Chapter 27

The STAR (Self-Testing And Repairing) Computer: An Investigation of the Theory and Practice of Fault-Tolerant Computer Design[1]

Algirdas Avižienis / George C. Gilley / Francis P. Mathur / David A. Rennels / John A. Rohr / David K. Rubin

Summary This paper presents the results obtained in a continuing investigation of fault-tolerant computing which is being conducted at the Jet Propulsion Laboratory. Initial studies led to the decision to design and construct an experimental computer with dynamic (standby) redundancy, including replaceable subsystems and a program rollback provision to eliminate transient errors. This system, called the STAR computer, began operation in 1969. The following aspects of the STAR system are described: architecture, reliability analysis, software, automatic maintenance of peripheral systems, and adaptation to serve as the central computer of an outerplanet exploration spacecraft.

Introduction: Chronology and Rationale

This paper presents a summary of the theoretical results and design experience obtained in an investigation of fault-tolerant computing which is being conducted at the Jet Propulsion Laboratory (JPL). Initial studies (1961–1965) led to the conclusion that dynamic (also called standby) redundancy offered the greatest promise in the design of fault-tolerant digital computer systems [Avižienis, 1967a]. The *dynamic* redundancy [Short, 1968] approach requires a two-step procedure for the elimination of a fault: first, the presence of a fault is determined; second, a corrective action is taken (e.g., replacement of failed unit, repetition of program, reconfiguration of systems, etc.). The alternative to the dynamic approach is *static* (masking) redundancy [Short, 1968], which was already being utilized in existing component-redundant [Lewis, 1963; Kuehn, 1969] and triple-modular-redundant (TMR) [Kuehn, 1969; Anderson and Macri, 1967; Lyons and Vanderkulk, 1962] computers. Early analytic studies of dynamic redundancy with idealized series-parallel system models indicated that mean life gains of an order of magnitude and more over a nonredundant system could be expected from dynamically redundant systems with standby spares replacing failed units [Reed and Brimley, 1962; Kruns, 1963; Flehinger, 1958; Griesmer, Miller, and Roth, 1962]. This gain compared favorably with the mean life gain of less than two in the typical TMR systems. Other qualitative advantages of the dynamic over the static redundancy were: (1) greater isolation of catastrophic (non-independent) faults which is especially important for densely packed microelectronic circuitry; (2) survival of system until all spares of one type are exhausted; (3) ability to eliminate errors which are caused by transient faults by the use of program rollback; (4) ready adjustability of the number and type of spare units; (5) utilization of the potentially lower failure rate of unpowered components in spare units; (6) avoidance of the circuit-related problems of static redundancy: increases in fan-out, fan-in, power requirements, and the need for isolation and synchronization of separate channels; and (7) facilitation of the check out of spare units by means of standard diagnostic programs.

The attainment of the apparent advantages of a dynamically redundant system had been shown to depend very strongly on the successful execution of the detection and replacement operations [Flehinger, 1958; Griesmer, Miller, and Roth, 1962]; these observations have since been formalized as the concept of "coverage" [Bouricius, Carter, and Schneider, 1969].

The second phase of the investigation (1965–1970) was focused on the identification and solution of the problems involved in the design of a general-purpose digital computer possessing the properties attributed to the abstract model of a dynamically redundant computing system. Three major areas of investigation were: (1) an investigation of fault-detection methods; (2) a study of computer architecture with emphasis on partitioning into subsystems with minimal interconnection requirements; and (3) a study of the "hard-core" problem, i.e., the alternate technologies and logic organizations for implementing the detection and switching functions. The choices among feasible alternatives in all three areas are strongly affected by assumptions on the available component technology and on the computing tasks to be required of the computer. In order to retain contact with the practice of computer design, it was decided to design and construct an experimental general-purpose digital computer which would incorporate dynamic redundancy (i.e., fault detection and replacement of failed subsystems) as integral parts of its structure. The design objectives have been carried out and the system, called the STAR (self-testing and repairing) computer, began operation in 1969. The modular nature of the STAR computer has allowed systematic expansion and modifications that are still being continued.

The first objective of the design is to study the class of problems which are encountered in transforming the theoretical model of a self-repairing system into a working computer. State-of-the-art

[1]*IEEE Trans. on Computers*, vol. C-20, no. 11, November 1971, pp. 1,312–1,321

integrated circuit and memory technology was employed in the design. The STAR computer characteristics were chosen to satisfy all predictable requirements of a spacecraft guidance, control, and data acquisition computer which would be used in the very long (ten years and more) unmanned missions exploring the outer planets of the solar system [Long, 1969]. The second objective was to provide a tool for laboratory studies of fault-tolerant computing, including the injection of transient as well as permanent faults of catastrophic nature. Very extensive displays of registers, manually controlled clocking, and provisions for convenient modification of subsystems were incorporated into the experimental STAR computer breadboard (Fig. 1).

The STAR computer employs a balanced mixture of coding, monitoring, standby redundancy, replication with voting, component redundancy, and repetition in order to attain hardware-controlled self-repair and protection against transient faults. The principal goal of the design is to attain fault tolerance for a variety of faults: transient, permanent, random, and catastrophic. The actual construction (rather than simulation) of the STAR breadboard has two significant advantages. First, the design process has uncovered interesting new hardware-related problems and led to numerous improvements. Second, the computer serves as a vehicle for further experimentation and refinement of the recovery techniques.

Fig. 1. The STAR computer.

During the studies of fault-tolerant architecture and the design of the STAR computer, concurrent investigations were being conducted in other closely related areas of fault-tolerant computing, including studies of software, reliability prediction, and extension of dynamic redundancy to peripheral devices [Avižienis et al., 1969]. A complete redesign of the STAR computer is being performed to match the exact requirements of a control computer for the thermoelectric outer planet spacecraft (TOPS) [Astronaut., 1970]. This effort led to the evaluation of additional fault-recovery techniques. The results of the efforts described above are summarized in the following sections of this paper.

