NINTH EUROPEAN ROTORCRAFT FORUM

PAPER No. 39

FAULT TOLERANCE ASPECTS OF THE A-129 INTEGRATED MULTIPLEX SYSTEM

DR. BARRY J. JOHNSON
DR. PAUL M. JULICH
HARRIS GOVERNMENT AEROSPACE SYSTEM DIVISION
HARRIS CORPORATION
MELBOURNE, FLORIDA, U.S.A.

AND

DR. ATTILIO SOLDAVINI
COSTRUZIONI AERONAUTICHE GIOVANNI AGUSTA S.P.A.
GALLARATE, (VA), ITALY

SEPTEMBER 13-15, 1983

STRESA, ITALY

ASSOCIAZIONE INDUSTRIE AEROSPAZIALI

ASSOCIAZIONE ITALIANA DI AERONAUTICA ED ASTRONAUTICA

ABSTRACT

The Agusta A129 Integrated Multiplex System (IMS) is a highly reliable data processing system designed to implement automated flight control, navigation, system monitoring, and other flight-critical and mission-related tasks in the A129 helicopter. The reliability of the IMS has been achieved by developing software-implemented fault-tolerance features which take advantage of several unique architectural and hardware characteristics that have been designed into the system. This paper describes the fault-tolerance philosophy, the resulting IMS architecture, and the built-in-test/redundancy management features which were developed to provide automatic fault detection and system reconfiguration.
A Markov reliability analysis which was used to quantify the reliability of the system is also presented.

FAULT-TOLERANCE ASPECTS OF THE A-129 INTEGRATED MULTIPLEX SYSTEM

DR. BARRY J. JOHNSON

DR. PAUL M. JULICH

DR. ATTILIO SOLDAVINI

INTRODUCTION

This paper describes the design of a fault-tolerant Integrated Multiplex System (IMS) for the Agusta A129 combat helicopter. The A129 IMS is a multi-computer data collection and processing system which is designed to implement automatic flight control, navigation, fire control, engine minitoring, communications, and other mission-related tasks, as illustrated in Figure 1. The IMS shares its enclosures with the tail rotor fly-by-wire electronics. In the event of multiple failures within the tail rotor electronics, the IMS has the ability to dictate reconfigurations of the tail rotor electronics. However, the majority of the tail rotor fault-tolerance features is independent of the IMS and is the subject of a future paper. The present paper will address only the IMS.

Because one of the fundamental goals of the IMS is to give the A129 a significant improvement in mission success reliability and

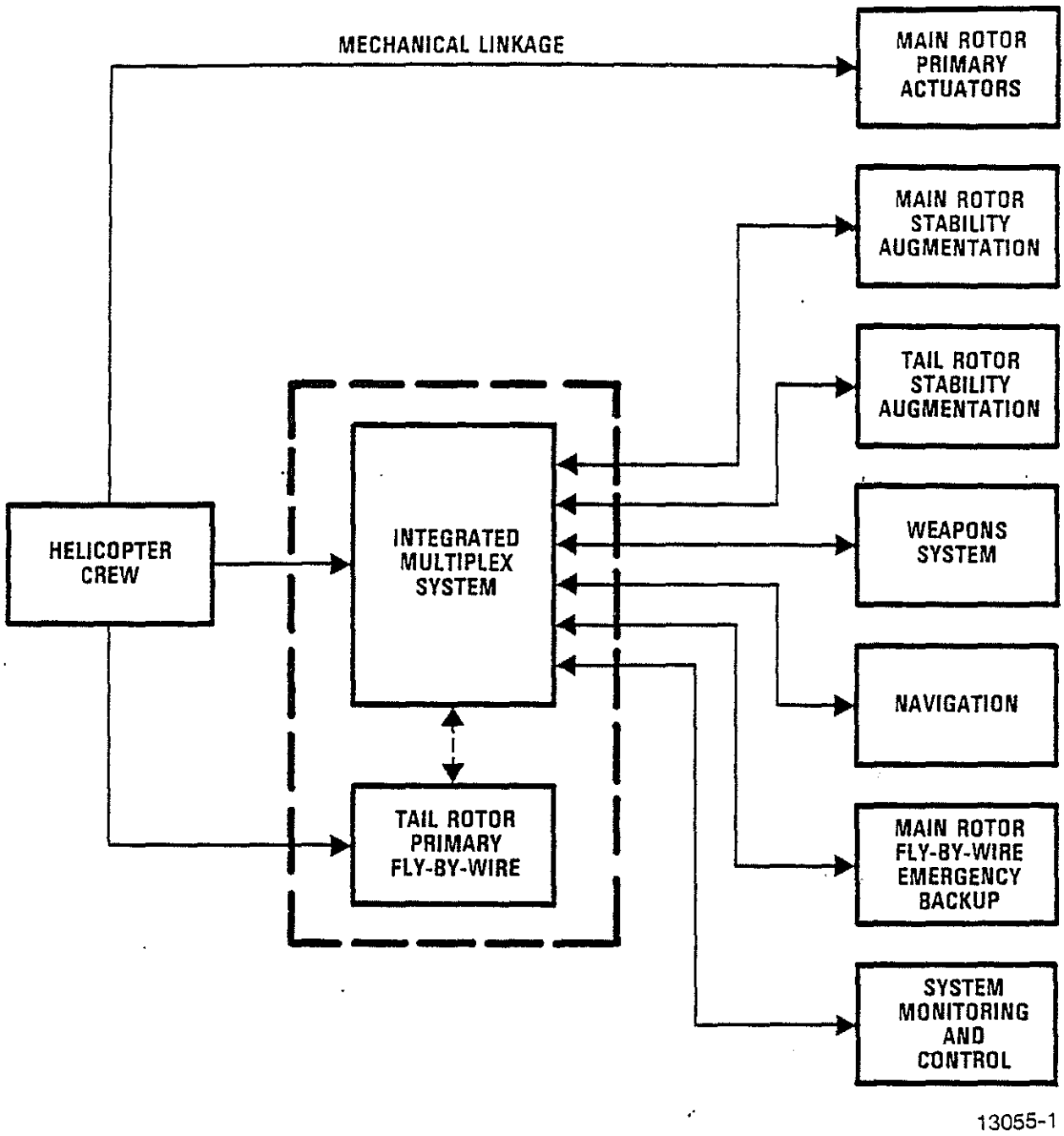

Figure 1. Primary Functions of the Integrated Multiplex Syst

survivability, stringent reliability and fault recovery requirements must be met. The reliability and fault recovery specifications for the IMS may be divided into three major areas which include maintenance assistance, flight critical functions, and mission critical functions. In the area of maintenance assistance, the IMS is required to autonomously detect and locate, to the line replaceable unit, 90 percent of all failures.

Relative to flight critical functions, the IMS has two major requirements. First, any single IMS failure which affects the capability of either the stability augmentation system or the main rotor fly-by-wire emergency backup system must be automatically detected and reconfigured in 95 percent of all cases. Second, there should be no more than one IMS failure, during 15,000 three-hour missions, which results in the disabling of the stability augmentation system. Considering purely random failures, this failure rate corresponds to a reliability of $.9_4 3$ for any three-hour mission.

The next specification relates to mission critical functions. The IMS must experience no more than one failure, in 220 three-hour missions, which affects the capability of the crew to successfully complete its desired mission. Considering only random failures, this failure rate corresponds to a reliability of .995 for any three-hour mission.

The fault-tolerance specifications have been met by developing a carefully defined mix of hardware and software features which provide for the detection of failures in critical system elements and the minimization of the effect of those failures on the overall performance of the helicopter. Failures within the IMS and, in many cases, in equipment that is interfaced to the IMS can be accommodated.

SYSTEM OVERVIEW

The development of the IMS was based on several underlying design philosophies. First, and foremost, it was realized that fault tolerance cannot be designed into a system as an "add-on" feature; fault tolerance must be considered from the inception of the design process and factored into the architecture, the hardware, and the software of the system. Second, it was desired to develop the system such that a maximum amount of fault isolation would be inherent in the design of both the hardware and software. For example, faults which originate in one function should not be allowed to catastrophically propagate to other distinct functions. Third, the flight crew should maintain the capability to perform manual reconfiguration in the event of catastrophic multiple failures. Because fault coverage can never be perfect, the flight crew provides a means of reconfiguring the system if undetectable or unreconfigurable faults occur. Fourth, maximum flexiblity is provided to the system if the majority of the fault-tolerance features are software implemented. This includes fault detection, fault diagnosis, fault isolation, and system reconfiguration. Finally, because the A129 is an agile attack/scout helicopter, the design was severely constrained by weight and size limitations. A series of tradeoffs involving ballistic tolerance, weight, and reliability were performed and resulted in the selection of a dual redundant architecture for the IMS. Additional redundancy was judged to be too costly because of weight limitations.

