X 1 17

IBM Report No. 66-894-0005

DEVELOPMENT AND UTILIZATION OF COMPUTER TEST PROGRAMS FOR CHECKOUT OF SPACE VEHICLES

INTERNATIONAL BUSINESS MACHINES CORPORATION

Federal Systems Division
Space Systems Center
Huntsville, Alabama

DEVELOPMENT AND UTILIZATION OF COMPUTER TEST PROGRAMS FOR CHECKOUT OF SPACE VEHICLES

by

Edward A. Robin, Manager
Vehicle Test Programming Department

PRESENTED

AIAA Conference
XVIIth International Astronautical Congress
Madrid, Spain
October 10-15, 1966

INTERNATIONAL BUSINESS MACHINES CORPORATION
Federal Systems Division
Space Systems Center
Huntsville, Alabama

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

DEVELOPMENT AND UTILIZATION OF COMPUTER TEST PROGRAMS FOR CHECKOUT OF SPACE VEHICLES

by

Edward A. Robin, Manager
Vehicle Test Programming Department

INTERNATIONAL BUSINESS MACHINES CORPORATION
Federal Systems Division
Space Systems Center
Huntsville, Alabama

OBJECTIVE

A computer system was designed to allow test engineers to progressively employ automation in the checkout of the Uprated Saturn I and Saturn V space vehicle programs and still allow manual control of the checkout process. A two-computer system was selected by National Aeronautics and Space Administration, and the International Business Machines Corporation was chosen to provide the programming engineering necessary to implement these objectives. Space vehicle checkout, prior to launch, may be characterized by controlling, monitoring, and testing the vehicle and its subsystems through the use of ground support equipment (GSE).

METHOD OF APPROACH

The use of computer programs to execute tests has several advantages over a purely hardware automated system (a system without a general purpose digital computer):

> Programming is easier to modify (thus more versatile) than hardware is to redesign, and programming can easily accommodate more complex logical processes than hardware.

Two general purpose digital computers connected by a data link transmission system were selected to equip the launch site GSE for Uprated Saturn I. An additional display computer system was added to create the Saturn V checkout system. [1]

---

1. F. W. Blum, IEEE Regional Convention, Atlanta, Georgia, April 11, 1966.

To implement this philosophy using these Uprated Saturn I and Saturn V hardware configurations, IBM designed and developed a General Purpose Operating System program.[2] The Uprated Saturn I program was written in two parts, one to run in each computer and designed to run asynchronously in each computer.

The Uprated Saturn I Launch Computer Complex is composed of two computers: one in the Blockhouse, Blockhouse Computer System, connected to another computer in the Automatic Ground Control System Complex (AGCSC). (See Figure 1.) These computers are connected by a data link. Each computer can receive vehicle measurements via a digital data acquisition system (DDAS). The blockhouse electrical support equipment is connected to the Blockhouse Computer via discrete-in and discrete-out relays. Vehicle stimuli and response sensing are executed via discrete-in and discrete-out relays in the AGCSC.

GSE INCLUDING COMPUTER HARDWARE

The computers used for checkout are similar to those used for scientific computation with the exception of some special input/output equipment that is used to communicate with the test engineers' display consoles and the vehicle.

Each of the computers has 32,000 24-bit words of high-speed memory storage and a magnetic drum having a 32,000-word capacity. The other familiar input/output equipment consists of five magnetic tape stations, a card reader, a card punch, and a line printer. Six remote terminal cathode-ray tube display consoles with alpha/numeric keyboards furnish the main method of communication between the test engineer and the computer. The test engineer types the command at the console to have his test performed, and is informed of the results on the cathode-ray tube.

The vehicle and ground measurement equipment consists of a DDAS, which contains a high-speed memory that is continually updated with the latest vehicle analog measurements. The programs in the checkout computers must request information from the DDAS by measurement address whenever the values are needed. This information is then transferred from the DDAS memory into the checkout computer memory. The maximum sampling rate in the DDAS equipment for a vehicle measurement is approximately 8.3 milliseconds. Discrete signals may also be received over the DDAS.

