# MULTIPROCESSOR SELF DIAGNOSIS, SURGERY, AND RECOVERY IN AIR TERMINAL TRAFFIC CONTROL

W. Walther
Transportation Systems Software
Defense Systems Division
Sperry Univac

## 1. INTRODUCTION

### 1.1 GENERAL CONSIDERATIONS

The rapid growth of global aviation for business and pleasure has created the need for automated terminal systems of increasing complexity and capability. Continued increases in the aircraft population will require higher levels of automation. Sperry Univac is responding to this challenge with a multiprocessing system, including hardware and software, currently under development which will enable controllers to safely manage the crowded skies.

A new and unique multiprocessor executive is the heart of Sperry Univac's next generation, automated terminal system. The multiprocessing system configuration consists of a maximum of eight processors and sixteen 16,384 word memory modules. This special purpose executive provides three essential functions:

- Automatic self diagnosis of all processors and memory upon detecting a failure.
- Automatic partitioning (surgery) of a failed element (processor or memory module) from the system.
- Automatic restart of the system without loss of critical controller information.

The restarted system will operate with a functional capability commensurate with the remaining processor and memory elements. System simulation techniques are utilized to model system operating characteristics (e.g., task scheduling, data base structuring, etc.) as a means of evaluating executive overhead and memory requirements.

### 1.2 PRESENT DAY AUTOMATION

Early in 1969, Sperry Univac's Defense Systems Division was awarded a multi-year contract to provide the Federal Aviation Administration with 64 Automated Radar Terminal Systems designated ARTS III. Sixty-two of these systems are presently in operation at the country's busiest airports. The remaining two systems are installed at Federal Aviation Administration training and experimental facilities. The primary function of ARTS III is to assist controllers in safely and efficiently controlling aircraft within a designated terminal airspace.

There are three subsystems within ARTS III which combine to display vital realtime aircraft position, velocity, and altitude data for the controller. The three subsystems are the Data Acquisition Subsystem, the Data Processing Subsystem, and the Data Entry and Display Subsystem.

- The Data Acquisition Subsystem accepts analog beacon code signals (transmitted by a transponder on board the aircraft) from which it generates a range and digital code number, and transmits these, plus azimuth, to the Data Processing Subsystem for further processing.
- The Data Entry and Display Subsystem includes cathode ray display and keyboard entry devices which provide the man/machine interface between the controllers and the ARTS III automation equipment. The cathode ray displays present to the controller conventional radar and beacon sensor video supplemented with Data Processing Subsystem generated alphanumeric data. The keyboards permit the controllers to manually enter flight data into the Data Processing Subsystem and to request and control the display of the alphanumeric data.
- The Data Processing Subsystem accepts beacon reply data from the Data Acquisition Subsystem, flight information from adjacent Air Route Traffic Control Centers, and data entries from the controllers. The Data Processing Subsystem performs realtime tracking of beacon-equipped aircraft in the terminal airspace, processes flight information and controller entries, and provides Data Entry and Display Subsystem processing.

Figure 1 is a simplified diagram of an ARTS III single beacon tracking system showing the primary paths of information flow.

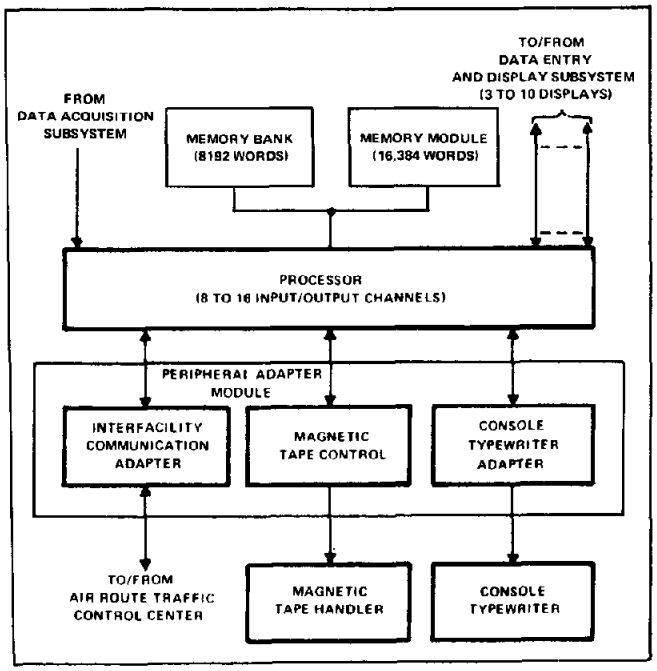


**Figure 1. ARTS III Single Beacon Tracking System**

An extremely significant feature of ARTS III is its capability for expansion in practical "building block" modules to meet additional air traffic requirements. However, in the basic system configurations, no automatic failure recovery capability exists. In these systems, if a processor or memory failure occurs then data processing stops and the controllers must revert quickly to the manual control procedures used before ARTS III. The controllers growing acceptance and dependence upon today's automated systems rapidly brings to focus the requirement for a fully automatic recovery capability.

