Operating Experience with Raydac

FRANKLIN R. DEAN

The Raydac acceptance tests were run from June 4 to July 19, 1952, in five sessions of 3 days each. The tests were divided into three phases. Phase 1 consisted of programmed test routines which caused each unit of the computer to go through its various combinations of operations. This included tests of the terminal equipment, central control, and internal memory; the arithmetic unit; and the external memory. Phase 1 also included an exhaustive test of all checking and error-detecting circuits. Phase 2 was a single mass-data-handling problem calling for large volumes of input and output data with relatively little actual computation. Phase 3 consisted of the solution of a full-length physical problem to which a reliable numerical solution was already available.

Phase 1

Terminal Equipment Tests

The terminal equipment tests involved the operation of the problem preparation unit, the output printer, the control console, and the directly connected printer.

Problem Preparation Unit. This test routine consisted of orders and data which were punched on paper tape and transferred onto magnetic tape by the conversion unit. To determine whether the preparation unit performed correctly, the magnetic tape was then put on the output printer, and the information printed out. Agreement between the printed listings and the original manuscripts was to be considered a demonstration of satisfactory operation. The original manuscript consisted of 32 orders and 160 numbers.

Three attempts were required before this test was passed. The first attempt failed because the speed of the transmitter-distributor governing the preparation of the first paper tape was set at an incorrect speed, through an oversight of maintenance personnel. This caused transfer weight count check errors to occur while preparing the second punched paper tape. The second attempt failed as a result of malpositioning of the magnetic tape on the paper to magnetic tape converter unit. The third attempt passed successfully without error.

It was also demonstrated that all combinations of errors occurring in the problem preparation unit were detected by the transfer-weight-count-error detection system.

Output Printer. This routine was designed to test the output printer and its several modes of operation, using a variety of formats and numbers. Specific instructions for the setting of the output printer controls were given on a master tabulation sheet which showed the form and contents of the printed results. Agreement between these results and the tabulation sheet was to be considered satisfactory performance for this test. Various formats were tested by separately adjusting the controls of the 10 printed page columns. This routine was completed without error on the first attempt.

Control Console. This routine was designed to test the use of the major operating switches and keyboards on the operator’s console, together with the operation of the directly connected printer. Forty-three words (orders and numbers) were inserted in the internal memory through the use of the order and word keyboards and the manual controls. The directly connected printer automatically printed these orders and numbers as they were entered. The orders and numbers entered comprised a routine which was started by the normal start routine of the computer. The computer performed this routine and, during its course, 16 numbers were printed on the directly connected printer using both decimal and octal printing modes. This test was performed successfully on the first attempt with no errors. It took approximately 1/2 hour to insert the information manually in the internal memory, and approximately 1 minute of computation time. It was also demonstrated that errors in the control console and directly connected printer were detected by the transfer weight count and printing error detecting systems.

Central Control and Internal Memory

This test consisted of 15 separately coded routines which tested the variables that effect the processing of an order by the central control. These variables are the type of address (regular addresses of the internal memory, special addresses of static registers and void addresses) and the time of selection of these addresses. Orders consisting of all combinations and permutations of the different types of addresses were programmed. The time of selection was varied by deliberately placing the operands in certain relative word positions. In the course of the routines all internal memory positions were read from and written into at least once. The program of these routines was such that, upon completion of the last routine, the first routine was inserted again. One cycle of the routine took approximately 36 seconds. To pass this test, the computer was to operate continuously for at least 5 minutes without error. The test was completed successfully on the first attempt, running for 26 minutes and 45 seconds. The machine stopped at this time as a result of an arithmetic unit error that was detected by the arithmetic weight count check. It was also demonstrated that all operational and transfer weight count error-detecting circuits in the central control performed properly.

External Memory

This routine was devised for the express purpose of testing the external memory of the computer in its several modes of operation. The four external memory orders; hunt-prepare-to-read, hunt-prepare-to-write, read, and write, were used in a variety of combinations. Program checks were inserted to verify the correctness of the machine’s operations. Hunting, reading, and writing were performed on the magnetic tapes of all four external memories. The test was self-cycling, requiring no operator intervention at any point. To pass, the computer was to operate continuously without error for a period of 5 minutes. This test was passed successfully on the first attempt, running without error for 1 hour and 23 minutes. It stopped as a result of a transfer weight count error of one of the operands transferred to the arithmetic unit. During this test, approximately 7,000 36-bit words were written on and 6,300 36-bit words read from the magnetic tape. After the completion of this test, it was also demonstrated that all operational and transfer weight count error detecting devices operated satisfactorily.

