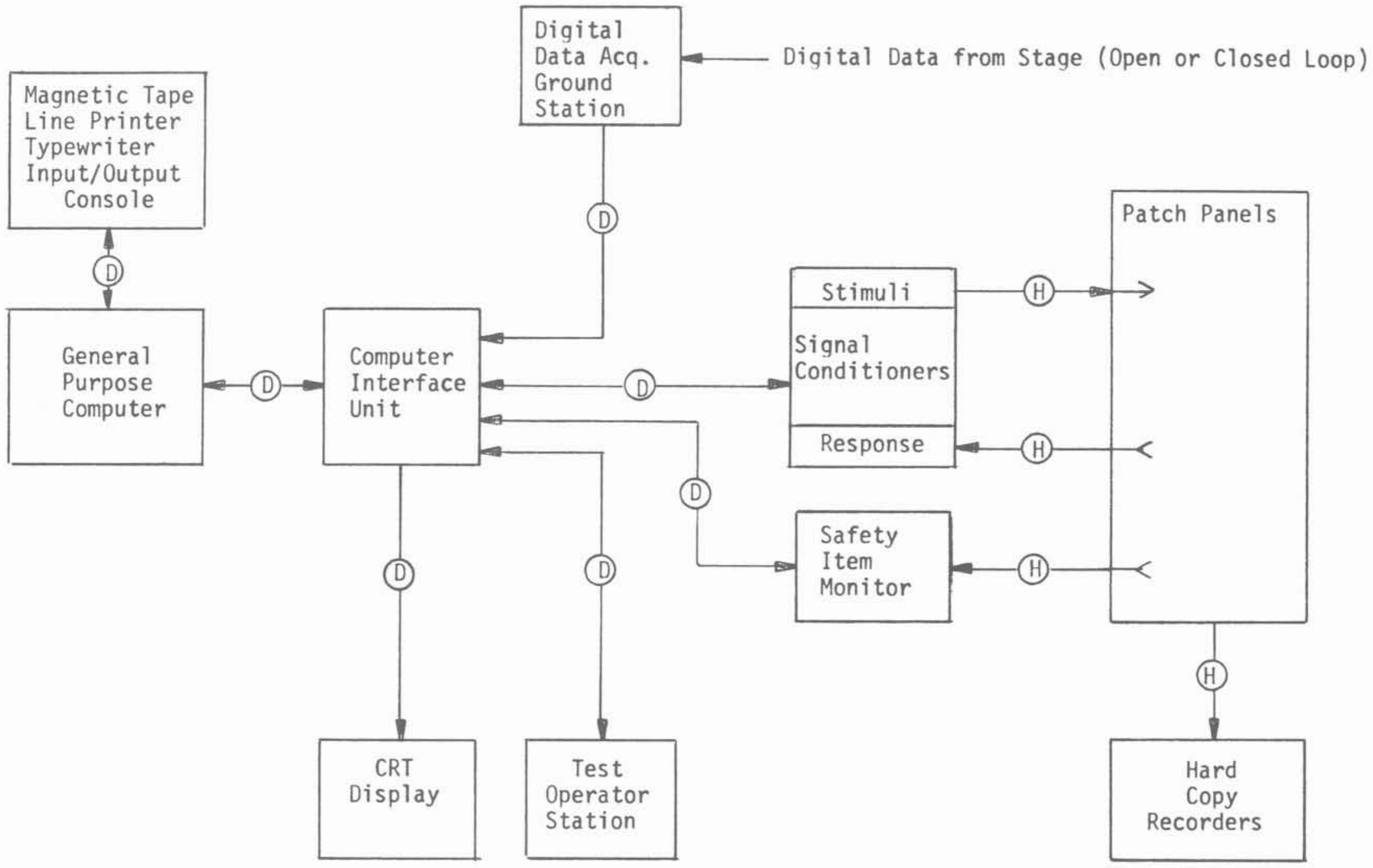

TOTAL SYSTEM DEVELOPMENT
SOFTWARE DEVELOPMENT
HARDWARE DEVELOPMENT
TEST PROGRAMS
BASIC SOFTWARE SYSTEM
BASIC HARDWARE SYSTEM
STAGE/FACILITY INTERFACE AND ASSOCIATED EQUIPMENT
GSE Self Tests
Ground Equipment Test Set Test
Stage Subsystem Tests
Overall System Tests
Propellant Loading
Acceptance Firing
Dictionary
Data Description Tape
Compiler
Assembler
Executive
Input/Output Monitor
Utility Routines
Test Results Processor
Computer
Computer Peripherals
Computer Interface Unit
Signal Conditioners (Stimuli and Response)
Safety Item Monitor
Test Operator's Station
Digital Data Acquisition Ground Station
Hard Copy Recorders
Patch Panels (less removable patch boards)
CRT Display
Removable Patch Boards
Signal Distribution Units (Stage)
Auxiliary Distribution Unit (Facility)
Manual Control and Monitor Panels
Ground Equipment Test Set
Special Telemetry Ground Stations
Stage External Power Supplies