### 1.3 PLANS FOR THE FUTURE

With the ARTS III systems providing a well-established base, the Federal Aviation Administration initiated a research and development program whereby the ARTS III systems will ultimately be enhanced through design, development, checkout, and testing in several important hardware and software areas. Major activities include:

- Radar tracking improvements encompassing both hardware and software.
- Beacon tracking system improvements.

- Multi-sensor tracking.
- Air terminal traffic control automation system simulation studies.
- Display hardware enhancements.
- Computer aided metering and spacing of aircraft into and out of the terminal area.
- Conflict detection and collision avoidance.
- Data processing subsystem multiprocessor modifications.

The pressing need for automatic failure detection and recovery, together with the addition of expanded capabilities and improvements, has declared requirements for a failsafe-failsoft multiprocessor executive program. As a result, a unique executive system was developed; it provides the software basis for the major ARTS III expansions mentioned in the preceding paragraph, while providing for automatic failure detection and recovery.

The Federal Aviation Administration has established an experimental facility at the Minneapolis-St. Paul, Minnesota International Airport. This facility is being utilized to evaluate these expansion activities.

This paper elaborates on the failsafe-failsoft multiprocessor executive system hardware and software which makes multiprocessor self diagnosis, surgery, and recovery in air terminal traffic control a reality. Before beginning, a definition of both failsafe and failsoft would be helpful in understanding the sections that follow.

- Failsafe -- In a failsafe system, even though major elements of the system fail, total system capability is maintained. The system will be capable of detecting specific failures in active elements and partitioning the failed element from the active system. Backup elements will be switched into the active system to replace a failed element. The failed element will be switched to a maintenance mode. All switching is automatic and will be accomplished without loss of operational performance.
- Failsoft -- In a failsoft system, when major elements of the system fail, system capability is maintained at a degraded mode. The system will be capable of detecting specific failures in active elements and deactivating the element. Backup elements will not generally be available. System performance will be degraded in a predetermined manner corresponding to the magnitude of the failure. The failed element will be switched to a maintenance mode.

## 2. MULTIPROCESSING SYSTEM DEVELOPMENT

### 2.1 GENERAL CONSIDERATIONS

The end product of the multiprocessing system development was a failsafe-failsoft multiprocessor executive program operating in and controlling a configuration consisting of Central Processor Modules (CPM), Input/Output Processors (IOP), Memory Modules, and a Reconfiguration and Fault Detection Unit (RFDU). The executive provides the overall control of the multiprocessor configuration, failure detection and failure recovery logic, and the execution of operational programs called tasks. These tasks perform the air traffic control function.

Development of the executive evolved through two major phases. The first phase concluded with a basic multiprocessor executive designed to operate in and control a configuration consisting of a maximum of eight processors (a combination of IOPs and CPMs) and up to sixteen 16,384 word memory modules. Formal demonstrations of the executive capability were provided after two multiprocessor IOPs were available at the Minneapolis-St. Paul, Minnesota experimental facility and again after the first CPM was installed. The software aspects of this phase were primarily concerned with providing effective control of air traffic control tasks in a multiprocessor environment. The major hardware aspects of the first phase were to design, develop, and install a Central Processor Module in the experimental facility and modify two ARTS III Input/Output Processors to incorporate several new multiprocessor-oriented instructions. The modifications provided the functions necessary to permit the IOP and CPM to operate efficiently in a multiprocessing environment. Instructions that were added include: relative address allocation and memory protection, relative interrupt steering, table access control, executive call, and processor biased load and store.

The emphasis of the second phase was to expand the basic multiprocessor executive to include the failsafe-failsoft capability. A failsoft executive was developed, tested, and demonstrated as an intermediate step leading to the failsafe-failsoft executive system. Phase two processor modifications provided the functions necessary to permit IOPs and CPMs to operate as a failsafe-failsoft failure detection and recovery system. A Reconfiguration and Fault Detection Unit was designed, developed, and installed in the experimental facility. Additional processors were also installed during the second phase. The failsafe-failsoft capability was demonstrated in a four IOP, two CPM, and six memory module configuration.

The failsafe-failsoft executive evolved through these phases so that an intermediate capability could be made available for operational use during the later stages of the total development effort. This approach has many benefits, one of which is that a complex program, such as this executive, was developed and proven step-by-step, with each step built on a firm predecessor. Additionally, operational software was developed and tested concurrent with continued executive development and evaluation.