Arithmetic Unit

The variables in the arithmetic unit test routine were the 26 operations and...
the known configurations of the operands. There was a separate test for each of the arithmetic operations. For example, more than 1,000 additions and 1,000 subtractions were performed. If each routine was processed correctly, its operation code number was printed by the directly connected printer. If the operation being tested was not being performed correctly, the machine would stop shortly after the performance of the incorrect order. The test was self-cycling, requiring no operator’s intervention at any point, and took approximately 3 minutes for one complete cycle. To pass, the computer was to operate continuously without error for a period of 5 minutes. The test was passed on the first attempt. It ran without error for 1 hour, and 52 minutes, during which time approximately 11,000,000 operations were performed. It stopped as a result of a transfer weight count error in the external memory. After the completion of this test, it was demonstrated that all operational, transfer weight count, and arithmetic weight count error circuits in the arithmetic unit operated satisfactorily.

Phase 1.2

Two supplementary tests, designated phase 1.2, were added to phase 1 for the purpose of producing conclusive evidence of everything tested, mainly by printing results on the directly connected printer. The first routine was designed to test the ability of the arithmetic unit to perform each of its built-in operations, the ability of the machine to transfer data from one random memory position to another, and the ability of the internal memory to retain data. This was accomplished by performing each of the built-in operations of the arithmetic unit using operands designed to test the circuits as exhaustively as time permitted and, in general, printing from the directly connected printer the operation code, the two operands, and the result of the operation. All lines and word positions of the internal memory and the upper reservoirs were used to transfer randomly, using each different kind of transfer to do this. The word was printed before and after it was transferred. Fifteen minutes after the last printing, the transferred numbers were reprinted to demonstrate the ability of the internal memory to retain data. This test was programmed to last 1 hour and 13 minutes when performed without interruption. The directly connected printer was the limiting factor in this case. Approximately 1,657 words were printed. The test was passed with two errors occurring during the operation. Both of these errors were due to loss of information in the internal memory and were detected by the transfer weight count error check. In each case, the information was restored by transfers from the operator’s console, and the test restarted at a previous point in the routine.

The second part of phase 1.2 was designed to test the selectivity of the central control. The Raydac selection circuits provide for the selection of the address that is available first, for the immediate selection of special addresses, and for bypassing the selection of void addresses. Ten separate tests were performed to demonstrate the ability of the central control to make address selections in this manner and in the exact time stated. Each test was timed for a specific length of time. The calculated number of operations was then checked with the actual number of operations performed.

For instance, to demonstrate an average time of selection of operands, one order consisting of four regular addresses was cycled for 6 minutes. This order, which was a counting order, was performed at the rate of approximately 1,640 operations per second. At the end of 6 minutes, the order had been performed 589,553 times, as contrasted with the calculated number of 589,824 times that it should have been performed. This small discrepancy can be attributed to the time lags in relation to starting and stopping the machine, and starting and stopping the stop watch.

To demonstrate a fast selection of addresses, 5 orders which contained a number of void addresses were cycled. This was also programmed so as to count the number of times the routine was performed. The orders were performed at the rate of approximately 8,200 per second. At the end of the 1 1/2 minutes, the five orders had been cycled 541,239 times as contrasted to the calculated number of 540,672. More than 2,700,000 operations were performed in the 1 1/2 minutes.

Phase 2

Phase 2 was a problem which called for the evaluation of a rational algebraic function of two independent variables for an uninterrupted period of 5 minutes. The function was of the type \( Q = f_1(x,y)/f_2(x,y) \) where \( f_1 \) and \( f_2 \) were both polynomials. The values of the independent variables were generated in advance of the test by the computer, and recorded on magnetic tape. Eighteen hundred blocks consisting of 28,800 pairs of values of \( x \) and \( y \) were recorded. The results, along with the values of the independent variables, were recorded on tape. At the conclusion of the problems, values were chosen at random and then printed from the tape by the output printer. The results were checked by hand computation. This test was passed on the first attempt with no errors, all selected values agreeing precisely with the hand calculated values.

Phase 3

Phase 3 consisted of a problem of 14 simultaneous nonlinear differential equations. Solutions of the 14 dependent variables and their derivatives were computed and recorded on magnetic tape for many values of the independent variables. During the course of this problem, approximately 660 numbers were printed out on the directly connected printer. The results were to agree within 3 per cent of those which were produced on another machine. There was no limitation regarding machine error except that the problem was to be successfully concluded within an 8-hour period. This problem was solved on the first attempt in 1 hour and 3 minutes without error or interruption.