Most of the discretes are received over the discrete input/output data channel. They are received from the test console switches by the engineer requesting a function on the vehicle or from the vehicle to signal the status of equipment. Discrete signals are also issued from the computer to the engineers' status lights and to the vehicle functions. The discrete signal equipment has the capability of issuing discrete signals

---

2. E. A. Robin and J. E. Prescott, "A General Purpose Computer System Using a General Purpose Computer System Program for a Special Purpose Task — Space Vehicle Checkout," Instrument Society of America Proceedings, New York, October 1966.

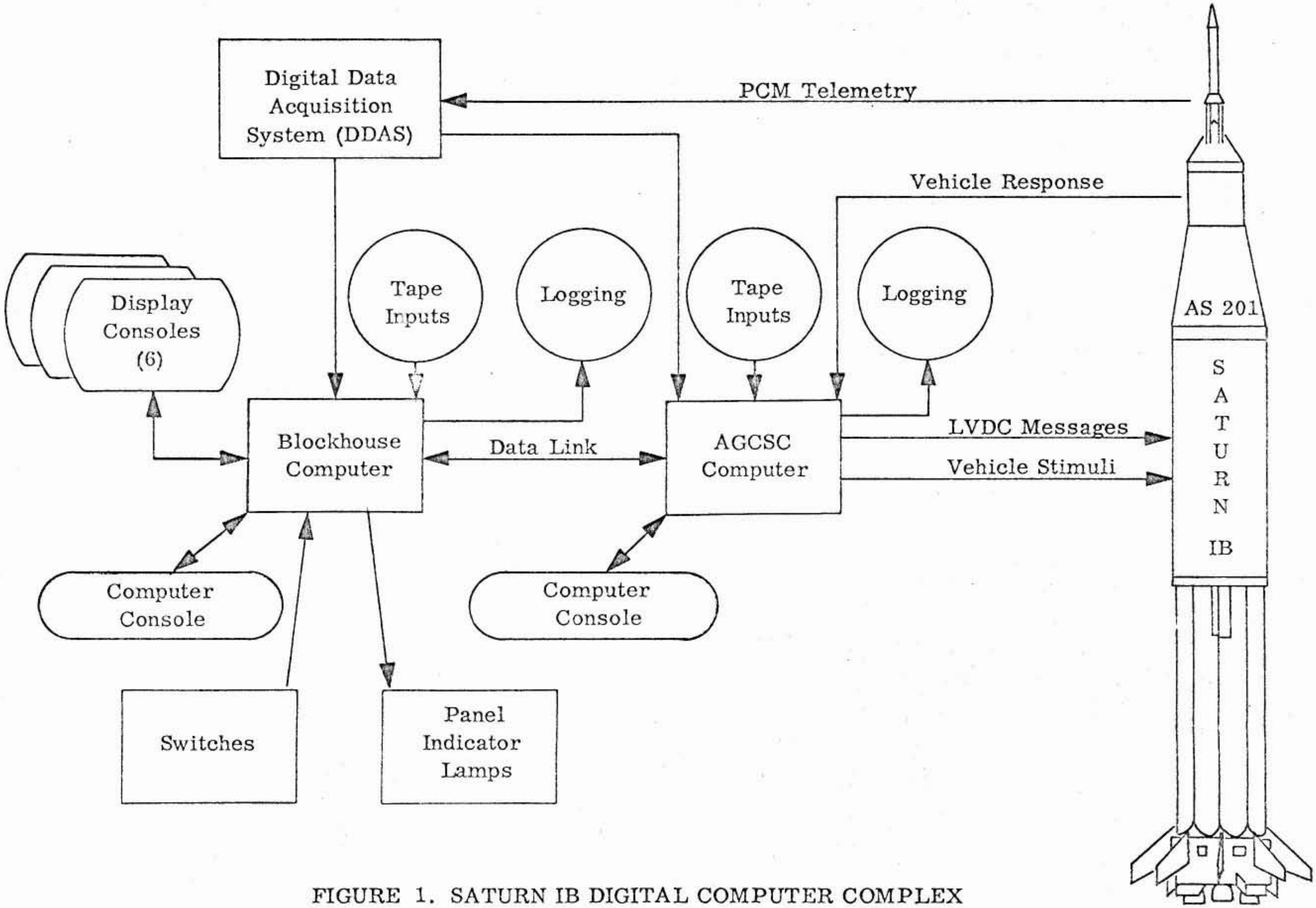

FIGURE 1. SATURN IB DIGITAL COMPUTER COMPLEX

or receiving discrete indications. This equipment uses the high-speed memory of the checkout computers as its storage for posting the discrete-received status. A bit within a computer word depicts the status of each discrete. The discrete signal is either on (24 volt) or off (zero volt). Once the discrete signal scanning is initiated by programming, the bit positions corresponding to the discretes are continually updated until the scanning is stopped by programming. One scan cycle of the incoming discretes (to the computer from the launch vehicle, test engineer control console, or other GSE) requires a minimum of 2 milliseconds.

The discrete equipment posts to the computer memory the occurrence time and the discrete number (coded) for each discrete received. The programs must then react to the discrete occurrence: the normal reaction is to issue a discrete. In this way the computer system resembles a giant junction box. All discrete information is recorded on magnetic tape that can be used for post-test analysis. The status of the outgoing discretes is maintained by programs. Outgoing discretes are those issued from the computer (by programming) to the launch vehicle, test engineer control console, or other GSE. The discrete equipment records the time of issuance and number of each discrete issued. These data can be used for post-test analysis.

Another section of the computer system is the data link connecting the two computers. This cable is full-duplex for control word transmission and half-duplex for data transmission. The transmission rate for data is 180 microseconds per 24-bit word. All information from one computer to the other travels over the data link by program actions.

Two features of the computer that facilitate the implementation of the system are the priority interrupt feature and the programmable interval timers. An example of an interrupting unit is the programmable interval timer. The program stores a binary number into the timer; and, when the timer counts down to zero, the hardware transfers control to a specific memory location. Each computer has three timers set to interrupt at different intervals to accomplish periodic tasks. Discrete signals entering the computer cause interrupts, and input/output equipment terminating transmission causes interrupts. The test engineer requesting a keyboard entry also causes an interrupt that must be serviced by programs.

COMPUTER SYSTEM PROGRAMMING