System Architecture

The architecture of the IMS is based upon a dual redundant, MIL-STD-1553B bus, as illustrated in Figure 2. The system consists of two master units (MU1 and MU2), two "local" units (LU1 and LU2), each of which is colocated with and shares the same power supply with a master unit, two remote units (RU1 and RU2), and two 1553B buses (A and B). The master units provide dual redundant processing capabilities while the local units and the

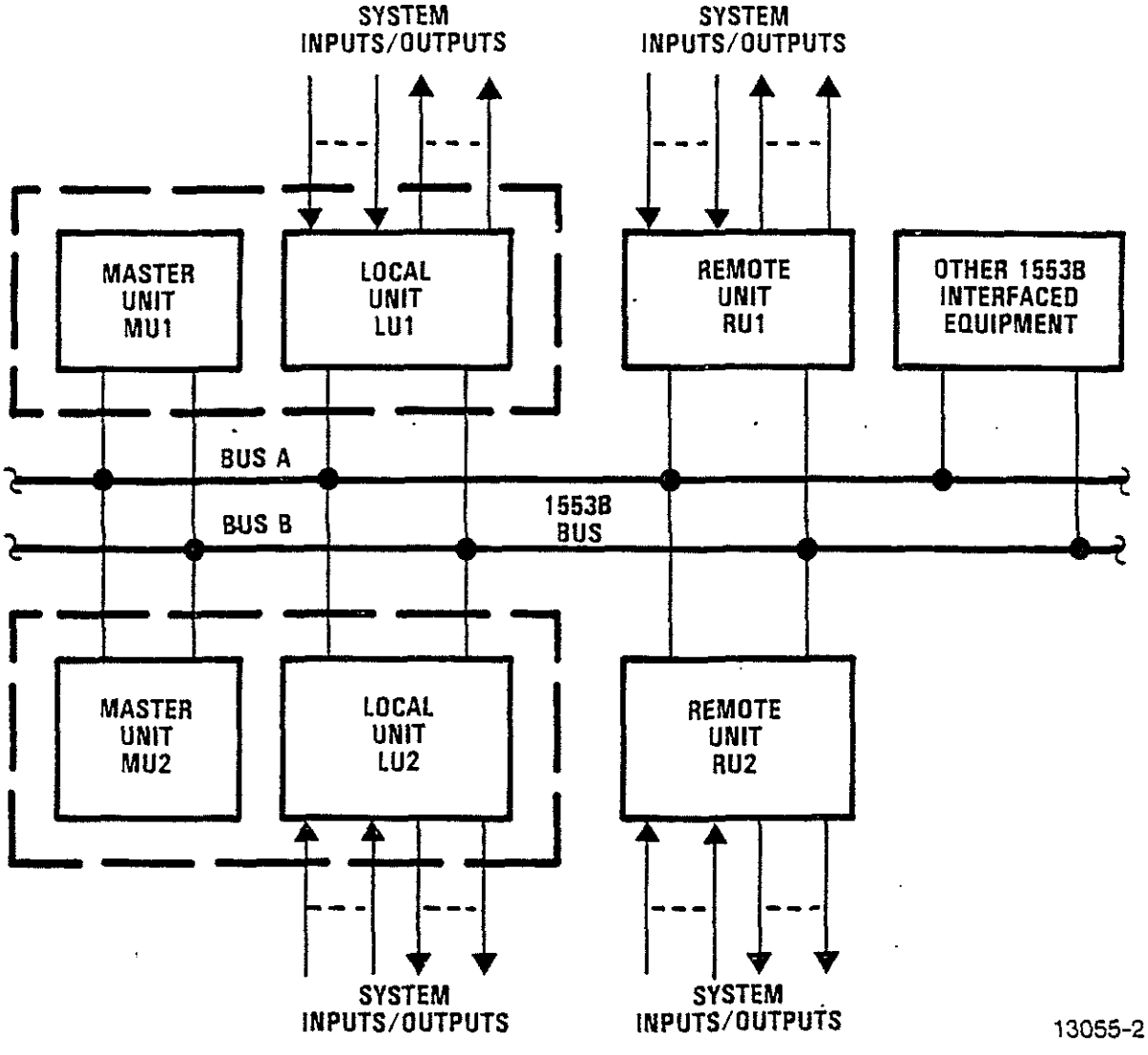

Figure 2. Architecture of the Integrated Multiplex System

remote units form a dual redundant set of interfaces between the master units and the external flight control sensors and actuators. On power-up, a contention procedure causes one master unit to assume mastership of the 1553B bus. The second master unit serves as a "hot" backup in that it monitors the same information over the 1553B as the master and concurrently performs the same computations. Equivalency checks are performed between the two units to guarantee the validity of the critical computations. If the master fails, it ceases to issue polls, and the slave unit (backup) senses the silence and assumes mastership. All communications between the master units and the local and remote units occur over the 1553B bus, consequently any combination of a master unit (MU1 or MU2), a local unit (LU1 or LU2), a remote unit (RU1 or RU2), and a 1553B bus (A or B) forms a completely functional system. Degraded modes of operation, however, may be obtained with fewer functional units.

Each master unit has the internal architecture shown in Figure 3. The master unit consists of four or more processors which have independent program memories, data memories, and clock sources. In addition, the processors have access to global memory consisting of a shared RAM and a nonvolatile RAM. Nonvolatile memory is used for global data storage and the retention of critical configuration and control parameters. Shared memory provides input/output (I/O) buffers for 1553B data bus transfers and for interprocessor information exchanges. Shared memory also serves as a global data storage area. The software in the IMS master unit uses global memory as a "bulletin board" for the transfer of information between processors and between subsystems within the IMS. The software which performs a particular computation will place the result in a specified global memory location where it may be accessed by other processors or the bus controller.

The bus controller serves as an interface between the processors within the master unit and the 1553B bus. The bus controller is a

