AD-A014 102

DEVELOPMENT PROGRAM FOR AN AIRCRAFT RELIABILITY AND MAINTAINABILITY SIMULATION (ARMS) MODEL. VOLUME I. PROGRAM DESCRIPTION

William C. Friese

RAIL Company

Prepared for:

Army Air Mobility Research and Development Laboratory

July 1975

USAAMRDL-TR-75-26A

251074

DEVELOPMENT PROGRAM FOR AN AIRCRAFT RELIABILITY AND MAINTAINABILITY SIMULATION (ARMS) MODEL

Volume I - Program Description

RAIL Company
Executive Plaza III
Hunt Valley, Md. 21031

July 1975

Final Report for Period June 1974 - December 1975

Approved for public release; distribution unlimited.

Prepared for

EUSTIS DIRECTORATE
U. S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY
Fort Eustis, Va. 23604

142

EUSTIS DIRECTORATE POSITION STATEMENT

A simulation model has been developed which permits operational, reliability, and maintainability analyses to be made for an aircraft system throughout the life cycle of that system. This model, the Aircraft Reliability and Maintainability Simulation (ARMS) model, is a flexible analytical tool that can be applied to a wide range of simulation experiments without the need for reprogramming. The model is augmented by a series of input programs that perform extensive data checking and provide diagnostics to aid the user in preparing data for the model. An output program is also provided to give the user control over the output data selection and formatting process. The entire series of programs is designed to be used without any knowledge of the programming languages involved.

The conclusions and recommendations contained herein are concurred in by this Directorate. The simulation model described herein is a tool for examining the impact of proposed policies on the reliability and maintainability parameters of an operating system. It is therefore useful in performing studies for project managers and other agencies, and should also aid R&M engineers involved in the design of large systems.

The technical monitor for this contract was Mr. Timothy D. Evans, Military Operations Technology Division.

DISCLAIMERS

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

Trade names cited in this report do not constitute an official endorsement or approval of the use of such commercial hardware or software.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return it to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: USAAMRDL-TR-75-26A	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): DEVELOPMENT PROGRAM FOR AN AIRCRAFT RELIABILITY AND MAINTAINABILITY SIMULATION (ARMS) MODEL, Volume I - Program Description	5. TYPE OF REPORT & PERIOD COVERED: Final Report June 1974 - December 1975
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): William C. Friese	8. CONTRACT OR GRANT NUMBER(s): DAAJ02-73-C-0090
9. PERFORMING ORGANIZATION NAME AND ADDRESS: RAIL Company, Executive Plaza III, Hunt Valley, Maryland 21031	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62205A 1F162205A119 07 005 EK
11. CONTROLLING OFFICE NAME AND ADDRESS: Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia 23604	12. REPORT DATE: July 1975
	13. NUMBER OF PAGES: 142
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Volume I of two volumes

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Military Aircraft
Army Aircraft
Reliability
Maintenance
Aircraft Maintenance
Models
Mathematical Models
Computerized Simulations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Aircraft Reliability and Maintainability Simulation (ARMS) model concept was developed by the U. S. Army Air Mobility Research and Development Laboratory, Eustis Directorate. The ARMS model is a management tool which permits observation of the impact of a proposed action prior to implementation. The model is used to simulate aircraft operating in user-defined operational and maintenance scenarios. It is designed to allow the user almost complete

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 - continued

flexibility in defining aircraft components with their associated failure rates and repair requirements, and in defining necessary resources such as ground support equipment.

The ARMS model can be applied throughout the life cycle of an aircraft system from the conceptual phase, through the developmental, and during the operational phase. It can be used to determine the systems level impact of changes in reliability and maintainability parameters at the component level, to determine the effect of various TOE combinations on aircraft availability, to determine the optimum mix of maintenance resources, and to determine the effectiveness of alternative maintenance concepts. Additional uses include evaluation of the R&M aspects of proposed modifications and the evaluation of the effect of new safety and survivability design concepts on aircraft combat damage maintenance requirements and mission completion rates. The ARMS model can also provide quantitative R&M data for input to cost and effectiveness analyses.

The use of the ARMS model will provide a structured analysis of the interrelationships between reliability, maintainability, availability, utilization frequency, mission success, maintenance policies, maintenance manpower and logistics support. In addition, the ARMS model will permit evaluation of the risk factors associated with failure to meet various levels of future operational goals.

Potential users will find sufficient detail in this report to provide a complete understanding of ARMS. Each major model program is described in three increasingly detailed sections. A general operation section provides an overview of each program. A narrative is provided along with logic flow diagrams to provide details of what each subroutine in the model will accomplish. Finally a detailed logic description is presented which fully explains how each routine is logically implemented.

PREFACE

This report presents the results of a development program to produce a user oriented, highly flexible simulation model performed under Contract DAAJ02-73-C-0090 with the Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia. Mr. Timothy Evans served as the Contracting Officer's Technical Representative and made significant contributions throughout the program. Messrs. Howard Bratt, Larry Sackman and Frank Tabor also provided valuable assistance and guidance during various phases of the effort.

The project engineer for RAIL Company was Mr. William Friese. Principal investigators were Messrs. John Florence and Richard Engelhardt.

The Bell Helicopter Company was a co-contractor in the development program. Significant contributions were made by Messrs. Donald Laingor, George Knudson and James Marsh.

TABLE OF CONTENTS

Preceding page blank

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

INTRODUCTION

BACKGROUND

As operational Army aircraft increase in sophistication, the estimation of their reliability, maintainability, required support and operational capabilities also becomes increasingly complex. In particular, operational relationships of these factors become impractical or impossible to analytically investigate. Simulation techniques, however, can be utilized to better understand these problems and to gain an insight into the significant interactions between the aircrafts' reliability, maintainability, operability and supportability characteristics.

Simulation techniques have been previously employed to successfully estimate these parameters for Army aircraft. Prior models, however, required significant revision and computer language reprogramming to change from one simulation experiment to the next. This process is costly and involves skills not necessarily readily available in analysis groups.

The ARMS model concept has evolved from the previous simulation experience of USAAMRDL. An R & M simulation of a UH-1 helicopter scenario was developed under Contract DAAJ02-72-C-0090. This model was improved and made more generalized under Contract DAAJ02-73-C-0031. Even the more generalized model lacked the flexibility to easily accommodate revised aircraft types or maintenance and operational scenarios.

OBJECTIVES

The present program was undertaken to provide an analysis tool with sufficient flexibility so that a wide range of simulation experiments can be made without the need for model programming revisions. An additional objective was to establish the model input formula in "English language". This would eliminate the requirement for simulation language coding of input data. A standard set of input data sheets was derived that can be converted directly to key punch input after the analyst has defined his particular simulation experiment.

The final major program objective was relative to the simulation model output. Since the model was to be extremely flexible and capable of handling a wide range of experiments, it followed that collecting and printing all output which might be of interest would result in voluminous data. This is not always desired. The ability to select beforehand the key data outputs required, automatically suppressing the remainder, was set as a final major program objective.

APPROACH

A program was established that would maximize the probability of achieving established objectives. Three major efforts were undertaken. First, a definition phase was initiated to investigate, specify and bound the logical limits of the simulation model. This phase established the boundaries of flexibility that could be provided in the model. A set of use-oriented input data forms was prepared.

The second phase involved two parallel efforts: model logic programming and data preparation. These were timed to provide a complete set of input data and a runnable model concurrently. The data preparation effort was under separate contract, and reporting of its outcome is beyond the scope of this report.

Phase three was the execution phase. The model and input data were merged and underwent extensive debugging. Verification of model logic was obtained by comparing output results to the known aircraft operational characteristics reflected by the input data base.

Presented in this report are comprehensive descriptions of the simulation model produced. Logic is described in a narrative sense which will provide a detailed description of what the model will do. Also provided are descriptions of computer program logic to a degree of detail that will enable a reader with prior knowledge of the GPSS programming language to follow how the capabilities described in the narrative are logically implemented.

MODEL OVERVIEW

GENERAL

The model produced by this development program provides an analyst a flexible tool that can be exercised with minimum operator intervention to perform simulation experiments over a wide range of scenario and reliability/maintainability data input. This objective is achieved by combining various programs cataloged as permanent data sets with input cards, tape input/output and temporary data sets. Job control can be established so the program will execute, barring errors, from start to finish.

Implementation of the model required programs in two computer languages: FORTRAN and GPSS. FORTRAN was used for data input and report output processing. The actual simulation program is written in GPSS. Interface between input programs and the GPSS program is accomplished by use of the IBM cataloged IEUB update routine to create a temporary data set consisting of processed input data merged with a permanent data set containing the simulation program.

Output interface is achieved by use of the GPSS HELP option. This allows transfer of GPSS output data arrays to magnetic tape for processing by the output FORTRAN routine.

In all, the complete program contains five major components:

- SSDDIP - Simulation Scenario Description and Data Input Program - FORTRAN
- ARMS - Aircraft Reliability and Maintainability Simulation - GPSS
- ARMDAT - ARMS Data Output - FORTRAN (via GPSS HELP option)
- FORMSOUT - Format of SDOSFI Output - FORTRAN
- SDOSFI - Simulation Data Output Selection, Format and Identification - FORTRAN

Table 1 contains the computer assets required to exercise the various programs. The values given are for a nominal one-month scenario and can vary considerably depending on the complexity of the scenario to be simulated. In particular, computer assets required are most sensitive to the number of elements and mission building blocks defined by input. Appendix IX presents algorithms which are useful in estimating computer assets.

TABLE 1. COMPUTER ASSET REQUIREMENTS

Program	Nominal Computer Core	Nominal CPU Time
SSDDIP	206K	12 Minutes
ARMS and ARMDAT	580K	50 Minutes
SDOSFI and FORMSOUT	242K	5 Minutes

TOP LEVEL OPERATION AND FLOW

A clearer understanding of total model operation can be obtained by reference to Figure 1. This figure depicts schematically the flow of information in the total ARMS program.

Preprinted data input forms are prepared by the analyst to describe the simulation experiment. These forms have the capability of varying the reliability and maintainability parameters, operational scenario, maintenance assets and maintenance scenario. A set of data input forms is contained in Volume II. The completed input forms are processed by a key punch operator to produce an input deck of cards. The card deck is then read into the computer to create a temporary data set for disc storage. As an option, an input tape file may be created. This can be of help in cases where repeated experiments are to be made with minor input changes. It reduces the time required to process large numbers of input cards through a card reader.

Having created a temporary data set from tape or cards, the program proceeds by merging this information with the SSDDIP program stored on discs as a permanent data set. Raw input data is also stored on partition one of a magnetic tape file.

The merged data sets are then processed by the three phases of the SSDDIP program. These three phases are Card-Edit, Cross-Edit and Arithmetic routines. The program will automatically transition from one routine to the next if no errors are detected. The output from each routine is also stored on separate partitions of the magnetic tape file. Each succeeding routine uses information created by the preceding routines.

