R. H. Doyle
R. A. Meyer
R. P. Pedowitz

Automatic Failure Recovery in a Digital Data Processing System

Abstract: This paper describes a program which will enable a complex digital data processing system to give "first aid" to itself. Ordinarily, when an error occurs during system operations, the computer must be stopped for corrective maintenance. The FIX program, however, automatically compensates for computer malfunctions so that recovery from errors may be effected with a negligible loss of operational time. Some equipment features used by the FIX program are briefly outlined prior to a detailed discussion of the structure and function of the program itself. In its initial application in the SAGE system, FIX provided automatic recovery from more than 90% of all failures occurring during the period studied.

Introduction

Perfect reliability in digital computers has not yet been achieved by simply designing ruggedness into the equipment components. Nevertheless, it is essential for a computer to perform dependably under all conditions. In certain computer applications, errors resulting in unscheduled maintenance delays can be tolerated, but only at the cost of expensive computer time. In some special military and civil applications, such as the SAGE System and air-traffic control systems, poor equipment reliability can be disastrous, since input information not processed when the system is inoperative can become obsolete during the time required for manual recovery.

Although it is virtually impossible to guarantee that failures will never occur, it is possible to maintain high over-all reliability of the system by immediately recovering from these failures with a negligible loss of operational time.

The FIX program was designed to effect automatic recovery from failures by either:

a) re-initiating the operation that failed,
b) preventing the operational program from processing incorrect data, or
c) determining the effect that a particular failure would have on a word of information and then modifying the information to compensate for this failure.

The error-detection circuitry of the computer is relied upon to indicate the existence of an error in computer operations. When an error is detected by this equipment, the FIX program will be operated in an attempt to diagnose the failure and to compensate for it.

Although FIX was specifically designed to work with the Air Defense Program of the SAGE Computer, the technique employed may be modified for other operational or production systems.

Several other methods for maintaining system reliability have already been developed. Some of these methods will be briefly outlined in the preliminary section of this paper, followed by a detailed description of the structure and operation of the FIX program.

Reliability techniques

In a complex computer system, component quality standards are necessary but cannot in themselves insure complete reliability. To approach the goal of high reliability, a more sophisticated viewpoint has been taken in designing both the equipment and the computer programs.

In the SAGE System, for example, the complete central computer has been duplexed, and the two computers alternately perform the operational program and a standby program on a 24-hour schedule. Special alarm circuits provide for alerting the standby computer when the active computer breaks down, so that the standby machine will prepare to assume the active role. A portion of the standby-computer time is devoted to attempting to predict potential failure conditions before they occur. This technique, known as "marginal checking," consists in operat-

ing and testing various circuits while an abnormal voltage is supplied to them. In this simulated aging of the equipment, the potential failure spots are anticipated.

Modern computing equipment is usually designed with built-in circuitry[1] that will automatically detect the majority of errors that occur during system operation. Many operational programs are written to take advantage of this circuitry by including alarm-interrogation routines which will automatically repeat any operation that generated an alarm.

Error-checking routines have also been incorporated directly into operational programs.[2] In programs where it is necessary to store blocks of information on auxiliary storage drums or tapes before re-using it, the accuracy of the transferred information may be checked by comparing the arithmetic sum of the block before it is stored to a similar sum obtained after the block is brought back from storage. If the two check sums are not equal, the reliability of the information block cannot be depended upon and the program should be re-run. If the program is of considerable length, this task may be shortened by periodically saving the environment of the program as it operates. This will provide a convenient recovery point should it be necessary to regenerate a particular block of information.

Elaborate equipment and coding systems, such as the Hamming Code,[3] can provide for automatic self-correction of errors and for detection of multiple errors. This is accomplished by dividing the information to be checked into groups of bits and by parity-checking each group. The groups of bits are chosen in a manner such that an error in any bit in the entire word will generate alarm indications for a unique combination of these groups. Conversely, incorrect parity counts for any combination of these groups will uniquely identify the erroneous bit in the word. Since the incorrect bit can be identified, circuitry can be provided to correct the error. This ingenious coding system achieves excellent results, but only at considerable expense. Channel capacity of the equipment must be increased to provide for enough checking bits to represent a number equal to the total number of information bits plus the checking bits.

Although FIX incorporates some of the techniques described above, the distinguishing feature of the FIX program is that it achieves automatic failure recovery by means of programming techniques after an error has been detected by machine circuitry. While variations of the FIX concept will be necessary for other operational systems, depending upon the error-detection circuitry of the computer and upon the form of the operational program, this paper will serve to illustrate the general principles of the FIX technique.

Errors in a computer can occur either during the actual processing of data, such as sorting, collating, arithmetic computations, et cetera, or during the transfer of information between the central computer and the various auxiliary drum storage units. Since the Air Defense Program requires a large storage area, it is stored on auxiliary drums, and a considerable number of information transfers continually occur during normal operations as the various subprograms and their data tables are brought into core memory to be operated. It is extremely important that these transfers be performed correctly; hence a large portion of this paper discusses the technique of monitoring and correcting errors in such transfers.

Errors incurred during either central computer or transfer operations may be either transient or "solid" in nature. Errors which are due to high stresses of voltage, temperature, shock, et cetera, and which have a low probability of recurring, will be referred to as "transient errors." Those errors which are a result of a persistent equipment malfunction, and which can continually be expected to reappear whenever the submarginal area of equipment is used, will be referred to as "solid errors." The FIX program has achieved a high degree of success in automatically recovering from most of the classes of errors described above.

Program design

Storage requirements for the present version of the FIX program are 50 core-memory registers and 5000 auxiliary-drum-storage registers. This represents approximately 3% of the total storage available in the SAGE Computer. During any alarm condition, the short FIX routine that is permanently stored in core memory will save a portion of the operational program in order to provide a working area for the main section of FIX. This same routine will then read the appropriate diagnostic FIX routine into core memory.