The three tasks of controlling, monitoring, and testing the vehicle can be accomplished through these general purpose computers and their associated input/output devices. The General Purpose Operating System program requirements were similar: it must be able to control, monitor, and test. This Operating System is similar to an operating system used for a business or scientific application in that it schedules and controls the input/output devices, schedules and controls the central processing unit, schedules and controls memory storage, maintains event trails, and reports computer system status to the computer operators.

Controlling via Switches and Keyboard Inputs

Information is received from the display console keyboard and electrical support equipment switches, as well as standard punch card and magnetic tape inputs. Under Operating System control, these inputs must generate discrete output signals to the electrical support equipment and the launch vehicle. When the engineer presses a button on his console, a discrete is generated into the Blockhouse Computer. This action causes an interrupt in the program currently being executed, and the hardware transfers to the Operating System program to process the discrete action. The program must examine the entry posted by the discrete input/output data channel in the high-speed memory table to determine which of the discretes changed state. Once the discrete number and the new state are determined, the program must decide the response to the discrete. This is accomplished by a table-look-up operation (normally, a discrete must be issued to the vehicle); the program sends a message to its counterpart in the AGCSC Computer to issue the discrete to the vehicle. A discrete received from the vehicle causes the process to act in reverse. This action normally results in a discrete being issued from the Launch Control Computer to the status lamps of the test engineers' consoles.

Monitoring via the Computer

In its monitoring role, the Operating System program must monitor vehicle system and subsystem reaction to these signals. The requirement was to monitor both discrete responses and analog measurements.

The test engineer indicates to the Operating System the functions to be observed, the periods they are to be observed, the states (discrete) or limits (analog) to be noticed, and the action to be taken. To accomplish these tasks, the engineer communicates with the program within the Operating System by using his display control console keyboard.

An example of a monitoring function would be to read a specified DDAS measurement each fifteen seconds; and, if it reads less than two volts or greater than four volts, to display a message to the test engineer and issue specified discretes to place the vehicle in a safe state.

Testing via Subsystem Test Programs

The third requirement was to test the vehicle and its associated subsystems. This function is provided by the execution of test programs that run under executive control of the Operating System program. The relationship of the test program to the Operating System is similar to that of a mathematical matrix inversion program to a General Purpose FORTRAN Operating System.