Figure 1

BASIC HARDWARE SYSTEM



Ⓓ Digital Data Transfer

Ⓗ Hardline Connections

Figure 2

STAGE/FACILITY INTERFACE AND ASSOCIATED EQUIPMENT

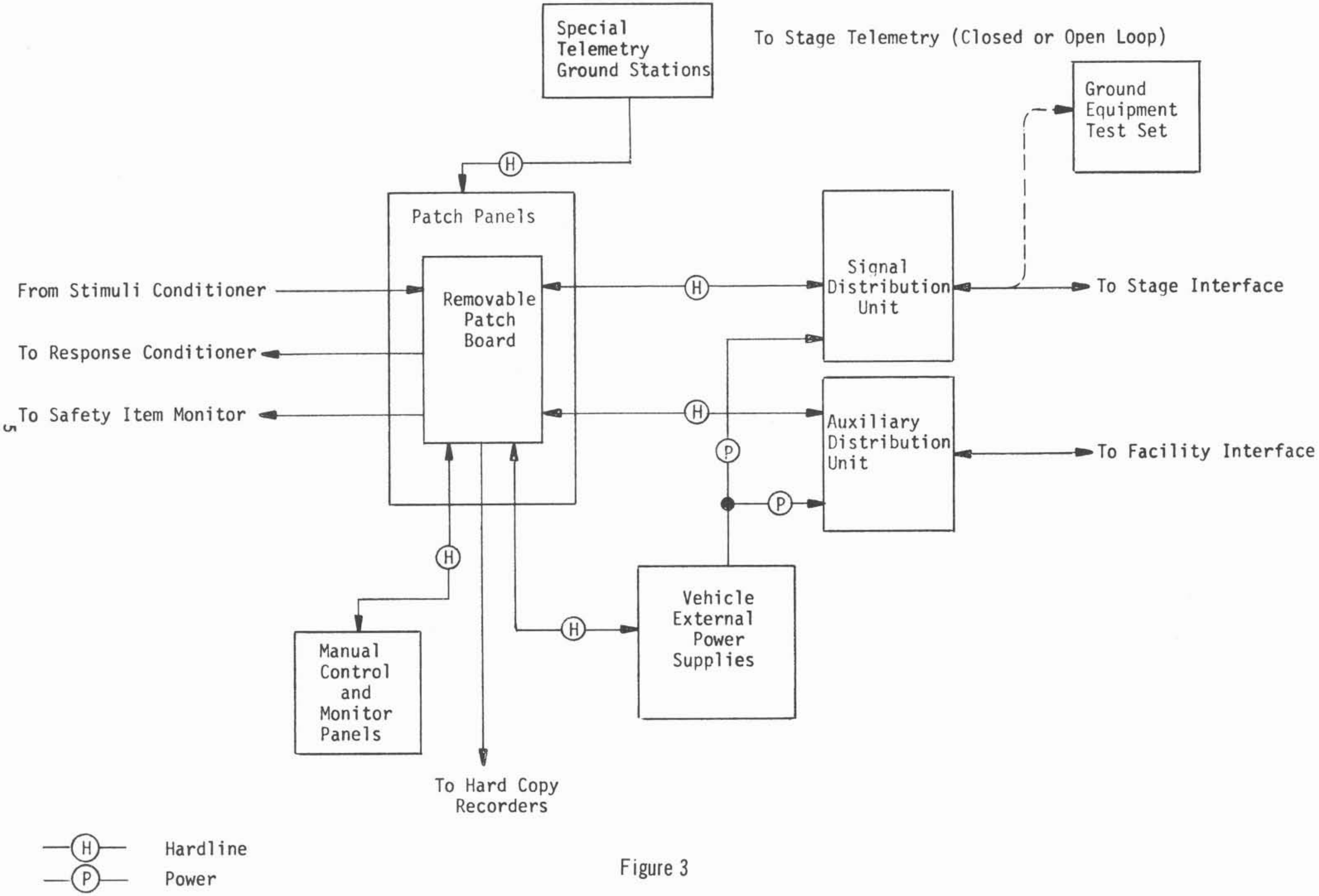


Figure 3

shelf." The development of this capability may take place early with respect to the stage design and fabrication. This development, however, is not complete until the stage peculiar portion has also been developed and it has been determined that the capability provided is in fact sufficient for the application.

Basic System Development

The development of the basic hardware/software system starts with the installation and checkout of the computer and its peripheral equipment. This is generally done by the computer manufacturer with manufacturer supplied diagnostics. These are not necessarily application oriented. At this point certain portions of the basic software may be checked out, particularly that part associated with input/output.

In parallel with this operation, the other pieces of hardware shown in Figure 2 are installed and preliminary checks performed. These checks consist of continuity tests, calibration, and functional testing. The degree of functional testing depends on how much individual self-test capability has been designed into the various end items. In the case of the S-IVB equipment, the capability was provided to operate all logical functions, though not necessarily at system clock speed.

The computer Interface Unit is then connected to the computer. At this time the operational executive begins its development process. Special test programs are written to check out the interface unit using those parts of the executive applicable to that interface. The other end items are then connected to the Computer Interface Unit. Additional special test programs are used to check out these items and those portions of the executive applicable to each end item. These programs do not really involve simulation, but merely drive the end items in all their functional modes to demonstrate the operational readiness of the basic system.