The Sperry Univac Input/Output Processor and Central Processor Module both have the following general characteristics: one's complement integer subtractive arithmetic, 32-bit words (30 bits of data and 2 parity bits), internal control memory and nondestructive read only memory, memory addressing of 262,144 memory locations (either whole or half word), two accumulators and seven index registers, hardware instructions for stacks and queues, and relative addressing and memory protection.

The Input/Output Processor contains input/output chaining, while the Central Processor Module contains floating point and extended arithmetic and overlapped memory access. The load/store/add/subtract time is approximately two microseconds without overlap.

The failsafe-failsoft executive and the new hardware operate concordantly to perform the following functions under automatic processor control:

- Detect failures in the processor and memory hardware.
- Determine and disable the failed element.
- Reconfigure the processors and memory modules.
- Continue operation with the remaining elements.
- Provide the capability to diagnose and repair the isolated failed element, while the air traffic control system continues in operation.

### 2.2 SOFTWARE

2.2.1 GENERAL CONSIDERATIONS. Simplicity and cooperation guided the development of the executive system software.

Experience has shown that overemphasis on grandiose executive capability results in horrendous side effects: schedule slips, cost overruns, excessive executive busy time, and often times empty technical promises. Although highly sophisticated multipurpose executives can and do provide an environment quite suitable for the development, checkout, and implementation of operational programs, they do so at some expense. This executive is application-oriented in the sense that its notable functional capabilities are designed to satisfy explicit air traffic control goals. This dedication to rather specific objectives provided the framework to achieve simplicity, while providing all essential capabilities.

The ultimate failsafe-failsoft failure detection and recovery system was achieved only though highly coordinated efforts in each of the basic areas; hardware, executive software, and the air traffic control tasks. For example, in order for the executive to engage in effective failure detection and recovery action, it must make certain constraints on task programs. In turn, the executive must provide the tools necessary for the tasks to perform their assigned function effectively. The executive shares responsibility with the hardware for failure detection and recovery from IOP, CPM, RFDU, or memory failures. The tasks have the responsibility for failure detection and recovery from peripheral device failures. The executive plays a subordinate role in peripheral device failures. For example, a duplexed display channel may fail. If so, the task detects loss of communication, makes the decision to switch channels, and requests the switch via services provided by the executive.

2.2.2 OBJECTIVES. The following objectives were established as the major considerations for the specification of the executive:

- Have a storage requirement of less than 16,384 words of main memory.
- Have a common executive at all sites.
- Structure the executive to facilitate the modification, insertion, or deletion of tasks and data bases that comprise a system program (i.e., executive, tasks, and data bases).
- Structure the executive to facilitate failure monitoring, failure protection, and recovery.
- Structure the executive for failure monitoring, failure protection, and ease of recovery.

2.2.3 CONSTRAINTS. The following constraints were established to set forth how the objectives were to be achieved.

- The executive shall reside totally within online main memory.
- The executive shall be capable of being executed simultaneously by combinations of IOPs and CPMs.
- The order of task execution shall be specified by a lattice which will be represented as a sequence table in main memory.
- The executive shall control execution of periodic tasks, called planned tasks, defined in the lattice sequence table.
- The executive shall permit the assignment of specific processors to execute specific tasks.
- The executive shall permit the execution of aperiodic tasks, called popup tasks, specified in a table in main memory.
- An entire system program shall reside in main memory.
- There shall be no dynamic relocation of tasks in the system.
- The executive shall provide a variety of control and input/output services for tasks.
- The executive shall be capable of failsoft operation.
- The executive shall be capable of failsafe-failsoft operation (provided a backup IOP, CPM and memory module are available).
- Manipulation of input/output data and chains will be a task responsibility, whereas routing of input/output instructions to the proper IOP and proper channel shall be performed by the executive.
- Instruction modification shall not be permitted within either the executive or task instruction sets.
- A task must provide for its own initialization and must be restartable.
- The executive shall provide debugging aids and debug mode operation to permit the checkout and integration of tasks.

2.2.4 EXECUTIVE FEATURES. The major components of the executive are called modules. They are: Builder, Initializer, Scheduler, Executive Services, Interrupt Control (i.e., non-task input/output control), Debug and Recovery. The executive is divided into these definitive sections principally to provide functional and physical modularity.

The builder module operates in a unit-processor configuration and is strictly a pre-operational process. The recovery module is loaded and executed only after a failure has been detected. During execution of the recovery module no other executive modules or air traffic control tasks are executed. The balance of the modules constitute the operational executive. The scheduler, executive services, and debug modules may be executed in parallel and concurrently by two or more processors.

1. Builder Module — The builder provides the capability to process preamble data, process scheduling data, load and link the relocatable executive modules and relocatable tasks, load and process operator prepared corrections to tasks and data bases, and write a system program on magnetic tape in absolute format. Preamble data provides the builder with task loading information such as symbolic task or data base name, symbolic name of each task segment, base address of each segment, and length of each segment. Preamble data provides information used operationally including the designation of processors eligible to execute the task, memory lockout constraints, extension register values required at task entrance, type of task and associated entrance address, and data bases referenced. The scheduling data provides operational information describing the lattices for planned task execution.