Architecture of the STAR Computer

Methods of Fault Tolerance

The STAR computer is a replacement system that provides one standard configuration of functional subsystems with the required computing capacity. The standard computer is supplemented with one or more spares of each subsystem. The spares are unpowered and are used to replace operating units when permanent faults are discovered. The principal methods of error detection and recovery are the following.

1. All machine words (data and instructions) are encoded in error-detecting codes and fault detection occurs concurrently with the execution of the programs.
2. The computer is divided into a set of replaceable functional units containing their own instruction decoders and sequence generators. This decentralization allows simple fault location procedures and simplifies system interfaces.
3. Fault-detection, recovery, and replacement are carried out by special-purpose hardware. In the case of memory damage, software augments the recovery hardware.
4. Transient faults are identified and their effects are corrected by the repetition of a segment of the current program; permanent faults are eliminated by the replacement of faulty functional units.
5. The replacement is implemented by power switching: units are removed by turning power off and connected by turning power on. The information lines of all units are permanently connected to the buses through isolating circuits; unpowered units produce only logic "zero" outputs.
6. The error-detecting codes are supplemented by monitoring circuits which serve to verify the proper synchronization and internal operation of the functional units.
7. The "hard core" test and repair processor (TARP) is protected by triplication and replacement of failed members of the triplet.

Hardware System Organization

The block diagram of the STAR computer is shown in Fig. 2. Communication between the units is carried out on two four-wire buses: the memory-out (M-O) bus, and the memory-in (M-I) bus. The abbreviations designate the following units.

COP	Control processor, contains the location counter and index registers and performs modification of instruction addresses before execution.
LOP	Logic processor, performs logical operations on data words (two copies are powered).
MAP	Main arithmetic processor, performs arithmetic operations on data words.
ROM	READ-ONLY memory, 16,384 permanently stored words.
RWM	READ-WRITE memory unit with 4096 words of storage (at least two copies powered; 12 units are directly addressable).
IOP	Input/output processor, contains I/O buffer registers.
IRP	Interrupt processor, handles interrupt requests.
TARP	Test and repair processor, monitors the operation of the computer and implements recovery (three copies are powered).

The functional units (processors and memories) of the STAR computer communicate by means of the M-I and M-O (four-wire) information buses. The 32-bit words are transmitted on these two buses as eight bytes of four bits each. Three control signals are sent from the TARP on the three-wire control bus to synchronize the operations of the functional units and to initiate recovery. Otherwise the functional units operate autonomously. Unless otherwise noted, one copy of each unit is powered at a given time. The decentralized organization allows a standard interface between each unit and the remainder of the computer. Each STAR unit interfaces with the computer by the means of 14 signal lines. Eleven lines, both in active and spare units, are permanently connected to the computer system buses, and three are connected to the TARP array. An unpowered unit cannot produce logic one outputs. The external connections of a STAR unit are shown in Fig. 3.

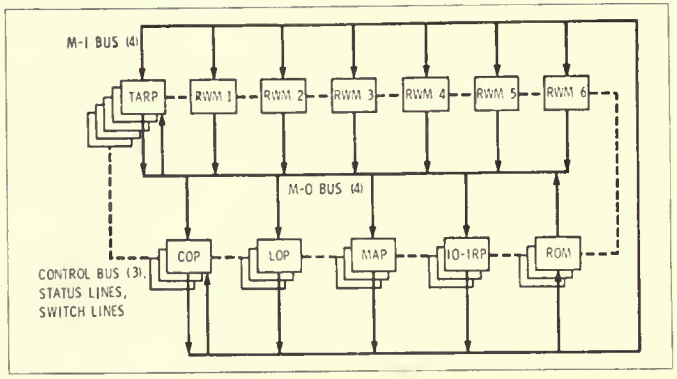

Fig. 2. STAR computer organization.

The four input and four output lines are connected to the data M-I and M-O buses. They receive and send coded machine words in four-bit bytes. The power switch control input causes power to be applied to the unit. The three control bus input signals are: CLOCK, a basic timing input: SYNC, a periodic synchronization signal; and RESET, a signal that forces the unit into a standard initial state. Two unit status lines send information on the internal operation of the unit to the TARP. These lines carry multiplexed information which will be discussed in a following section. Each functional unit is autonomous and contains its own sequence generator as well as storage for the current operation code, operands, and results. The internal design of a unit may be altered without affecting other units as long as the interface specifications are observed.

It is to be noted that the IOP and IRP units are shown combined in Fig. 2.

Standard Operation

The STAR computer has two modes of operation: the standard mode and the recovery mode (under TARP control). During the *standard mode* the stored programs are carried out. The TARP processor issues the principal CLOCK signal and SYNC signal which occurs when a new step is initiated in the execution of an instruction. Ten CLOCK periods form the basic time unit (cycle) of the computer. During the first period, a four-bit "step-code" (in 2-out-of-4 encoding) is issued by the TARP to the M-O bus. The next eight periods are employed to transmit or manipulate one eight-byte machine word. During the tenth period a four-bit "condition-code" byte may be broadcast by one of the functional units. The ten-period cycle is needed because of the series-parallel organization of the computer.

One instruction is executed in two or three steps. In the first step, the address of the instruction is sent from the location

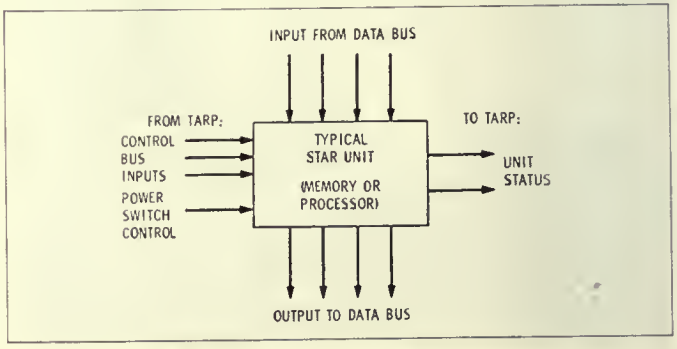

Fig. 3. Functional unit of STAR computer.

counter in the COP to the memory (ROM and RWM) units. In the second step, the addressed memory unit broadcasts on the M-O bus the operation code and address of the instruction to all functional units. The address is indexed in the COP which transmits it to the M-1 bus if necessary. The appropriate units recognize the operation code, store the address, and initiate execution. In the third step the instruction is executed: an operand is placed on the appropriate bus and accepted by the destination unit. The first two steps require one cycle each; the duration of the third step depends on the instruction and requires 0, 1, or more cycles. Program interrupts begin without the first step. During the second step an instruction is broadcast by the interrupting unit (IO-IRP or TARP).