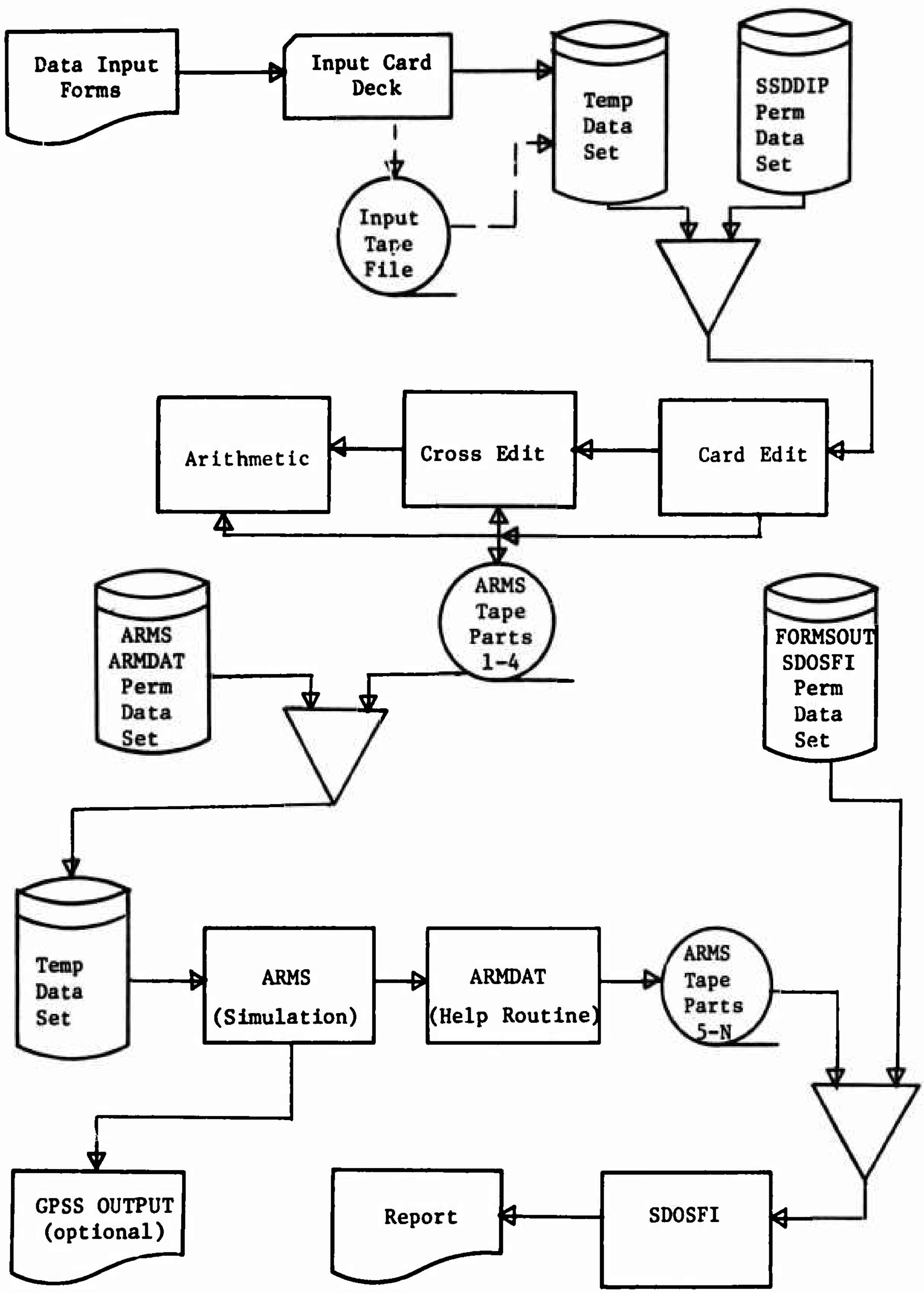

Figure 1. Top Level Flow.

The Card-Edit program checks individual input cards for error conditions. Errors caused by contradictory or missing information from separate cards are detected by the Cross-Edit routine. The Arithmetic routine processes input data and creates the GPSS format necessary for the simulation model.

At the conclusion of the SSDDIP program, the ARMS tape will contain four partitioned data sets. Partition four contains the program output formatted for input to the simulation model. This information is merged with the ARMS simulation program stored on discs as a permanent data set to create a temporary data set containing all input data and program logic.

From this point the simulation will proceed in accordance with the user-specified run control information. Output is produced at the interval specified by the user. At each output interval the ARMDAT routine is used to format GPSS output for storage on partitions five and up of the ARMS magnetic tape. The ARMDAT program is accessed via the GPSS HELP option. A full set of GPSS output may also be obtained for each simulation interval.

Partitions five and up of the ARMS magnetic tape create the input to be used by the SDOSFI program. Tape information is merged with program logic stored as a permanent data set on discs, and the output program is exercised to create a report in accordance with user-specified instructions. Volume II contains a set of the user forms which are used to specify the desired outputs.

Each of the program steps mentioned above is described in much greater detail in the following sections of this report. In each case a narrative will be provided which describes the programs so that they may be understood by readers unfamiliar with programming languages. Also provided are sections which give detailed logic descriptions such that a reader familiar with the programming languages can obtain a complete understanding of the logic implemented in the model.

ARMS

This section of the report presents a description of the Aircraft Reliability and Maintainability Simulation (ARMS) model. Three levels of detail are provided. A general operation section provides a quick, top level overview of the simulation. More detail of individual routines and subroutines is presented in a narrative section. Both of these sections may be understood by readers without knowledge of the programming language. Finally, a detailed logic description is provided that will enable a reader with a knowledge of GPSS programming to understand how the simulation is implemented.

GENERAL OPERATION

ARMS, an aircraft simulation model, was developed to analyze the capabilities and support requirements of Army aircraft. The potential of this model lies in its capability to simulate a complex scenario and provide numerous output data concerning the aircraft's capability to perform in the given environment.

GPSS V, the model language, employs logic, numeric, and Monte Carlo techniques and features broad logical power, reasonable computer running time, and relative ease of model construction and use. The language permits the total weapon system to be analyzed dynamically by evaluating the capability of the weapon system and its support system to meet mission requirements. Since the intricate complexities of the mission and support system are constructed in the model itself, only standard analysis inputs and statistics are required.

Considerable deliberation was given to the selection and development of the various model routines, the intent being to introduce maximum flexibility and to enable revision by merely changing values of input data.

A high degree of realism is employed in the logic and flow of ARMS. The input data (MTTR, MTBF, Maintenance personnel quantities and skills, etc.) may be defined in the model at a level consistent with the user's purpose (i.e., aircraft definition at the subsystem level may be sufficient for a conceptual study, whereas an operational study would probably require an aircraft definition in greater detail). A reasonable number of influential perturbations and events have been logically introduced throughout the model to account for such things as GSE delays, maintenance actions, preventive maintenance items, etc. These events may be introduced with an appropriate distribution and are subjected to Monte Carlo techniques to obtain a high degree of realism. Finally, the model output has been modularized so that the user may specify the quantity and types of information desired.

The ARMS model consists of three major groups of logic: Control Logic, Aircraft Mission Logic and Aircraft Maintenance Logic. Each logic group is subdivided into various routines and subroutines. Details of each group are outlined below.

I. CONTROL LOGIC BREAKDOWN

   A. Stabilization Routine
   B. Run Control Routine
   C. Regular/Surge Control Routine
   D. Shift Control Routine
   E. Inactive Time Routine
   F. Plot Data Routine

II. AIRCRAFT MISSION LOGIC BREAKDOWN

   A. Aircraft Generation and Initialization Routine
      1. TBO Initialization Subroutine
   B. Mission Schedule Control Routine
      1. Scheduled Mission Schedule Subroutine
      2. Random Mission Schedule Subroutine
      3. Continuous Mission Schedule Subroutine
      4. Alert Aircraft Schedule Control Subroutine
   C. Mission Generation and Launch Routine
   D. Mission Calls by Other Missions Routine
   E. Aircraft Selection Routine
   F. Aircraft Mission Events Routine
      1. Aircraft Reconfiguration Subroutine
      2. Aircraft Ground Events Subroutine
      3. Aircraft Flight Events Subroutine
   G. Aircraft Failure Events Routine
      1. Aircraft Failure Determination Subroutine
      2. Red X/Blue X Subroutine
      3. Aircraft Replacement Subroutine
   H. Mission Manpower Routine
      1. Manpower Determination Subroutine
      2. Mission Manpower Subroutine
      3. SMA Manpower Subroutine

I. Aircraft Accounting Routine

1. Mission Accounting Subroutine
2. Failure Detection Subroutine
3. Aircraft Return Subroutine

III. AIRCRAFT MAINTENANCE LOGIC BREAKDOWN

A. Supply Routine

1. Incoming Parts Subroutine
2. Probabilistic Supply, NORS Subroutine
3. Deterministic Supply, NORS Subroutine
4. Cannibalization Subroutine

B. Scheduled Maintenance Routine

1. Scheduled Maintenance Initialization Subroutine
2. Scheduled Maintenance Induction Subroutine
3. Scheduled Maintenance Planning Subroutine
4. Daily Inspection Control Subroutine

C. Aircraft Repair Routine

1. Incorrect Diagnosis Subroutine
2. Parallel Maintenance Subroutine
3. MOS Assessment Subroutine
4. GSE Acquisition Subroutine
5. GSE Return-Repair Subroutine
6. Off Equipment Repair Subroutine

D. Aircraft Maintenance Accounting Routine

1. Maintenance Action Logging Subroutine
2. Maintenance Control Subroutine
3. SMA Control Subroutine
4. Test Hop Control Subroutine

The control logic contains all routines necessary to perform overall timing and coordination in the model. It can be seen from the outline that these routines operate independently but have control over a large portion of the model. For example, the shift control routine acts independently to regulate working hours as specified by input data. It indirectly impacts any other model logic requiring manpower by establishing what MOS is available at any given time.

The aircraft mission logic implements the major portion of the operational scenario specified by the user. Various routines establish and control mission schedules and launches. Aircraft are selected, flown and tested for failure and failure consequences. Manpower is obtained and controlled for all mission-related events, and accounting is maintained on individual aircraft.

The third major logical area, Aircraft Maintenance, implements the scheduled and unscheduled maintenance scenarios. It controls the usage of all maintenance assets including supply, manpower and GSE. Accounting is maintained for all maintenance-related information.

The ARMS Top Level Logic Diagram (Figure 2) depicts the interrelation between the major routines and some of the major functions and decisions considered during the course of a simulation. The model time increment at which the master clock is updated has been set at 1 minute. This time increment provides the degree of output data definition required for trade-off studies, parametric analyses, etc., where time deltas are normally small.

Referring to Figure 2, the operational scenario is controlled by mission scheduling logic. Flying schedules for all types of missions are established 48 hours in advance. Aircraft are called for the beginning of mission ground events at a time computed in the model. This computed time is self-adaptive to maximize the number of launches successfully achieved. When queueing situations occur, the time of call is moved further from takeoff time.

Aircraft are selected for flight based on pending scheduled maintenance requirements. They are obtained primarily from the ready pool, but may be obtained elsewhere when circumstances dictate.

Missions are defined by input data in "segments". Each segment has a time interval and a segment designator of ground, flight, combat, or remote ground. This combination allows the user to construct a wide variety of mission types by combining the defined segments. The model logic processes each mission segment by segment. Failures occur and are detected on a Monte Carlo basis at each segment. Aborts and their consequences are also established by Monte Carlo process. Consequences are selected based on input probabilities and the segment designator. For example, a "crash" consequence is not allowed to result if the aircraft is in a ground segment. Capability exists for introducing combat damage in addition to reliability failures.

When an abort occurs, it may require additional missions to be flown. If the group of aircraft flying the mission is reduced by the abort below a specified minimum, an attempt is made to obtain a replacement. Also, several abort consequences require an aircraft, for example, a rescue mission. Capability also is provided for each type of mission to generate a demand for other missions. For example, a recon