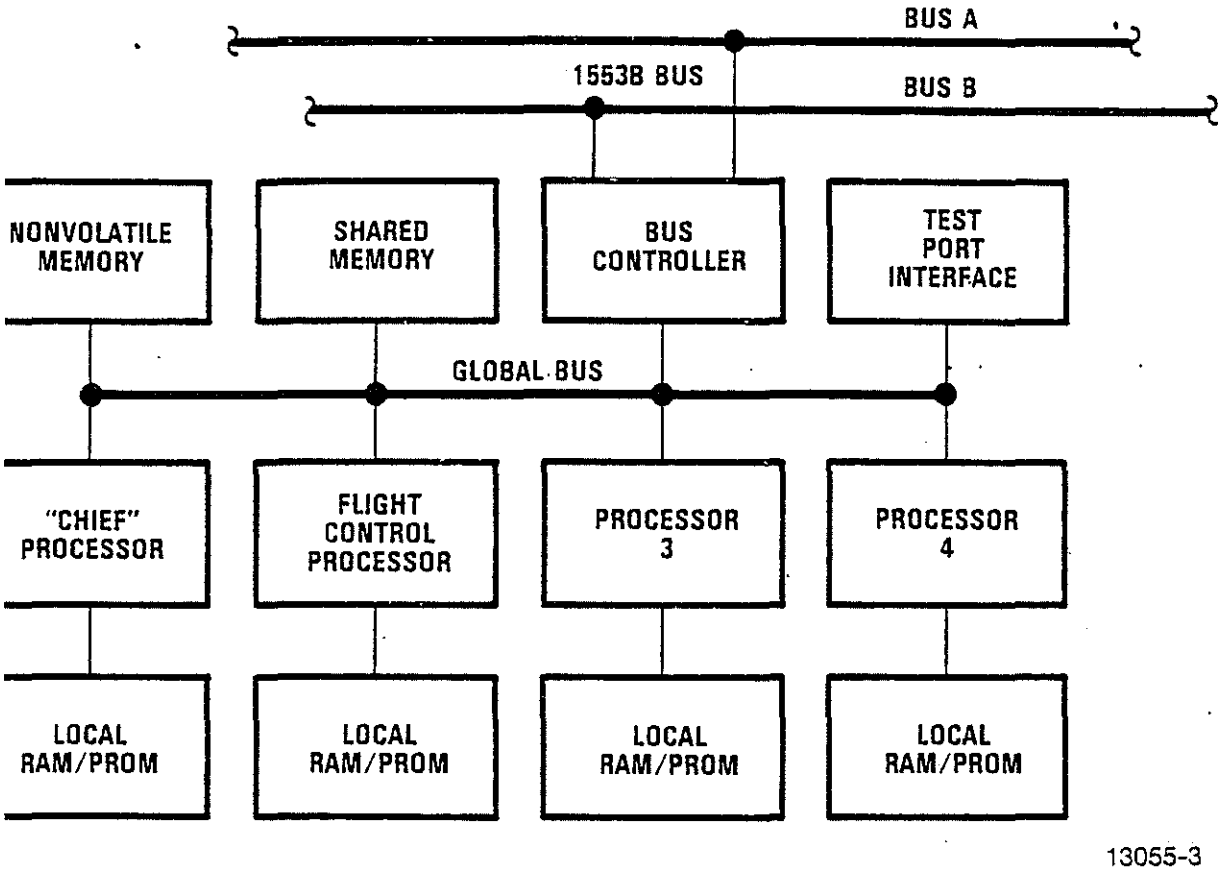

Figure 3. Architecture of Each Master Unit

dedicated hardware unit which handles 1553B protocol for both transmission and reception. During transmission from the master unit, the bus controller accesses output buffers in either nonvolatile memory or shared memory and converts the information into standard 1553B format. During reception, incoming information is transformed from 1553B format and placed in specified locations in shared memory. This technique allows the software within the master unit to remain free of 1553B formatting and protocol issues. The processors need only access proper locations in global memory to transmit or receive information over the 1553B bus.

A general philosophy designed into the master unit is that the processing elements should have to continuously prove their operational capabilities to serve as bus master. This is accomplished by two flags which are incorporated into the bus controller. A master unit which is functioning properly must signal the bus controller that it is capable of performing as bus master by setting a dedicated flag. If both master units indicate that they are healthy, a contention mechanism determines which is to be master. Because hardware clears the flag at a 180 Hz rate, the master unit must refresh the flag at that rate to maintain bus mastership.

A second control flag must also be set by the master unit's software at 180 Hz to prevent the bus controller from resetting all of the processors in the master unit. The master unit's software will cease setting this flag when fatal errors, such as a memory parity error, have been detected by the master unit's built-in-test features. When a reset command is issued by the bus controller, the master unit will go off-line and reinitialize itself to attempt to reach back-up status. This will be long enough to allow the backup master unit to assume mastership of the bus. This is the normal means of transferring mastership from one master unit to the other.

Each local unit and remote unit has the basic architecture illustrated in Figure 4. Information coming from the external sensors is placed in local memory where it can be accessed by the bus interface unit (BIU) for transmission to the master unit. The BIU handles 1553B bus protocols for both transmission and reception. Information coming from the master unit is received by the BIU, transformed from 1553B format, and stored in local memory where the I/O controller may access it for transfer to the flight control actuators.

System Synchronization

System synchronization is maintained by the bus "master" issuing a Broadcast command at the beginning of each frame (30 Hz). The Broadcast command is not acknowledged, therefore, the bus controller's redundancy management will automatically alternate the bus over which the Broadcast is issued. A unit which cannot hear over one bus will receive the Broadcast every other frame (15 Hz). This rate will maintain an adequate degree of synchronism of the units.

Processing within the IMS master unit is synchronized by the use of a global interrupt clock which occurs at a 180 Hz rate. The occurrence of this interrupt initiates the beginning of a new "superframe," as illustrated in Figure 5. Six superframes form a single frame which occurs at a 30 Hz rate. Events can be scheduled to run at multiples of the 180 Hz clock source; for example, an event occurring every two superframes would be running at a 90 Hz rate.

At the beginning of each superframe, the master unit's bus controller goes to a polling list which is contained in PROM to determine the I/O operations that must be performed during that superframe. The bus controller will then either load inputs from remote devices into shared RAM or read outputs from shared RAM and transmit the information to the appropriate remote device. After