The instruction set consists of 180 single-address instructions, about one-third of which are indexable. It includes fixed-point arithmetic, maskable logic, and shift operations. Loop-facilitating and subroutine link register instructions are provided. There are 28 interrupts which can be masked out and tested under program control. A special class of instructions aids in fault tolerance. They include diagnostic instructions which exercise unit status messages and the fault-location logic in the TARP. Others perform updating of the "rollback" register in TARP units, name assignment and cancellation of RWM units, power control of spare units, duplexing of ROMs and processors, and absolute read or write operations in RWM units.

Computer Words: Formats and Encoding

There are two possible effects of logic faults upon the operation of a digital computer. First, a data word or an instruction word may be altered during storage, transmission, or processing. The effect is a *word error*. Second, during the execution of an instruction a processor or a memory module may act incorrectly, act out of turn, or fail to act at all. The effect is a *control error*. Both classes of errors are detected in the STAR computer. The present section considers coding techniques for word error detection; control errors are considered later.

Complete duplication offers the simplest word-error detection at the highest cost. Low-cost arithmetic error-detecting codes [Avižienis, 1967b] are attractive because they are preserved during arithmetic processing and mandatory duplication of an arithmetic processor is avoided. An intensive study of error codes led to the choice of modulo 15 arithmetic checking which is especially effective for a byte-organized computer with four-bit bytes [Avižienis, 1971].

All words in the STAR computer are encoded as shown in Fig. 4. The 32-bit numeric operand word [Fig. 4*b*] consists of the 28-bit binary number b, and a 4-bit check byte $c(b)$. The check byte is a binary number which has the value

$$c(b) = 15 - |b|_{15}$$

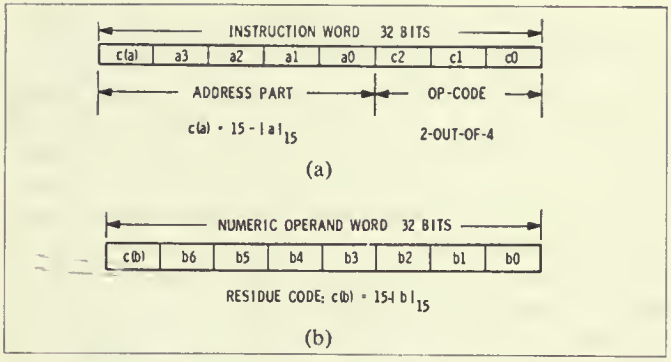

Fig. 4. *(a)* STAR instruction word format. *(b)* STAR operand word format.

where $|b|_{15}$ means "the modulo 15 residue of b." This check byte causes the 32-bit word to be a multiple of 15. The checking algorithm casts out 15s, that is, it computes the modulo 15 residue of the entire coded word. A zero residue, represented by 1111, indicates a correct word; all other values of the residue indicate a fault. The casting out 15s is implemented with a four-bit "end-around carry" adder and takes place concurrently with the transmission of a word on the bus.

The 32-bit instruction word [Fig. 4*a*] consists of a 12-bit operation code and a 20-bit residue-coded address part. The 16-bit address is encoded in the same residue code as the operands, and the same checking algorithm is used. The operation code is divided into three bytes, and each byte is encoded in a 2-out-of-4 code. This code permits each byte to be checked individually. There are six valid forms of each byte, giving a total of 216 valid op-code variants. The structure of a bus checker circuit which performs word checking is shown in Fig. 5. The single step-code and condition-code bytes also use the 2-out-of-4 code and are checked by the bus checker.

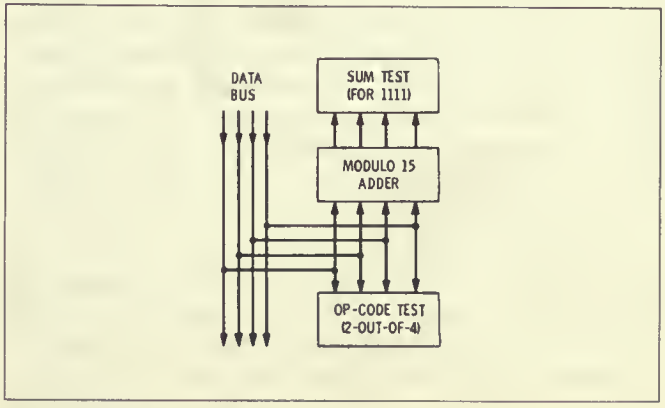

Fig. 5. The bus checker circuit.

The initial choice of error codes in the STAR computer emphasized variety for the purpose of comparison and evaluation, and the arithmetic product (or AN) code was used for operands [Avižienis, 1967b]. Two reasons for the change to the present encoding of operands were: (1) the residue code is separable and allows the use of the more efficient two's complement algorithms for binary arithmetic, and (2) multiple precision and floating-point arithmetic is much more readily implemented with residue encoding. Residue encoding is also suitable for operation codes in STAR instructions. Its advantage is that an identical checking algorithm is applied to instructions and operands; an explicit identification is not required for checking, and loading of programs is facilitated. The drawback is that the bytes of the op-code cannot be checked individually as in the 2-out-of-4 coding.

Control Error Detection

It has been observed that a large number of faults which cause control errors also cause word errors and are detectable by the use of error codes. Some critical control errors, however, do not fall into this category and require other methods of detection.