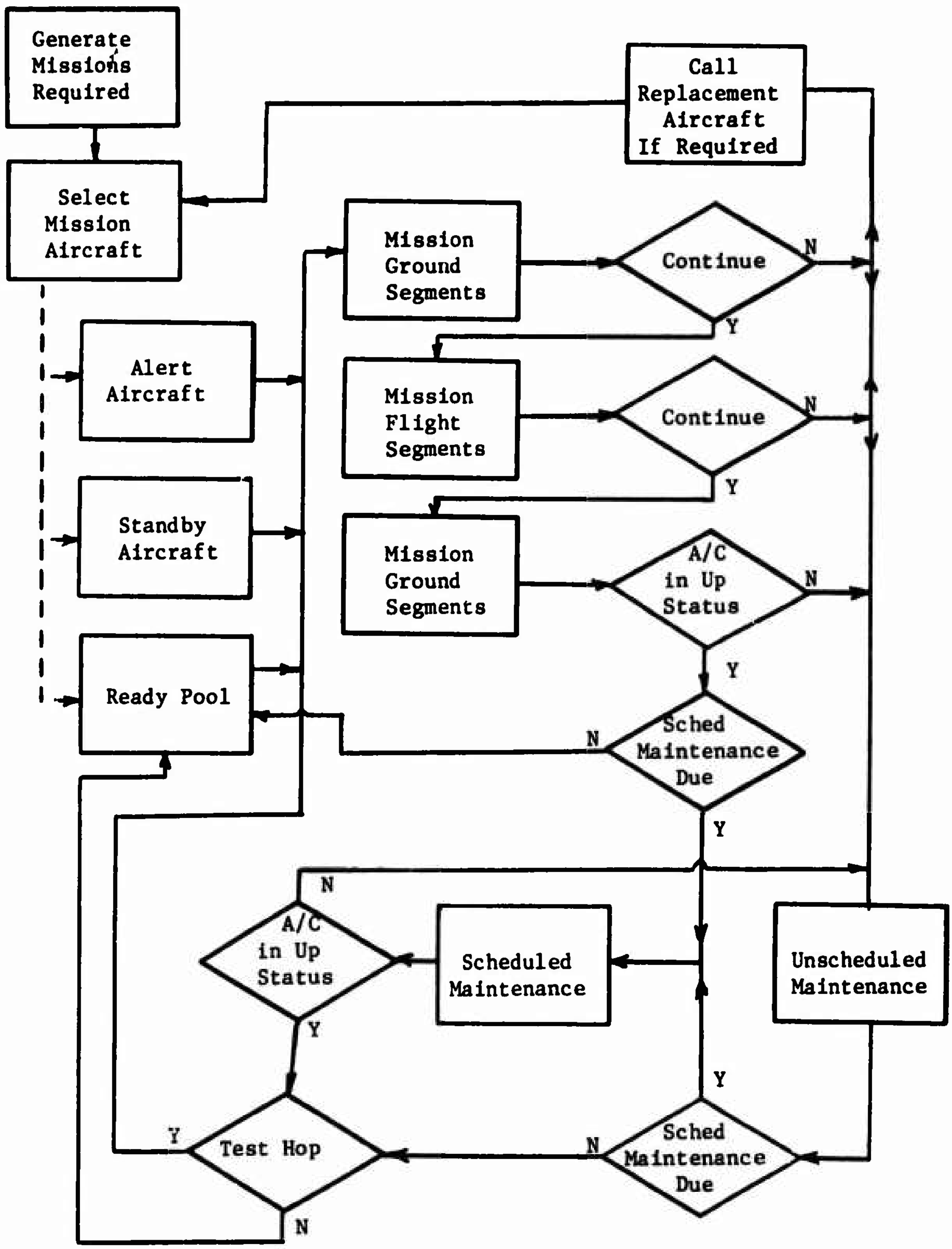

Figure 2. ARMS Top Level Logic Diagram.

mission may call a strike. These calls are established by Monte Carlo process using input probabilities.

As each aircraft returns to base, from either an abort or a completed mission, its maintenance status is interrogated to establish one of three destinations. These are the ready pool, unscheduled maintenance or scheduled maintenance.

Scheduled maintenance requirements are completely specified by the user. Up to twenty different scheduled events plus a daily may be defined. Inspection frequency is a function either of flight time or of calendar time. In addition, time between overhaul (TBO) items may be defined. Assets required to perform th maintenance, manpower and GSE are defined by the user. The model tracks flight and calendar time on each aircraft and inducts them into scheduled maintenance as required.

The frequency of unscheduled maintenance is determined by Monte Carlo process from the reliability data provided for each aircraft element defined by the user. Reliability data is combined with operational data to achieve the specified failure per flying hour rate. All failures occur during mission segments and are detected during missions or scheduled maintenance in accordance with input data. Detected failures are classified as "Red X", "Blue X" or "upsquawk". Red and Blue X failures may cause aborts and always cause the aircraft to be inducted into unscheduled maintenance. Upsquawks are allowed to accumulate up to a user-specified limit before downing the aircraft.

Manpower, supply and GSE requirements are specified for each maintenance action. The user may classify each element's maintenance as a "remove and replace" or "repair in place" action. Provisions are also made to input probability of incorrect diagnosis and incorrect repair for each element. Off equipment repair may also be completely specified by the user. Maintenance at up to four echelons may be defined.

Supply data for each element may be provided in a probabilistic or deterministic manner. Probabilistic supply is handled by a Monte Carlo process. When a deterministic supply scenario is chosen, stock levels and reorder points are specified and parts usage is tracked by the model to determine element availability. Cannibalization logic is included and is used to provide parts if permitted by user-specified input.

Another feature provided by the model is the ability to define scheduled or unscheduled maintenance events as "significant maintenance actions" (SMA). This feature is provided so that extensive maintenance actions may be input in segments. Each segment may be defined in terms of duration, manpower and GSE.

Test hop requirements may also be input. These are accomplished, as defined, after scheduled or unscheduled maintenance is performed.

ROUTINE AND SUBROUTINE NARRATIVE DESCRIPTION

This section presents a narrative description of each routine/subroutine of the simulation model. Details of the decision processes and data input utilized are given. Also included are flow diagrams of each routine.

Stabilization Routine

The purpose of the stabilization routine is to exercise the model scenario, prior to recording data, until a steady-state maintenance posture has been achieved. This routine uses input data from card type "A". One of two stabilization techniques may be specified - automatic or programmed.

When programmed stabilization is chosen, the model is exercised for the time period specified and then continues until the next Sunday midnight. At this point, stabilization is flagged to other model logic, aircraft timing parameters are set, and output data is cleared. Control passes to the run control routine.

The automatic stabilization option samples the number of deferred maintenance actions three times each day. Once daily, the tabulated data is tested for stabilization. The model is considered stabilized when the smoothed exponential mean of the number of deferred maintenance actions is within the standard error of the mean value of sampled data.

Figure 3 is a flow diagram of the stabilization routine.

Run Control Routine

The purpose of the run control routine is to establish the data collection periods specified by the analyst. Data collection can be specified as a single time interval or divided into subintervals of equal time. Data for the run control routine is obtained from card type "A". Figure 4 presents the flow diagram for this routine.

Run control is exercised by a transaction generated at the beginning of the model and at the beginning of each data interval specified. The first transaction generated is

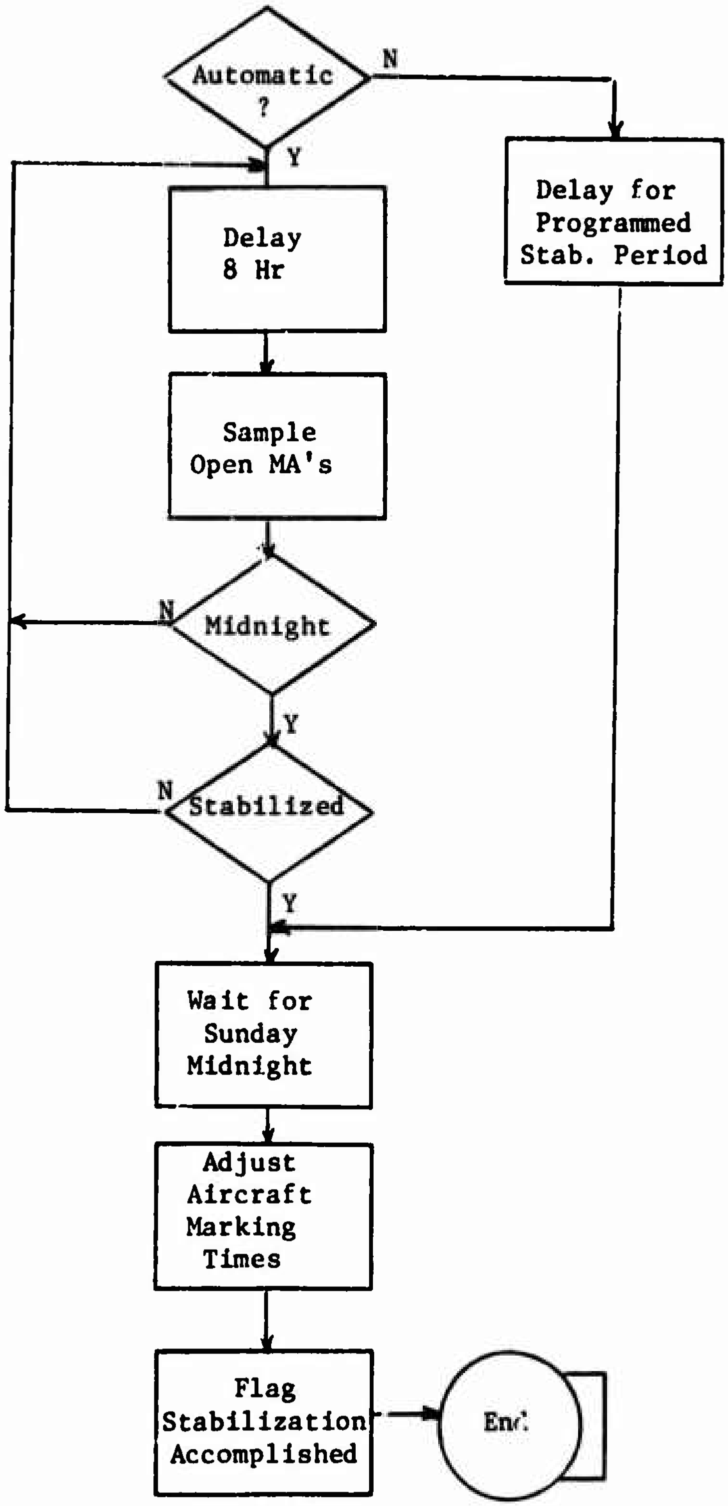

Figure 3. Stabilization Routine.

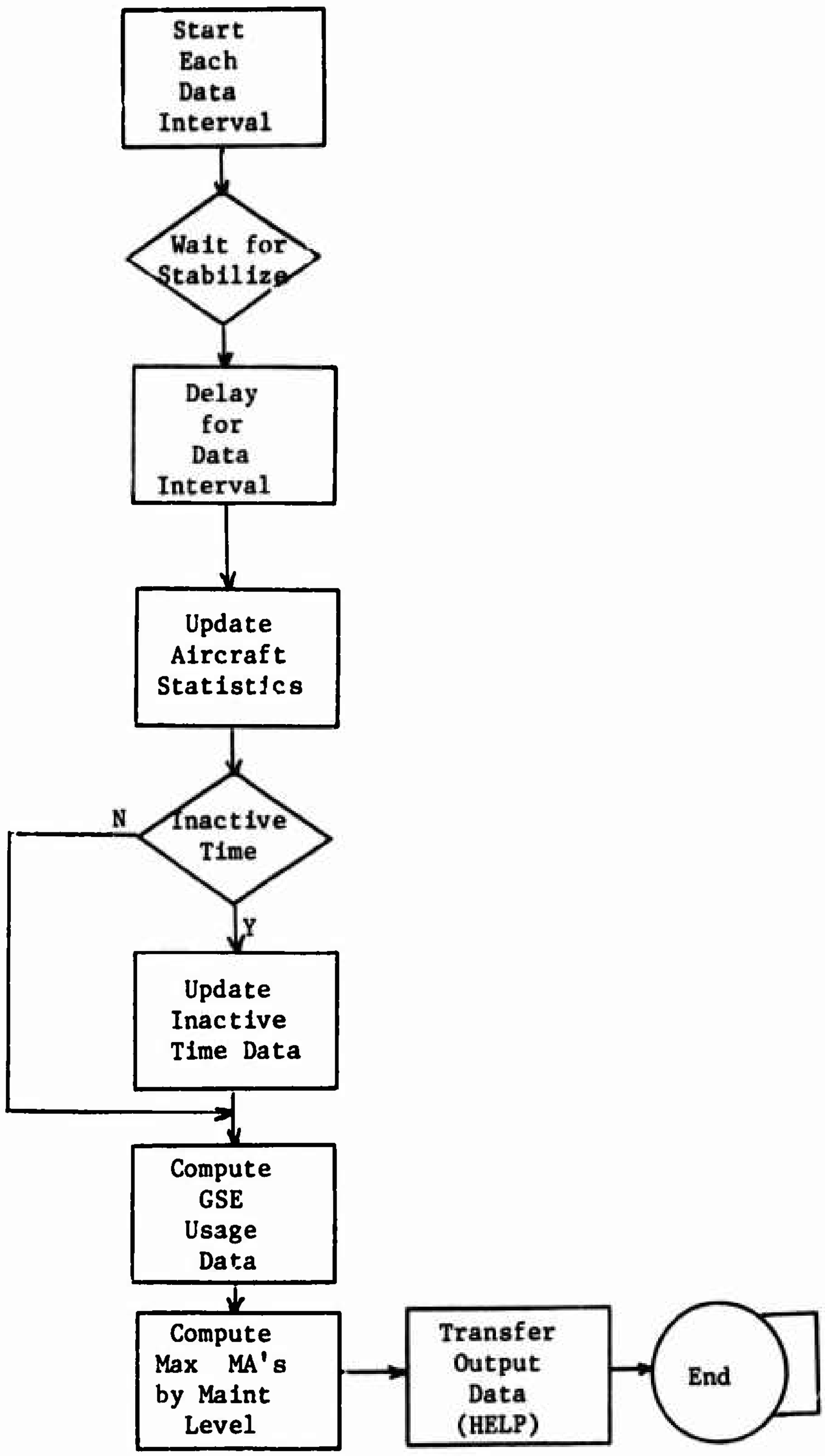

Figure 4. Run Control Routine.

delayed until stabilization has been achieved. After stabilization, an additional delay for the period of the data interval is introduced.