Figure 1 is a flow chart of the logical structure of the FIX program. This structure may be analyzed in terms of four functions:

a) Alarm Monitoring and Control
b) Diagnosis
c) Logging
d) Recovery.

These functions are closely related and, although the above list represents the over-all time sequence of the operations to be performed, there will be considerable overlapping in the detailed structure. Since the design of the FIX program is a function of the make-up of the operational program and of the system to be monitored, some of the features of the SAGE System, including the error-detection equipment of the computer and the structure of the Air Defense Program, will be discussed during the analysis of the FIX Program.

• *Alarm monitoring and control*

The operation of the FIX program is greatly dependent upon the means by which FIX can be notified of an error occurring in the monitored system. In the SAGE Computer, this is provided for by error-checking and alarm-control circuitry.

Self-checking is performed by the use of parity-code generation and checking circuits that determine if the correct number of bits in a binary word have been transferred from register to register during the normal data-processing operation. This is accomplished by increasing

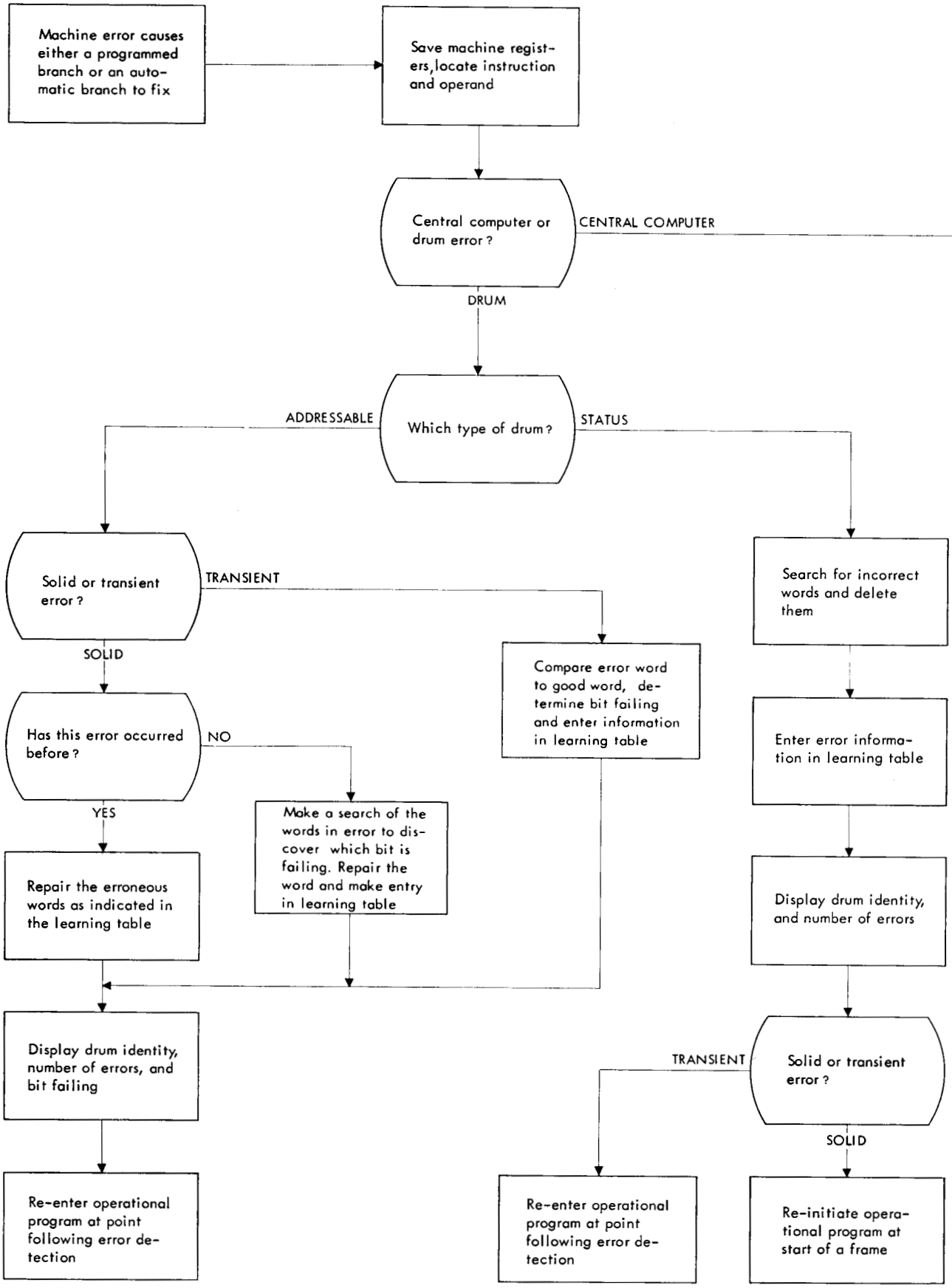
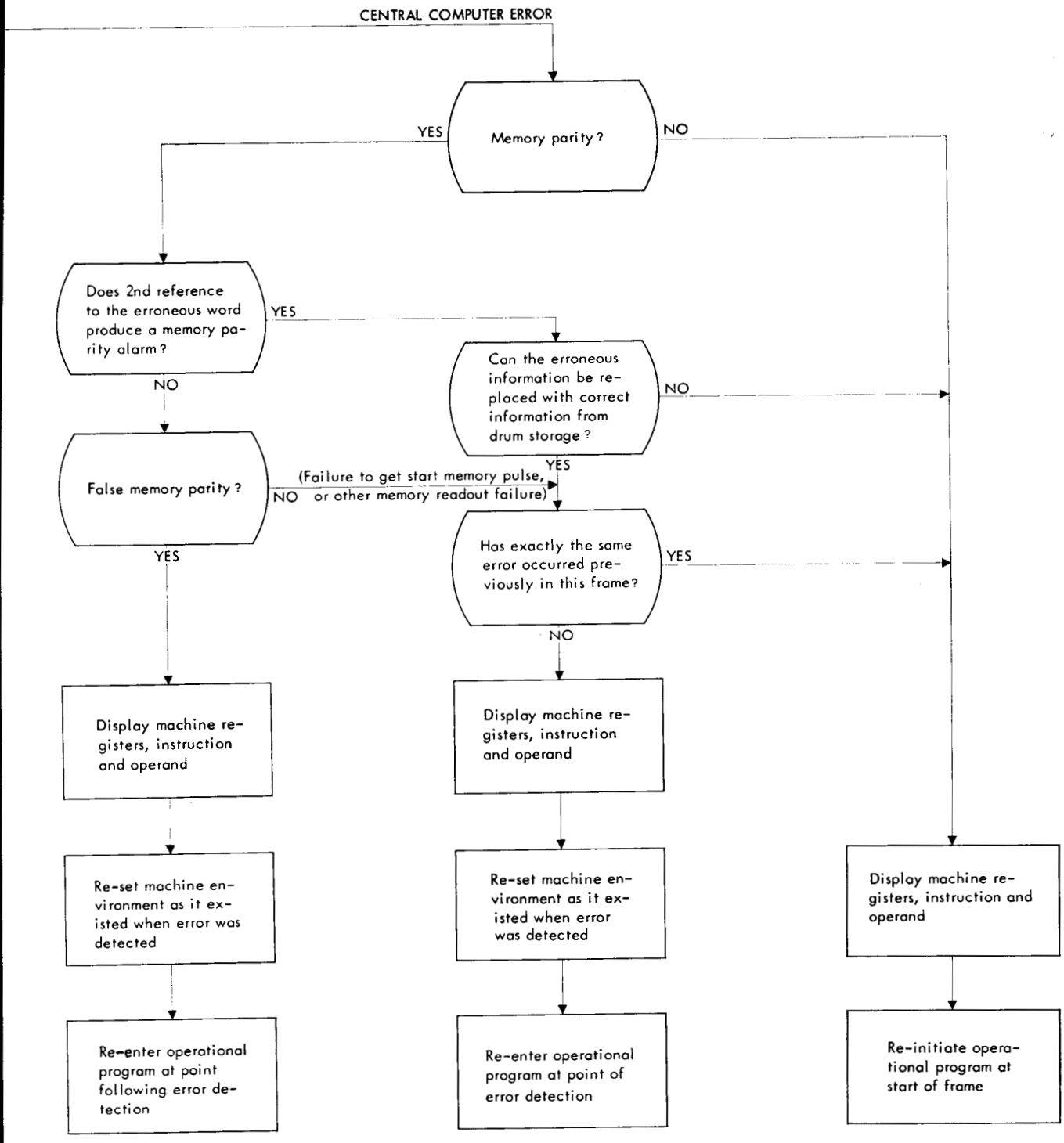