Error Diagnosis

In the design of the Raydac, considerable emphasis was placed on circuits and methods for providing self-checking. This was done to minimize the number of undetected errors, and to provide means for error diagnosis. As a result, the operator’s console contains a large number of error indicators, status lights, and controls for obtaining diagnostic information.

When an error occurs, computation is automatically stopped. The operator can immediately determine the order being processed and the point in the order at which the error occurred. In most cases, the operator can simply push the reset and then the normal start button. This starts the computation with the same order in which it has failed. If the error was of a transient type, the problem might proceed normally. If the error reoccurs, the operator should try to determine the cause of the error and the circuit involved. Some of the means of obtaining information available to the operator for error diagnosis is as follows:

1. Twenty-eight error lights which indicate the circuit or unit causing the error.
2. A group of lights which indicate the status of the central control, including the half-order being processed, the addresses...
that were selected, and the tag of the last address selected.

3. A group of lights which indicate the operation and, in the case of compound operations, the suboperation being performed.

4. A group of lights which indicate the status of the arithmetic unit. These lights indicate the exact step in the arithmetic processes at which the error was made.

5. A group of lights which indicate the status of the external memory; that is, if they are in a read or write state and if a hunting operation is in progress.

6. A monitor oscilloscope by which it is possible to read the contents of all internal memory addresses and all special address registers. Thus it is easy to determine such information as the order being processed, the previous second half-order that was processed, and operands of the order, and the content of the arithmetic unit registers.

To assist in determining the cause of an error, be it due to a component failure or some other factor, there are also available many manual controls for operating specific units of the computer. For example, it is possible to cause the central control to cyclicly process a whole order or even a half-order without requiring the use of other major units. Also, such circuits in the central control as the transfer weight count adder and selection circuits can be operated individually. A typical error occurring in central control might be a selection error. Selections are checked in the Raydac by dividing each address processed in central control into its spatial and temporal parts. These parts are then compared with tags delivered by the matrix and the clock indicating the selections actually made. Disagreement between any corresponding digits of the address and the tags is indicated as an error and the computer is automatically stopped. If this error occurs, the operator can determine from the central control status lights the address selection that failed, and the tags that were delivered for this address selection. He can then look up the order residing in the central control and compare the address of the selection that failed with the tags. Agreement indicates a failure in the checking circuits. Disagreement indicates a failure of either the temporal or the spatial selections. Further to isolate the failure, a built-in routine for cycling an address selection can be used.

It is possible to step the arithmetic unit manually through each of its sub-operations and thus analyze the status of the arithmetic unit after each step. A switch arrangement is provided which shifts the control of the arithmetic unit over to special test equipment. This permits any operation to be cycled with any pair of operands.

The arithmetic unit status lights previously mentioned represent the nine stages of progress in the arithmetic unit operation. The first stage indicates that the arithmetic unit has been properly prepared for the operation. Stages 2 and 3 indicate that the two operands were delivered properly and that their transfer weight counts checked. Stages 4 to 6 indicate the progress of the main and check arithmetic unit computations. Stage 7 indicates agreement between the main arithmetic unit's result and the check arithmetic unit's weight count identity. Stages 8 and 9 indicate that the valid weight count has been appended to the result and that the central control has been sequenced. Failure in any of these steps prevents the transmission of the signal which would initiate the next step of the operation. Other check circuits prevent the arithmetic unit from beginning an operation if the central control has not completed all of its checks, and prevent the central control from beginning a new operation if an arithmetic unit error has occurred.

When the console indicates an arithmetic unit error, the operator first notes from the status lights whether the error occurred before or after the arithmetic unit began an actual computation. A typical failure before computation might be indicated by the arithmetic unit transfer weight count error light, and with the status lights indicating that only one valid operand was received. The operator can look up the operands in the memory by use of the monitor scope, and check the transfer weight count tags of these operands. If the stored tags are incorrect, he knows that this information in the memory has been garbled. If they are correct, the operator should check the tags of the operands as delivered to the arithmetic unit registers. If the transfer tags are incorrect here, the operator has isolated the trouble to the read-in circuitry and transmission path. If the tags of the operands in the arithmetic unit are correct, trouble is isolated to the transfer-weight-count checker.

A typical failure occurring during the arithmetic computation might be indicated by an arithmetic check error light, with the arithmetic unit status lights indicating that all main arithmetic unit computation was completed. In this case, the operator would look up the operands in the memory, manually perform the arithmetic unit operation, and check this result against the contents of the arithmetic unit result register.