Several types of output data are accumulated or updated at the end of each data interval. Aircraft statistics (NORM, NORS, etc.) are updated. If the data interval ends during inactive time, i.e., no manpower on duty, these statistics are updated. All GSE data are computed and maximum maintenance actions by maintenance level are recorded. The HELP C block is entered to access the ARMDAT FORTRAN routine which records the output data on magnetic tape for processing by the SDOSFI program.

Regular/Surge Control Routine

The regular/surge control routine will set regular and surge indicator flags that are used by many other routines in the model. Card type "A" supplies the input data for this routine. Figure 5 presents the flow diagram for this routine.

The routine is energized at the beginning of the model and enters a delay corresponding to a regular operational period. Note that the model must begin on a regular period. Also, regular and surge periods must start and end in even-day multiples. At the end of the regular period delay, a surge flag is set and a surge period delay introduced. Following this, the regular operational period flag is again set and the entire process repeated for the number of cycles specified by input data.

Shift Control Routine

This routine is used to control the start time and duration of up to three shifts for four maintenance echelons. Input data for this routine is extracted from card type "Q". Figure 6 presents the flow diagram for this routine.

At the beginning of the model, a routine is entered which makes all manpower unavailable. Manpower is activated by a routine which is exercised once per day. If any of the three possible shifts are specified for work during that day, a transaction is created which delays until the specified shift start time. The proper shift is then activated and any jobs in the model requiring manpower from the oncoming shift are removed from their queues to compete for the oncoming men. A delay is then entered corresponding to the shift working hours, after which the proper MOS's are released. Note that this logic is entered for up to three shifts and four maintenance levels.

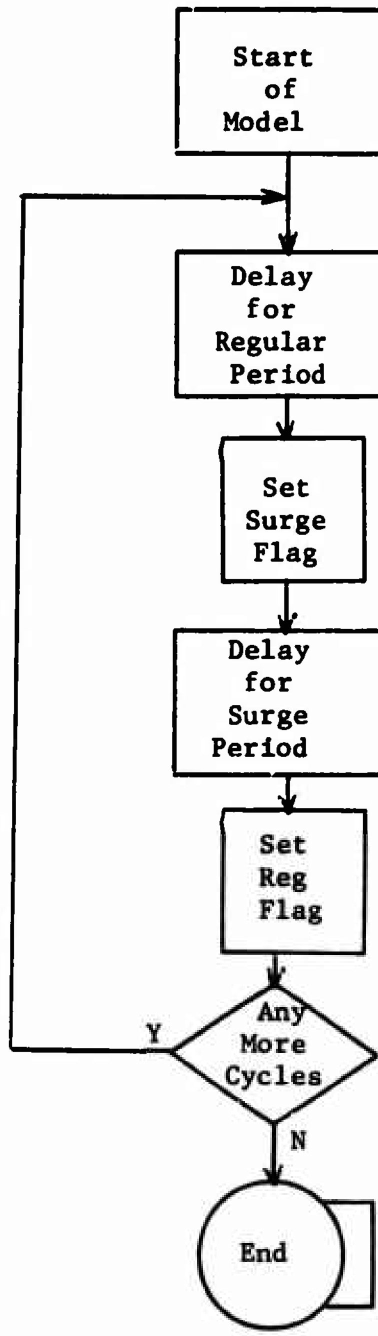

Figure 5. Regular/Surge Control Routine.

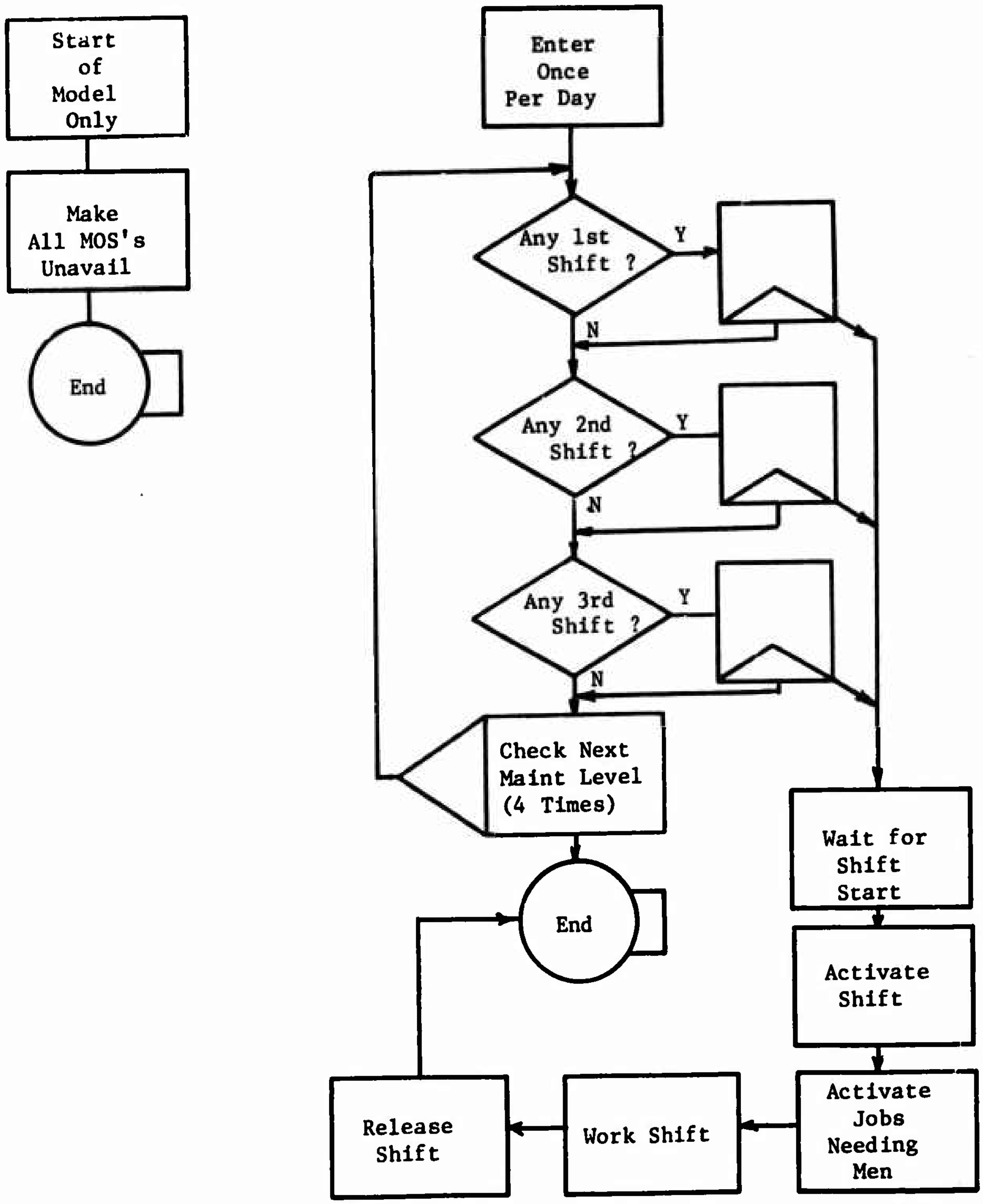

Figure 6. Shift Control Routine.

Each maintenance level can work different hours. Different working hours may be specified for each day of the week. Shifts may be eliminated on any day of the week. Completely different working hours may be specified for regular and surge operating periods.

Inactive Time Control Routine

The purpose of this routine is to compute and record the amount of inactive time contained in the model operational scenario. Inactive time is defined as those periods where no shifts are working at any of the specified maintenance echelons. This routine does not directly use data from the input cards. It is entered each time a shift change occurs and is activated by the shift control routine. Figure 7 presents a flow diagram for this routine.

At shift change, a test is made to establish if any men are currently working. If men are on duty the transaction immediately terminates. If not, it signals the beginning of an inactive time period. The transaction is delayed until the end of inactive time and then updates all inactive time statistics and terminates.

Plot Data Routine

The purpose of this routine is to accumulate output data that will be used to generate graphs in the SDOSFI routine. It is entered daily at midnight and records the various statistics necessary to create the plots. These are: Aircraft Readiness, Aircraft NORM, Aircraft NORS, Flight Data, Manpower Data, Primary MOS Queueing Information, Aircraft Ready Pool Delays and Mission Preparation Time. The transaction terminates after these data are recorded. Figure 8 presents a flow diagram for this routine.

Aircraft Generation and Initialization Routine

Model transactions representing aircraft of the proper configuration and with initial flying hours relative to scheduled maintenance requirements are created by this routine. The logic establishes the initial configuration and flying hours of each aircraft considered in the model. Up to four configurations may be specified.

Each aircraft will be initialized with flying hours and calendar days to establish their relative position in the scheduled maintenance cycles defined by input. These hours and days will be set either at random or at equally spaced intervals depending on input data.

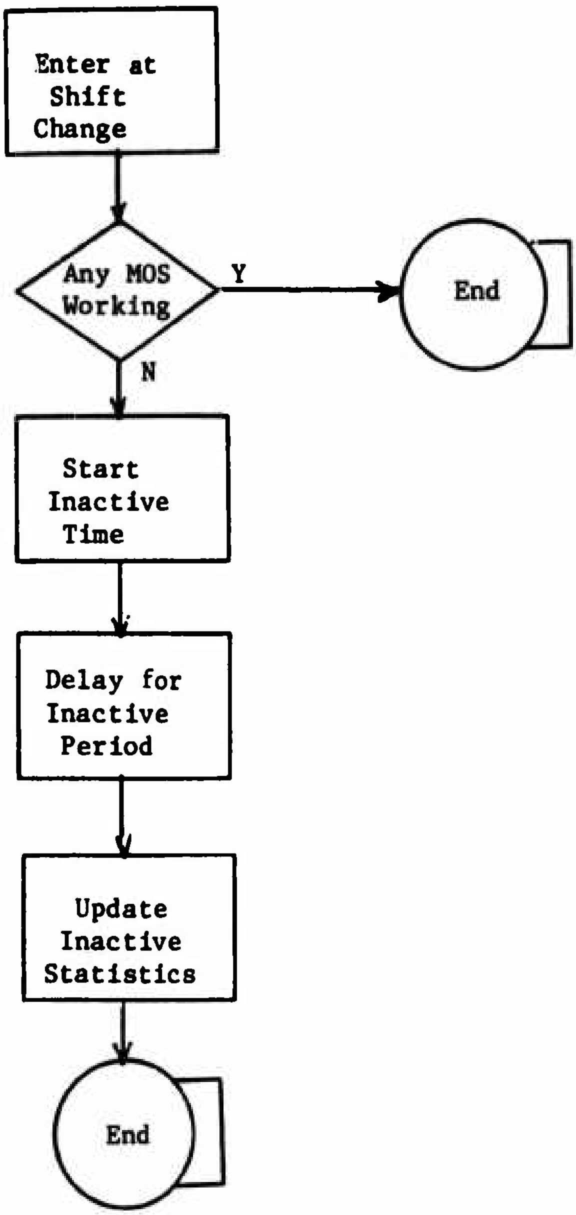

Figure 7. Inactive Time Control Routine.