The test engineer requests execution of his test program by using the display control console keyboard, and is informed of the status of the program by displayed messages from either the Operating System or the test program. The Operating System stacks the request, schedules the request, has the requested program loaded into high-speed memory from magnetic tape, gives computer control to the requested program, and monitors execution of the program.

An example of a portion of a test program would be to check initial conditions of a pump (discrete or analog), issue discretes to the vehicle, delay, look for expected response in a pressure measurement (discrete or analog), and analyze this DDAS measurement for trend.

Programming Interaction (Multiprogramming)

The priority interrupt feature of the computers allows the computer to perform many functions virtually at the same time. All actions performed by the system are triggered by interrupts. Some of these actions are handled immediately. Other actions must be remembered for later processing. The Operating System program stacks these delayed actions and assures that they are serviced by the responsible programs when time is available.

For instance, a test program may be executing vehicle testing; and, before completing the testing, may be interrupted by a message receipt from the other computer. Before this message is acknowledged on the interrupt, the program may be interrupted by the receipt of a switch action from a test engineer's control console. Before this switch action is completely serviced, another message may be received from another device; the Operating System would transfer control to service this request. In most cases, the time away from the test program would not have hampered its proper execution.

TEST PROGRAM DEVELOPMENT

Test programs are being developed and are operating to test such subsystems as the airborne computer, gyros, flight control, and the various network tests that must be performed in the vehicle checkout process. Programming techniques have been developed to allow the test programmer to easily direct the Operating System program in its servicing of input and output signals.

Test programs are designed to be stored on magnetic tape just as a data processing operating system program stores its tasks to be performed on tape. The major difference is that the Uprated Saturn I Operating System program has the ability to call and execute these programs as directed by test engineers. The data processing operating system program usually services its jobs (or tasks) in the sequence in which they are placed on the tape; whereas the Uprated Saturn I Operating System program searches out the test program that the test engineer requests and executes this test program as the checkout tests are needed. This versatility is part of the Operating System program.

This method of program organization allowed IBM to begin to develop a set of integrated test programs for the various vehicle subsystems. For example, a networks programming group was formed to work with NASA and stage engineers to develop automation plans. The group also designs and writes programs for automation of vehicle power-on and power transfer. Similarly, programming groups were formed for telemetry subsystems, propellant subsystems, electronic onboard subsystems, gyro subsystems, and flight control subsystem checkout.

The constant programmer-engineer interface needed for automation definition, in addition to the complexity of the checkout task, prompted IBM to select engineers for these programming assignments. Thus a new breed of programmer began to develop, who could be appropriately entitled an "Engigrammer." These professionals are as at home with stage wiring prints, onboard electromechanical devices, and electronic principles as they are with program listings and a multicomputer installation.

The Engigrammer brings the backgrounds of the engineer (chemical, electrical, and mechanical), the programmer in real-time programming, and the system analyst in understanding of multiprocessing (more than one computer) and multiprogramming (more than one program) systems to space vehicle automation. (See Figure 2.)

In the blockhouse the Engigrammer is invaluable. He understands the problems of the onboard hardware as well as the problem of the multicomputer program and Operating System. He understands and provides the man-machine interface between the stage contractors and NASA test engineers and the computer checkout system. New methods of troubleshooting and troublerecording were developed and implemented. These methods enable the blockhouse team to meet the NASA objectives of progressive automation.

In major overall vehicle test (Plugs Out), the data shown in Figure 3 were recorded. The Operating System serviced approximately 5,800 switches and issued about 5,190 stimuli to the vehicle. The vehicle responded with 178,711 responses. In addition, about 6,100 Launch Vehicle Digital Computer messages were communicated between the GSE checkout computer and the onboard computer. Approximately 21,000 display messages were transmitted to the display tube (console) for monitoring of vehicle data.

CONCLUSIONS

The first three Saturn I launches, AS-201, AS-202, and AS-203, employed this checkout system. Test programs were used to evaluate and ensure launch readiness of the varied vehicle subsystems, relating to each propulsion stage, the flight computer, calibrating equipment, and telemetry. These test programs represent about 15 percent of the ultimate achievable goal of total automated vehicle checkout. In the launch site environment, procedures will always necessitate a degree of manual control so that ultimate achievable automation will always include various degrees of manual intervention.