   A builder control routine, loaded via tape bootstrap, provides the necessary man-machine interface to enable a user-operator to "build" a system program. Additional capability is provided to prepare mass memory (disc or magnetic tape) storage for failsafe-failsoft operating environments.

2. Initializer Module — The initializer provides the control mechanism for the accomplishment of all hardware and software initialization. One IOP directs the initialization process. Each task is provided the capability to set up interrupt control words associated with peripheral devices that it controls, and to initialize internal flags and variable data areas.

3. Scheduler Module — The scheduler provides for the dispatch of program control to tasks within the system program. Two distinct task scheduling algorithms are implemented: the popup scheduler and planned scheduler. The popup scheduler provides a high priority entrance to tasks which can be executed without concern for other tasks currently being executed by other processors. Normally, these tasks have unpredictable (i.e., aperiodic) execution entrance times. The planned scheduler provides an entrance to tasks which are periodic in nature and generally dependent upon the prior completion of other tasks. The planned tasks are those tasks whose entrance criteria are known "a priori". These two scheduling algorithms provide maximum scheduling flexibility and fast response while keeping executive busy time at a minimum. The planned task scheduler and popup task scheduler are completely separate. The schedulers are table driven; that is, all planned and popup scheduling inputs are contained in unique main

memory tables that can be operated on by any processor. Task entrance conditions are maintained in a common table. The primary function of both schedulers is the release of control to tasks; however, the schedulers load the task set of memory lockout register values and task extension register values before transferring control.

4. Executive Services Module – Executive services provides the capability to process all executive service requests, to control all console typewriter input and output messages, and to control disc storage and retrieval. An extensive set of executive service requests is available to tasks for the general capabilities of input/output control, scheduling control, privileged instruction execution, disc data storage and retrieval, and normal task exit. The console typewriter handler provides input control and output queuing for task/operator and operator/executive interface. A magnetic tape handler, also a part of executive services, provides a backup magnetic tape read and write capability when the disc is inoperable.

5. Interrupt Control – Interrupt control provides the capability to process all interrupts other than the task controlled channel interrupts. Specific capabilities include the processing of the program error and interprocessor interrupts.

6. Debug Module – Debug provides a collection of basic operating software to assist users in the on-line realtime debugging of operational tasks. Debug functions provided include: task timing data collection, software breakpoint (instruction address only), memory dump, change memory, masked search, suspend selected processor scheduling, history data collection, and hardware breakpoint (instruction and operand address). Debug permits the user to repeat any debug operations or to release function requests from the system. Debug operates in a multiprocessor environment and permits multiple debug function requests to be in the system simultaneously. The user is provided the capability to select functions from the console typewriter or card reader using a well defined set of function mnemonics. Debug resides in main memory as part of the executive.

7. Recovery Module – Recovery performs processor and memory testing following error detection, performs RFDU tests, computes and generates a processor and memory module resource map that identifies all "healthy" processors and memory modules, and selects the system program compatible with the remaining resources. Recovery passes certain operating parameters to the newly loaded system program. The recovery module resides on disc during normal operation and is loaded into main memory only after a failure is detected.

### 2.3 HARDWARE

This section presents a brief functional description of the Reconfiguration and Fault Detection Unit and the important processor modifications implemented during the second major phase of the development.

2.3.1 EXPANDED NONDESTRUCTIVE READ ONLY (NDRO) MEMORY. This modification expanded processor unique NDRO memory from 96 words to 1024 words. The error recovery techniques employed require the larger NDRO memory. The program implemented in NDRO memory consists of a limited processor diagnostic and memory test. Additionally each IOP's NDRO memory contains bootstrap loaders for the disc and tape subsystems.

2.3.2 SCATTER INTERRUPT. The scatter interrupt is used to alert all processors of a failure. Each processor is forced to the starting address of its NDRO memory. The scatter interrupt occurs whenever a processor detects a hardware fault and can also be issued by the software.

2.3.3 STRANGLE INSTRUCTION. The strangle instruction is used to deactivate a processor. The processor's timing chain is stopped to prevent it from engaging in any further processing.

2.3.4 PUSH/PULL P INSTRUCTION. The push/pull P instruction provides the capability to stack the program address register for subroutine nesting and for re-entrant programming. The stack control word is contained in processor control memory. The stack itself is maintained in control memory as specified by the executive. Subroutines may, therefore, be allocated to memory areas where only read access is permitted. This instruction obviates the need for a general register or main memory location for subroutine referencing.