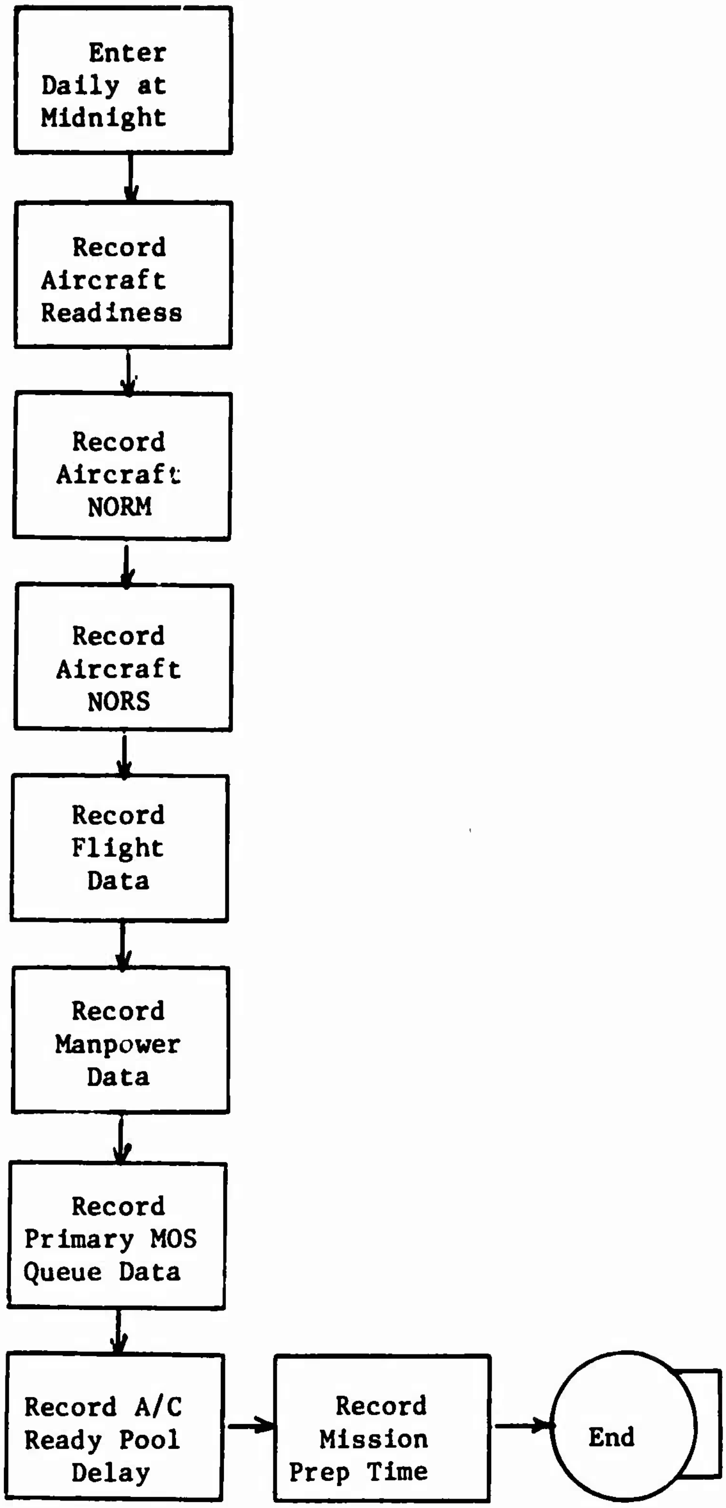

Figure 8. Plot Data Routine.

As an alternative, all new aircraft may be specified. In this case, calendar days and flying hours will be set to zero. If unlimited aircraft are specified by the input scenario, 99 aircraft are created and stored as "new" air-carft until required for usage by the operational logic.

Data from card "B" are used in this routine. Figure 9 is a logic flow diagram of the Aircraft Generation and Initialization Routine.

TBO Initialization Subroutine

This subroutine will establish the initial values of operating time for each TBO item. The subroutine is entered once at the beginning of each simulation experiment. All except "new" aircraft utilize this subroutine to establish the operating hours or calendar days since last overhaul on each TBO item. Position within the TBO interval is directed by input to be a uniform random distribution or to be in equal intervals by tail number.

All data in this routine are extracted from card "N". A flow diagram of subroutine logic is presented by Figure 10.

Mission Schedule Control Routine

This routine is used to create and maintain a 48-hour flying schedule in the model. Data from card type "Y", "Z", "1" and "2" are interrogated to create requirements for scheduled missions, random missions, continuous missions and alert aircraft. These input data are used to establish mission launch time and aircraft requirements for each mission in the operational scenario. In addition, this routine computes the mission preparation time required. Preparation time is dynamic. It is computed from the total time required for mission ground segments and increased in the model if queueing problems cause excessive cancellation of missions.

The routine is entered each day at midnight. If no change from a regular to surge or surge to regular operating period is indicated, the mission schedule is established one day in advance. When a change in the operational status occurs, however, this routine will cancel all previously established schedules for the current day and reschedule it along with one future day. Each mission demand created is assigned a call time, a launch time and the aircraft requirements. Each requirement is maintained in file until activated by the Mission Generation and Launch Routine.

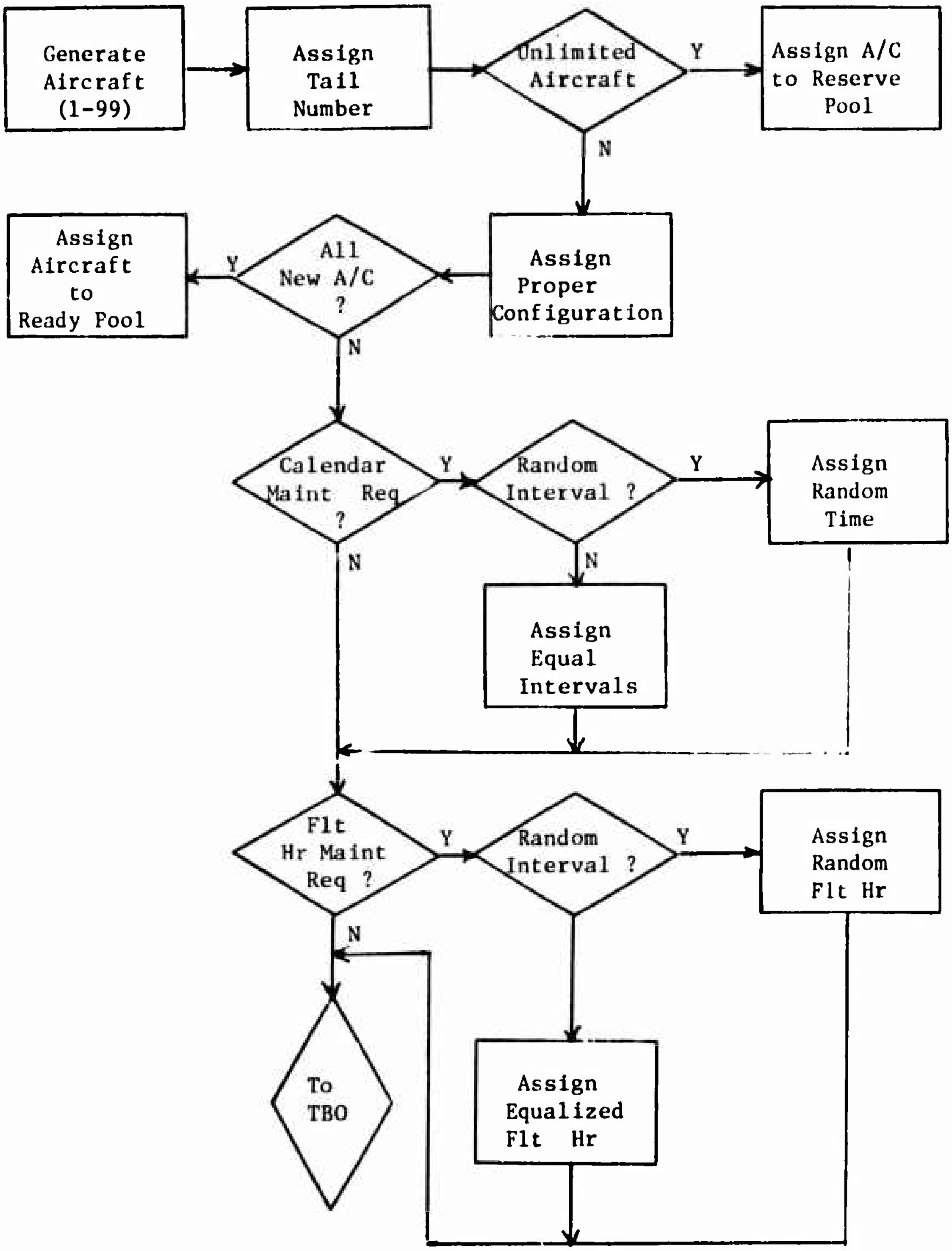

Figure 9. Aircraft Generation and Initialization Routine.

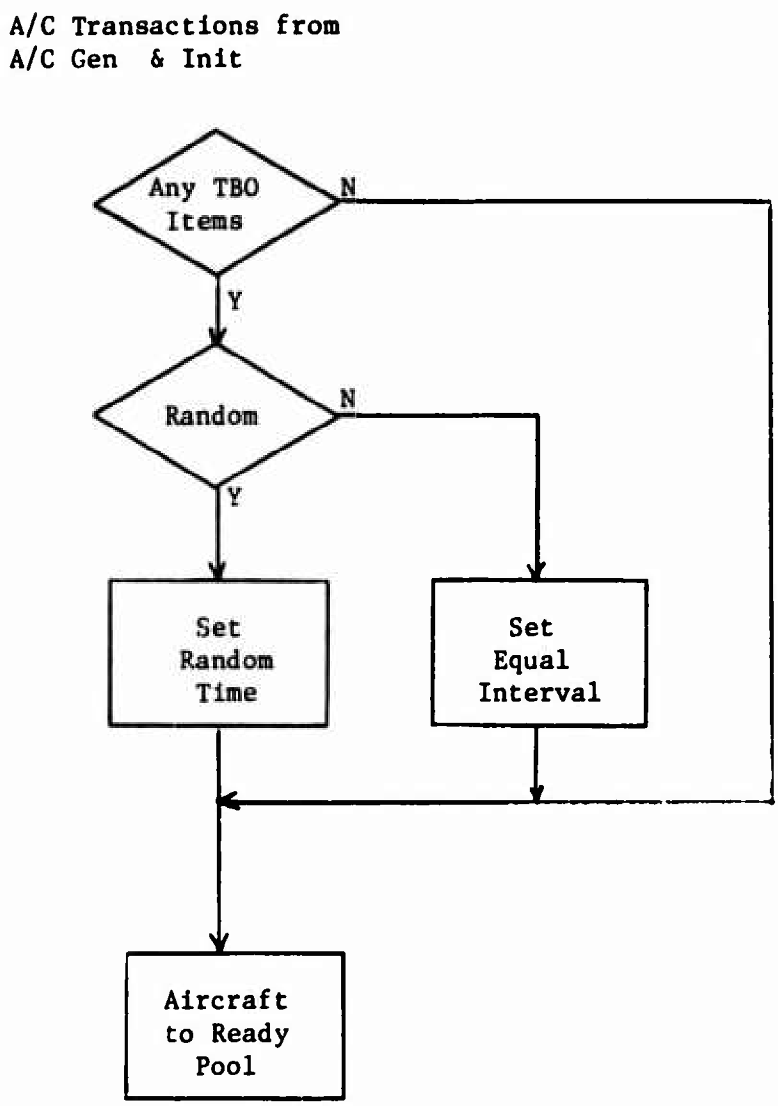

Figure 10. TBO Initialization Subroutine.

Figure 11 presents a flow diagram of the Mission Schedule Control Routine.

Mission Generation and Launch Routine

This routine controls the activation of mission requirements. It establishes timing for mission launch and monitors each mission until it is completed or cancelled. Data for this routine are extracted from flying schedules created in the Mission Generation and Launch Routine. Additional data are extracted from card types "Y", "Z", "1" and "2".

The established flying schedule is interrogated to extract the most eminent mission requirement. A delay is introduced to arrive at the computed mission call time. At this point the Aircraft Selection Routine is also activated to secure the required mission aircraft. Launch time is computed and the launch window opened after an appropriate delay. If sufficient aircraft have been prepared, the mission is launched and the routine will continue to monitor in-flight segments until the mission is either completed or cancelled. If enough aircraft are not available at the scheduled launch time, an additional delay is entered. This delay is computed from launch window input data. If enough aircraft are still not available, the mission will be cancelled. When sufficient aircraft are found, the mission is launched and its flight status monitored.

