COMPUTER-CONTROLLED HARDWARE TESTING
A Programmer's View of the Test Center

Jean W. Sherman
International Business Machines Corporation
Systems Development Division
San Jose, California

ABSTRACT

Computer-controlled hardware testing can be accomplished safely, dependably, and economically with the four-phase plan presented in this paper. The four phases are product design, product testers, tester programming, and data output. Usefulness of the plan increases if the designer, programmer, and management understand each others problems and do an adequate job of communicating.

INTRODUCTION

For many years whenever a designer had an idea for a new product, he would build a hardware model and try it out. If the model worked, fine; if it didn't, he went back to the drawing board and tried again.

Recently, computers have been used by designers to produce safer and more dependable and economical methods for designing and testing their ideas. Figure 1 shows a comparison of this method with the practice followed in the past.

The method described in this paper is one way a designer may utilize the computer. Known as the four-phase plan, it involves: (1) product design, (2) product testers, (3) tester programming, and (4) data output.

Communication ranks as the most important effort in the four-phase plan. Management, designers, and programmers must work together. They must understand each others problems and define the expected goals. They must document and communicate.

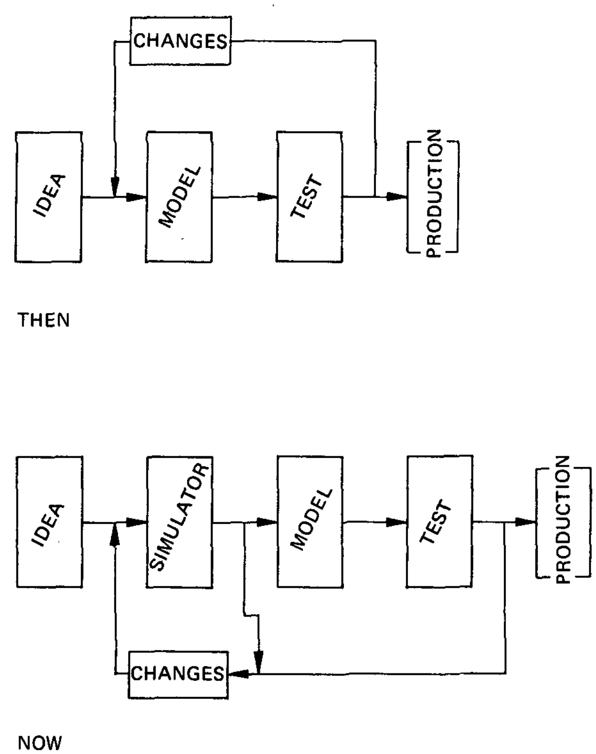


Fig. 1 Comparison of methods in designing products.

A description of the four-phase plan is presented in the subsequent section. Following this is an example of how the plan can be applied.

DESCRIPTION OF THE FOUR-PHASE PLAN

The first phase of the four-phase plan deals with the design of the product. Here, the designer and programmer fit the design to a simulator. In the second phase, the product tester is designed, debugged and installed. In the third phase, the designer defines the test of his product and the programmer writes the testing programs for the process-control computers. The test results are defined and edited in phase four.

Phase I -- Product Design

A designer begins by planning the new product and its test environment. When he has a plan, he brings in test personnel and the programmer to determine the best method for producing a well-tested product.

The designer defines all or a portion of his product to be processed by a simulator. Depending on the application, an existing simulator is used or one tailored to the task may be written.

The simulator performs four basic functions. These are design verification, logic diagnostics, input for product testing, and test diagnostics. Experience has shown that most simulators are good for verifying designs. However, most are weak in the logic diagnostic area and the capability of automatic generation of test data and test diagnostics.

Logic and test diagnostics include fault simulation where a bug is created and the results observed. Oscillation or logic breakdown prohibits further simulation after being recorded. This type of information is of value when testing the product and also for diagnostics when the product is in the field.

Test-data generation saves the time of designer and programmer when the product is ready to test. The input to the testers can be cards, tape or disk files. Disk file communication is the most direct for testers, process controllers, and large computers. Tape input is the most widely accepted although tapes do get lost or destroyed. Card input is the least desirable. Cards present storage problems and are more susceptible to damage, mixup and loss.

A graphic display device is a valuable tool for the designer. With it he can make temporary changes right on the face of the tube and them run through a partial or whole simulation. He can get hard copy of his design to use as documentation. Hard copy may be printer or plotter output.

Figure 2 shows a general simulator. This is how it may be used for an integrated circuit chip or module of chips: The logic is defined and tested for accuracy. All blocks without input/output and all nets not tied down are listed on the printer. A sample test is applied and all logic-block usage and switching during simulation are noted. Any nets or blocks not used or switched are printed upon request. All redundant inputs are listed. If the output pattern is given, each output is compared and discrepancies printed. The designer may request the status of all logic units at time of miscompare. The simulator, on request, will produce the inputs for future testing.