Machine error causes either a programmed branch or an automatic branch to fix
Save machine registers, locate instruction and operand
Central computer or drum error?
CENTRAL COMPUTER
DRUM
Which type of drum?
ADDRESSABLE
STATUS
Solid or transient error?
TRANSIENT
SOLID
Has this error occurred before?
NO
YES
Repair the erroneous words as indicated in the learning table
Make a search of the words in error to discover which bit is failing. Repair the word and make entry in learning table
Compare error word to good word, determine bit failing and enter information in learning table
Display drum identity, number of errors, and bit failing
Re-enter operational program at point following error detection
Search for incorrect words and delete them
Enter error information in learning table
Display drum identity, and number of errors
Solid or transient error?
TRANSIENT
SOLID
Re-enter operational program at point following error detection
Re-initiate operational program at start of a frame

Figure 1 **Flow chart of the logical structure of the FIX program.**



channel capacity to allow for one redundancy bit to be contained in the information transferred. As each instruction or data word is stored in the computer, it passes through a buffer register, which counts the number of "one" bits in the word. The parity bit associated with each word will be set to a "one" or a "zero" to give an odd number of "one" bits in the word, including the parity bit.

The parity bit will then be stored with the rest of the word. When this instruction or data word is referred to by the program, a parity-check count is again performed in the buffer register as the word is brought out of storage. If the total parity count is not still odd at this time, the word is presumed to be incorrect and a parity alarm will be generated. If no error is detected, the operation will continue and a new parity assignment will be performed prior to storing the word after it has been operated upon. The parity circuitry is used to check the correctness of all data transfers that occur in the system. It should be noted that a major shortcoming of the parity-checking system is that if two bits in the word are altered as the result of some failure, the odd parity count will not be disturbed and the error will not be detected by the parity circuitry. Such an error might remain unnoticed, in which case the final result would be incorrect, or it might result in other error indications which could be detected.

Automatic detection of other abnormal conditions in the SAGE Computer is also provided by circuitry. Sometimes because of circuit failure or an undetected parity error, or because of a peculiar set of environmental circumstances unanticipated by the program designer, the computer can begin a non-terminating cycle of meaningless operations, commonly referred to as an "illegal loop." Similarly, the computer might begin an inactive period during which it does nothing but wait for some anticipated event. If for some reason the event can never occur, the computer will remain in this inactive condition indefinitely. Special circuitry designed to impose time limits on such conditions can, upon sensing an illegal delay in computer operations, terminate the condition and, by means of an inactivity alarm, indicate to the computer that the delay existed. The inactivity alarm will be activated if a pulse is not generated by the program at regular intervals or if too many of these pulses are generated within a given time period, usually about eight seconds. The programmer must, therefore, insert the pulse-generating instruction at regular intervals throughout his program if he intends to use this circuitry. If the program operates normally, the pulses will be generated at regular intervals. If the program is "illegally" delayed in a routine, or if it continuously loops through a few instructions, either too few or too many signals will be generated, and the inactivity alarm activated.