The principal method of control fault detection in the STAR computer is the validation that every unit is active at the proper time and that the proper algorithm is carried out within the unit. The initial design [Avižienis, 1968] used a four-wire status line for every replaceable unit to transmit one of six possible "2-out-of-4" coded status messages. Experience has shown that the diagnostic logic in the TARP is significantly simplified when status messages are conveyed to the TARP at predetermined clock times within each ten-unit cycle of operation. In the revised design, each status message is conveyed on two wires (in 1-out-of-2 encoding) and each message covers the time interval between two messages of the same type. The status-message originating circuits are duplicated in each unit to allow the detection of a fault in the status message.

The "output active" message indicates that the unit has produced a nonzero output to the bus in the preceding time interval. It serves to identify improperly active units which otherwise would destroy the information being transmitted on a bus, and make it impossible to locate the source of error. The absence of an expected active message is also a fault condition, since the all-zero word is not a validly coded operand or instruction. The checking of output activity is the most critical of all status monitoring functions.

The other status messages are multiplexed and sent over the same pair of wires as the output active messages because the activity information is not required continuously in the byte-serial machine structure. The status messages which are listed below aid in increasing the probability of immediate detection of incorrect operation.

The "disagree with bus" message is needed for duplex operation (discussed in the next section). Two identical units produce outputs to a bus which acts as an OR gate. Each unit compares the bus word to its internally held output word and records a disagree message if a mismatch occurs. The message is conveyed to the TARP at a specified time. The bus checker result together with disagree message permits a rapid identification of a faulty unit. In simplex operation this message helps to identify improper activity of another unit.

The "complete" message is essential for functional units which have variable-duration algorithms. Memory units issue "write complete" and "read complete" messages which are essential for immediate detection of incorrect storage events.

The "internal fault" message is produced by internal monitoring circuits within each unit. Its function is to indicate incorrect internal algorithms detected by duplication of critical signals, special test circuits, and "inverse microprogramming" in which an operation is deduced from active gating signals.

In addition to the above listed four types of messages, time is provided for a "special" status message which varies for different units. For example, the IO/IRP uses it to report to the TARP the arrival of an external interrupt request.

Properties of Functional Units

The main arithmetic processor (MAP) input consists of an operation code followed by a coded operand, and the output is a coded result followed by a condition-code byte, indicating either one of three singularities (sum overflow, quotient overflow, zero divisor) or the type of a good result (positive, zero, negative). The control processor (COP) stores the condition code and uses it to implement conditional branches instructions. The COP also contains the location counter LC, two index registers, and a four-bit adder to implement indexing of residue-coded addresses and incrementing the LC. The logic processor (LOP) performs the bit-by-bit logic operations and code conversions on input words. The arithmetic coding is removed from the operand before the operation, since error codes are not preserved during logic operations, and the final result is again encoded. The LOP operation is checked by operating two copies which issue disagree status messages when their outputs differ. The IO/interrupt processor (IO/IRP) receives external interrupt requests, initiates allowable interrupts, and carries out input/output buffering functions.

The READ-ONLY memory (ROM) contains the permanent programs and the associated constants. The present machine uses a "braid" assembly of transformers and wires for the permanent storage of 16,384 words. Complete replicas of the ROM are used as replacements. Each 4096 word READ-WRITE memory (RWM) unit has two modes of operation. In the *absolute* mode a RWM unit recognizes its own wired-in absolute name. In the *relocated* mode a RWM unit responds to an assigned name. All relocated

units with the same assigned name store and read out the same locations simultaneously. In case of a disagreement with the word on the M-O bus, the RWM unit sends a disagree status message to the TARP. The relocated mode provides duplicate or triplicate storage for critical programs and data. When a RWM unit fails, its replacement unit can be assigned the same name, avoiding a discontinuity in addresses. Assignment and cancellation of assigned names is performed under program control; this provision allows selective redundancy of storage. A record of RWM name assignments is retained (in nonvolatile storage) in all active TARP units. The accessing of storage locations within a RWM unit is checked by permanently storing the 4-bit check byte of its 12-bit internal address in every location. This byte is read out and checked against the contents of the address register during every read and write operation.

In the STAR computer only the logic processor and the RWM memory unit containing critical system programs are duplexed for normal operation. For experimentation, complete provisions have been made for optional duplex operation of all memory and processor units under program control. The combination of duplication and coding offers detection of all errors as well as a fast identification of one faulty unit. In order to permit duplex operation of processor and ROM units, active TARP units hold a record of units which are operating in duplex.

The Test and Repair Processor (TARP) and Recovery Mode

The "hard core" monitor of the STAR system is designated as TARP (test and repair processor) in Fig. 2. The TARP monitors the operation of the STAR computer by two methods: (1) testing every word sent over the two data buses for validity of its code; and (2) checking the status messages from the functional units for predicted responses. An incorrect word or a deviation from predicted response causes an interruption of normal computing and an entry into the recovery mode of operation. The block diagram of one TARP is shown in Fig. 6. It is functionally divided into two sections. One section provides standard mode machine control and fault location, and the other controls the recovery mode operation and effects the switching of replaceable units.

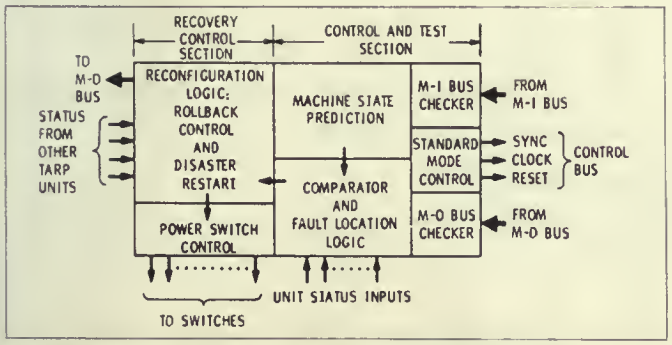

Fig. 6. Test and repair processor (TARP) organization.