Figure 12 presents a flow diagram of the Mission Generation and Launch Routine.

Missions Called by Other Missions Routine

This routine is used to establish requirements for missions called by other missions. It is entered under two conditions. First, each aircraft launched on a scheduled or random mission activates the routine so that it can be determined if additional missions are required. Data from card type "U" is interrogated for each of the nine missions. If the Monte Carlo process establishes the requirement for a mission, its launch time, mission type and aircraft requirements are recorded and the Mission Schedule Routine is entered.

The second circumstance which activates this routine is a requirement for a Red X consequence mission. In this case, data from card type "F" is used to set the mission definition and the Mission Schedule Routine is entered.

Figure 13 is a flow diagram of this routine.

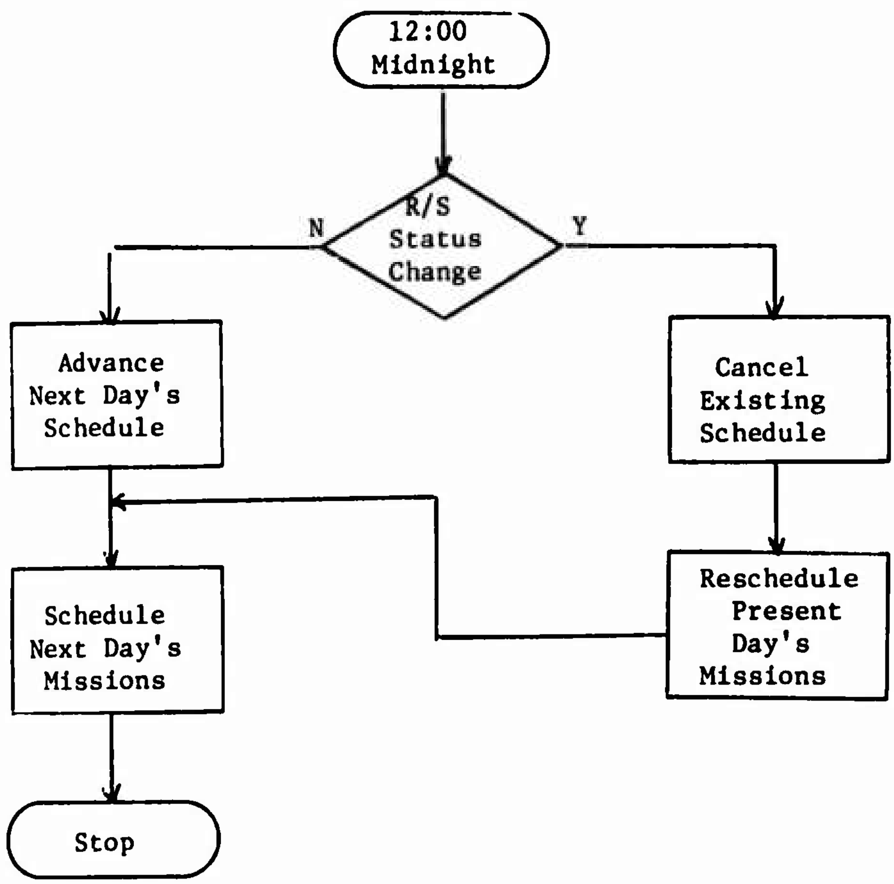

Figure 11. Mission Schedule Control Routine.

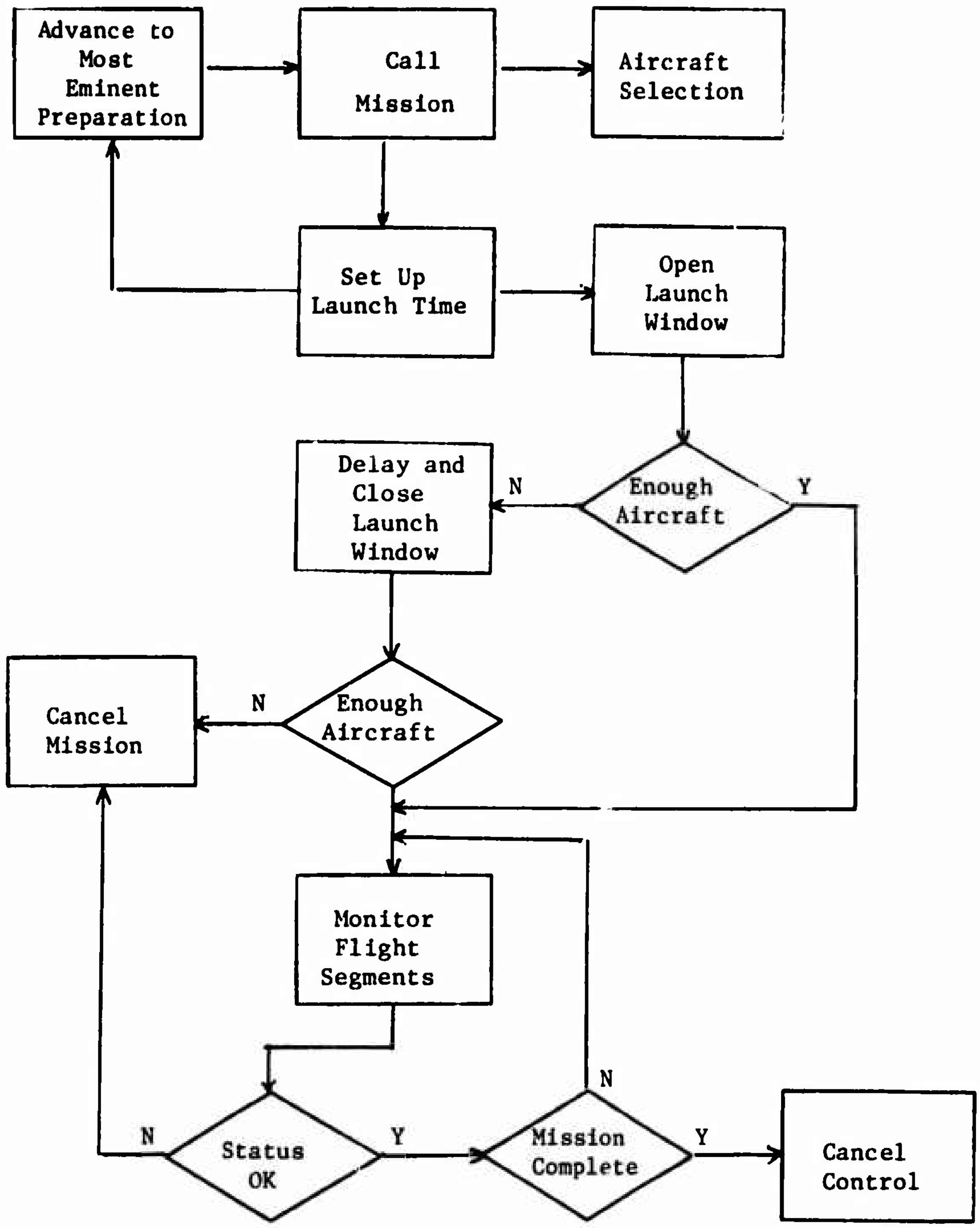

Figure 12. Mission Generation and Launch Routine.

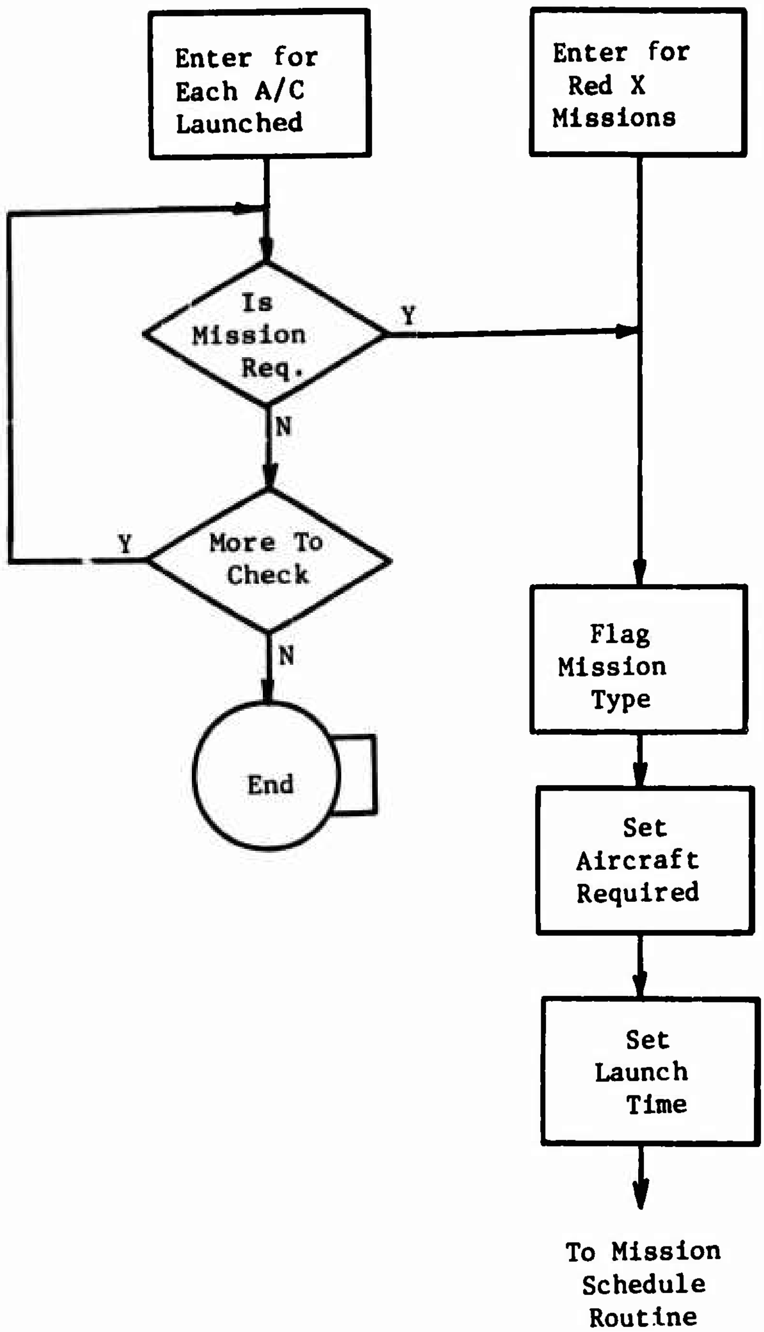

Figure 13. Missions Called by Other Missions Routine.

Aircraft Selection Routine

The Aircraft Selection Routine is used each time an aircraft must be obtained to perform a mission. There are 17 situations which can require aircraft selection. These are listed in Table 2. Aircraft may be obtained from the ready pool or the standby pool, or an alert aircraft can be used. These three sources are checked in a specific order of preference for each of the 17 possible mission situations. Table 2 also presents the order in which sources of aircraft are tested in order to satisfy mission requirements.

When an alert aircraft has been selected, a test is made to determine if input data requires replacement of the alert aircraft. If this is true, an additional requirement is entered as an input to the Aircraft Selection Routine.

When a ready pool aircraft is selected, a determination is made relative to its configuration. If no aircraft of the preferred configuration are available in the ready pool, input data is checked to see if reconfiguration is allowed. The time required to reconfigure is then computed. If the estimate indicates that an aircraft can be readied prior to expiration of the launch window the selection transaction will wait for reconfiguration to be accomplished.

When sufficient time is not available, or reconfiguration is not permitted, the model will delay the selection transaction until an aircraft is obtained or until time no longer permits the mission requirement to be satisfied. An exception to this is the case of an unlimited aircraft scenario. When this has been specified, a reserve aircraft will be inducted into the model rather than delaying selection of a mission aircraft.

Figure 14 is a flow diagram of the Aircraft Selection Routine.

Aircraft Mission Events Routine

The Aircraft Mission Events Routine is comprised of three separate subroutines. These are Reconfiguration, Ground Events and Flight Events. A narrative of each of these subroutines is presented in the following paragraphs.