Stage Peculiar Hardware Development

During the development of the basic system, the equipment associated with the stage/facility interface is installed and subjected to preliminary tests. These tests are continuity checks and very limited functional checks. The S-IVB system was designed such that the bulk of functional testing of this equipment is done using the computer and associated digital system. This is accomplished by configurating the removable patch panels shown (and similar patch panels which are incorporated in the distribution units as an integral part of the unit) to "turn the system around on itself." The amount of system self-testing that may be accomplished in this manner is a function of the self-test capability built into the system. In the case of the S-IVB system, the capability was designed into the hardware to allow isolation of faults between the stage and checkout system. This same capability provides thorough system self-testing during the development phase.

At the completion of the "system self-test," the programmable patch boards are exchanged for a set containing those patches necessary to interface with the stage to be tested. The primary differences between these two sets of boards exist in the distribution unit patch boards, with only minor changes required in the main patch panels. At this point, two aspects of system development have not been covered. These are: (1) the verification of the stage/checkout system interface, and, (2) development and verification of stage test programs.

A Ground Equipment Test Set (GETS) is used to verify the stage/checkout system interface. This device provides the capability to monitor signals from and provide signals to the stage umbilicals. This is accomplished with patch panels and a limited amount of switching hardware and electrical loads. No capability is provided to perform stage test programs using the Ground Equipment Test Set. The development of test programs is discussed as part of the description of the S-IVB Simulator and Systems Integration Laboratory.

"CONVENTIONAL" MISSILE/STAGE SIMULATORS

The Ground Equipment Test Set is in many ways similar to missile/stage simulators used on programs not having automatic checkout systems. These simulators were designed such that they would provide responses to inputs in much the same manner as the stage under test. Manual operations were provided for those cases where the output was a result of a non-electrical input, e.g., pneumatic or hydraulic. Manual procedures could be run with this type of simulator. This provided verification of the checkout system and the procedure at the same time. Certain modifications were necessary, of course, to account for the differences between the simulator and the stage, but these were readily handled within the manual procedure. In certain cases time delay circuits were also incorporated within the simulator for certain time critical functions to allow checkout of interlock or sequencing circuitry.

With the advent of automatic checkout systems, this type of procedure verification becomes invalid. This lack of validity is a result of:

1. Lack of Digital Data Acquisition System (Telemetry) Simulation.
2. Lack of facility/vehicle interaction simulation.
3. Lack of timing response characteristics.

Digital Data Acquisition System (DDAS)

An automatic checkout system provides for the integration of the airborne digital data acquisition system (a portion of the telemetry system) into the ground checkout system. In previous space programs the telemetry system has been treated as almost a separate entity from the control system. Checkout of the telemetry system was not a prerequisite for the checkout of any other subsystem. The only vital interface between the two was the requirement for the control system to be available to turn on and off the telemetry. Certain telemetry channels were monitored during the checkout of other subsystems to verify channel integrity, and to provide engineering personnel a "feel" for the operation of the telemetry. However, proper performance of the telemetry system was not a requirement of vehicle checkout per se since all checkout parameters were also brought out hardwire either through the umbilical or over special test cables.

The Saturn IB and V vehicles present an entirely different picture. With the advent of automatic checkout systems and the use of digital computers, it has become feasible to use the data from the telemetry as an integral part of the checkout. This has radically reduced the number of signals (particularly analog signals) which must be hardwired out of the stage. This is particularly true of upper stages. Therefore, the checkout of the digital data acquisition system has become an integral part of the overall checkout, and is a prerequisite for most other tests. This mode of operation requires that any simulator that is to verify checkout procedures must be able to supply this telemetry (digital data acquisition system) link to the checkout system. Further, parameter values transmitted over this link must be responsive to hardline commands from the checkout system.

Facility/Vehicle Interaction

The most difficult tests to develop for the S-IVB Stage were the propulsion subsystem tests and the overall (simulated flight) tests for factory and post acceptance firing checkout, and the propellant loading and static firing programs for acceptance firing. In these tests there is a direct interaction between the facility pneumatic supplies and the vehicle. The control and regulation of pneumatic supplies is an area requiring considerable development. Additionally, the operation of these systems presents a higher degree of danger to equipment and personnel than pure electrical tests. This situation is magnified in the case of the acceptance firing situation with its cryogenic loading and hot engine firing. Therefore, a high degree of development is required in the area of controlling these elements and providing appropriate safety shutdown routines. The typical pre-S-IVB GETS does not provide any simulation of the interaction of the facility and the vehicle except for manual inputs at the test set to simulate certain vehicle responses. Because of the large number of unknowns in this area it is necessary to have the facility and vehicle simulated in such a way that as much development of these programs as is economically justifiable can be carried out prior to actual operations.

Timing Response Characteristics

Finally, in an automatic checkout system the timing of responses to commands takes on additional significance. The computer is set up such that it allows a certain period of time for the response to a given command to appear. For example, if relays are used to simulate vehicle valves, the response characteristics change considerably from the valve operation itself. To a human operator monitoring a light indication, the change in response time to the command is not necessarily meaningful. However, to a computer which is working in small numbers of milliseconds, such a change is not allowable if a valid procedure is to be developed. Experience has shown that one of the major areas of test programming requiring modification as a result of the development process is that of timing between commands and their responses or between commands and other commands.