The Control and Test (CAT). This section contains the standard mode control logic consisting of an op-code decoder, a clock, and a counter which generates the step-code signals for standard mode operation. The machine-state prediction logic uses the current instruction and step-code to predict which status messages should be received from each powered functional unit. It also predicts the information source and the type of encoding expected on each bus. The fault location logic compares the status and bus checker (Fig. 5) results to the prediction. In most cases, it can localize an error to a particular functional unit. Upon detecting an error, the CAT section stops the machine and transfers its error information to the recovery control section.

Recovery Control (REC). This section of the TARP contains a "rollback point" address register which specifies the location of the instruction at which normal operation is to be resumed after a recovery. This register is updated under program control. Before every updating, the contents of all processor registers needed for recovery are stored in duplexed memory units. Upon receipt of an error message from the CAT section, the REC section issues the "reset" signal, which causes all powered units to be set to an initial state, and then broadcasts an unconditional jump instruction, which causes the program to be resumed at the "rollback" address. A repeated fault indication in the same unit leads to its replacement. The number of repetitions before replacement can be specified in the experimental TARP. To replace, power is turned off in the unit, a spare is turned on, and another reset (and jump) is issued. For cases of temporary power loss and other fault conditions which cannot be resolved by the fault location logic, the REC section contains a wired-in "disaster restart" procedure.

The TARP is the hard core of the system. Three fully powered copies of the TARP are operated at all times together with n standby spares ($n=2$ in the present design). The outputs of the TARPs are decided by a 2-out-of-$(n+3)$ threshold vote. When one powered TARP disagrees with the other two, the recovery mode is entered and an attempt is made to set the internal state of the disagreeing unit to match the other two units. If this TARP rollback attempt fails, the disagreeing unit is returned to the standby condition and one of the standby units receives power, goes through the TARP rollback, and joins the powered triplet. The computer is now restarted, a rollback performed, and standard operation continues. Because of the three unit requirement, design effort has been concentrated on reducing the TARP to the least possible complexity. Experience with the present model has led to several refinements of the design.

The replacement of faulty functional units is commanded by the TARP vote and is implemented by power switching. It offers several advantages over the switching of information lines which

connect the units to the bus. The number of switches is reduced to one per unit, power is conserved, and strong isolation is provided for catastrophic failures. Magnetic power switches have been developed which are part of each unit's power supply and are designed to open for most internal failures. The threshold function is inherent in the control windings of the switch. The information lines of each unit are permanently connected to the buses through component-redundant isolation circuits. The signal on a bus is the logic OR of all inputs from the units, and unpowered units produce only logic zero outputs. The power switch and the buses utilize component redundancy for protection against fatal "shorting" failures.

Comparative Reliability Analysis

This section considers the reliability (with respect to permanent failures) which can be expected for the STAR computer. The approach is to estimate the relative reliability with respect to an existing reference system. An absolute reliability prediction is not made because the failure rates for components which are being developed for a flight model are not yet adequately established.

The reference computer for reliability estimation is the nonredundant Mariner Mars 1969 (MM'69) computer, which was the on-board computer for the successful Mariner 6 and 7 missions to Mars. It was chosen because a detailed description and extensive failure rate data are readily available. With respect to computing performance it must be noted that the MM'69 computer is a bit-serial machine with a bit rate of 2.4 kHz and an instruction set of 16 op-codes, whereas the STAR is a byte-serial machine with a 0.5 MHz clock and an instruction set of 130 op-codes. This gain in performance is not used as a factor in reliability estimation.

Reliability models (1) the MM'69 computer, (2) a simplex computer equivalent in performance to the STAR, and (3) the STAR computer are shown in Fig. 7. The MM'69 computer [Fig. 7*a*] is assigned a complexity of unity. It is assumed that the simplex computer [Fig. 7*b*] consisting of eight functional units is 8×CF times as complex as the MM'69 computer. The relative complexity factor CF is defined as the ratio of complexity (component count) of a single STAR unit to the complexity of the entire MM'69 computer. The value CF=1/3 was established by