This automation plan is indicative of a new mode of automating launch vehicle checkout. The personnel needed to accomplish these tasks must be versatile in many new disciplines of programming, system engineering, and automation. Only through team effort can the ultimate goals of space vehicle automation be accomplished.

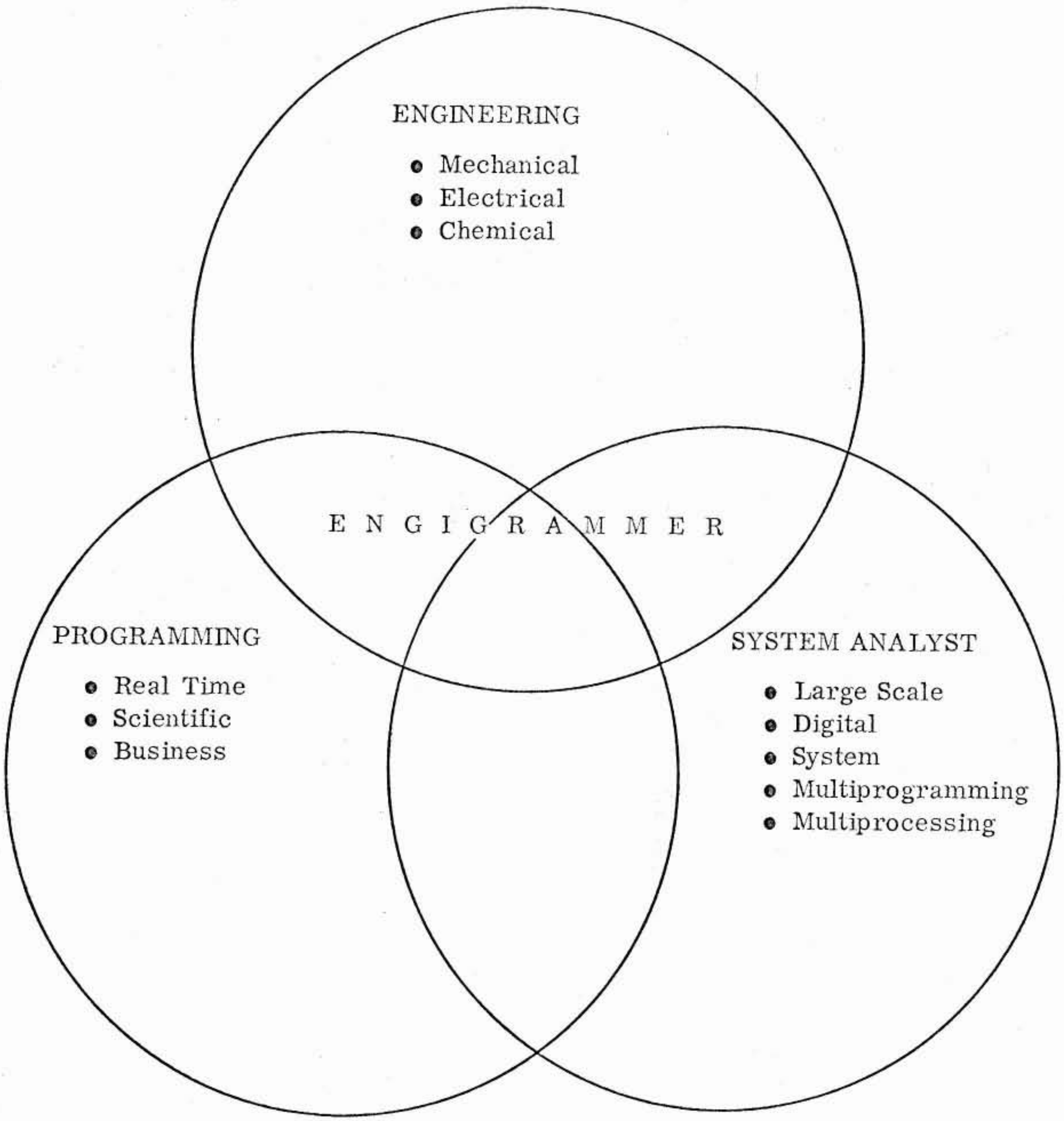

FIGURE 2. ENGIGRAMMER – COMPOSITE ENGINEER, PROGRAMMER, SYSTEM ANALYST

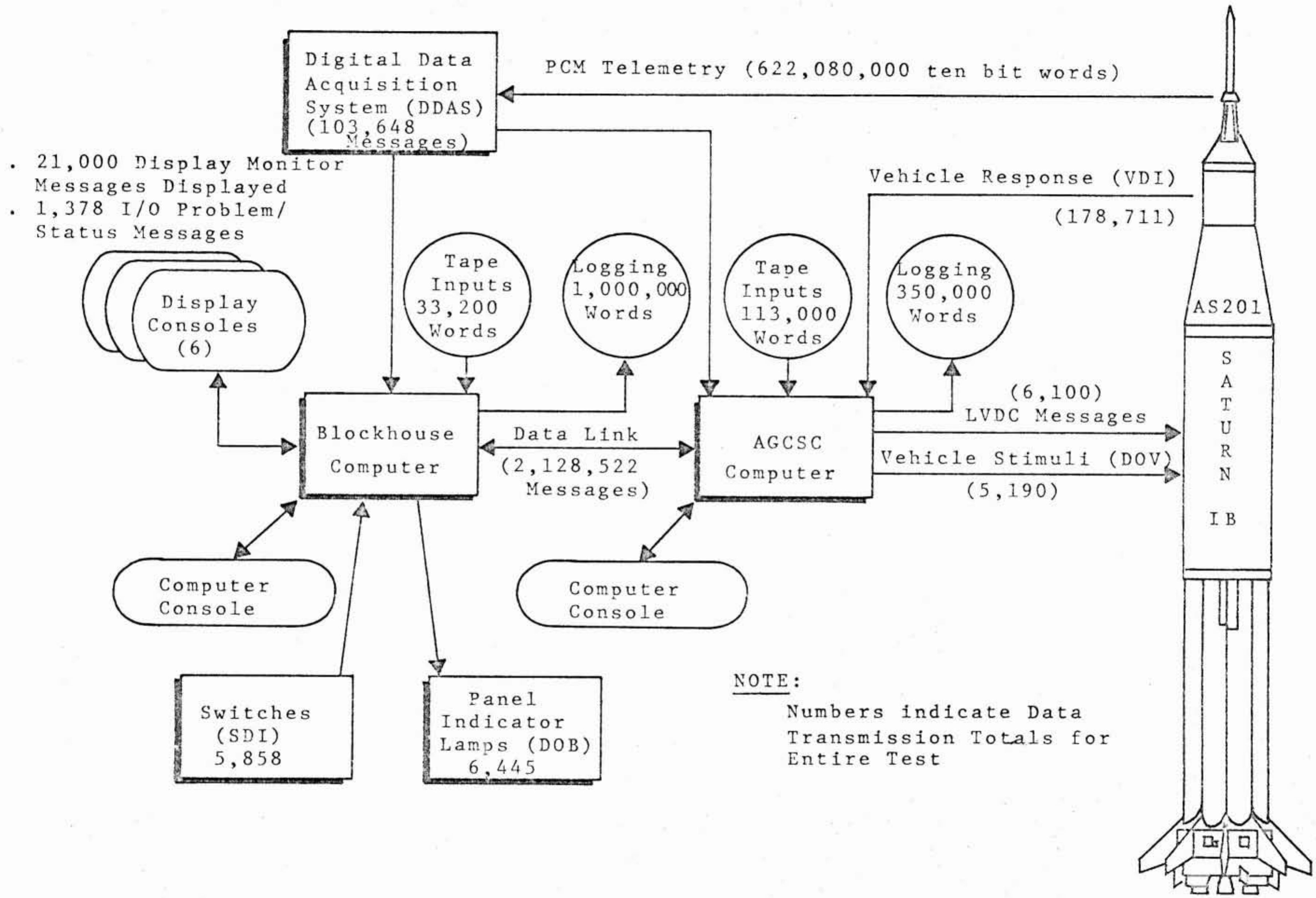

FIGURE 3. SATURN IB PLUGS OUT TEST USING THE DIGITAL COMPUTER COMPLEX

NOTES