In conclusion, it would be fitting to quote a summary from the report on the acceptance tests on the Raydac by Professor Francis J. Murray of Columbia University. Professor Murray was the chief examiner for the Navy during the acceptance tests. He wrote: "The Raytheon computer acceptance tests were run from June 4 to July 19 in five sessions of three days each. The computer passed all its tests, some in a brilliant manner. The percentage of 'down time' for the computer during these tests was reasonably good if one takes into account the fact that this is a new computer. Except for a transient difficulty in the phase 1.2 test, all tests were completed perfectly. The phase 3 test was particularly notable since the machine did in 1 hour and 3 minutes, without error or interruption, a computation which previously had required 20 hours on an automatic sequence calculator. "The checking features of the Raytheon computer were demonstrated in a very convincing manner. The computation procedures are simplified and the effective computing ability of the machine is greatly increased. The external memory input and output equipment are very effective pieces of equipment because of the combination of this feature and well-engineered construction.

"Other aspects of the design of the Raytheon Computer probably should be judged on the basis of actual use. I believe the machine is certainly in a state as good as any large scale electronic computer at the start of its computing life."

Dean—Operating Experience with Raydac

From the collection of the Computer History Museum (www.computerhistory.org)
I think that whenever it is important to have no error whatsoever in the preparation of data for feeding a machine, it would be advisable to provide a double preparation system, that is, where the actual data is prepared twice, such as program information. In cases where the amount of input information is large, I suppose it does not matter too much whether an occasional digit or character is dropped or picked up, in which case this would not be necessary. Certainly for program information and possibly for some kinds of data, I believe this should be continued.

K. M. Rehler (Raytheon Manufacturing Company): That is a good broad question. I will make my answer a little bit safer by saying that some of it is purely conjecture. We have studied defects quite a bit on the plastic base medium. We feel sure that a plastic base medium would be very suitable for data storage. We have looked at some of the defects that we have observed on the magnetic oxide and the material used for the colloidal suspension when they build a film of oxide on the base backing. They have observed that this has become a small hill. You can scrape these off, as I said in my paper, but the question is how to overcome this defect. The tape manufacturers are co-operating with us. We did some work on this, I believe, with Minesota Mining to investigate particle size of the magnetic oxide and the material used for the colloidal suspension when they build a film of oxide on the base backing. They have observed good improvement in the past 2 1/2 years. In the future, assuming that a plastic base magnetic tape might still have 10 flaws in 1,000 feet, how does one get rid of those flaws? Pre-pulsing techniques, in the future, may become more important. I believe the Bureau of Standards is working on this quite successfully. This requires that the sprocket or sync channel be permanently written, and never rewritten. Therefore, if a sprocket pulse exists somewhere, it indicates that all pulse locations across the tape are free from defects. This is a very good solution.

For those who use a computer with some time available, the computer probably represents the best minimum test equipment both for inspecting tape and perhaps preparing prepulsed tape. The inspection of tape used at Raytheon inspected 7 channels simultaneously and involved about 28 tubes. The effect of a splice joint is difficult to evaluate since, among other things, it is a function of the quality of the magnetic background, or erase, used.

C. V. L. Smith (Office of Naval Research): Do you ever have trouble with the magnetic tape units losing information due to random dust particles on the tape?

Mr. Rehler: At one time, we did. That kind of problem is one that occurs after all the easy-to-find electronic troubles are overcome. We have a dust cover over our tape drive, and a wiper of lens tissue.

R. D. Bronson (Consolidated Vultee Aircraft): Would you give more details on magnetic tape flaws, and indicate the trend in manufacturing techniques, and if it is possible to eliminate essentially all defects? What minimum test equipment would the user require to confirm the tape quality, and how does a good sprocket joint affect the double test routine?

K. M. Rehler: As far as polishing the tape is concerned, we have made it a practice to run a tape reel a couple of times before recording anything on it. It sometimes helps in smoothing down the bumps a little (the magnetic defects or clumps). It also helps, on occasion, to wipe the entire tape.

D. Haagons (Computer Instrument Company): What is the speed of output with the Teletype equipment? Was any thought given to higher output speed printing equipment?