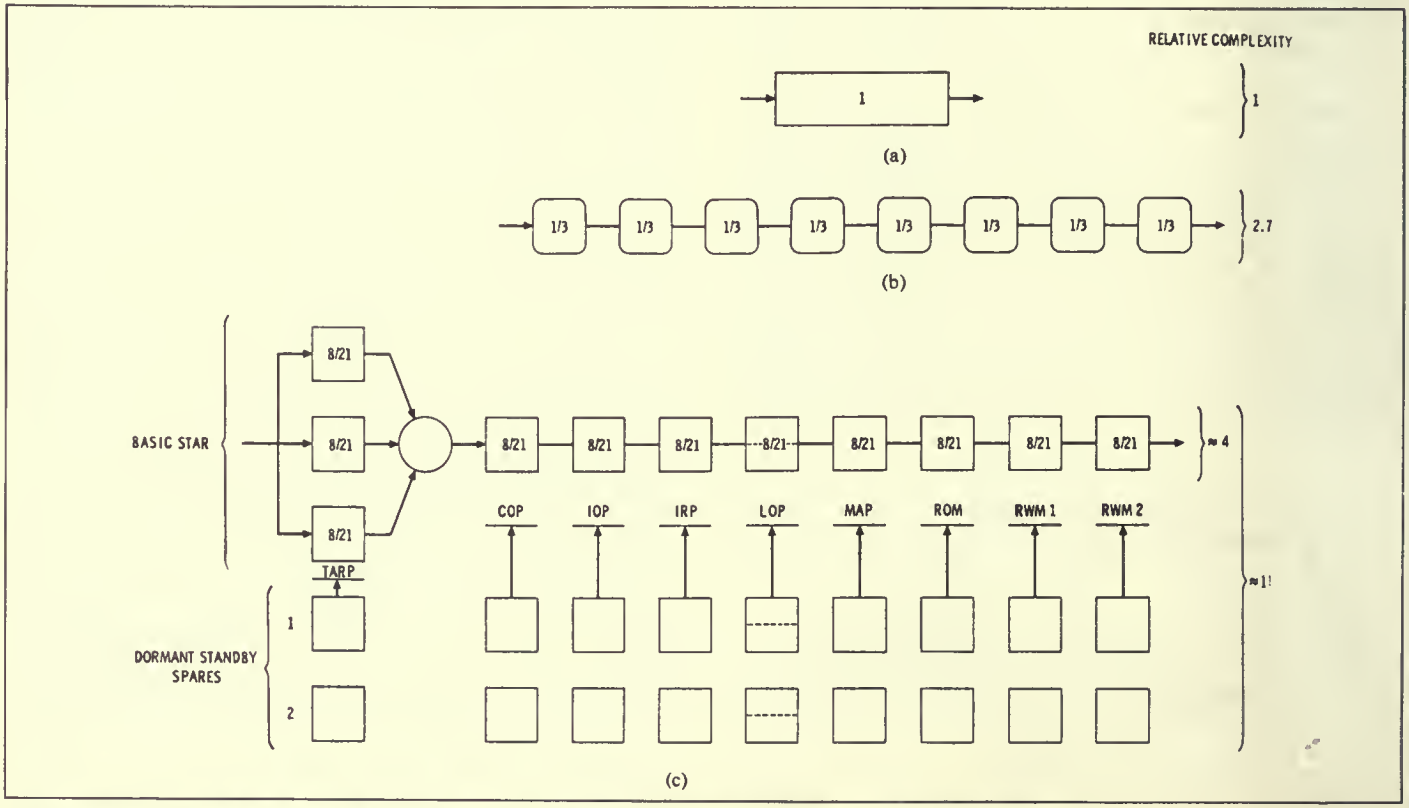

Fig. 7. Reliability models. *(a)* Mariner Mars 1969 computer, *(b)* Simplex computer, *(c)* STAR computer.

detailed comparison and is used in the subsequent analysis. The comparison is made with respect to MM'69 technology, i.e., it is assumed that the simplex and the STAR computers employ the same components and packaging techniques as the MM'69 computer.

The STAR model [Fig. 7*c*] consists of eight functional units plus the test and repair processor (TARP) array in series reliability. All units are considered to be of similar complexity and are allocated an equal number of spares. Results for $S=2$ and $S=3$ are presented. The reliability model applied to all units except the TARP is the standby-replacement redundancy model with dormant spares [Bouricius, Carter, and Schneider, 1969; Mathur, 1971a]. The TARP was modeled as a hybrid-redundant $H(3, S)$ system [Mathur and Avižienis, 1970]. Details of the reliability models and measures are presented in [Mathur, 1971a]. The logic processor LOP is assumed to have an internal duplication of the circuits which are not protected by the error-detecting codes. Two sets of three RWM units each are shown; this is a pessimistic assumption, since the computer can function with only one of the six RWM units surviving.

The fault coverage factor [Bouricius, Carter, and Schneider, 1969] in the STAR model is taken into account in two ways: (1) by including the fault detector and recovery initiator as a separate processor (the TARP), and (2) by applying a self-testing factor (STF) to the relative complexities of the units. Note that the simplex computer [Fig. 7*b*] does not contain a processor corresponding to the TARP in the STAR computer since the simplex computer is a computationally equivalent nonredundant machine without "test and repair" capabilities. Since 4 bits of the 32-bit STAR word serve for error detection, a STF equal to 8/7 was chosen. The STF expresses the overhead due to the self-testing and repairing features within each STAR unit, that is, a STAR unit has 8/7 of the complexity of the same unit in the "simplex" computer. Applying CF = 1/3 and STF = 8/7 a STAR unit has the relative complexity of 8/21 with respect to the entire MM'69 computer.

Examples of reliability predictions based on the MM'69 data are shown in Tables 1 and 2 and Figs. 8 and 9. The *lower bound* ($K = 1$) assumes equal failure rates of powered and spare units (K is the failure rate ratio). The *upper bound* ($K = 1$) assumes a zero failure rate of spare units. Two-spare ($S=2$) and three-spare ($S=3$) STAR systems are considered. Table 1 and Fig. 8 show the predicted reliability as a function of time. Table 2 shows the time (in years) for which the reliability remains above a specified value. Figure 9 presents the predicted reliability gain, defined as the ratio STAR reliability/MM'69 reliability.

The computing operations for the foregoing analysis, the generation of tables, and the plotting of graphs was done with the aid of the computer-aided reliability estimation (CARE) program [Mathur, 1971b], which was developed as a design tool during the reliability study. CARE is a software package developed on the Univac 1108 computer system at JPL. CARE may be interactively accessed by a designer from a teletype console to calculate his reliability estimates. The input is in the form of a system

Table 1 Reliability versus Time for Various Configuration (CF = 1/3)

Mission time (h)	*MM'69 computer*	*Simplex computer*	*STAR computer with S spares*			
			Upper bound ($K = \infty$)		*Lower bound ($K = 1$)*	
			S = 3	*S = 2*	*S = 3*	*S = 2*
4368 (≈ 6 months)	0.928	0.82	0.9999998	0.99997	0.999995	0.99982
43 680 (≈ 5 years)	0.475	0.14	0.997	0.97	0.966	0.87
87 360 (≈ 10 years)	0.225	0.019	0.96	0.79	0.71	0.45

Table 2 Mission Duration for Specified Reliability (CF = 1/3)

Desired mission reliability	*Mission duration in years*					
	MM'69 computer	*Simplex computer*	*STAR computer with S spares*			
			Upper bound		*Lower bound*	
			S = 3	*S = 2*	*S = 3*	*S = 2*
0.9	0.7	0.3	12.5	7.5	6.7	4.5
0.8	1.5	0.6	16.0	9.7	8.5	6.0
0.7	2.4	0.9	18.5	11.7	10.0	7.0
0.6	3.5	1.3	20.5	13.5	11.3	8.3