S-IVB STAGE SIMULATOR AND THE SYSTEMS INTEGRATION LABORATORY

In order to more nearly meet the needs of automatic system development, particularly in the area of test program development, the S-IVB Stage Simulator and the Systems Integration Laboratory (SIL) came into being. The Systems Integration Laboratory consists of the simulator and the prototype set of automatic checkout equipment. The prototype set, including the facility setup and pneumatic consoles, is functionally identical to the equipment installed for factory checkout and post acceptance firing. Early in the program a trade-off was made between that configuration and the configuration of facility and checkout equipment representing the static firing site. It was decided that the factory checkout configuration would be used, primarily for the following reasons:

1. The static firing configuration would have involved considerable extra expense and would still not have allowed total facility representation since cryogenics, water systems, etc., would still have to be functionally simulated.
2. The factory checkout configuration is the first configuration used to test each flight stage. Therefore, it is the most advanced with respect to schedule.
3. The factory checkout site and the Systems Integration Laboratory are both located at Huntington Beach, California, thus allowing maximum crosstalk between the two areas. Hence, the Systems Integration Laboratory could be as receptive as possible to factory checkout problems

The S-IVB Stage Simulator contains both electrical and mechanical components. In most cases these are either a prototypes or one of the early flight articles.

Propulsion Subsystem Simulation

Propulsion subsystems are simulated by special subsystem simulators designed specifically for that purpose. Figure 4 shows some of these simulation devices. The J-2 engine shown is a non-flight engine procured as Government Furnished Equipment after it had already been used as an early ground test article. All engine systems required for dry checkout with the exception of the gimbal devices are functional in this engine.

In the foreground of Figure 4 is one of the stage propellant tank simulators. The tank is capable of being pressurized, vented, etc., in much the same manner as the actual stage tank with considerably less gas being required because of its much smaller volume. The vent plumbing and valving is similar, and in some cases identical, to the flight system. That transducery which is functional during dry checkout is installed. The panels shown in the background of Figure 4 are the simulation panels for the various pneumatic subsystems (cold helium, ambient helium, etc.). Again, actual flight type valving, regulators, and transducery are used. Bottles are provided behind the panel to represent stage storage bottles. The primary difference is that only one bottle is used to represent the multiple bottle configuration of the flight stage.

Figures 5 and 6 show the electrical portions of the simulator representing the aft and forward skirts on which most of the stage electrical equipment is located. The electrical installation is connected to the propulsion simulation equipment in much the same way as on the stage. The black boxes shown are prototype of flight articles. The wiring installation is similar, and in many cases identical, to the flight wire installations. Components are located as they are on the stage taking into account the square shape of the simulator as opposed to the circular form of the vehicle. The