2.3.5 RECONFIGURATION AND FAULT DETECTION UNIT. Basically the RFDU provides automatic partitioning of processors and memory modules by means of an 8 x 16 reconfiguration matrix. The function of the matrix is to partition the failed elements and enable the backup elements under executive control. The RFDU has manual override switches to positively isolate elements from the system (for maintenance or other "off-line" work) so that even program control cannot reengage them. These switches provide a means of isolating elements from the realtime system operation in the event of a RFDU electronic failure.

The RFDU monitors processor and peripheral cabinet air flow sensors to provide an advance warning of an impending shutdown. The central memory access module is monitored to detect processor time outs. A central memory access time out implies a processor failed to complete a write or replace cycle to a memory module. The RFDU provides a time monitor on the system program. The executive interrogates the RFDU status register at periodic intervals. If the interrogation is not made within a specific time, RFDU assumes that the system program is lost because of some undetected failure and a scatter interrupt is sent to all processors forcing them to NDRO memory where recovery is initiated.

The RFDU provides a realtime reference so that the executive can determine if the system realtime clock is working properly. In addition, the RFDU assists a processor in the performance of self tests on certain hardware error detection and interrupt capabilities. This is accomplished by commands from the processor directing the RFDU to generate errors.

The RFDU provides an interprocessor interrupt disable to each processor so that interrupts from an errant processor may be disabled by the recovery program. These disables also have manual override switches so that an operator can positively disable interrupts from any processor.

The RFDU also provides an audible alarm whenever an error is detected by the system. An additional alarm can be connected remotely to alert operators or maintenance men not in the immediate area.

The RFDU does not exercise control over the system. It is controlled by the system itself. For communication with the processors, the RFDU is configured in a manner similar to a memory module. All program control of the RFDU is implemented over memory communication paths. The RFDU assumes a 16,384 word memory module address location in the addressing structure of the processors.

## 3. FAILSAFE-FAILSOFT RECOVERY SEQUENCE

### 3.1 GENERAL CONSIDERATIONS

The myriad of potential causes for a particular failure is not directly analyzed; rather, the self diagnosis process determines the operable processors and memory modules. This is the kernel of the failure resolution approach. This unique approach to failure recovery was chosen because a malfunctioning element can invalidate any resident program in a multiprocessing system. When a failure is detected, all portions of the multiprocessing system are considered in question and no attempt is made to immediately trace the cause of the failure. All failures are treated identically, they are all essentially catastrophic.

Failures may be detected by either software or hardware. Software detected errors consist of processor time outs and illogical conditions. Hardware detected errors include memory parity and resume, memory lockout (read, write, or write on input), and power errors.

Certain types of data are considered necessary to the task to effect an orderly recovery. This data is defined as critical data. Task programs control and process a great deal of rapidly outdated data, called regenerative data. Regenerative data includes, for example, radar data, beacon data, and display data. Data which are used to manipulate regenerative data is called

supportive data. Regenerative data is not included in the task critical data since it is replenished immediately upon system program start or recovery restart. Supportive data, such as tables which are used to process regenerative data, is included in the critical data. Supportive data includes, for example, display and keyboard configuration information, beacon reply processing parameters, and system program identification. There is no critical data required for the executive.

Task programs utilize services provided by the executive to define critical data elements, request recording of the elements and following a failure to request the loading of critical data elements. The executive performs the input/output processing and data recording verification.

Critical data serves as the basic link between the sytem program operating at the time a failure is detected and the system program after recovery. The data is recorded on a mass memory device (disc or magnetic tape) periodically during normal operation in anticipation of failure. When a processor or memory failure is detected, all processors are driven to their own NDRO memory where the failsafe-failsoft recovery sequence is initiated. When a processor is operating in NDRO memory during an error recovery sequence, it assumes main memory cannot be trusted as a temporary memory storage area ("scratch pad") since the cause of the error may have been a memory failure. Hence, processor control memory is utilized as scratch pad during the recovery sequence. Figure 2 presents a simplified flow diagram of the failsafe-failsoft recovery sequence.

### 3.2 AUTOMATIC SELF DIAGNOSIS

The technique outlined below, implemented in NDRO memory, is the initial attempt to determine the "healthy" processors and memory modules.

Step 1 —Each processor executes a minimal instruction self test including conditional jump, load/store, and basic arithmetic. A processor stops if any test fails.

Step 2 —Each processor performs an address translation and memory access test. A processor stops if it cannot access a minimum number of memory modules.

Step 3 —Each processor implants its memory map in all memory modules that pass its memory test.

Step 4 —Each processor waits a specified time to allow all other processors to complete Step 3.

Step 5 —Each processor locates the memory map of the other cooperating processors (i.e., those processors not at a stop).

Step 6 —Each processor forms a processor map in all memory modules it can access.

Step 7 —Each processor waits to allow all other processors to complete Step 6. Processor maps are checked for identity before proceeding.