Finally, in certain instances an error in a series of arithmetic operations may result in the attempted development of a sum or quotient which has increased in size beyond the physical limits imposed by the register capacity of the computer. This condition too, can be sensed by machine circuitry in the SAGE Computer and indicated by means of an overflow alarm.

The programmer can choose various modes of operation by using switch settings when planning the reactions of the SAGE Computer to these alarm conditions. These options can be set to have the computer automatically

a) Stop on alarms
b) Branch on alarms
c) Continue on alarms.

Under option (c) the program retains the ability to interrogate the alarms at some convenient time before taking any automatic action.

The mode of operation used by the FIX program was determined by the nature of the errors that would be encountered. Certain types of computer malfunctions demand immediate transfer of control to the FIX program. For example, if there is a parity alarm when the computer refers to its internal memory for a new instruction step or operand, further operational steps would be useless and might even destroy information. An inactivity alarm, too, will cause an immediate transfer, since to continue in this case means to continue the abnormal function. The overflow alarm can also cause an automatic branch to FIX, but this feature is designed so that the alarm may be suppressed in the operational program when it is known that overflows may occur during normal operation.

An automatic transfer of control to FIX is effected by setting the core memory parity, inactivity, and overflow alarm switches in the "active" position and the stop-branch switch in the "branch" position.

Transfers of data between core memory and magnetic drums may be monitored in another manner. Erroneous information that might be included in such a transfer cannot adversely affect the computer until used. Therefore, a drum parity alarm need not cause an immediate branch to the FIX program. Instead, the drum-parity-alarm switch is set to continue on alarm. At the conclusion of every block transfer, FIX checks the drum-parity-alarm indicator by means of a program instruction. Since the Air Defense Program was designed so that all transfers are controlled in one section of the program, the insertion of one interrogation instruction is the only modification of the operational program necessary to enable FIX to perform its entire monitoring function. If the alarm indicator is sensed inactive, there is no change in the normal sequence of events in the operational program. If, at the end of a block transfer of data into core memory, the appropriate alarm indicator is tested and found to be active, a programmed branch to the drum recovery section of FIX is effected.

- *Diagnosis*

At this point FIX will attempt to perform all diagnostic work necessary for recovery. Where time permits, FIX will also perform several diagnostic operations that are desirable for corrective maintenance studies. In all cases the results will be saved for logging and, depending on the circumstances, they may also be displayed immediately.

In the event of any type of alarm, initial FIX action would save the contents of all the computer registers such

as the accumulator, index registers, buffer register, program counter, et cetera, as they existed at the time of the error. This information is used as an aid to diagnosis, as part of the record of the error for maintenance purposes, and also enables FIX to restore the environment of the Air Defense Program prior to effecting a recovery. The type of error, such as would be indicated by a drum parity alarm or memory parity alarm, will be determined by considering the mode of entry to the FIX program and by sensing the various alarm indicators.

If there has been a memory parity alarm, FIX will refer to the program counter setting as it was at the time of the error and will locate the incorrect information that was being operated upon at that time.

When the erroneous information is referred to a second time, a second memory parity alarm may or may not be generated. Considering the case where a second parity alarm is not generated, FIX will continue its diagnosis by comparing this instruction or operand to the word that was parity-checked in the buffer register at the time of the error. If the contents of the buffer register match either the instruction or data word, FIX concludes that there was a false parity error, i.e., an error in the parity-checking circuitry itself, and that the operation was in fact completed correctly.

If, upon comparing the buffer register to the memory register, FIX finds that the buffer register was completely zero at the time of the error, this would indicate that the alarm was probably due to a failure to get a start memory pulse and that no operation had begun when the alarm was generated.

Finally, a condition may arise where the buffer register is neither all zero nor equal to the instruction or data word in memory that supposedly generated the alarm. This would indicate a memory readout failure and an incorrectly completed operation. If a second parity alarm is generated on the second reference to the instruction or operand in core memory, the error is considered genuine, and, once again, the operation could not have been completed correctly.

The results of the investigation of each memory parity error are included in a record for maintenance purposes and will also serve as a guide to proper recovery action. No diagnostic action is taken in the event of an inactivity or overflow alarm other than saving the contents of the computer registers, and recording the type of alarm and the alarm exit location for the maintenance records.

Errors incurred during the transfer of information from the magnetic drums to the central data-processing unit are treated according to the class of drum involved.

Input status drums represent the supply of new information to the computer from an external source, such as a radar site. Under ordinary circumstances, input data cannot be stored on a status drum by a program, nor is it possible to transfer the same information from a status drum to the central computer more than once. Consequently, a status drum is not normally available for complete testing and diagnosing by FIX without undue delay of the operational program.

Recovery from status-drum errors will vary according to whether the failure was transient or solid. Therefore, when a status-drum error is detected, FIX will examine the block of transferred information in core memory and, on the basis of the number of errors found, classify the failure as transient or solid. An erroneous status-drum transfer is classified as a solid failure if the number of errors contained in that transfer is more than five (an arbitrary figure). Further diagnosis for recovery and maintenance consists in determining the identity of the failing drum-input channels and the total number and frequency of similar errors.

Addressable drums serve as an auxiliary information-storage area. All data transfers to and from addressable drums are performed under program control. Addressable drums are therefore readily available for diagnosing by the FIX program.

Upon noting an erroneous transfer of this type, FIX will search the block of transferred information in core memory for the information in error. The original version of data transferred incorrectly will remain unchanged on the drum until it is deliberately replaced with new data. For this reason, when an incorrect word is found, FIX will locate the original information on the drum and repeat the transfer of the incorrect word to determine whether the error was transient or solid. If the second attempt succeeds, a correct version of the word is now properly transferred and, by a comparison of the correct and incorrect information, the exact cause of the failure may be determined and saved for logging. This process is repeated if a second word in the transfer is in error.