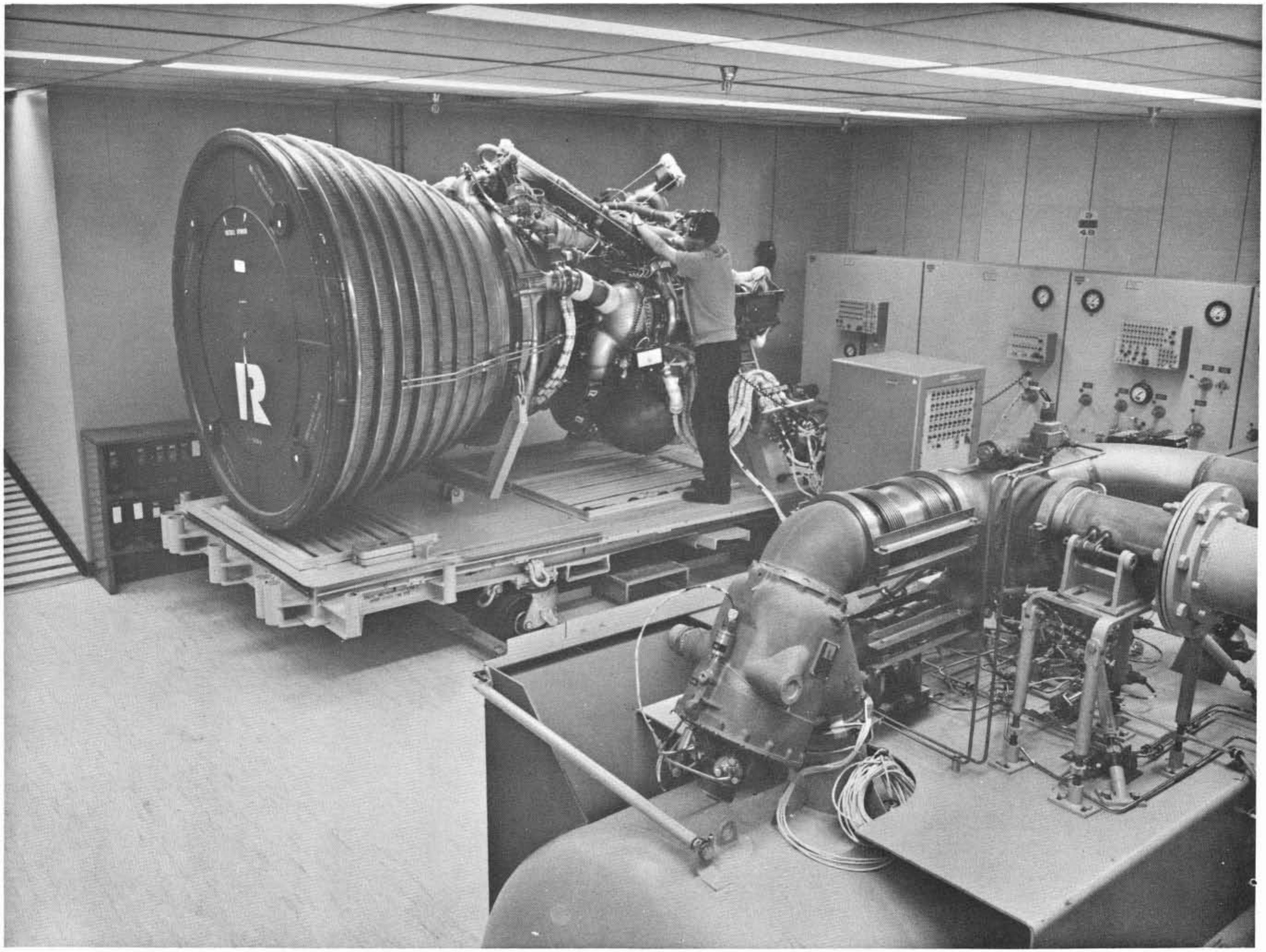


Figure 4.

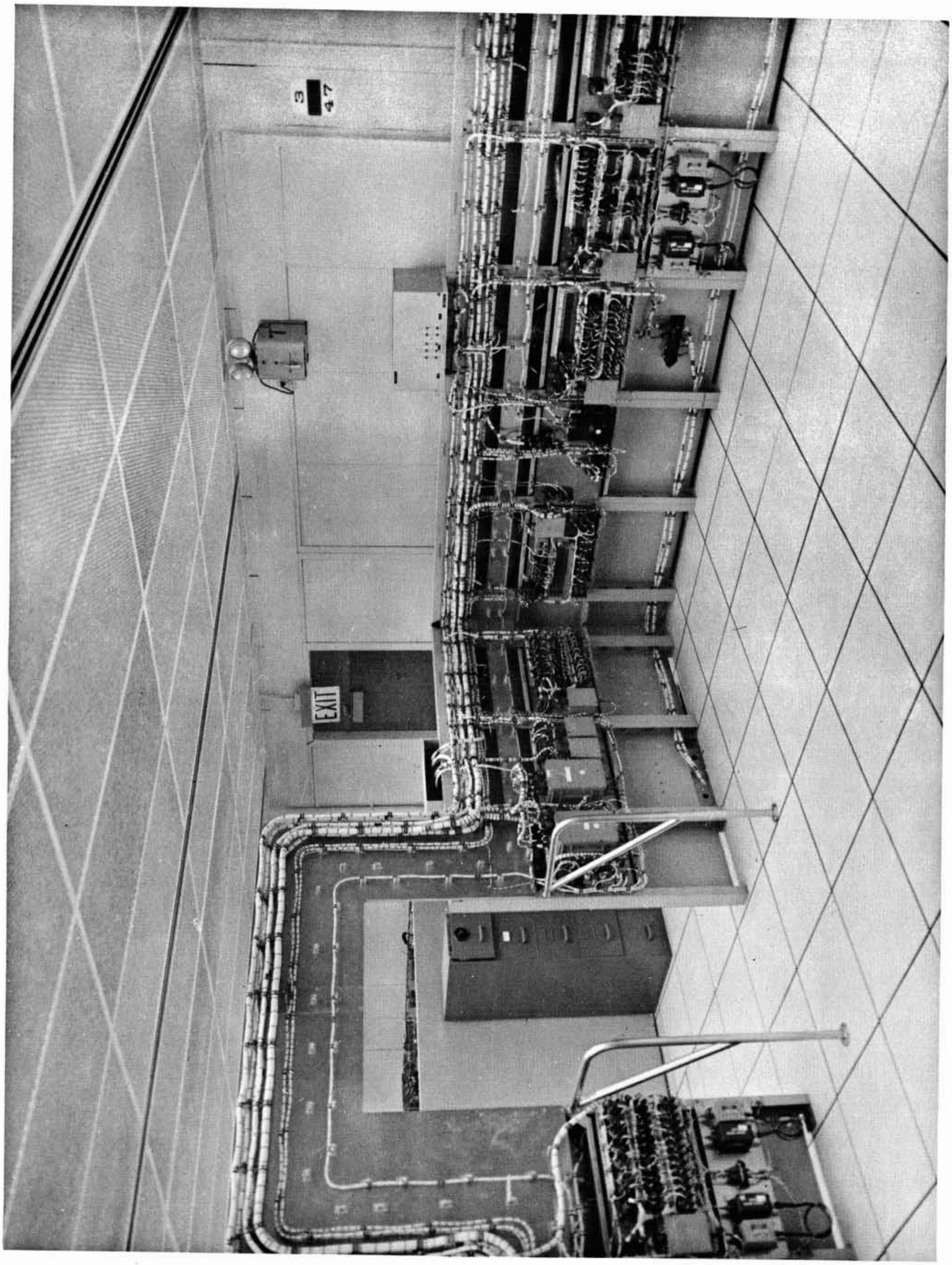


Figure 5.