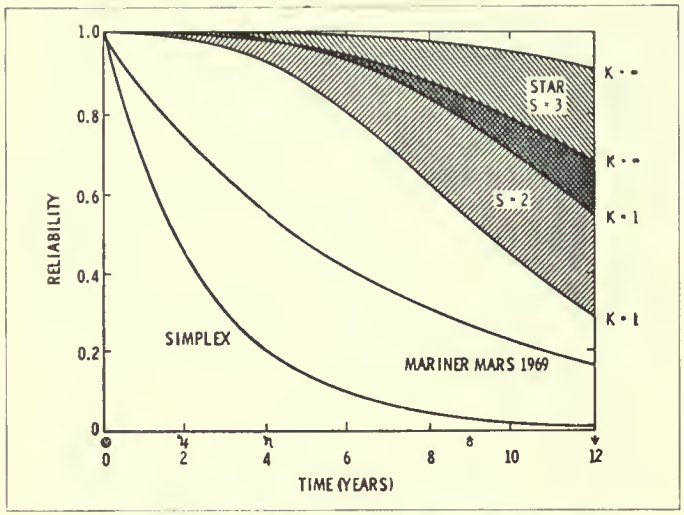

Fig. 8. Reliability versus mission time MM'69, simplex, and STAR computers.

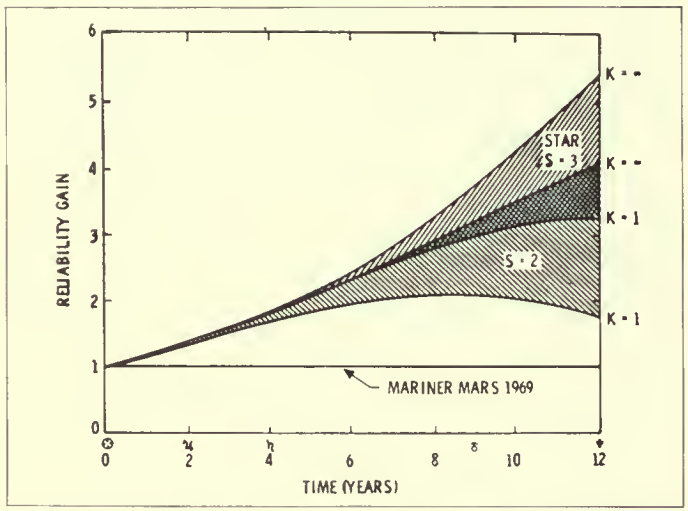

Fig. 9. Reliability gain of STAR computer with respect to the MM'69 computer.

configuration description followed by queries on the various reliability parameters of interest and their behavior with respect to mission time, fault coverage, failure rates, dormancy factors, allocated spares, and partitioning. The CARE program is extensible, and it may be updated to incorporate new reliability models as they become available.

STAR Computer Software System

Early in the design of the STAR computer it became evident that the fault-tolerant architecture would impose unconventional constraints on its software. The development of the software system for the STAR computer was initiated in 1968 and closely followed the hardware development. It is partitioned into two subsystems. The programming subsystem consists of three modules: an assembler, a loader, and a functional simulator. An executive program facilitates coordinated use of these modules. The operating subsystem consists of two modules: the resident executive module and the applications programs module. The programming subsystem has been implemented on the Univac 1108 computer of the Scientific Computing Facility at JPL. The first version of a resident executive for the STAR computer is nearing completion.

SCAP (the star computer assembly program) is the first module of STAR software. Programs for the STAR computer are written in the assembly language SCAL. SCAP is a traditional two-pass assembler incorporating machine instructions, pseudo-operations, and macrofacilities. A unique feature of SCAP is the encoding of instruction and data words as required by the STAR computer. SCAP calculates the code required and generates the encoded value of the word. Another feature of SCAP is the COMPILE pseudo-operation which implements automatic compilation of simple arithmetic statements by the assembler.

The second module LOAD (the loader) reads the program into the simulated STAR computer memory. After all decks have been read, a COMMON area is allocated, relocation is completed, and external linkage is accomplished. A map and cross-reference table are printed to aid in debugging and documenting the program. The third module of STAR software is the functional simulator, which is modular in nature and follows the latest STAR hardware configuration. Two special features are incorporated in the simulator. The first is the facility to simulate hardware errors in order to test the software aspects of error recovery. The second feature provides STAR register and memory dumps. An executive program facilitates the coordinated use of the assembler, loader, and simulator.

The modules of the operating subsystem of the STAR computer software system consist of the resident executive module and the applications programs module. The STAR resident executive augments the self-testing and repairing features of the hardware in addition to its normal functions. The standard features include interrupt control, input/output processing, and job scheduling. Novel features incorporated due to the fault-tolerance architecture of the STAR computer include a "cold start" capability, reconfiguration processing, rollback assistance, and diagnosis of faulty units. The cold start capability resets the hardware and software after a disaster restart as well as prior to an initial load.

Reconfiguration processing is required for memory replacement, since software assistance is required to load a newly activated memory unit. All programs running on the STAR computer require rollback (recovery) points. The resident executive provides rollback status storage and controls events which are nonrepeatable i.e., they may not occur more than once even if a rollback takes place. Finally, it implements diagnosis for faulty units to determine the cause and extent of failures for possible partial reuse. The present application programs module includes floating-point arithmetic subroutines, and test and demonstration programs. The applications programs which will be required for space missions are a part of the TOPS control computer subsystem project discussed later in this paper.

Extension of STAR Techniques to Peripheral Systems

The STAR techniques of fault tolerance can be systematically extended beyond the boundaries of the computer to effect automatic maintenance of various peripheral systems that communicate with the computer. The case which was investigated in connection with the STAR computer development is the implementation of automatic maintenance for a simplified model of the JPL thermoelectric outer planet spacecraft (TOPS) which is being proposed for the exploration of the outer planets [Astronaut., 1970]. The potentially lower failure rates of unpowered spare units and the constant power demand of a replacement system are exceptionally important in missions requiring a ten year survival of the spacecraft under very strict power constraints.