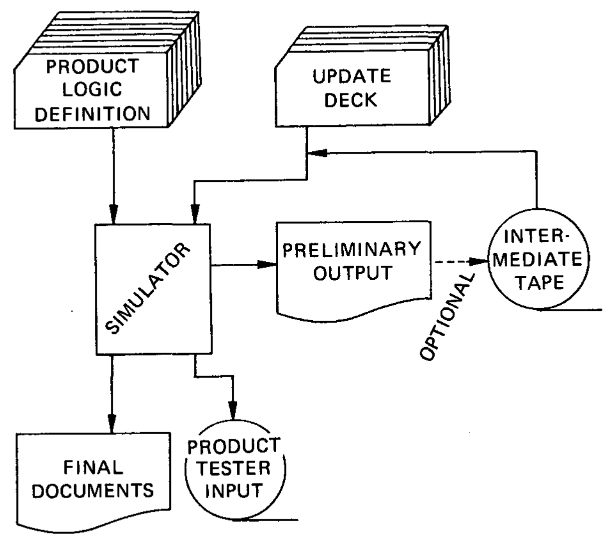


Fig. 2 A general simulator.

The ideal simulator is limited only by the computer on which it is run. Simulators that are independent of the technology can be used by many people during all stages of design.

Phase II -- Product Tester

During the design of a new product, the tester is also designed and built. The tester and product determine the format of the output data required from the simulator.

While the tester is being built, diagnostic programs are written to exercise each phase of construction. Diagnostic programs are also written to check and help maintain the tester.

Some areas for such diagnostic programs are wiring, mechanics, and tester/computer communication. All voltages controlled or monitored by the computer should be physically verified. All moving parts of the tester should have exercise programs to verify accuracy of the movement. Tester/computer communication such as digital and analog I/O should have diagnostic programs. These programs are valuable during construction of the tester, as well as preventative maintenance programs or tester failure diagnostics to be run when the product is being tested.

Always, the tester must have built-in interlocks and immediate manual control to prevent damage that may be caused by malfunctioning hardware or software.

Phase III -- Tester Programming

Tester programs are defined by the designer. There are several modes of testing. One is the go/no go mode for production testing which terminates the test when the first error is encountered. Another is a failure output mode, that is, when all failures are recorded and testing is continued to end of test. Check points may be inserted to stop the test if some number of failures occurs. This prevents volumes of useless data being printed in case of severe failure.

Some products lend themselves to failure diagnostics while others do not. A failure diagnostic mode can be useful for determining reasons for failures.

One of the most common diagnostics is the "loop on error" or "stop on error" test. This allows an engineer to probe, scope, or otherwise visually check existing conditions to determine the cause of the error.

Another possibility is to run a simulator in parallel with the test. Often an error is a result of a condition that happened several time elements prior to recognition. All conditions are compared at each point in time and miscompares are printed.

The programmer should be aware of hazardous conditions of test equipment. Programs should continually monitor voltages, liquid flows and mechanical operations during the test. Any condition not meeting specification should cause an immediate halt to the test and shut-off of all switches and valves. The reason for such a halt is printed after appropriate action has been taken.

Phase IV -- Data Output

All of the foregoing efforts are useless if the output is unusable. Data reduction and report generation is one of the most important areas of designer/programmer communication. Histories of performance serve as valuable aids in spotting trends. Performance plots provide a good means for maintaining long range activity and are relatively simple to program and support.

Test printouts are most useful when brief. Diagnostic outputs vary according to type. Most can be brief with the option remaining for full status printout which may become quite extensive.

APPLICATION OF THE FOUR-PHASE PLAN

As mentioned earlier, in applying the four-phase plan, a designer first conceives the new product and then determines its test environment. The make-up of such an environment is shown in Figure 3. Block (a) consists of computer-controlled testers and tables, (b) the process controller for block (a) and immediate reports, and (c) the large systems for design simulation and handling data reduction programs used for record keeping.

Diagnostics

As the testers and rework tables are installed, each installation step is checked out by diagnostic programs. Some of these

(a) COMPUTER - CONTROLLED TESTERS AND TABLES

CHIP TESTER	MODULE TESTER	CARD TESTER
BOARD TESTER	LASER TESTER	REWORK TABLES

(b) 1800 PROCESSOR - CONTROLLER

1810 DISK DRIVES
2401 TAPE DRIVES
DIGITAL AND ANALOG I/O

(c) OS 360/75

2250 CRT
2741 COMMUNICATIONS TERMINAL
2311 DISK (SHARED WITH 1800)

Fig. 3 An example of a test center.

diagnostic programs are usable by all on-line units. For instance, a general program to sèt and read specified voltage levels, another to open and close relays and measure current. For the testers that use a liquid, there is a diagnostic to measure flow and levels. The testers with moving parts have diagnostics to exercise the movements, such as a probe or filter rotation, arm placements, etc.

Some typical diagnostic routines are described for three of the testers in Figure 3. These are the chip tester, module tester, and laser tester.