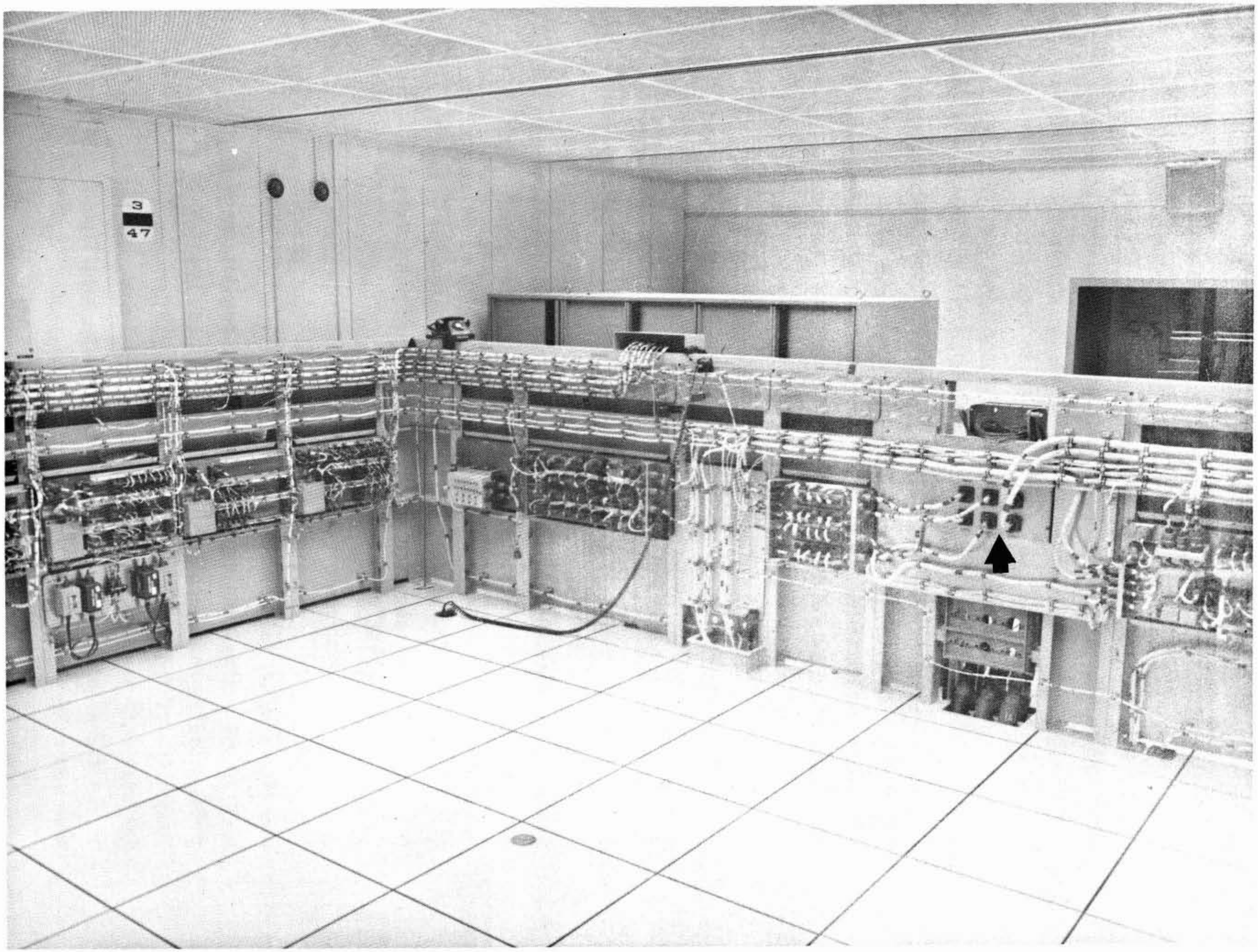


Figure 6.

major difference between the electrical installation in the simulator and the stage is the absence of cold plates or other environmental control devices used on the stage. As can be seen in Figure 6 (see arrow), an umbilical plate is provided on each skirt, which is functionally identical to the stage umbilical to allow connection of the checkout system to the stage in the same manner as in the factory checkout configuration. A means for disconnecting the umbilicals during simulated flight is also provided. The two skirt simulators are connected by harnesses under the floor representing the stage tunnel. A forward and aft interstage connection provides the interface to the upper and lower stage substitutes which are part of the checkout system.

Simulator Characteristics

The inclusion of transducery, multiplexers, signal conditioning, and associated telemetry equipment allows the checkout of the telemetry system and the simulation of the interaction of the telemetry (digital data acquisition system) with the control system in the stage. That data which comes out of the stage over the DDAS link for checkout is provided in the same form by the simulator. Thus, the full integration of the DDAS and control system for checkout is accomplished in the Systems Integration Laboratory as it is on the flight stages.

The design of the propulsion modules are such that pneumatic operations using factory checkout type facilities can be carried out. The techniques for control and regulation of these supplies have been developed with this equipment. Though the control of cryogenics and the operations required for static firing have not been developed in this area, many of the techniques evolved for pneumatic system operation have led directly to that development at the static test site.

The use of valves, transducers, pressure switches, motor driven switches, etc., has allowed the development of timing requirements prior to testing of the flight stage. The true timing response characteristics provided by the simulator have materially aided in providing working procedures to the operational areas.