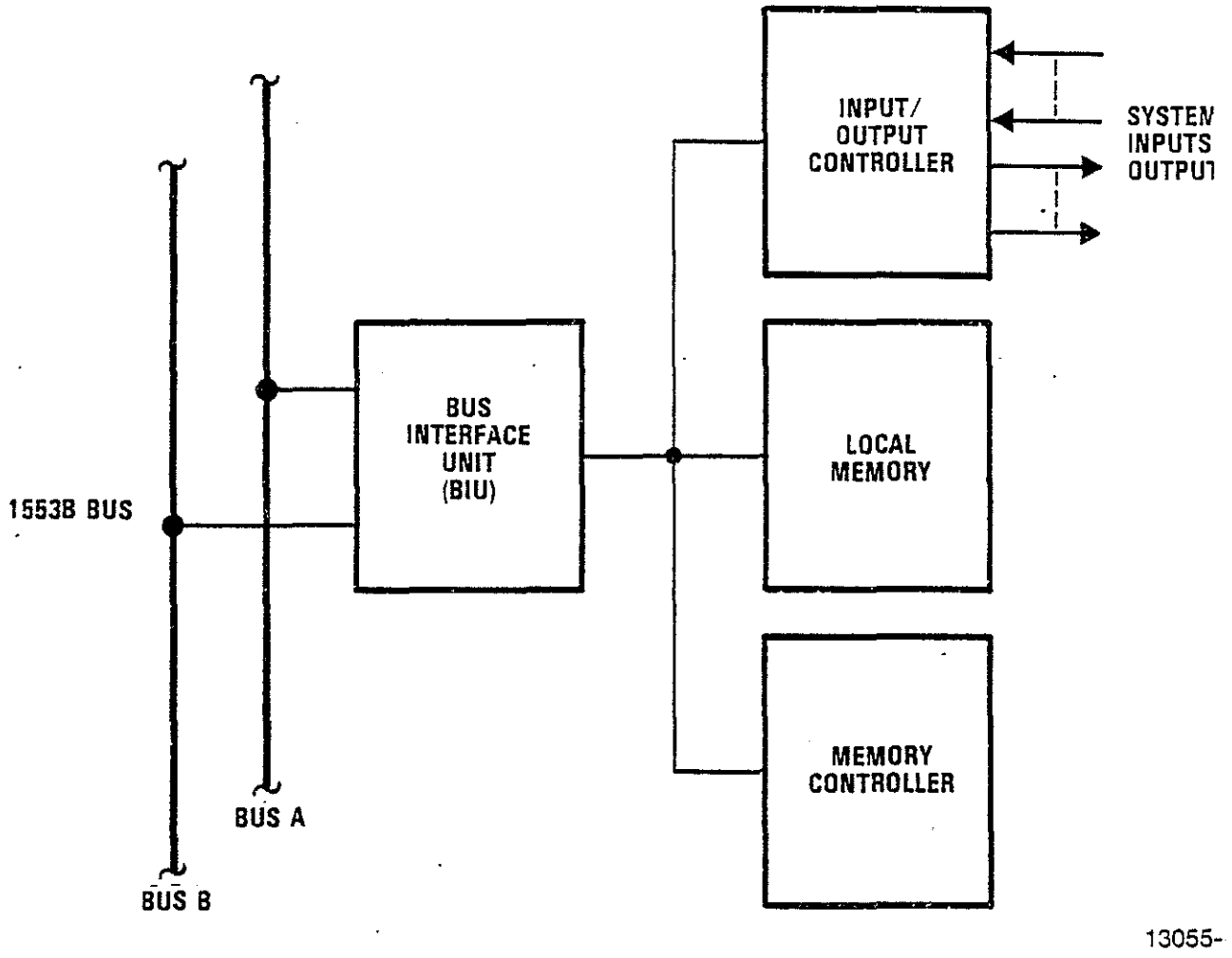

Figure 4. Architecture of Each Local and Remote Unit

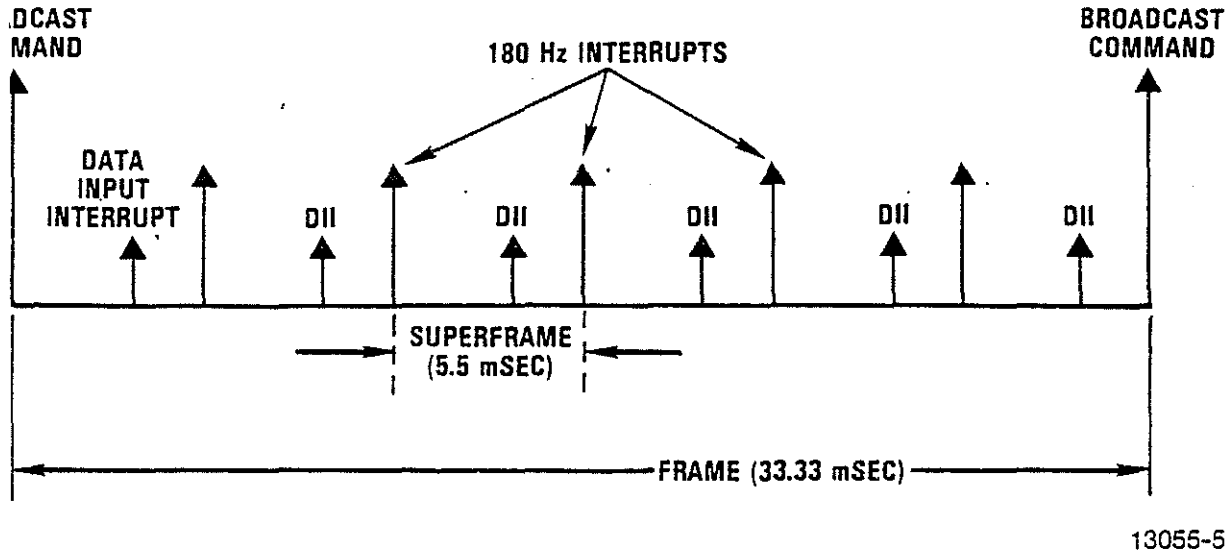

Figure 5. Integrated Multiplex System Synchronization

all of the current superframe's inputs are processed, the bus controller will generate a "data input interrupt" (DII) to signal the master unit to begin processing the new data. Once each item in the polling list has been processed, the bus controller will halt until the occurrence of the next 180 Hz interrupt, at which point the process will be repeated using the next superframe's polling list. The polling list is independently programmed for each of the six superframes in each frame.

BUILT-IN-TEST AND REDUNDANCY MANAGEMENT

The built-in-test and redundancy management (BIT/RM) features of the A129 IMS consist of a collection of carefully defined hardware and software which provides fault-tolerance capabilities. Because the system has been constrained to dual redundancy, the basic BIT/RM philosophies must be carefully defined if the desired high reliability and reconfiguration capabilities are to be obtained. Voting between the two master units, for example, can detect failures but cannot identify the faulty unit such that reconfiguration can occur. Consequently, voting cannot be relied upon as a primary fault tolerance mechanism.

A major function of the IMS is to provide flight controls for the A129, and a major function of BIT/RM is to provide fault-tolerance capabilities for flight controls. As a result, BIT/RM is mainly concerned with the elements necessary to implement the flight control algorithms. These elements include the master unit, local unit, 1553B bus, remote unit, and the required flight control sensors and actuators. Figure 6 specifies the heirarchy of elements under the jurisdiction of BIT/RM. Because the majority of BIT/RM features are implemented in software, the master unit forms the highest level of the hierarchy. Lower levels in the hierarchy may experience failures within their elements which, if not handled at the level of occurrence, may propagate through other levels of the hierarchy to the master unit which then detects the error condition and performs a redundancy management

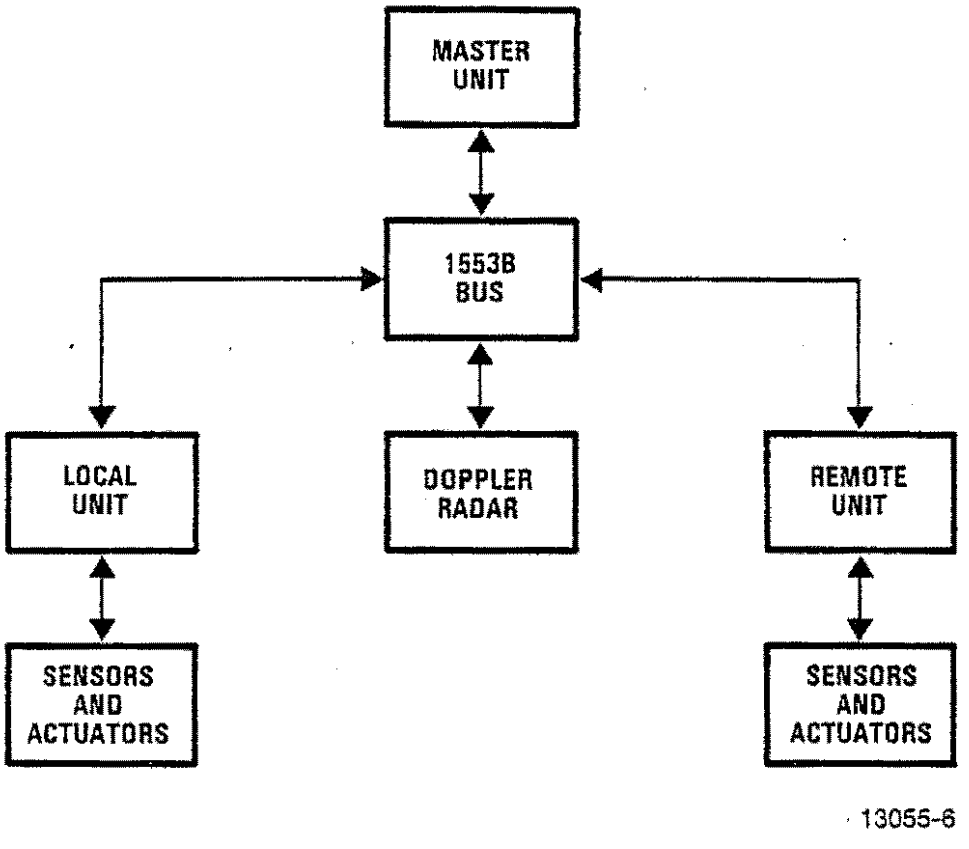

Figure 6. Hierarchy of Elements Covered by BIT/RM

action. For example, the effect of a failure within a flight control actuator must propagate from the actuator through the remote unit and the 1553B bus to the master unit before being identified and handled.

The basic philosophy of BIT/RM is to provide a collection of tests to concurrently check the operation of each level in the hierarchy and to check the paths between levels to guarantee fault-free performance of the system. The results of these tests are then used to form a decision concerning the health of each level and the paths between each level in the hierarchy, and to develop a redundancy management action if any element of any level is not functioning properly. The possible redundancy management actions include: (1) master unit swapover, (2) bus swapover, (3) local/remote unit swapover, (4) sensor swapover, (5) degradation of the system, and (6) passivation of the system.

Master Unit Built-In-Tests

The basic concept of the master unit's built-in-test is that each processor runs a series of tests to verify its own capability. Each processor then uses the results of the tests to form an independent decision concerning its own health. The resulting decisions are transferred to the "chief" processor which combines the information with the results of several additional master unit tests to form a decision concerning the total master unit health. If the master unit is determined to be faulty, the "chief" processor will cease setting the master request flag for the bus controller, and a mastership swapover will occur, provided that the second master unit is healthy. The bus controller of the faulty master unit will force a processor reset and a recovery to back-up status will be attempted.