Reconfiguration Subroutine

The Reconfiguration Subroutine is utilized to change aircraft configurations when the Selection Routine determines that time will permit reconfiguration. Reconfiguration data is obtained from input card type "C". The

TABLE 2. AIRCRAFT SELECTION ORDER

	Ready Alert A/C	Standby Aircraft: Same Mission	Standby Aircraft: Different Mission, Same Msn Type	Standby Aircraft: Different Mission, Different Msn Type	Avail Ready A/C	Wait for Ready A/c
Scheduled Mission A/C	-	-	-	-	1	2
Random Mission A/C	-	-	-	-	1	2
Continuous Mission A/C	-	-	-	-	1	2
Other Mission A/C	1	2	4	5	3	6
Rescue Mission A/C	1	2	4	5	3	6
Repair Mission A/C	1	2	4	5	3	6
Air Evac. Mission A/C	1	2	4	5	3	6
Rescue & Air Evac Mission A/C	1	2	4	5	3	6
A/C for Ready Alert	-	-	-	-	1	2
Scheduled Mission Replmt A/C	3	1	2	5	4	6
Random Mission Replmt A/C	3	1	2	5	4	6
Continuous Mission Replmt A/C	-	1	-	-	2	3
Other Mission Replmt A/C	3	1	2	5	4	6
Rescue Mission Replmt A/C	3	1	2	5	4	6
Repair Mission Replmt A/C	3	1	2	5	4	6
Air Evac. Mission Replmt A/C	3	1	2	5	4	6
Rescue & Air Evac Msn Replmt A/C	3	1	2	5	4	6

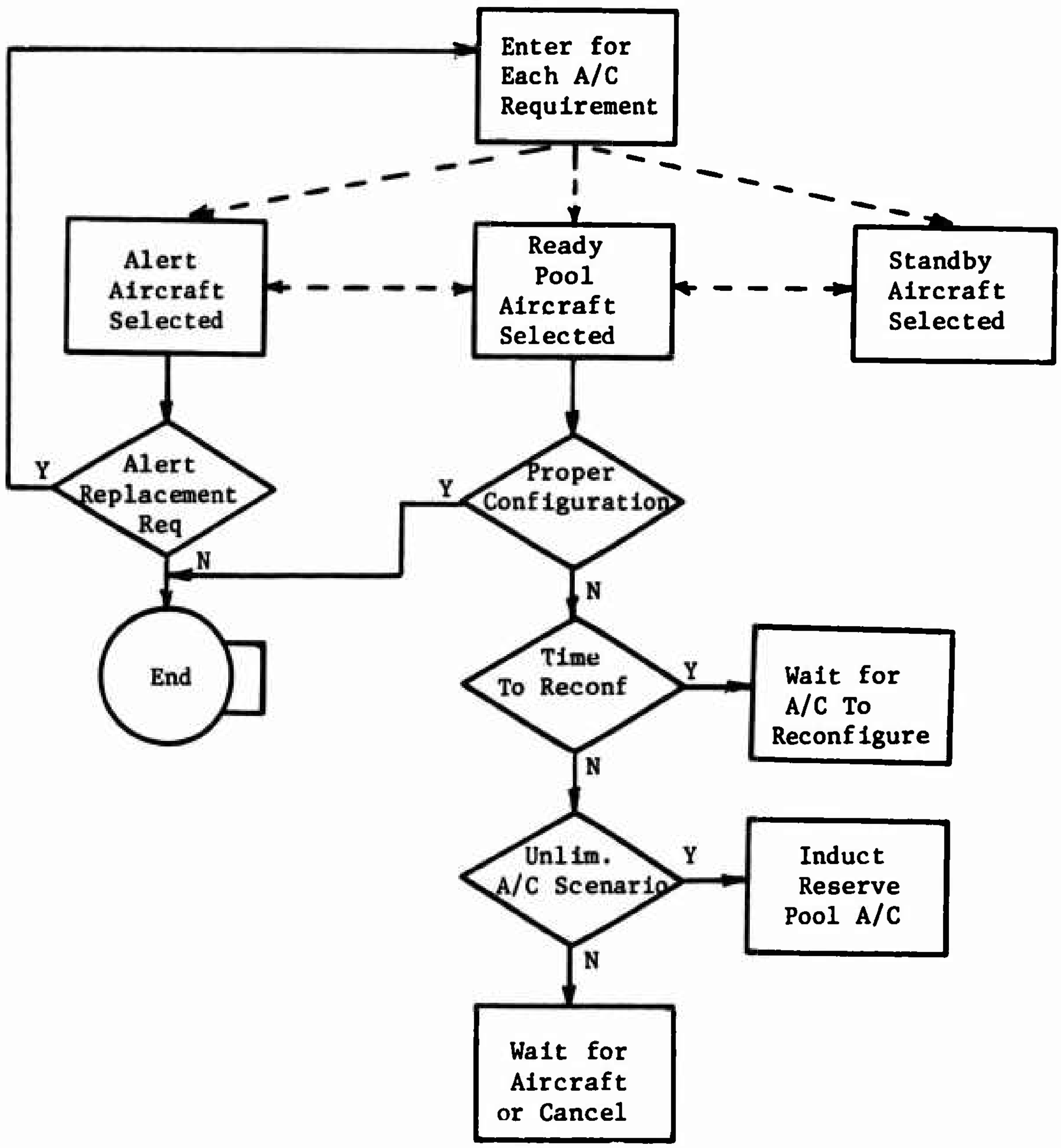

Figure 14. Aircraft Selection Routine.

routine computes the removal time for the present aircraft configuration and, after obtaining the necessary manpower and GSE, delays in accordance with the computer removal time. If input data specifies an SMA requirement to install the new configuration, the aircraft is routed to the Ground Events Subroutine, where reconfiguration will take place. If an SMA is not required, the aircraft is routed to the ready pool, where it will be obtained by the Aircraft Selection Routine. Figure 15 is a flow diagram of this subroutine.

Ground Events Subroutine

This subroutine is entered whenever an aircraft is starting a mission, when a ground event segment is encountered during a mission, or when an SMA is required. For all ground segments, the aircraft will obtain the input specified manpower and GSE and encounter a delay for the required segment time. During this delay, other model routines establish failures and potential aborts. When an abort is indicated, the aircraft is routed to the aircraft return logic. If not, the next segment is entered. This process is repeated until input indicates other than a ground segment. Two cases exist here: First, if no further segments are indicated, the aircraft is routed to the return routine. If more segments are indicated, aircraft will obtain the required in-flight manpower and GSE. An exception to this is the preparation of a ready alert aircraft through its ground segments. Ready alert aircraft having completed their ground segments are routed to an alert pool for possible call-up.

Aircraft which have obtained manpower and GSE are routed to the ready-to-launch chain unless they are in the process of an SMA or test hop. In this case, they proceed directly to the Flight Events Subroutine.

Figure 16 is a flow diagram of the Ground Events Subroutine.

Flight Events Subroutine

The flight events logic is entered from the Ground Events Subroutine for each mission requiring in-flight segments. Flight segments include combat and remote ground segments. For each combat segment a delay of one-half the segment time is introduced. A Monte Carlo process determines the number of hits sustained by the aircraft. When hits are sustained, the probability of

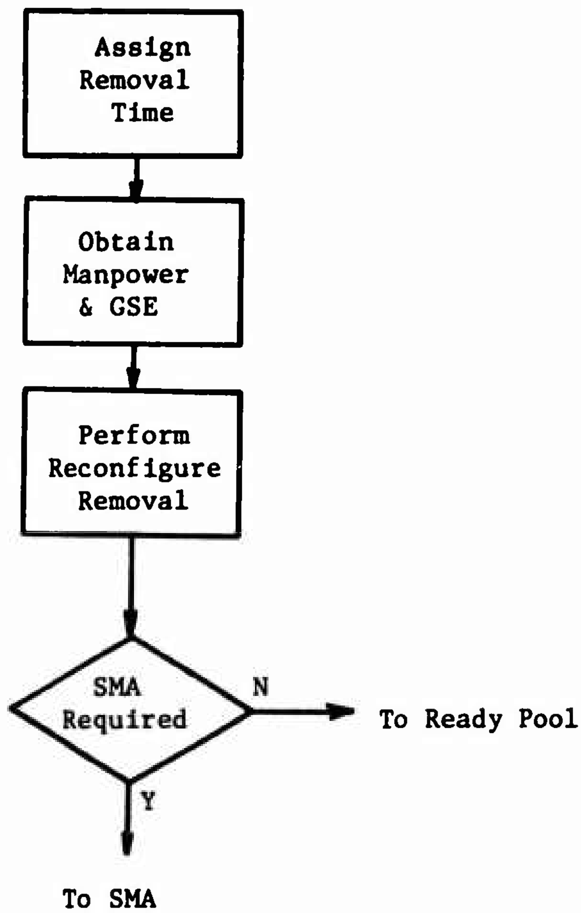

Figure 15. Reconfiguration Subroutine.

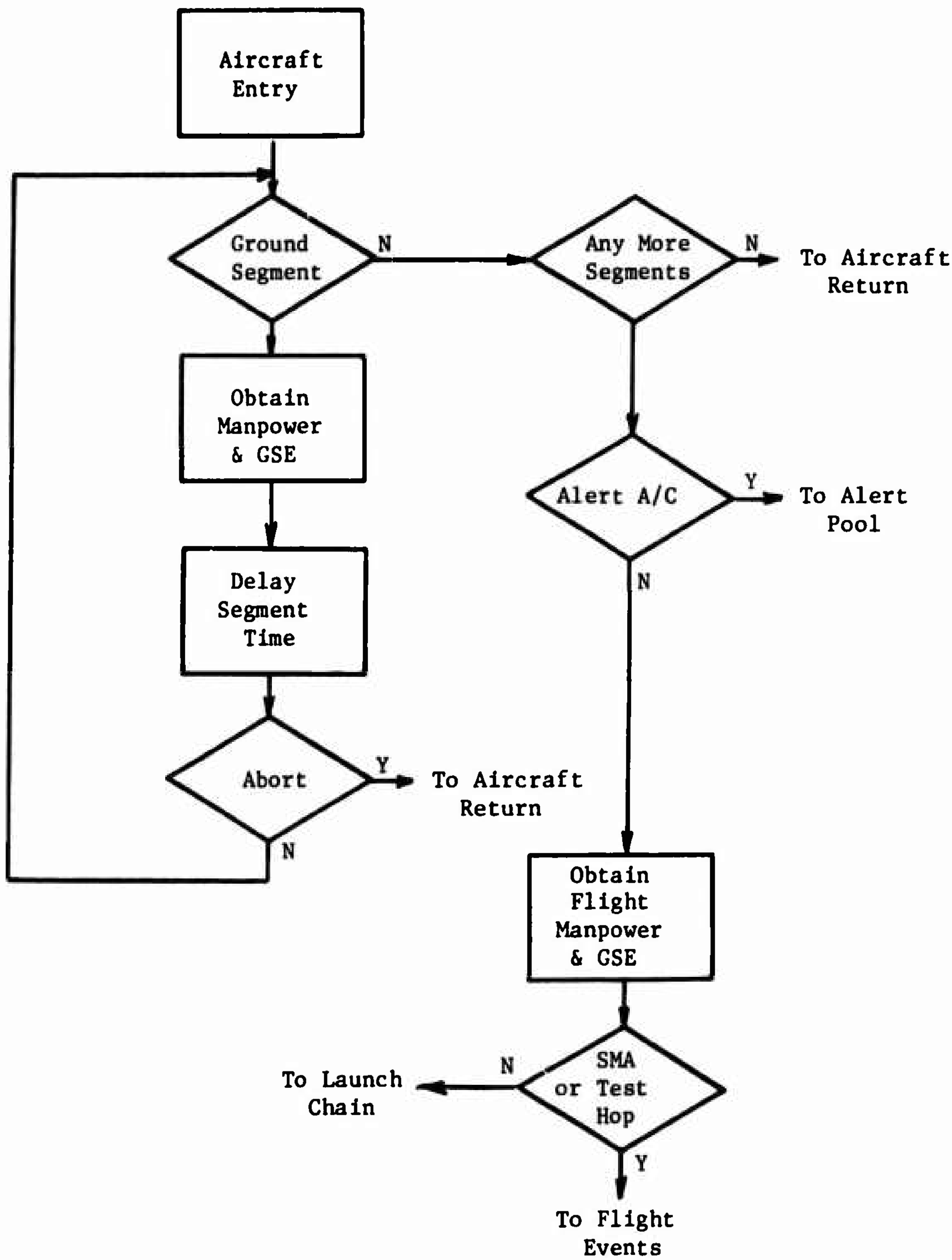

Figure 16. Ground Events Subroutine.

mission aborts is used and the aircraft will be transferred to abort consequence logic if appropriate. If hits are not indicated or an abort is not appropriate, the remaining segment time is utilized and the mission scenario tested for the presence of additional segments. The aircraft will either repeat the next flight event segment or be transferred to aircraft return logic.