The simulator described is considerably more costly than the GETS-type of missile simulator. Additionally, in order to maintain its usefulness it requires updating at periodic intervals to maintain functional identity with successive configurations of flight stages. Several attempts have been made to evaluate the cost of the Systems Integration Laboratory (SIL) versus the advantages provided by such an operation. In order to aid in this evaluation an outside consultant was brought in to look at the SIL operation. The following are excerpts from that report resulting from that effort:

"In other words, SIL is a test bed in which paper designs of programs and hardware can be tested before checkout in the VCL." (Author's note: VCL is the Vehicle Checkout Laboratory referred to in this discussion as factory checkout.) "It is a design tool for faster delivery and for earlier correction of errors in software and hardware. Its value, then, should be judged in how well it reduces the elapsed time of a stage in the VCL, when every minute may be critical.

"To determine meaningfully whether SIL pays for itself, we must analyze its contribution and its cost to the Saturn Program. The main contributing factors are:

1. Reduction of cost in VCL
    a. Manpower and equipment usage
    b. Premature wear-out of deliverable parts

2. Faster overall delivery
3. Others
    a. Operator training
    b. Programming research
    c. Support activity

The cost factors are the cost of SIL as reflected in its monthly equipment and manpower charges, and the cost of possible errors in VCL introduced by temporary fixes in SIL that have not been removed . . . . no errors uncovered in VCL may be traced to temporary fixes in SIL.

"In support of this study, the errors found by SIL from 4 January to 7 May 1965 were tabulated by error type, and potential savings to VCL were estimated. Error types that saved a delay in VCL are vehicle and GSE hardware errors, test requirement errors, and programming errors. Other errors were peculiar to the SIL operation and saved no time in VCL. Table I shows the time spent in SIL on each of these errors for the 2001 vehicle." (Author's note: The 2001 vehicle was the first flight S-IVB stage.)

"The pessimistic estimate in Table I assumes that each delaying error from SIL is discovered in VCL only after the previous delay had been fixed . . . . The optimistic estimate assumes that the right expert is in VCL at all times and can fix the program using a minimum of red tape . . . . Between 1000 and 1300 hours saved for the problems discovered on the 2001 vehicle may be a realistic value for the first configuration.

"Table I Actual SIL Time and Estimated VCL Delay Savings for the 2001 Vehicle

Problem Corrected	SIL Time Hrs	Est. Savings in VCL Delays Pessimistic Hrs	Est. Savings in VCL Delays Optimistic Hrs
Vehicle	6.5	200	100
GSE	27.5	976	150
Test Requirements	29.2	446	75
Procedure Program	62.9	1550	173
SIL	74.7	--	--
Total	200.8	3172	500"[1]

Another indication of the contribution of the SIL and the Simulator to the S-IVB checkout system development can be found in Space/Aeronautics magazine on the subject of the Apollo program.

> " . . . within limits, each of the major Apollo contractors was allowed to develop his own checkout approach, and it is generally agreed that Douglas chose the one involving the highest degree of automation. Yet checking out the first S-IVB, Douglas reports, proved 'an order of magnitude' easier than expected. In fact, the company was surprised to find that its equipment took the automation more readily than did its engineers.
>
> NASA officials, too consider the S-4B's GSE problem essentially solved."[2]

There is little question as to the value of the SIL and the associated stage simulator in the development of the automatic checkout system and the associated test programs. There is a general feeling among those personnel connected with the development of the S-IVB GSE that the development would have been "an order of magnitude" more difficult, had

---

[1] Excerpted from IM 217-1, Effectiveness of the Systems Integration Laboratory (A Critical Review), September 15, 1965 by G. L. Hollander, Hollander Associates, Fullerton, California.

[2] "Aerospace in Perspective, A Special Issue," Space/Aeronautics, January, 1966, p. 72.

the SIL and simulator not been available. The question remaining is one of determining that point in the program where such a development tool is no longer necessary. This would eliminate the need for further modification of the simulator to keep up with flight stage configurations. This is now being studied by both Douglas and NASA.

SOFTWARE SIMULATORS

The simulator described in the preceding section has considerable cost, not the least of which is the engineering and manufacturing time required to update it to successive stage configurations. The obvious question of whether the same capability could not be provided at less cost and with greater configuration flexibility through the generation of a software simulator is a legitimate one and warrants examination.

A software simulator, as the name implies, is one or more computer programs. It is designed to run on the test processing computer, another functionally identical computer, or a second computer connected to either of the first two. Two general types of software simulation may be considered:

1. Driver/Monitor Simulation Programs
2. Functional Simulation Programs

Driver/Monitor Simulation Programs

A driver/monitor simulation program is a program that is resident in the processing computer (or functionally equivalent machine). It acts as a supervisory executive over the normal executive routine causing various activities to take place in "stop" time. That is, the operational executive is allowed to process the test program for a short period, usually some small number of milliseconds, and then time is stopped by the simulation executive and various parts of the program are manipulated or checked.

The simulation executive is used in conjunction with a simulation input program. This program may be written in the same language as the test program (a test oriented language as opposed to "machine" language). There is a statement-by-statement correlation between the test program and this simulation input program. For example, if the test program is required to measure a parameter at a certain point in the program, the simulation input program must provide a value for that parameter at that point. Values provided may be either in or out of tolerance depending upon whether the simulation is driving the program down a main path or out through a fault isolation path.