The primary tests for the master unit are illustrated in Figure 7 and consist of three hardware implemented tests and seven software implemented tests. Hardware provides for parity generation and

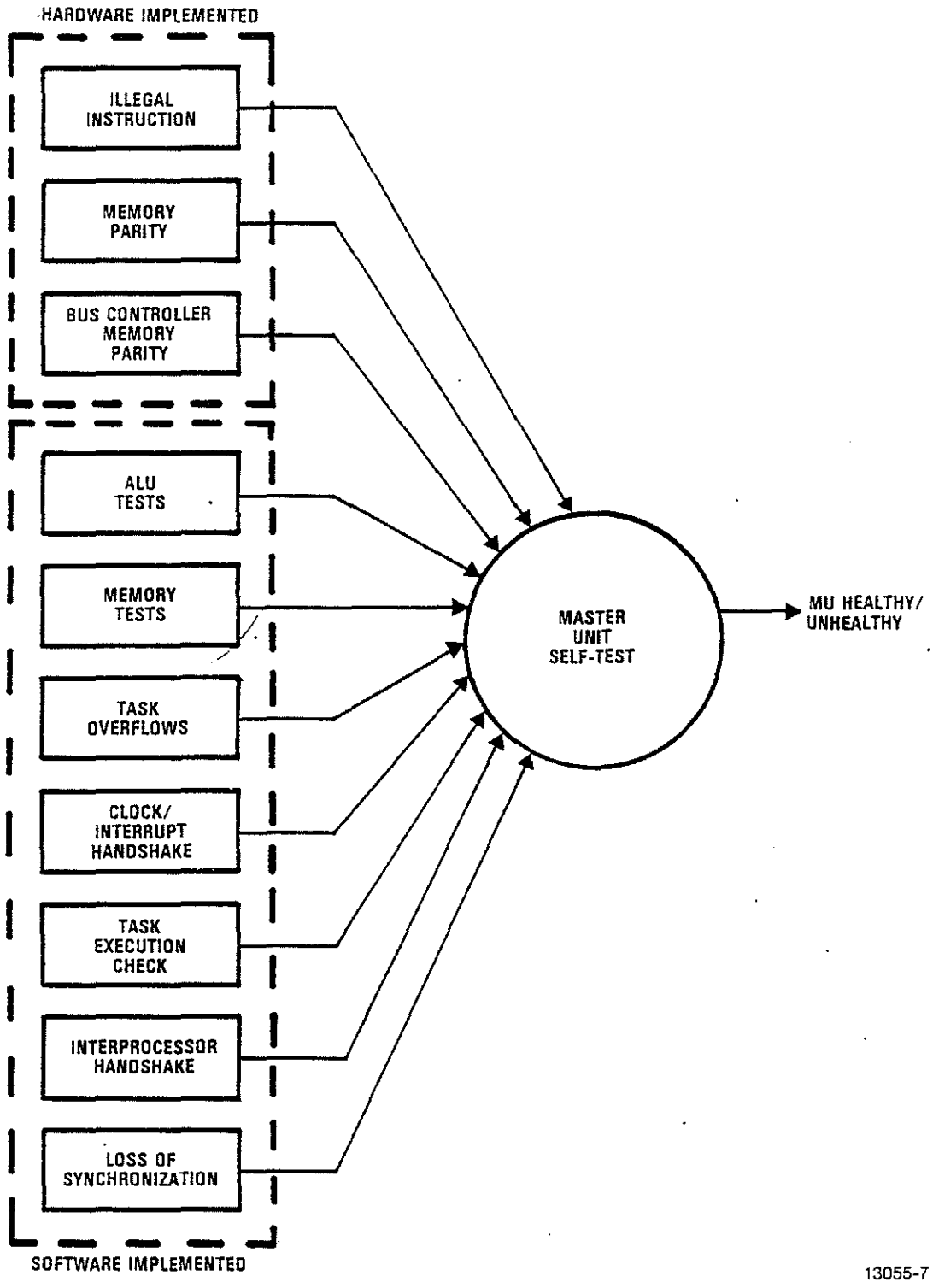

Figure 7. Built-in-Tests for Each Master Unit

checking on all memory transactions for both the processors and the bus controller. If a parity error occurs, a hardware interrupt is generated which causes the failed MU to attempt a restart (resulting in MU swapover). Hardware also provides protection against invalid operation codes. Certain bit patterns which are possible when the instruction codes are in error will produce illegal codes that can be detected by special hardware within the processor chips. If an illegal code occurs, an interrupt is generated and the MU will attempt a restart.

Possibly one of the most important software-implemented tests is the interprocessor handshake. This test requires each processor to set a flag in shared memory each superframe to indicate the processor's health. The "chief" processor then reads all flags and does not set the master request flag for the bus controller if any processor fails to indicate that it is fully functional. The "chief" processor immediately resets the flags after reading them so that all processors must refresh the handshake flags each superframe.

Additional tests are run in background mode on each processor's arithmetic logic unit (ALU) and on the memory contained within the master unit. Software within each processor checks the performance of the ALU by executing a representative subset of the processor's arithmetic and logical instructions and comparing the results to prespecified values stored in PROM. This test is also designed to utilize all of the processor's general purpose registers. Memory tests on nonvolatile memory, shared memory, and local memory are also performed on a real-time basis by writing specified patterns to certain memory locations and reading those locations to verify the results. This test is designed mainly to check interfaces between the processors and memory. Parity is the main technique for detecting memory cell failures.

In addition to the above tests which are primarily designed to detect hardware failures, several tests have been provided to

protect the system against latent software errors. For example, the task overflow tests and the task execution test are designed to first assure that all tasks are executing and second, to assure that tasks, such as the flight control algorithms, are executing within their alloted times.

The final two master unit tests, the clock/interrupt handshake and the loss of synchronization test, are intended to validate the performance of the bus controller and to assure proper system timing. The clock/interrupt handshake verifies that the 18Ø Hz interrupt clock and the data input interrupt, which is generated by the bus controller during its polling sequence, occur in the proper order. Any discrepancies are handled via a mastership swapover. The loss of synchronization test verifies that the processing units and the bus controller are operating on the same superframe count. This is accomplished by the bus controller generating a superframe count status word for the "chief" processor. A loss of synchronization between the processors and the bus controller is sufficient reason to implement a mastership switchover.

1553B Bus Built-In-Tests

The bus controller handles the redundancy management for the 1553B bus. If one of several tests is failed, the bus controller will automatically switch to the other bus when the next communication with the affected remote device occurs. If a bus itself is faulty, all remote devices will toggle to the opposite bus after each remote device has been polled once on the faulty bus.

The primary tests which verify the operation of the 1553B bus are illustrated in Figure 8 and consist mostly of hardware-implemented tests. Time-out counters, for example, are provided to monitor the time of response for all remote devices on the 1553B. Each remote device must respond within a specified period of time when polled by the bus master. If a time-out occurs, the next poll to

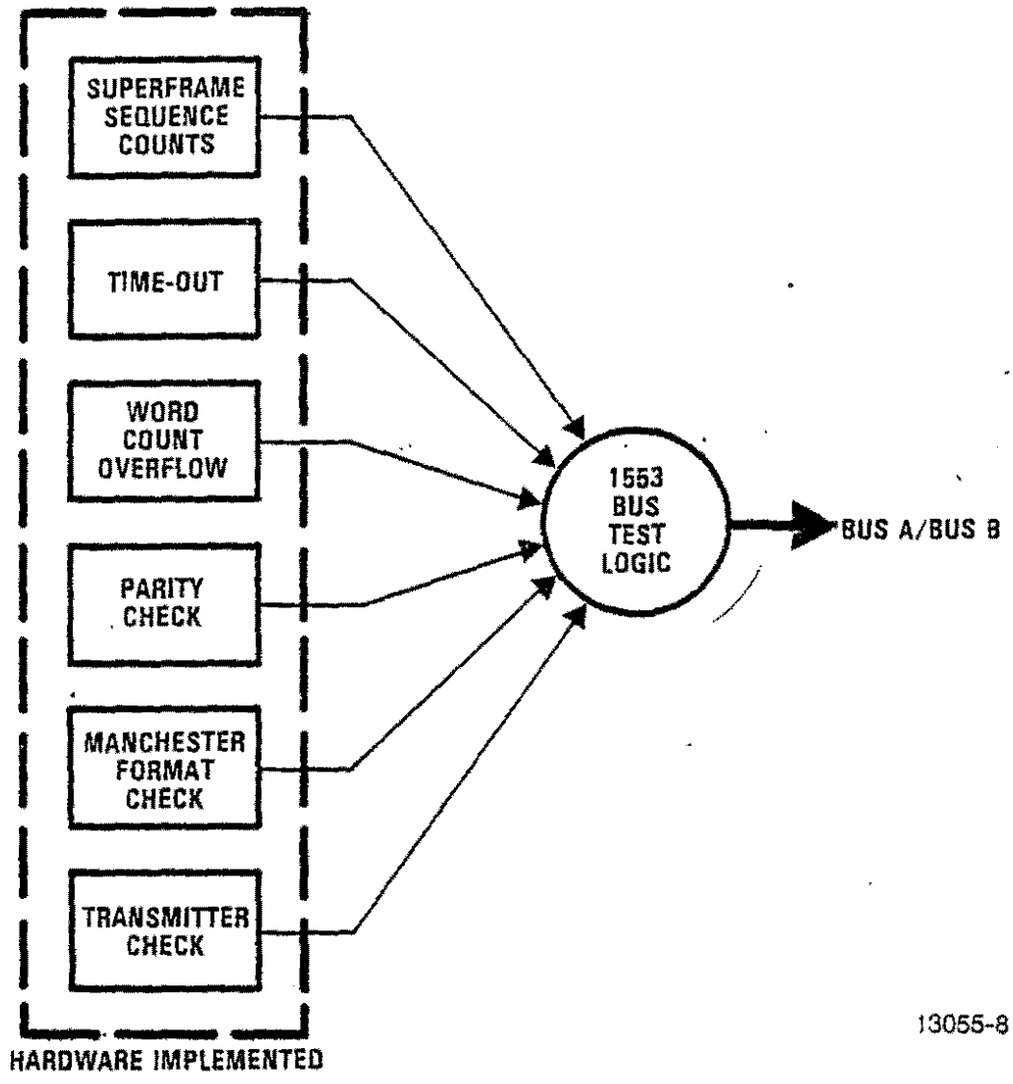

Figure 8. Built-in-Tests for Each 1553 Bus

the affected device will automatically occur over the opposite 1553B bus. If all remote devices fail to respond on both buses, then the current bus master will relinquish mastership and allow the backup master unit to attempt to communicate with the remote devices.