Step 8 —Each processor computes an intersection of all memory maps to determine the memory modules common to all processors.

Step 9 —Each processor loads its relative address registers corresponding to the result of Step 8.

When all processors complete Step 9, each surviving IOP cooperates to load only the recovery module from mass memory (disc or tape) into main memory. All processors then enter the recovery module thus completing the NDRO memory self diagnosis.

In some cases, the result of the self diagnosis is that all processors and memory modules are declared healthy. The cause of a failure may not be isolated to a processor or memory. In any case, however, the recovery module is loaded and the failure recovery sequence is continued.

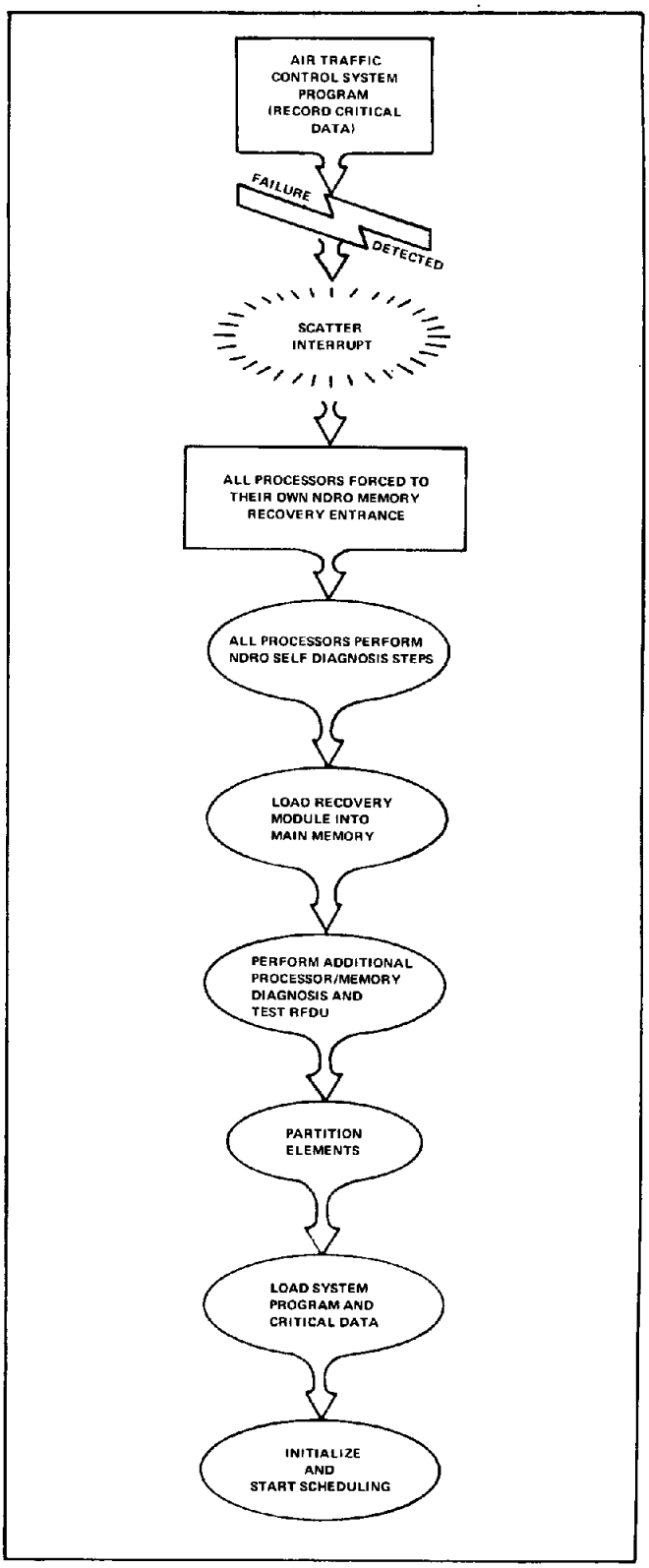


**Figure 2. Failsafe-Failsoft Recovery Sequence: Simplified Flow Diagram**

### 3.3 AUTOMATIC PARTITIONING (SURGERY)

The recovery module controls the execution of additional memory and processor instruction tests. This testing is designed to complete testing not treated by the NDRO memory self diagnosis program.

Processor testing is conducted in a progressive manner until all instructions and registers are thoroughly exercised successfully. Each phase of the test builds upon previously completed phases so that logic elements are tested before they are used. Processors that complete the test are further exercised to insure that each processor can communicate with each other processor via the interprocessor interrupt. This test is designed to insure that not only will each processor perform as a unit processor but also that each processor will operate in a multiprocessing environment.

Every memory module is tested by storing discrete bit patterns into each memory location followed immediately by checking each memory location for the correct pattern. A memory module must perform all operations correctly or it is excluded from the hardware system.