The methodology of extending the STAR techniques consists of several steps: (1) identification of the replaceable peripheral units; (2) selection of internal error detection functions which are economically feasible within the units themselves; (3) identification of possible functional redundancy, in which either another type of peripheral unit or the computer itself can take over the function of a failed unit; (4) algorithmic description of the monitoring and recovery procedures to be performed for each unit by the computer; (5) development of fault-tolerant communication between the peripheral units and the I/O and interrupt processors of the computer; (6) translation of the monitoring and recovery procedures which have been assigned to the computer into computational requirements: speed, instruction set, storage size, input/output and interrupt system complexity; and (7) estimation of reliability and mean life attainable for each peripheral unit. Several iterations of the design process lead to a system for which a balanced gain in reliability has been attained by means of computer-controlled automatic maintenance. A detailed case study of the application of these techniques is presented in Gilley [1970].

The investigation has identified and quantized the computing capability required from the STAR computer in order to effect the automatic maintenance of the TOPS spacecraft. Furthermore, the results have shown that: (1) the fully automatic maintenance of a complex long-life spacecraft is feasible through a systematic extension of STAR techniques, and (2) the automatic maintenance requirements of the spacecraft systems can be algorithmically described to the detail required to produce computer programs for their implementation. The results of the investigation have systematically extended dynamic redundancy to various peripheral subsystems of an information processing system. Beyond the specific example of a spacecraft, the methodology is applicable to computer-controlled automatic maintenance of other complex data processing, communication, and control systems.

Design of the TOPS Control Computer

The most recent step in the development of the STAR computer concept has been the design of a control computer subsystem (CCS) for the thermoelectric outer planet spacecraft (TOPS) [Astronaut., 1970]. After the TOPS requirements were quantified as described in the preceding section, the CCS design had still to meet four major externally-imposed constraints: (1) the weight of the subsystem was not to exceed 40 lb; (2) power consumption was not to be greater than 40 W; (3) probability of successfully completing a 100,000 h mission was to be equal to or greater than 0.95 (using TOPS approved part failure rates, and (4) it could not, as a consequence of any single internal fault, result in a failure mode catastrophic to the mission.

Because of these constraints, it was not possible merely to "shrink" the STAR computer into a flight package. The STAR design was simplified by retaining only the capabilities needed to meet the TOPS functional requirements. The entire self-test and repair ability of the larger machine has been retained; in fact, the TOPS CCS has expanded failure detection and recovery capability. A variety of advances arising from the years of work on the STAR computer that preceded the TOPS effort have been incorporated into its design.

The CCS operates at a clock frequency of 500 kHz. The CCS word is the same length as the STAR word, 32 bits. The word-processing cycle, ten byte-times long in the STAR computer, has been reduced to nine in the CCS: eight for processing or transferring information and one (two in STAR) for the messages and decision making between words. The execution (including fetch) of an instruction requires one to three cycles. The STAR instruction set with over 200 variants has been reduced to less than 100. To detect word errors, the CCS uses the same residue code as the STAR computer. Unlike the STAR, however, the CCS employs the residue encoding also for operation codes of instructions. In addition to these failure detection measures, the CCS

incorporates dual control logic and clocking, memory address checking simultaneous with all memory accesses, and a nondestructive read-after-write option on all store instructions.

The CCS consists of the seven STAR computer functional units designated the COP, LOP, IOP, IRP, ROM, RWM, and TARP (Fig. 2). The IO/IRP has been split into independent IOP and IRP units in order to improve failure detection and isolation in a completely unattended environment. The MAP is deleted because software multiplication and division are sufficient, while addition and subtraction are done in the LOP. Simplifications in the instruction set have resulted in reduced hardware in the COP, LOP, IOP, and IRP. Conversely, there is increased hardware in the RWM and TARP for added failure detection. A 4096-word ROM and two 4096-word RWM units constitute the program storage capability of the CCS. In addition, another 4096-word RWM (designated SHM) is shared (by use of two independent ports) by the CCS and measurement processor subsystem (MPS). All the CCS RWM units are identical; any one of them can be assigned either as a CCS internal memory or as the SHM. The SHM contains the MPS operating program and the most recent samples of spacecraft variables gathered by the MPS. Because the SHM is available to the CCS as part of its own memory, these samples are conveniently available to it for fault diagnosis and monitoring of spacecraft activity [Gilley, 1970].

Current Research

The research and development program which led to the STAR computer is continuing in several directions. The design of several improved second generation STAR functional units is under way, including a new arithmetic processor, a control processor for medium-scale integrated-circuit implementation, and the shared READ-WRITE memory unit for the storage of automatic maintenance information from the spacecraft telemetry system. Analysis of automatic maintenance algorithms and design of a command/data bus for their implementation are under intensive study. Other current investigations are concerned with the following areas: (1) hardware-software interaction in a fault-tolerant system with recovery, especially the interaction of the TARP and the operating system; (2) studies of advanced recovery techniques, i.e., post-catastrophic restart, TARP replacement schemes, recovery from massive interference, partial utilization of failed units; (3) advanced component technology, especially methods to attain bus and power switch (i.e., hard core) immunity to faults; (4) heuristic studies of fault tolerance by interpretation of extensive experiments with the STAR breadboard as the instrument; (5) design of a second-generation STAR-type computer with universal processor and storage modules, and their implementation by large-scale integration; (6) Computational utilization of the spare units for supplemental tasks in a multiprocessing mode.

At the present time it is evident that the STAR computer design and construction effort has led to valuable new insights into the problem of fault-tolerant computing; further results in this field are expected from the research program in the future.

References

Anderson and Macri [1967]; Astronaut [1970]; Avižienis [1967a]; Avižienis [1967b]; Avižienis [1968]; Avižienis [1971]; Avižienis, Mathur, Rennels, and Rohr [1969]; Bouricius, Carter, and Schneider [1969]; Flehinger [1958]; Gilley [1970]; Griesmer, Miller, and Roth [1962]; Kruus [1963]; Kuehn [1969]; Lewis [1963]; Long [1969]; Lyons and Vanderkulk [1962]; Mathur and Avižienis [1970]; Mathur [1971a]; Mathur [1971b]; Reed and Brimley [1962]; Short [1968].