A hardware counter has also been included in the bus controller to compare the number of data words requested from a remote device with the actual number of words received. As with the time-out condition, the first redundancy management action when a word count error occurs is to communicate with the affected device over the opposite bus during the next transaction. Once again, if all remote devices produce word count errors, the bus master will relinquish mastership and allow the backup master to attempt to correctly operate.

Parity checks and Manchester format checks are performed on information which passes over the 1553B bus. If either a parity error or a Manchester format error occurs, the bus controller will automatically discard the data that is affected. The next transaction with the affected device will be over the opposite bus.

Hardware timers are provided to detect a transmitter which continuously babbles over the bus. The result of this failure is a bus swapover for all affected remote devices. If the condition is detected on both buses, the current bus master relinquishes mastership and allows the backup master unit to gain control of the bus.

Local Unit and Remote Unit Built-In-Tests

The redundancy management actions for the local/remote units may be divided into two categories; one which applies to system outputs and one which applies to system inputs. For the system outputs, any failure of the local/remote unit tests will cause the

outputs of the affected local/remote unit to be passivated to a safe state. Normal system control will then be provided through the alternate local/remote unit. If the error condition clears, the local/remote unit will be allowed to recover thus returning full redundancy. For the system inputs, any failure of the local/remote unit tests will cause the inputs, where possible, to be read from the opposite local/remote unit.

The primary built-in-tests for the local and remote units are illustrated in Figure 9. Command feedback is provided on selected stability augmentation and fly-by-wire commands to check the capability of the analog-to-digital and digital-to-analog converters. The command generated by the master unit is converted from digital to analog to provide a drive signal to the servo amplifier. The resulting analog command is then routed back through an analog-to-digital converter and transmitted to the master unit where it is compared with the original commands. A significant difference between the generated command and the returned command indicates a fault in either the digital-to-analog converter, the analog-to-digital converter, the multiplexers, or the path that the data must travel upon. Command feedback is also provided on selected discrete commands, such as the solenoid valve commands.

To provide additional protection against analog-to-digital converter and analog multiplexer failures, fixed voltage inputs have been applied to several multiplexer channels. The BIT/RM software monitors these references to guarantee that they remain within specified tolerances.

Several tests which were designed primarily to detect bus errors are also used to provide protection against remote/local unit failures. For example, time-outs or word count overflows which occur on both buses, but for only one remote/local unit may be used to identify the affected remote/local unit as failed. In addition, the perceived status of sensors and actuators is used to

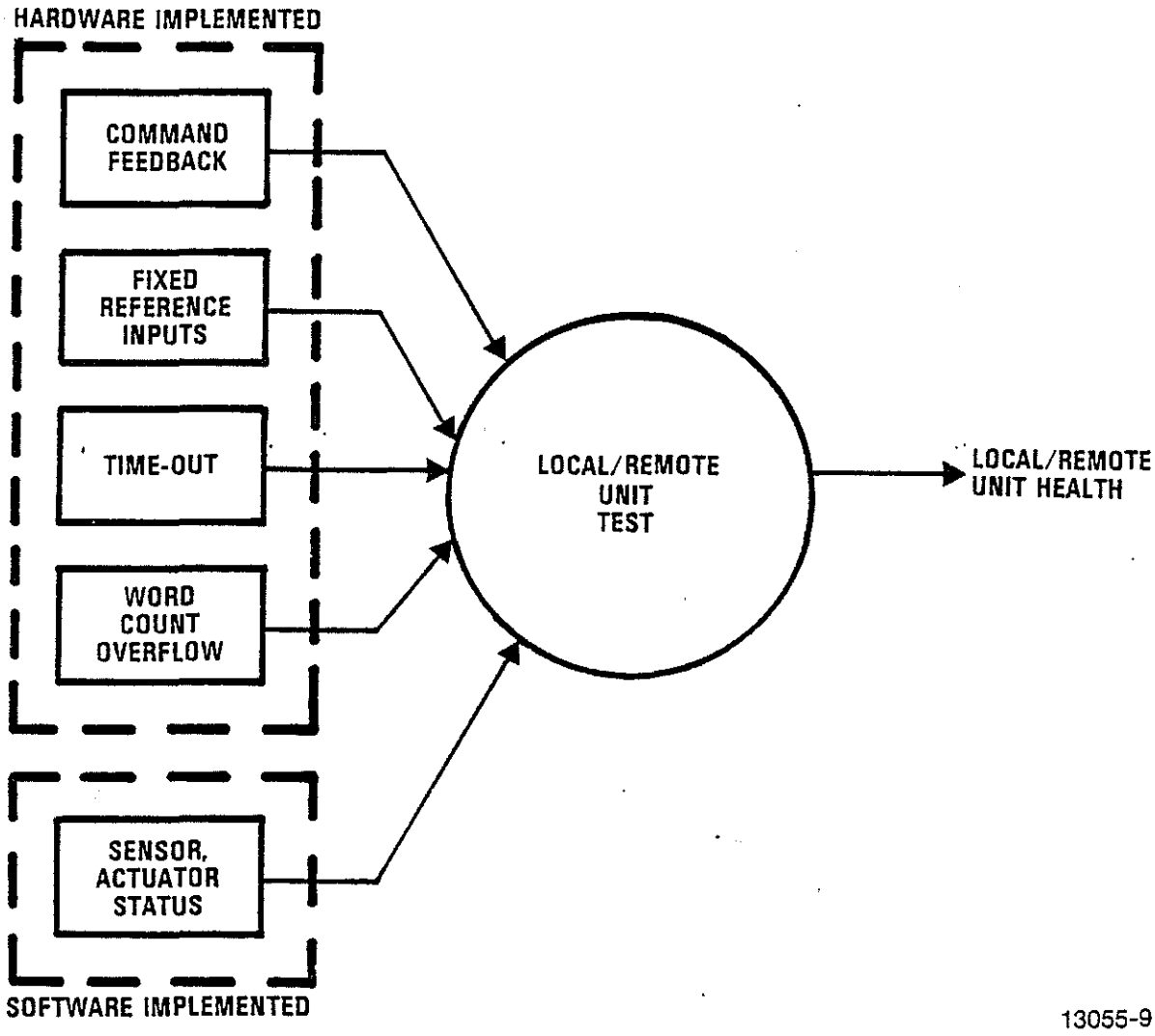

Figure 9. Built-in-Tests for Each Local and Remote Unit

isolate failures to remote/local units rather that the sensor or actuator.

Aircraft Sensors and Actuators Built-In-Tests

The BIT/RM subsystem uses three primary techniques to validate the performance of the flight control sensors and actuators. These include actuator models, sensor comparison, and dedicated hardware built-in-test, as illustrated in Figure 10. Sensor comparison is conducted on the outputs of sensors which are deemed critical enough to make redundant. For example, the system has two vertical gyros and an attitude indicator gyro. The outputs of the vertical gyros are compared, and if a disagreement occurs, the attitude indicator gyro is used to distinguish between the good and the faulty vertical gyro.

Dedicated hardware is also provided for selected sensors to obtain added fault coverage. For example, rotary variable differential transformers (RVDT's) which are used to detect crew stick movements are provided with hardware to detect shorts and open-circuit conditions. This hardware produces a single bit which indicates the perceived status of the RVDT.