Mr. Gray: The output speed varies, as I mentioned in the last part of the paper. The shortest printing time for a word is 2.8 seconds. The shortest printing time for a word is 0.6 second. As to going into higher speed printers—yes, we would like to, but we did not. We stayed with the Teletype digit-by-digit printing system. This offers one advantage, notably, that we obtain a check on each character on the page as it is printed on the page. This, we feel, is a very worthwhile feature.
Engineering Organization of Input and Output for the IBM 701 Electronic Data-Processing Machine

LOUIS D. STEVENS

The International Business Machines Corporation (IBM) type-701 electronic data-processing machine is to be a general-purpose high-speed electronic calculator utilizing the Williams type of electrostatic storage. The binary system of notation is used internally with a word size of either 36 bits (a full word) or 18 bits (a half word); transmission is parallel throughout the machine except to and from magnetic tape. Instructions are of the single-address type with 32 possible operations, and a maximum address capacity of 4,096 half words; each instruction occupies one half word of storage capacity.

In addition to the high-speed electrostatic storage with a maximum capacity of 2,048 full words, the machine is provided with a slower access time magnetic drum storage of 8,192 full words, and a completely integrated and flexible input and output system consisting of one card reader, one card recorder, one alphabetic printer, two twin magnetic tape reader-recorders, and manual input and output by means of the operator's control panel.

A more detailed description of the general characteristics and of the organization of the machine has been presented previously.

Input-Output System

The input-output system of type 701 is so designed that the use of an input or an output device by the calculator is accompanied by the performance of five distinct functions:

1. Selecting. An input-output unit to execute either a reading, a writing, or an auxiliary operation such as rewinding a tape.
2. Interlocking the operation of the selected input-output unit with the execution of the calculator program.
3. Copying data to and from the electrostatic memory and the input-output units.
4. Synchronizing the signals between the input-output unit and the calculator.
5. Disconnecting the input-output unit from the control of the calculator after the operation for which the unit was selected (and connected) has been completed.

These five functions are performed in such a manner that the inactive waiting periods normally associated with the use of an input-output unit may be utilized for useful operations by the calculator. This is accomplished by allowing the calculator to continue with the execution of its program during these otherwise inactive periods, and to converse with the input-output equipment only at those times when these devices may be in a position to transmit or receive data in full-word increments to or from the electrostatic memory. The following general description of the use of an input-output unit by the calculator makes the operation clear.

An input-output operation is initiated by the execution of a Select instruction (read or write) which places the desired unit under calculator control and sets the unit into motion to perform the function required. After the execution of select instruction the calculator may continue with its program, until the selected input-output unit has reached a position where it requires a word to write or a place to store a word which has been read. Just previous to this point, the calculator must have executed an instruction which is known as copy and skip or simply as copy. This instruction provides an address in electrostatic memory at which a word to be written may be located or at which a word that has been read may be stored.

Upon reaching a copy instruction in its program, the calculator will delay the execution of further instructions until the selected input-output unit has reached a position that will allow the memory address associated with the copy instruction to be utilized. Since most transfers of data between the storage and input-output units consist of more than a single word, successive copy instructions are required, one for each word to be transferred. These successive copy instructions only need be available just previous to the time an input-output unit may utilize the storage address, thus making the time between successive copy instructions available for useful operations.

From the foregoing it can be seen that the copy instruction not only supplies the necessary storage address for data transfers between the storage and input-output but also provides a synchronizing and interlocking function between the slow-access input and output devices and the faster access electrostatic storage. The skip function of the copy instruction will be described later as part of the discussion of the card reader and magnetic tape reader and recorder.

Since the calculator may continue with its program after an input-output device has been selected for reading or writing, it is necessary to provide an interlock, which will remember that a unit has been selected, and should the program arrive at another select instruction, its execution will be delayed until the previously selected unit has completed its operation. This interlock is known as the input-output interlock. It is turned on simultaneously with the selection of an input-output unit and will remain on as long as the selected unit is in operation. It will remember the unit selected and the function (reading or writing) for which it was selected. The need for this is due to the manner in which information is transferred to and from the electrostatic storage and the input-output devices. The storage address is supplied by a copy instruction, and the direction of information flow is determined by the setting of the input-output interlock.

Information transfer between the electrostatic storage and the input-output equipment is always routed through the multiplier-quotient register (M/Q). These transfers of information take place in single full-word increments and one copy instruction is required for each full-word transferred. Figure 1 shows, in block diagram form, the interconnection of the various units with the M/Q register of the calculator.

Each time a selected input-output unit is in a position to transfer information, the program must have arrived at a copy instruction just previously, to supply the required storage address. Should no copy instruction be available when required, a disconnect signal will be generated which will turn off the input-output

Stevens—IBM 701 Electronic Data-Processing Machine

This paper is a report of the efforts of many people in the IBM Engineering Laboratory. While it is impossible to name all of them at this time, I wish to recognize, and add my words of appreciation for, the fine contributions they made.