In addition, a RFDU test is performed to determine its status. A processor and memory module resources map is constructed reflecting the results of the combined self diagnosis (NDRO) and recovery modules testing. The RFDU acts as an electronic surgeon, electrically disconnecting those processors and memory modules from the system not contained in the computed resources map. All RFDU partitioning actions are commanded by a processor. The memory modules identified in the resources map are numbered 0 through N using the processor relative addressing registers.

### 3.4 AUTOMATIC RECOVERY

3.4.1 RELOAD. The automatic recovery sequence design is based on having various unique system programs stored on a mass memory device. The disc is used as the primary mass memory device. A system program is selected by the recovery module that provides the greatest functional capability for the remaining resources. The system is designed to permit one processor (specifically an IOP) and two memory modules as the minimum configuration. The system programs are preconstructed (by the builder module) and consist of the executive, tasks, and data bases in absolute format for direct loading into fixed main memory locations. A system program is loaded into main memory every time a failure recovery sequence is performed. Magnetic tape serves as backup to the disc.

The recovery module passes certain operating parameters and test results to the executive contained in the system program being loaded. The data is output on the console printer for operator evaluation after the recovery sequence is complete.

3.4.2 RESTORE. The recovery module transfers control to the executive, which is entered by all available processors. After the executive is initialized, it will release control to air traffic control tasks for task initialization. The tasks call for the retrieval of the latest critical data that was recorded before the failure. The tasks insert (reformat and modify if necessary) this data into task data bases.

3.4.3 RESTART. Task initialization is completed and scheduling reinitiated by all processors. Appropriate RFDU (switch settings) messages are output on the console printer, this being the final act of a recovery sequence.

The design goal for the completion of the recovery sequence was six seconds assuming 10,000 words of critical data and 90,000 words of program are loaded from the disc and 60 seconds if the magnetic tape system is used. The tape recovery time estimate is dependent upon the number of available system programs and the number of tape transports deployed. The recovery sequence is measured from the time a failure is detected until the time the executive enters its normal scheduling of tasks after reload.

### 3.5 FAILSOFT VERSUS FAILSAFE-FAILSOFT

The second phase of the executive development proceeded through two steps as stated in Section 2. First, a failsoft executive was developed providing automatic failure detection and recovery capability without the aid of a RFDU. However, the failsoft executive does utilize the processor modifications. A reconfiguration switch panel was implemented to provide manual partitioning capability. It should be noted, however, that the manual partitioning (i.e., isolating a failed processor or memory module) does not take place during the recovery sequence but after the recovery sequence is completed. The emphasis during the failsoft step was to develop the NDRO memory self diagnosis program, the recovery module, and provide the capability to recover critical data.

Secondly, the failsoft executive was upgraded when the RFDU and a redundant IOP, CPM, and memory module were available in the experimental facility. The fundamental concept of failsafe-failsoft is the availability of backup processors and memory modules. The failsafe-failsoft executive provides the capability to detect a failure in a processor or memory module, perform the recovery sequence, and restart without a loss in computing capability.

A maximum processor and memory module configuration is illustrated in Figure 3. The Data Acquisition Subsystem and Data Entry and Display Subsystem are shown to illustrate the dual path arrangement. These redundant data paths permit continued operation of the system in spite of a failure of one of the data paths. After restart the system reverts to the failsoft capability since a backup IOP, CPM, or memory module may not be available to switch into the system.

The capability is provided, as part of the failsafe-failsoft executive, for an operator to manually call diagnostics into main memory from the disc for offline or online testing of processors, memory modules, and peripheral equipment. The failsafe-failsoft executive segments the testing elements from the system, and the operator will manually insure segmentation utilizing the RFDU manual switches. After testing is complete, the malfunction corrected, and the correction verified, the operator notifies the executive of the availability of the repaired element. The executive will acknowledge the availability of the element.