The primary test for the flight control actuators involves the use of actuator models. Selected actuators within the system have been dynamically modelled and recursive equations defined to predict the movement of the actuators based on the drive signals provided to the actuators. The predicted actuator position is then compared to the measured actuator position, and if a significant discrepancy exists, the actuator drive is classified as failed. It should be noted that the models provide coverage on not only the actuators but also the data paths and the sensors.

RELIABILITY ANALYSIS

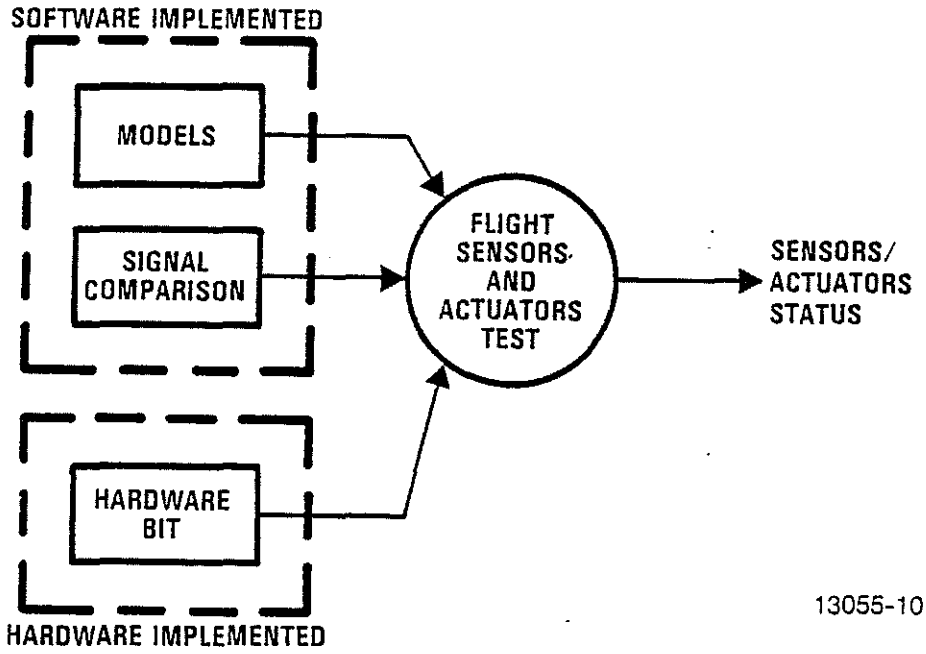

Figure 10. Built-in-Tests for the Flight Control Sensors and Actuators

This section describes one of the analysis techniques used to predict the reliability of the IMS. Specifically, this analysis determines the probability, as a function of time, that the IMS will autonomously maintain a completely functional system of at least one master unit, one local unit, one remote unit, and one bus. Under these circumstances, the crew would have the capability to complete any mission; consequently, this analysis represents a prediction of the mission success probability. The analysis is accomplished by using Markov modelling techniques [1].

The IMS is modelled as a cascade of stochastically independent subsystems including two master units, two remote units, two local units, and two 1553B buses. It is assumed that full operational capability is maintained as long as one of each subsystem remains functional, and all failed units are automatically identified and reconfigured out of the system. A state vector for the Markov model is defined as

$$S = (M, L, B, R)$$

where

M = number of fault-free master units
L = number of fault-free local units
B = number of fault-free 1553B buses
R = number of fault-free remote units

The system is assumed to begin in state (2, 2, 2, 2) and to transition to other states as failures occur and reconfiguration is implemented. For this analysis, it is assumed that all failures are permanent, failures occur one at the time, and consecutive failures are separated by a sufficient length of time to allow reconfiguration to occur. In other words, the system does not encounter a failure while in the process of recovering from a failure. The state transition probabilities are determined from the various units' failure rates and fault coverage factors.

Because the state vector has four elements, (M, L, B, R), and each element has three possible values, (Ø, 1, 2), the resulting Markov model has 81 total states. However, many of these constitute a system failure and can be combined into a single "failed" state. After reduction, the complete model has a total of 17 states. The possible transitions from the initial state are illustrated in Figure 11. One of the interesting transitions is characterized by probability, p_7, and illustrates the failure of a single power supply unit. Because a master unit and a local unit are housed in the same physical enclosure, they are powered from a single supply, and a failure within that supply can disable both a master unit and a local unit.

The general form of the Markov model is

$$\underline{P}(t + \Delta t) = \Delta t A \underline{P}(t)$$

where

$\underline{P}(t + \Delta t)$ = probability state vector at time $t + \Delta t$
$\underline{P}(t)$ = probability state vector at time t
Δt = time step
A = state transition matrix

The model may be solved by iterating to determing $\underline{P}(t)$ at Δt time increments. The main interest is in the element of $\underline{P}(t)$ which specifies the probability of being in the failed state. Figure 12, for example, shows the probability of system failure for various coverage factors assuming that all units have equal coverage factors. The failure rates used are specified in Table 1. The bus failure rate has been assumed to be negligible because it covers only the cable and connectors (drivers/receivers are included in each unit's failure rate), and therefore it is small compared to other failure rates within the system. Figure 13 illustrates the reliability of the IMS using the specific coverage factors which were determined during the analysis and are

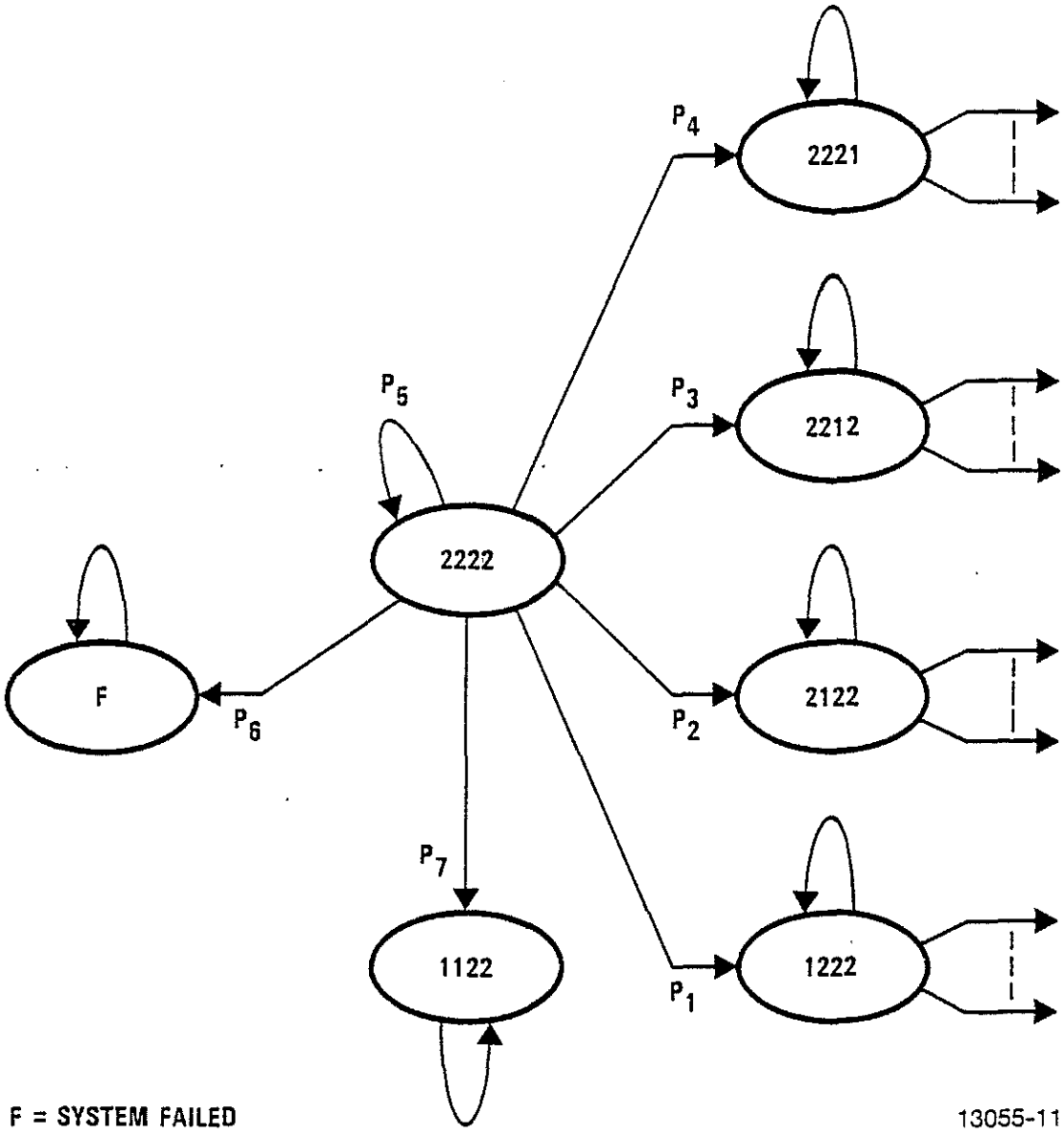

Figure 11. Transitions from the Initial State

presented in Table 1. As illustrated in Figure 13, the reliability at the end of any three hour mission is .9996 which corresponds to the IMS being able to complete approximately 2463 missions without encountering a failure which affects the capability to successfully complete a mission.