Chip Tester

- Relays -- Specific relays are opened and closed. Readings are taken to verify the operation.

- Analog input -- Specific inputs are read either once and printed or a thousand times and a distribution printed.

- Resistance -- A specific voltage is set and total resistance is calculated and compared to a maximum.

- Power supply control -- A voltage is set to a give voltage and the analog input is read one hundred times. The average and deviation is printed.

Chip parameters are stored in tables to maintain flexibility of limits and chip types. Sockets are checked and tests performed. All data is held for later output. Indicator lights on the tester indicate a "pass" or "fail" condition.

Module tester

- I/O lines -- Check DC levels, transition time and relative delay.

- Coolant temperature -- Allow the temperature to rise and verify tester shut off when temperatures reaches a maximum.

- Probe operation -- Move probe to a specified point and verify accuracy of move.

- Power supply -- similar to chip tester.

Laser tester

- Current -- The current is turned on with minimum resistance and the system comes to a pause. The current is manually read and verified. Resistance is increased until the maximum, pausing at each step for verification.

- Motors -- Each motor is sent the number of pulses to move the mechanism a given distance. The distance is manually measured.

- Switches -- The switch for each laser is turned on and the system pauses for a manual check. The switch is then turned off, and the step is repeated for all switches.

- Limit switches -- The limit switch input is read and printed. When the limit switch is activated the print out changes.

The laser tester is a laser life test operation. It is run 24 hours a day with a unit test performed once a day, in the early morning hours. The unit test is a measurement of each laser output. A solar cell is rotated 180 degrees in 9-degree increments, and the laser output at several levels is read. All data is recorded on tape to be sorted and reduced later in the day on another computer. Every minute during the

24-hour period, the tester is operationally monitored by the process control computers. Any condition not meeting specifications causes a shut-off and the values are printed along with time of day.

Tester Development and Programming

During this time the programmer works on the actual test program. He debugs each portion logically with dummy data if necessary. Also, he writes a series of debug tests to check out the results of modifications as the definition is modified. The programmer continually documents and writes operating procedures so others may use the programs with a minimum amount of training.

A dummy product of known values is available for all types of test equipment. When a programming change is made, the complete test is rerun to verify that the change solves the problem and does not create a new one.

This procedure is useful also when the accuracy of the system is questioned.

Data Output

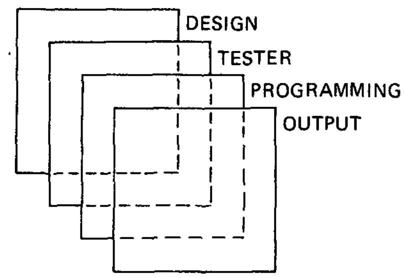
Daily reports for the laser unit test (Fig. 3) indicate which laser has failed or revived during the last 24 hours. The output tape from the 1800 is taken to the 360 for data reduction. Data is packed and the information is averaged for a two-day period. Once a week a performance plot is printed.

Other testers keep a log of part numbers and a pass or fail condition. The program will upon request print the full status or failure conditions for any part number.

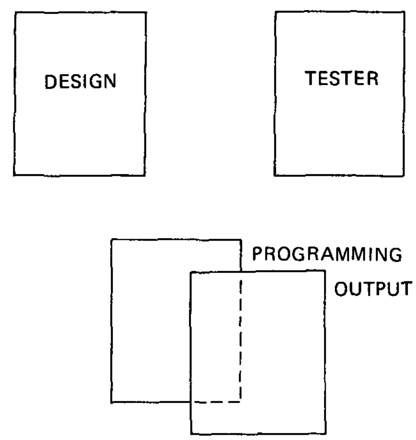
SUMMARY AND CONCLUSIONS

The four-phase plan presented in this paper offers a safe, dependable, and economical method for designing and testing ideas. In the first phase, the designer and programmer work together fitting the design to a simulator. In the second phase, they design and install the product tester. Next, they are involved with tester programming and, finally, in the fourth phase they are concerned with data output and record keeping.

If the people involved have worked together, the flow from phase one to phase four is smooth and integrated as pictured in Fig. 4a. Otherwise, the test center effort is disjointed as illustrated by Fig. 4b, often with chaotic results.



(a) BEST RESULTS



(b) LESS THAN BEST RESULTS

Fig. 4 Results of design efforts based on the quality of communication and co-operation between the designer and his co-workers.

Most design engineers are familiar with high-level programming languages. A programmer should use these languages so the engineer may follow the software progress. When the designer/programmer communication level is satisfactory, then the programmer should rewrite and/or continue programming, using the assembler level language. The reason for this is that most process control computers are short on core storage and efficient use of available storage is mandatory. When the programmer presents well-documented and tested routines for the designer's use, the high-level step may be passed.

Good communication during all phases of test center operation will appreciably lessen the possibility of accidents, tester and/or program failures. GOOD COMMUNICATION INCREASES THE USEFULNESS OF TEST CENTERS.