The timing requirements of the operational program will not permit the luxury of individually treating more than two such words solely for maintenance purposes. If more than two words in a given transfer are found to have been in error, the remainder of the erroneous transfer is repeated at once, sacrificing additional diagnostic information for increased speed in recovery.

If any one of the recovery transfers is not successful on the second attempt, the error is considered to be of a solid nature. Further diagnosis is necessary, but recovery can still be achieved. Test data of known structure may be transferred over the same channels and checked by return transfer. Since the failure is solid, these transfers will also fail, but this time the exact nature of the failure can be determined. FIX maintains a history of results obtained in this way in a "Learning Table." This table is used by FIX to compensate for future errors and also serves as a guide for corrective maintenance.

Figure 2a represents an abbreviated word, consisting of five bits plus a parity bit, which FIX has determined had been incorrectly transferred into core memory from an addressable drum. The total number of "one" bits, including the parity bit, must be odd in order to be correct. The parity alarm which identified this word for FIX was a result of a check which indicated only that an even number of "one" bits was transferred in this word. Therefore, further diagnosis is necessary to determine which bit had been modified in transit.

Figure 2b illustrates the standard method of testing the transfer channels to and from an addressable drum. By using a pattern of all "ones" and then of all "zeros," the channels may be tested for evidence of bit modification. This method, however, precludes the possibility that a unique pattern of bits in a word contributed to the failure of one particular channel. Investigation has established that failures may sometimes be uniquely associated with one word pattern and not with another. In the critical circumstances under which FIX is activated, it was felt that the extreme importance of being accurate has justified using a more detailed testing procedure than the common test pattern. FIX checks the transfer channels by using the original pattern of bits in the incorrect word as nearly as possible.

When a solid failure is detected, FIX first checks the Learning Table for a history of solid errors on this particular drum. If such records exist, FIX checks each transfer channel indicated by the Learning Table as having failed before. Figure 2c illustrates a channel being tested in this manner. A bit which was suspected of having been lost in transit, is changed to a "one" in core memory, and the entire word is then transferred to and from the drum to test this channel. If this bit fails to return as it was sent out, in this case as a "one," this discrepancy is recorded. The Learning Table search will usually take no more than about 50 milliseconds. At the completion of the Learning Table test, if only one bit is found to be erroneous in this word, the results will be used to effect a recovery. If the Learning Table examination is not fruitful, recovery may still be achieved by further diagnosis.

In this event the next step would be for FIX to conduct a complete examination of the word. This is basically the same as the Learning Table test, except that the entire word is tested for evidence of failure instead of only those channels indicated by the Table. Each bit in succession is complemented, transferred to and from the drum along with the rest of the word, tested for evidence of modification in transit, and restored to its original state. Any bit that fails to return in the same form as it was sent out is recorded. The bit-by-bit findings are accumulated until the end of the examination, which takes about one second. If the complete examination discloses that only one bit has failed in this word, the results will be used to effect a recovery.

• *Logging*

All information gathered by the FIX program will be either printed on the teletype monitor, displayed immediately, or recorded in the Learning Table. The recovery that will follow is meant to improve the system reliability, not to shield equipment failures. If failures were not logged when recovery is achieved, the equipment could deteriorate with age until, without warning, catastrophic failure occurred.

During any alarm condition, where automatic transfer occurs, FIX saves the current contents of all the computer registers for logging. The various alarm indicators will be tested to determine which type of alarm occurred. This information, together with the identity of the operational routine interrupted by the alarm, the data that was being processed at that time, and the details of the error as diagnosed, will be logged on the teletype printer immediately after the error occurs. A record of the number of such errors will also be maintained on a display.

Status- and addressable-drum errors are internally recorded in the Learning Table and are also displayed as they occur. Each time a status-drum error occurs, the drum field in error and its input channel are recorded and displayed, together with the total number of such errors recorded up to this time. The record and display for addressable-drum errors includes the drum field, failing transfer channel and the nature of the failure, i.e., whether the erroneous bits were "ones" or "zeros," whether the failures were solid or transient, and the total number of such errors.

Enough pertinent information concerning each failure incident is logged to permit maintenance study teams to attempt to duplicate the trouble and to keep detailed statistics on the reliability of the circuits in the system. Maintenance personnel can resolve machine difficulties only if this kind of logging is done. As experience is gained, the difficulties will recur less frequently, since equipment and program design improvements will be suggested by the statistics.

• *Recovery*

The function of the recovery sections of FIX is to perform all operations necessary to restore control of the computer to the operational program. The choice of the method depends on the following factors:

a) The nature of the interrupting malfunction
b) The results of the diagnosis
c) The time spent in detecting and diagnosing
d) The number and frequency of this type of failure incident.

Some failure incidents in the central computer may be rectified by restoring the contents of the computer registers and internal memory to their original values, as saved by the alarm monitoring and control sections of FIX, and then transferring computer control back to the operational program at the point of interruption. This recovery method is most efficient, requiring up to about 30 milliseconds, and is used, wherever possible, in the case of memory parity errors.

If diagnosis indicated that the operation was completed correctly, as in the case of a false parity error, the environment of the Air Defense Program will be restored, and recovery will be effected by reinitiating operations with the next program step following the one that operated at the time of the error. The exception to this would be if the interrupted instruction involved a transfer operation, in which case program control would be returned to the same transfer instruction.