It is interesting to note that the analysis considers a bus to be the physical connectors and wires used to form the bus. Transmitters and receivers, for example, are associated with the master unit, local unit, or remote unit that they are located in. The analysis assumes that if a transmitter or receiver fails in a unit, the unit is completely failed. This is not true, however, because dual transmitters and receivers are provided in each box such that a box can tolerate a failure of one transmitter or one receiver without being adversely affected. This makes the reliability estimate pessimistic because the system actually possesses more redundancy than the analysis accounts for.

CONCLUSIONS

This paper has presented a systems level description of the design and analysis of the fault-tolerance characteristics of the A129 Integrated Multiplex System. The basic philosophy has been to implement the majority of the fault detection and redundancy management features in software as opposed to hardware. The resulting system has been shown, via a Markov analysis, to possess high reliability during a three-hour mission.

The general architecture of the IMS provides for significantly more fault-tolerance capability than has been utilized in the present design. Concepts such as software reconfiguration within each master unit may be implemented. Any hardware increases, however, would have to be traded off by examining items such as cost, weight, power consumption, and reliabilty.

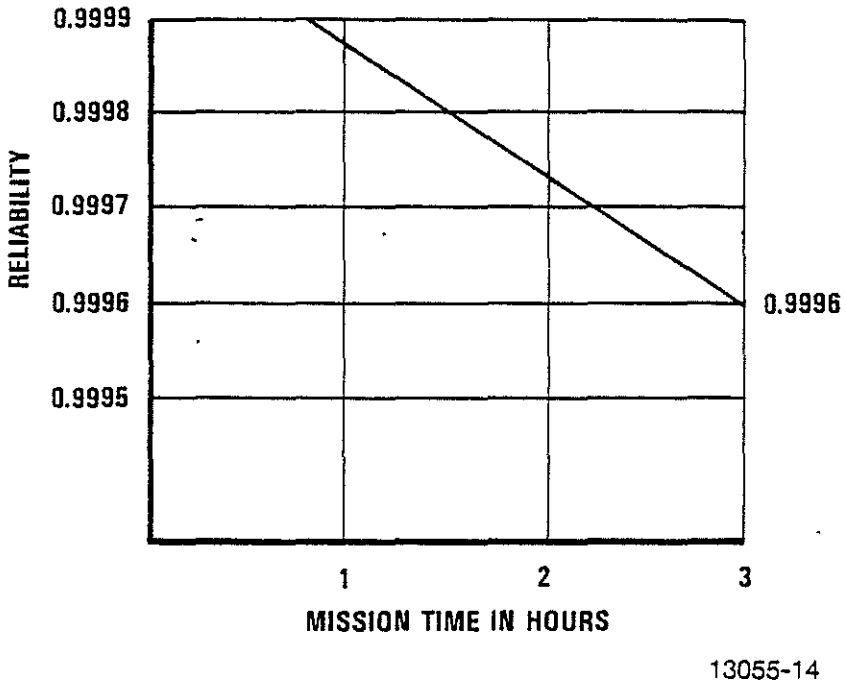

Figure 13. System Reliability as a Function of Time

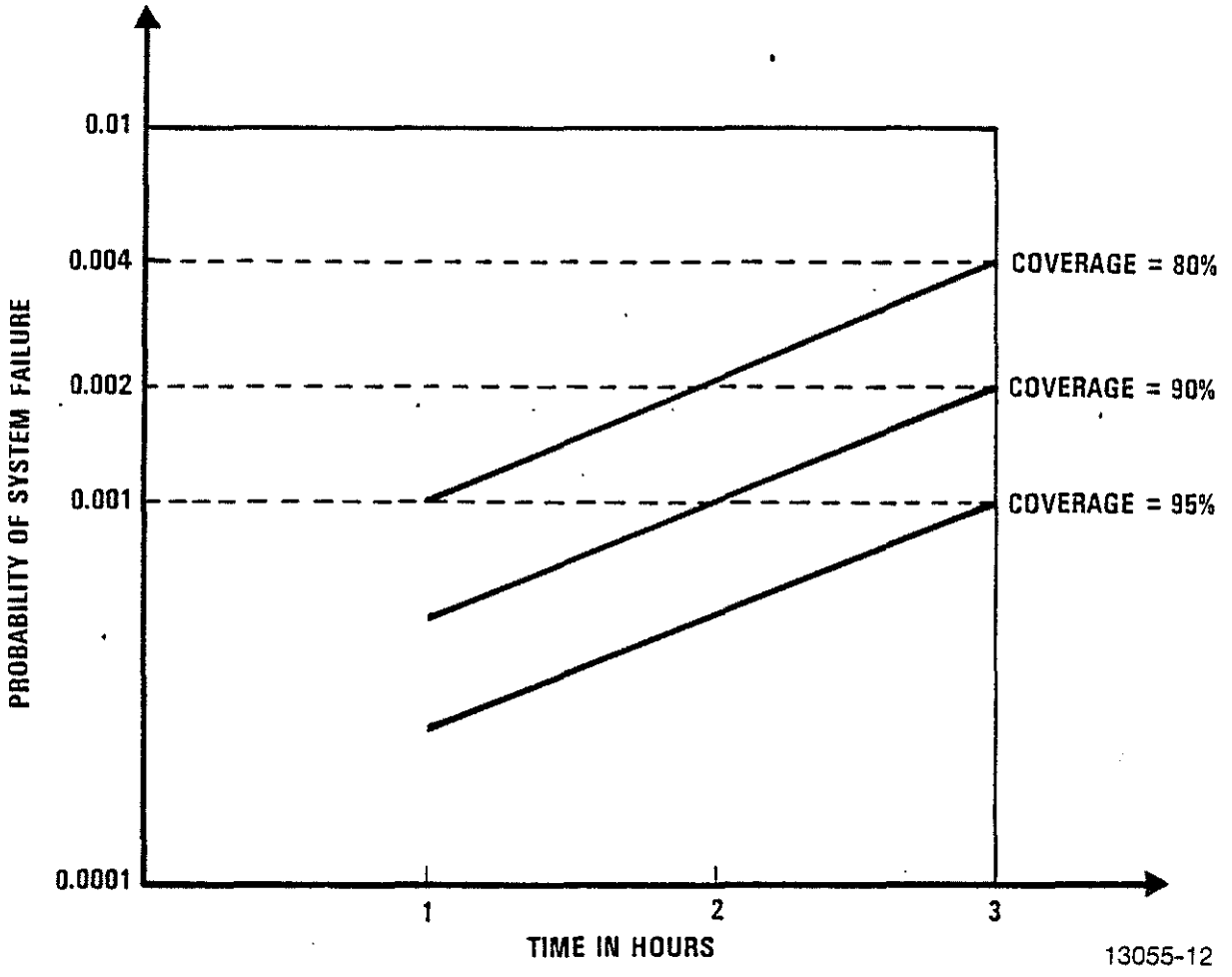

Figure 12. Probability of System Failure Versus Time

Table 1. Failure Rates and Coverage Factors for Each Element the Integrated Multiplex System

UNIT	FAILURE RATE (FAILURES PER HOUR)	COVERAGE (%)
MASTER UNIT	2.02×10^{-3}	96
LOCAL UNIT	8.80×10^{-3}	95
REMOTE UNIT	3.84×10^{-3}	99
POWER SUPPLY	56.2×10^{-6}	99

13055-13

REFERENCES

1. Daniel P. Siewiorek and Robert Swarz, The Theory and Practice of Reliabile System Design, Digital Press, Bedford, Massachusetts, 1982.