In its most sophisticated form this type of simulation can drive the program down all possible paths. At the same time checks may be built into the simulation executive to allow it to automatically monitor much of the test process. In its simpler form, the simulation executive merely provides the capability of monitoring the processing of the test program by recording certain parameters within the machine. They may be retrieved and printed out later for programmer evaluation.

A rather simple version of this type of simulator was written for the S-IVB system. In conjunction with simulation input programs written in the test oriented language, it had the capability of driving the program down all paths and recording various information for evaluation by the test programmer. Several shortcomings were discovered in this approach which made its use extremely limited. The two most pertinent were:

1. The simulation input routine was written from the same requirements document as the test program. This meant that the simulation was only as good as the requirements document. This did not provide

any check of the real capability of checking out the stage itself. Any errors in the requirements document were carried through the simulation process.

2. The time required to write the simulation executive was relatively small (less than one-half man year). However, the time to write the simulation input program for each test program was about half the time required to write the test program itself. This meant that the time required to generate a test program with simulation was 1.5 times that required if simulation were not used. Due to schedule restrictions it was not possible to allow this additional time in the programming schedule.

As indicated, this type of simulator is relatively simple to write and use. However, its use should probably be limited to those situations where no other form of program verification is available or where program verification is not critical to the overall project success.

FUNCTIONAL SOFTWARE SIMULATION

Experience with the driver/monitor simulation approach and its shortcomings led to consideration of a more sophisticated type of simulation. The simulation considered would provide a comparable degree of verification as the hardware simulator and eliminate the additional burden on the test language programmers.

The simulator would be written to run either on the test processing computer, a functionally identical computer, or a computer connected to one of the first two. Its prime characteristic is that it would be written from the unit under test design information independently of the test requirements or test programs. If possible it would be written to run in real time without the necessity of stopping time to keep up with bookkeeping, etc. To the test operator there would be little difference between running against this type of simulator and running against a hardware simulator (hence the name "functional simulator).

This is accomplished by writing the simulation program such that it actually emulate the systems and subsystems within the stage. In its most complete form it would provide for all the interactions between the various subsystems and would be capable of responding to incorrect inputs in the same way the stage would. To do this the representation of the subsystems, or parts thereof, must do more than merely provide an overall transfer function for expected sequences of inputs. If a bus voltage is low (either by design or by test program error) the subsystem should react to this low voltage and provide appropriate deviations in its output.

A functional simulation program would be a very powerful tool in the development of an automatic checkout system and the verification of test programs for such a system. It could be reproduced and used in many areas. It would have far more flexibility than a hardware simulator, and cost less to maintain once it was established. However, it is an extremely difficult programming effort. Many people in the aerospace field are presently looking into this form of simulation; but, to the author's knowledge, no serious attempt has been made as yet to write such a program for something of the size of a stage or vehicle.

It should be noted that the functional simulator as described has many of the characteristics of simulation programs used in other disciplines, principally operations analysis. However, the degree of complexity and the number of interactions involved are probably an order of magnitude higher than that presently handled in those fields. This does not say that it can or should not be used in the aerospace field, but only that it is a difficult achievement at this point in time.

SUMMARY AND CONCLUSIONS

Experience has indicated that the use of simulation in the development of automatic systems and their associated operating procedures provides an economic advantage over the cost of operations in a formal checkout area using a flight stage; an advantage of time utility since the development can take place prior to the completion of manufacturing on the stage; and the advantage of ease of measurement and observation since special capabilities may be built into the simulation equipment for that purpose.

To date, the simulators used for this purpose have been primarily hardware oriented with the degree of similarity to prime hardware being highly variable depending upon the application. It appears to be clear that the degree of automation used in the process and the degree of similarity required between prime equipment and simulator equipment are directly proportional to each other. This implies that as automatic systems become more sophisticated, so must the simulation used. It is not at all clear that hardware simulators can maintain a useful advantage under those conditions.

In other areas functional software simulation has been used as a tool of operations analysis for studying product flows, service facility distribution, production line organization and other related process operations. To date, the number of parameters involved and the complexity of the prime systems has been low compared to space systems. As these operations become more complex (for example, the operation of a nuclear power plant), they will take on many of the characteristics of space systems. Present simulation programming techniques will have difficulty coping with this increased complexity.

What then is the future of simulation of complex systems, whatever their application? The cost of prime systems, the degree of difficulty in the development of automatic techniques, and in some cases (such as space environments), the inability to get to the prime operation or system until a point in time when the development must already be accomplished, almost certainly means that simulation is a necessity in the development of future systems.

In order to fulfill its role to the greatest advantage, simulation as used in the future must combine the techniques of all the present forms. It is probable that the simulator of the future will resemble a software functional simulator. However, it must be done such that it has the validity of the hardware simulator, the flexibility of present software simulators used with simpler systems, and the capability of handling complex functional relationships. Whether this simulator uses a digital computer, an analog computer, special hardware, or some combination of these, will depend on the application and the degree of development in each of those areas. The primary ingredient required, however, for its development is the firm belief that such simulation providing all the advantages inherent in its use can, and must, be accomplished.