If the instruction was never completed correctly, as in the case of a failure to get a start memory pulse, or a memory readout failure, recovery will be attempted by

Figure 2a **Even parity count indicates word incorrectly transferred, but does not indicate which bit failed.**

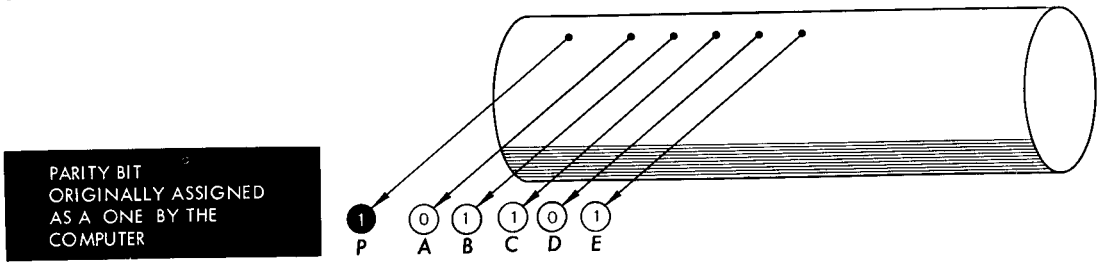


Figure 2b **Standard test pattern technique for checking transfer channels. Comparing *before* and *after* words discloses the discrepancy.**

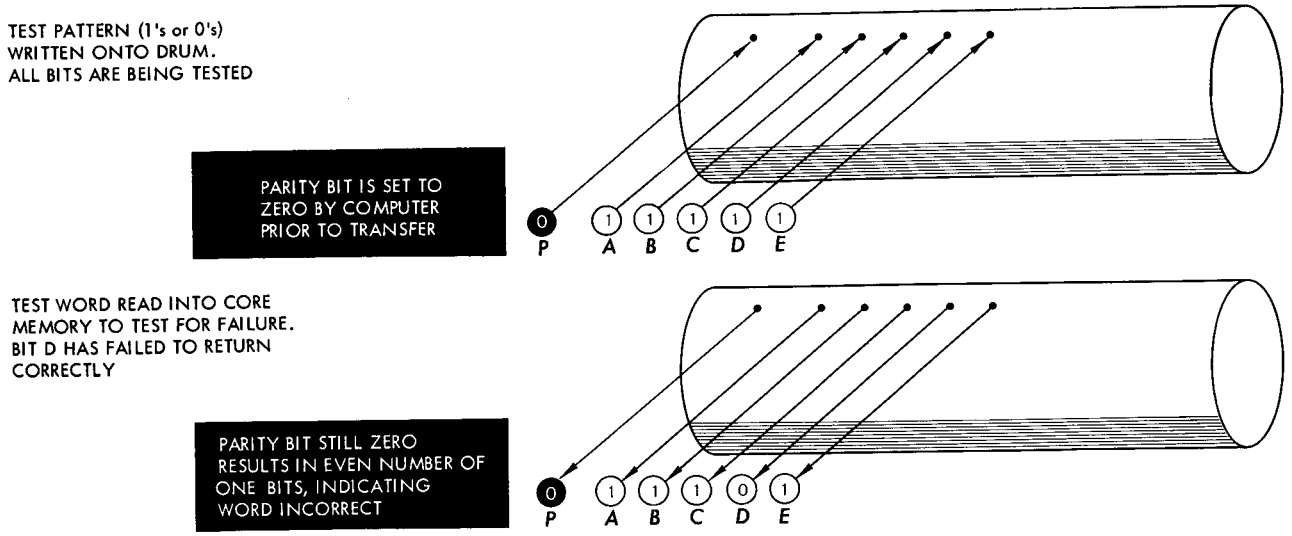
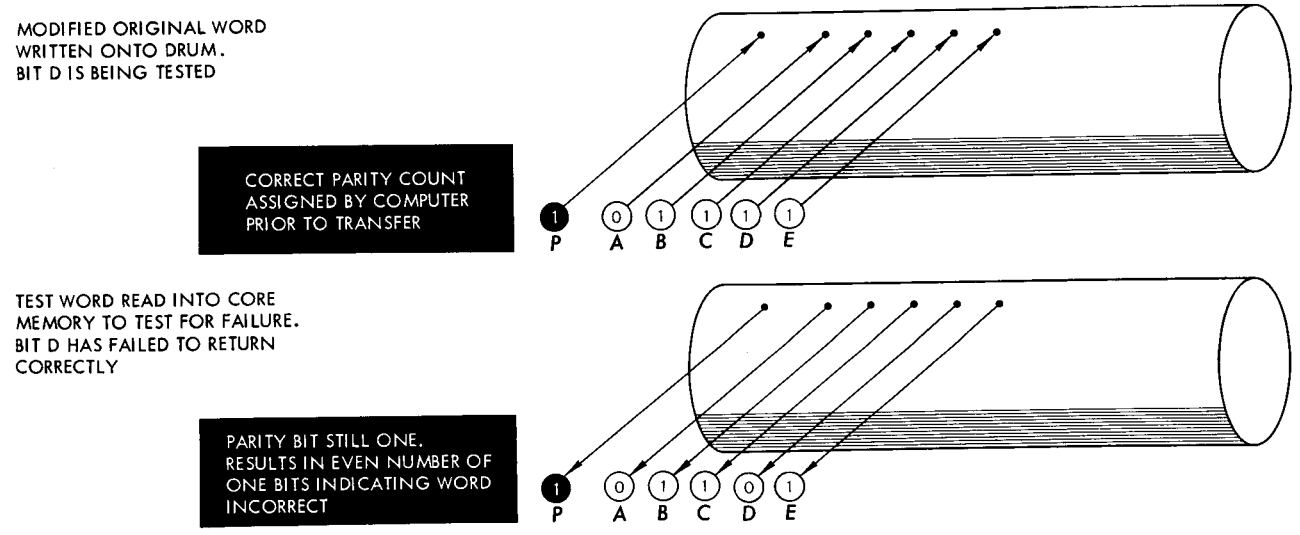


Figure 2c **FIX technique using original word (Fig. 2a) to check one channel at a time. The bit discrepancy is double-checked by *before* and *after* comparison.**



reinitiating the Air Defense Program with the instruction that originally failed.