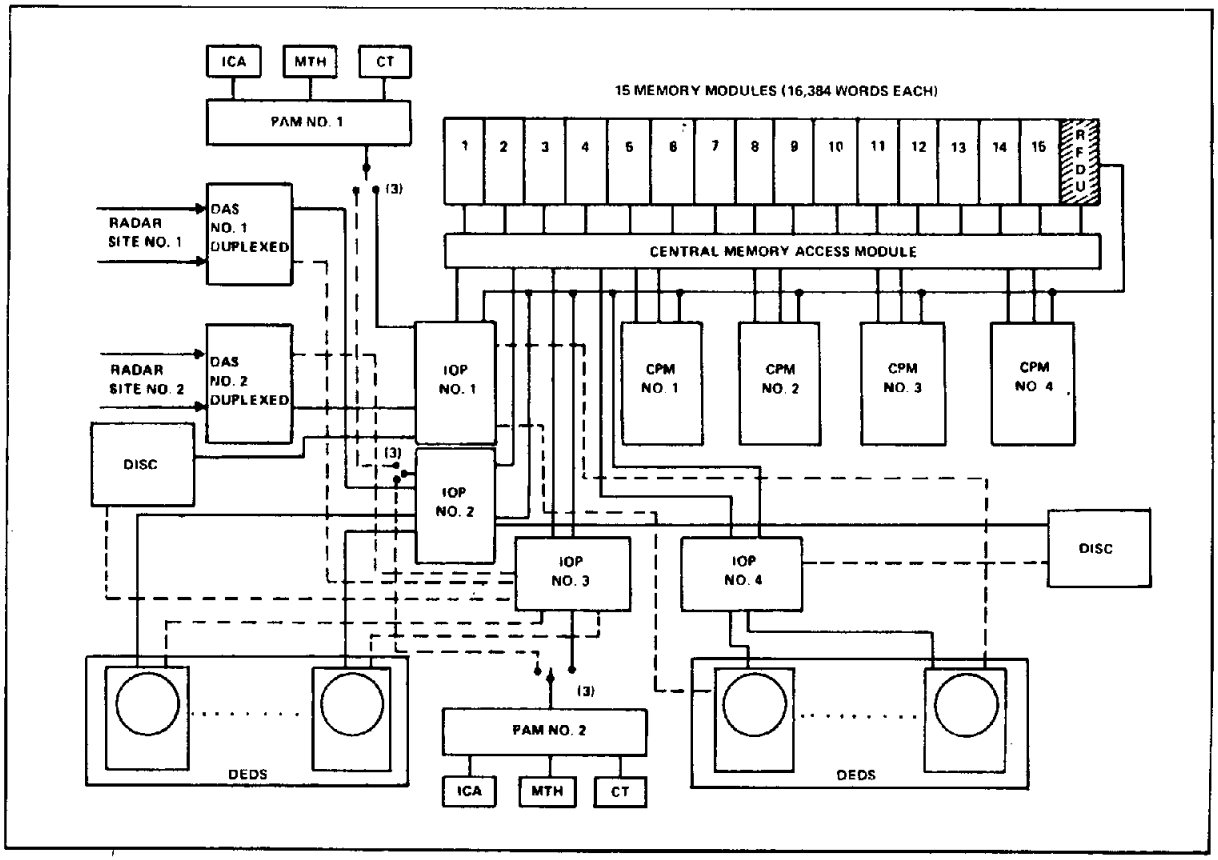


Figure 3*. Failsafe-Failsoft Configuration

*Legend for Figure 3

PAM – Peripheral Adapter Module
MTH – Magnetic Tape Handler
CT – Console Typewriter
ICA – Interfacility Communication Adapter
DEDS – Data Entry and Display Subsystem
DAS – Data Acquisition Subsystem
IOP – Input/Output Processor
CPM – Central Processor Module
RFDU – Reconfiguration and Fault Detection Unit

## 4. SYSTEM SIMULATION

Simulation via computer modeling affords certain major advantages in the design and evaluation of the air traffic control systems. It permits before-the-fact system-software evaluations and performance projections. It also provides data to be utilized in construction or modification of the various failsafe-failsoft system programs and configurations. A software simulator (computer model) was developed and used for configuration analyses and to provide statistical data to evaluate the executive program design. The executive portion of the model was developed concurrent with the actual executive program development.

This system modeling activity provides data concerning executive busy time, input/output handling (processing), and overall system utilization. Other system operational aspects addressed included memory map and task lattice structure evaluations. The memory map evaluations assisted in the reduction of queuing at memory. Modeling of the task lattices verifies or enhances lattice structures by assuring that all tasks are executed within required time increments, while providing an acceptable level of redundant capability for failure response. The system model was further used to evaluate the feasibility or impact of adding enhancements to the total air traffic control system.

## 5. SUMMARY

In order to meet the increasing need for more complex automated air terminal systems, Sperry Univac developed a unique failsafe-failsoft multiprocessor executive program that will serve as the software basis for enhancements to present day ARTS III systems. The executive provides the overall control of the multiprocessor data processing subsystem, failure detection and recovery sequence, and the execution of the air traffic control tasks.

This paper describes the need for and background of the development of the multiprocessor executive; in addition, it presented failsafe-failsoft multiprocessor failure detection and recovery design.

## 6. ACKNOWLEDGEMENTS

The author wishes to acknowledge A.A. Westerhaus, ARTS Enhancement Software Project Engineer and K.D. van Duren, Principal Systems Design Engineer for many of the concepts and techniques described in this paper. For encouragement and advice in the preparation of the paper, the author thanks J.T. Rye, Senior Program Analyst, Transportation Systems Software.