When an error is diagnosed as a genuine memory-parity error, the erroneous word must be corrected in core memory before the operational program can be resumed. At the present time FIX cannot automatically regenerate a correct version of a faulty word in core memory. However, since all of the program instructions and most of the data that is used by the Air Defense Program is stored on magnetic drums, FIX will attempt to locate a correct copy of a word in auxiliary storage and substitute it for an erroneous word in core memory. Whenever this is possible, recovery will then be achieved by reinitiating the Air Defense Program with the correct version of the instruction that originally failed.

If any of these methods are not successful in achieving recovery, i.e., if the identical failure is immediately encountered, or if these methods are not feasible, as in the case of inactivity or overflow errors, an alternate method of recovery is available.

The Air Defense Program consists of a series of subprograms, each of which operates in its turn upon the latest input data fed to it. The entire process is an iterative one. After the last program has been completed, the first program will be called on to repeat its function upon the latest available data.

When an operation cannot be resumed at the point of interruption, recovery can often be achieved by reinitiating the Air Defense Program at the beginning of a cycle or frame of operations, so that completely new input data may be processed. The startover procedure takes approximately five seconds. Thus, an abnormal condition which was the result of a transient failure will not degrade the performance of the system. A more serious or solid failure in the central computer will, of necessity, cause repeated restarts of the operational program. It is left to the discretion of the operator to impose limits on the number or frequency of the attempts to recover in this manner.

FIX does not distinguish between central computer errors that occur while the Air Defense Program is operating and those that occur during FIX operation. If a second error is encountered while one error is being corrected, the later error will take precedence. FIX will attempt to correct a central computer error within itself in exactly the same manner as an error occurring within the Air Defense Program. The original error, however, will then be disregarded, recovery of the Air Defense Program being achieved via startover. Of course, a serious error in a vital section of FIX will preclude automatic recovery, and manual intervention will be necessary.

Recovery from transient status-drum errors is achieved in another manner. In the event of a status-drum failure, the testing procedure is limited by the fact that there is no practical way to make an experimental transfer between the central computer and a status drum without a prolonged interruption of the Air Defense Program. Since FIX cannot determine exactly what failed in this type of transfer, it is not possible to estimate what the data should have been.

The solution is to render the erroneous data harmless by eliminating it from core memory, adjusting the Air Defense Program's records to compensate for the decrease in the amount of input information to be processed, and then returning to the logical operation in the main program that would normally follow the status-drum transfer. The momentary loss of some input data to the computer from a radar site, for example, will have no more effect on the Air Defense Program than would the slight interruption of radar fixes that are expected to occur during normal operation. Such temporary losses, or "miss fixes," as they are commonly called, are not unusual and the Air Defense Program provides for this by extrapolating or filling in for missing radar fixes when they occur. In this manner, an accurate plot of the velocity of a hostile ship can easily be maintained despite the fact that a few positional fixes are missing, if the available fixes are dependably correct. A much smaller number of incorrect fixes can destroy the accuracy of a coarse plot if no means is provided to prevent these fixes from being included in the plotting computations and, therefore, the elimination of incorrect input data is much more desirable than treating such information as valid.

FIX cannot allow the situation to continue where a large number of consecutive errors reduces the flow of information to the Air Defense Program below an acceptable minimum. An excessive number of errors in one status-drum transfer will require that recovery be attempted by reinitiating the Air Defense Program at the beginning of a new frame of operations so that new input data can be called for and processed.

A dynamic display of all facts which are pertinent to this type of error is up-dated each time an error occurs. By observing the display, maintenance men may be able to determine the input channel from which most of the errors are coming. They may thus be able to eliminate or reduce the quantity of status-drum errors by substituting a spare input channel without interrupting the operational program.

Recovery from addressable-drum failures may be achieved by modifying a bit in the incorrect word or words in core memory according to the nature of the failure. Transient errors are corrected during the diagnosing operations.

In a solid failure during the transfer of a block of information to core memory from an addressable drum, many words may be expected to have transferred incorrectly. In an operational period, time does not permit that FIX be allowed to diagnose each word before correcting it. When FIX is satisfied that the Learning Table test or the complete examination has disclosed the failing transfer line for one word, it will use this information to correct this word and the remaining words in error in the same transfer. In Fig. 3a, let us say that Words 1 through 10 were incorrect in the block of transferred information shown. Suppose that a complete test of Word 1 indicated that Channel *A* had failed, and that the bit in Position *A* should have been a "one". After correcting Word 1 in core memory as shown in Fig. 3b, FIX would continue

to examine the remaining incorrect words. If a check of bit position *A* in each incorrect word indicates that it was possible that the identical error occurred in each incorrect word, these words would also be corrected in the same manner as Word 1 (see Fig. 3b, Words 2 through 7).

In an actual transfer, several thousand words might be involved. If a solid failure occurred, the number of words in error could be expected to be quite large. Consequently, if the error initially found in the complete test did *not* apply to all of the incorrect words in a transfer, it is felt that this fact would soon be obvious. Continued examination of the remainder of the incorrect words should reveal at least *some* words which could *not* have failed in the same manner. In Fig. 3a, Words 8 and 10 now contain a "one" in bit Position *A*. Therefore, that bit could not have dropped during the original transfer. This would indicate that another channel had failed in transferring these words, and might also have failed during the transfer of any of the previously "corrected" words. This inconsistency would invalidate the "corrections" made earlier to this transferred data. In such a situation, recovery is effected by restarting the operational program at the beginning of a frame, so that new input data may be processed. If more than one channel is found to have failed, this information will be immediately logged prior to initiating a startover of the Air Defense Program. The maintenance men may then determine whether or not it will be possible to permit the Air Defense Program to continue to process new input information.

The main section of the FIX program is also stored on an addressable drum. If an error is encountered in reading FIX into core memory, the diagnostic and repair sections cannot be used to correct this error. Instead, the permanent FIX routine in core memory will initiate a startover of the Air Defense Program. If this technique is unsuccessful for recovering from an error, manual intervention will be necessary.

• *Results*

After a six-month study of failures during 87 missions of the Air Defense Program of the SAGE Computer, it was determined that the causes for failures were distributed as follows:

Drum parity (status and addressable)	53%
Memory parity	10%
Inactivity	21%
Miscellaneous (power failures, stopped by operator, et cetera)	16%

On the basis of these results, FIX theoretically is capable of automatic recovery from 84% of all failures occurring in the period studied.

Shortly after this study FIX was employed for an extended number of evaluation missions on the SAGE Computer.

During this large trial period of computer operation, FIX provided automatic recovery from more than 92% of the failures. Of the remaining errors, only about 2% required unscheduled maintenance, the other 6% being due to operator and program errors.

It appears that the inclusion of FIX in the tests increased computer efficiency. In addition to the improvement in system performance achieved with FIX, a long-term gain should be realized in basic equipment reliability and useful operational time because of the increased accuracy of the maintenance information supplied by FIX.

	INFORMATION BITS					
PARITY BIT	A	B	C	D	E	
1	1	1	0	1	0	10
1	0	0	0	0	1	9
0	0	0	1	0	0	
1	1	0	0	1	1	8
0	0	0	1	0	1	7
1	0	0	0	0	0	
0	0	1	1	1	1	6
0	0	1	0	0	1	5
1	0	0	1	1	1	4
1	0	0	1	1	0	
1	0	1	0	1	1	3
1	0	1	1	1	0	2
0	0	1	0	0	0	
1	0	1	1	0	1	1
0	0	0	0	1	0	
1	0	1	1	0	0	

Figure 3a **A portion of a block of data transferred into core memory from an addressable drum. Words numbered 1 through 10 were incorrectly transferred (note the even parity count in each erroneous word). An examination of Word 1 indicated that bit *A* had dropped in transit. This bit would be corrected in Word 1 *and all other incorrect* words which could have failed in the same manner.**

	PARITY BIT	INFORMATION BITS A	B	C	D	E
10	1	1	1	0	1	0
9	1	0	0	0	0	1
	0	0	0	1	0	0
8	1	1	0	0	1	1
7	0	1	0	1	0	1
	1	0	0	0	0	0
6	0	1	1	1	1	1
5	0	1	1	0	0	1
4	1	1	0	1	1	1
	1	0	0	1	1	0
3	1	1	1	0	1	1
2	1	1	1	1	1	0
	0	0	1	0	0	0
1	1	1	1	1	0	1
	0	0	0	0	1	0
	1	0	1	1	0	0

Figure 3b **Bit position *A* has been corrected in words numbered 1 through 7. The inconsistency in Word 8 prevented further attempts at recovery in this transfer.**

Table 1 **Summary of FIX action for each type of error.**

Type of Error	*Recovery Procedure*
False memory parity: within Air Defense Program and/or FIX.	Reinitiate program with next instruction following the one that operated at the time of the alarm.
Fail to get start memory pulse, or other memory readout failure: in Air Defense Program and/or FIX.	Reinitiate program by repeating instruction that operated at the time of the alarm.
Genuine memory parity: in Air Defense Program and/or FIX.	Replace incorrect word with good copy from auxiliary storage drum and reinitiate program by repeating instruction that operated at the time of the alarm.
Inactivity, overflow, genuine memory parity that cannot be corrected, solid memory parity: within Air Defense Program and/or FIX. Solid status-drum failure (an excessive number of errors in one transfer) in Air Defense Program only. Drum parities while bringing in FIX: FIX only.	Reinitiate Air Defense Program at the beginning of a new frame of operation. FIX imposes no limits on the number of "startovers" that may be initiated in attempting to recover. The operator must determine if an excessive number of restarts is cause for manual intervention.
Transient status-drum parity: in Air Defense Program only.	Eliminate erroneous information from core memory, adjust records of Air Defense Program and continue operations at point following transfer.
Transient addressable-drum parity: in Air Defense Program only.	Repeat transfer. Reenter Air Defense Program at point following transfer operation.
Solid addressable-drum parity: in Air Defense Program only.	Correct erroneous words in core memory. Restart Air Defense Program at point following transfer. If diagnosis is inconclusive for purposes of correcting error, restart the Air Defense Program at the beginning of a new frame of operation.
Any catastrophic error from which FIX does not successfully recover.	Manual intervention.
Errors which do not result in memory parity, drum parity, inactivity or overflow alarms.	None; FIX not activated except by the alarm circuitry of the computer.

In summary, the advantages offered by FIX are these: Recovery from most errors is accomplished automatically, almost immediately, and accurately, thus assuring the correct results with a negligible amount of lost operational time. Table 1 is a summary of FIX action for each type of error.

The Learning Table record permits more immediate correction of some errors the second time they occur. Solid errors in addressable-drum transfers, for example, have been virtually eliminated as a source of reduced computer efficiency.

The logging feature of FIX affords a detailed record of all errors exactly as they occurred, as an improved aid to corrective maintenance.

Although the advantages and results obtained so far have been limited to one specific application in the SAGE System, it is felt that variations of the FIX concept can be successfully applied to other operational and production programs written for a data processing system.

We wish to acknowledge the cooperation of those who assisted with the first trials of the FIX program, which were made at Lincoln Laboratory, Lexington, Mass.

References

1. C. J. Swift, "Machine Features for a More Automatic System for Digital Computers," *Journal of Association for Computing Machinery*, **4**, No. 2, 172 (April 1957).
2. J. H. Brown, J. W. Carr III, L. Boyd and J. R. McReynolds, "Prevention of Propagation of Machine Errors in Long Problems," *Journal of Association for Computing Machinery*, **3**, No. 4, 348 (October 1956).
3. R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, **29**, 60 (1950).

Revised manuscript received October 7, 1958