If the flight event is a remote ground segment and involves a repair mission, the segment time is determined by the repair time. Normal flight segments are delayed for the amount of time specified by input. After each segment the aircraft is routed either to abort consequence logic, to the next flight segment, or to Aircraft Return Routine.

If additional segments are indicated that are not flight segments, the aircraft will transfer to the Ground Events Subroutine for mission completion. Input data from card types "W" and "X" are utilized in this subroutine.

A flow diagram of the logic is presented in Figure 17.

Aircraft Failure Events Routine

This model routine consists of three subroutines. They are Aircraft Failure Determination, Red X/Blue X, and Aircraft Replacement. Contained in this routine is the logic required to establish failures, to control the consequences of in-flight aborts, and to obtain replacement aircraft for missions suffering aircraft losses. Separate narratives are presented for each of the three subroutines.

Aircraft Failure Determination Subroutine

Logic in this subroutine is used to control the flow of aircraft during missions. It uses the basic element failure rate data to determine when the next failure will occur. If the mission being flown is a repeat test hop, the failure rate used to compute time of failure is reduced by the amount specified in input data.

When the failure rate has been established, the inverse transform methodology described in Appendix VIII is used to compute time of failure. If the failure will occur within the present mission duration, a delay until time of failure is introduced. At this point, the element causing the failure is established and a Monte Carlo process determines if the failure is detected.

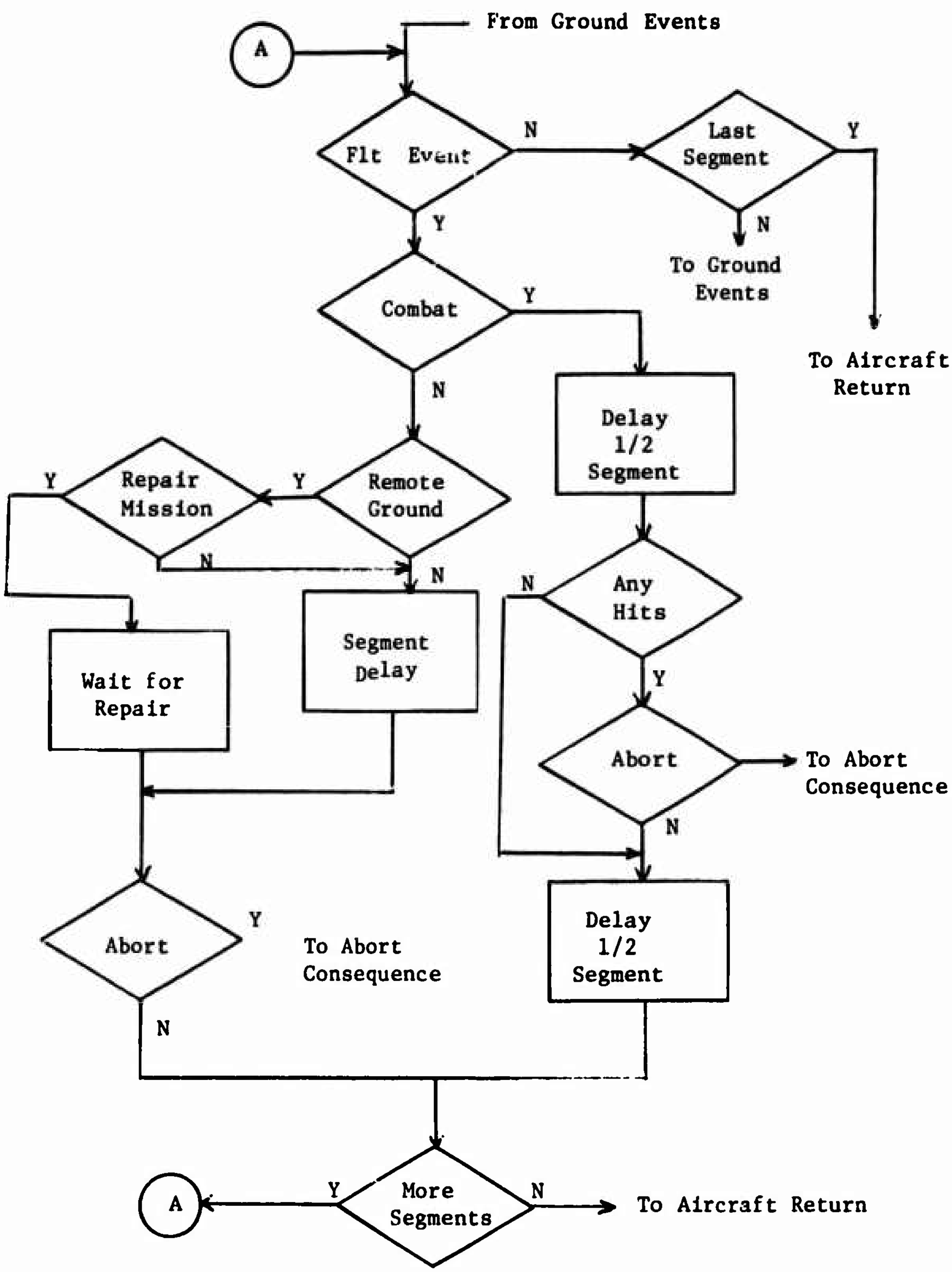

Figure 17. Flight Events Subroutine.

Undetected failures are recorded and the logic is re-entered to compute time until next failure.

When failures are detected, the probability of Red X and Blue X is tested. If the failure is determined to be in the upsquawk (deferred maintenance) category, it is recorded and the failure logic reentered.

When downing maintenance is indicated, the failure consequence logic is activated to establish the outcome. Possible outcomes range from continue mission to total loss of the aircraft. If continue mission is indicated, the failure logic is reentered to compute the time until next failure. More serious consequences will activate the Red X/Blue X subroutine where the appropriate action will be taken. The entire failure determination process is repeated until no further failures occur on the present mission.

Figure 18 is a flow diagram of the Failure Determination Subroutine logic.

Red X/Blue X Subroutine

Each Red X/Blue X failure that is detected at time of occurrence may have seven consequences specified by input data. This input data is supplied on card type "F". Logic in this subroutine implements the actions indicated by the failure consequence.

If continue mission is indicated, data is recorded and control is returned to the Flight Events Subroutine. An abort to base consequence causes a delay representing the flight time to base, after which the aircraft is transferred to aircraft return logic.

A crew repair consequence introduces two delays before the aircraft enters aircraft return logic. These delays represent the time on the ground to accomplish repair and the flight time required to return to base.

A repair mission consequence creates the demand for an additional aircraft. The original mission aircraft is delayed until the repair aircraft can reach the same mission segment. Additional delays to accomplish the repair and to fly to base are introduced before entering the aircraft return logic.

Another possible consequence is the requirement for a

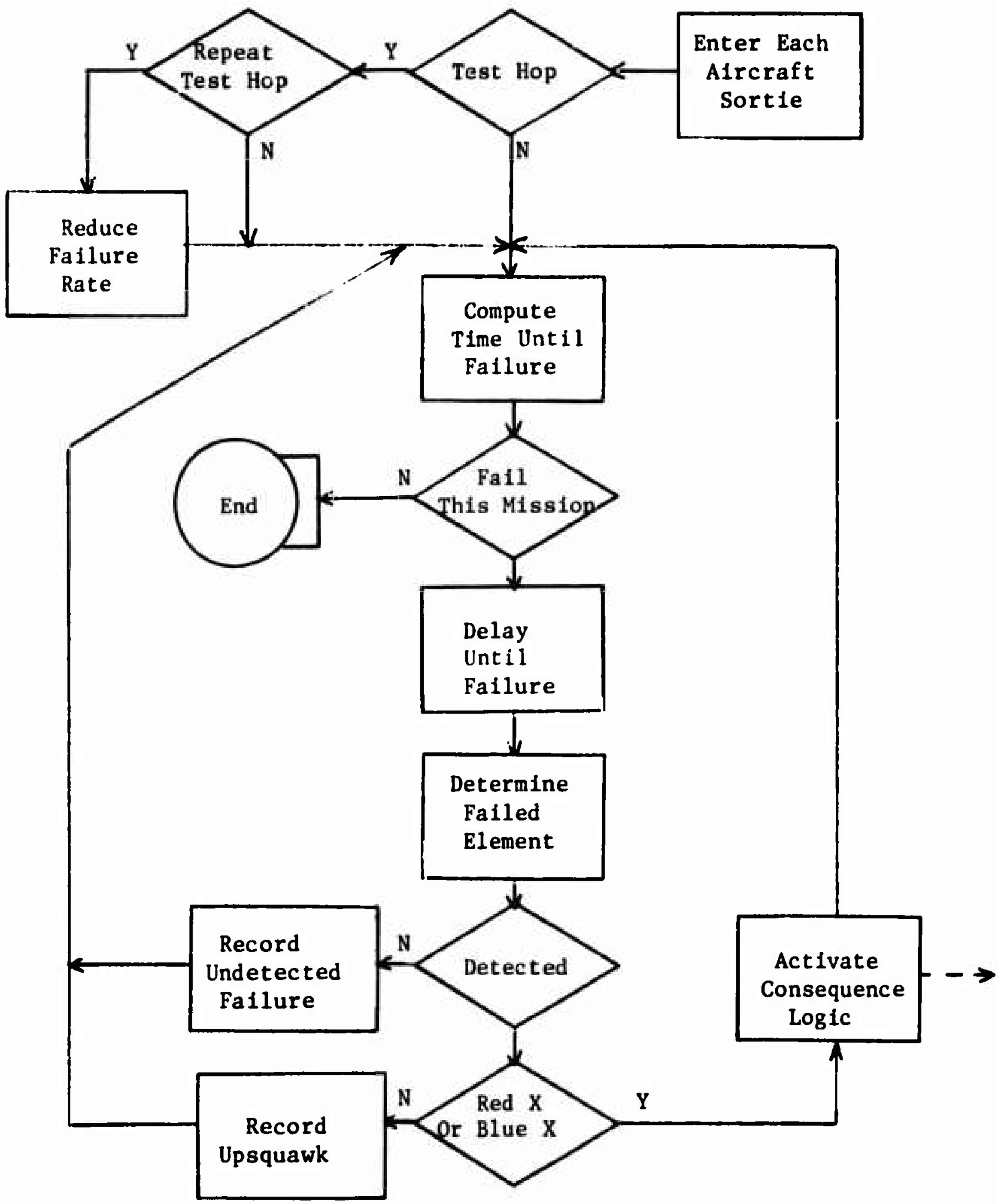

Figure 18. Failure Determination Subroutine.

rescue and air evacuation mission. The user may specify two separate missions to accomplish this. Two logical paths are followed. The mission aircraft is delayed until the evacuation aircraft can reach the same mission segment. From this point a delay for returning to base is added prior to entering aircraft return logic. If manpower and GSE were included in the original mission scenario, these assets remain in use until the requested rescue aircraft arrives and the delay for returning to base has expired. Manpower and GSE assets are then made available for other model usage.

A rescue mission consequence indicates that the basic mission aircraft is a total loss but that manpower and GSE assets are held unavailable in the model until the rescue process is completed. Input data may specify that an attrition aircraft is to be replaced. If this is true, the input specified time to obtain a float is used and an aircraft is returned to the ready pool. When aircraft replacement is not to be accomplished, the original mission aircraft is removed from additional model participation.

The final consequence possible is a total aircraft loss. In this case, manpower and GSE are returned immediately for additional model usage. Aircraft replacement is accomplished if input data indicates.

Figure 19 is a flow diagram of Red X/Blue X logic.

<u>Aircraft Replacement Subroutine</u>

Each mission flown in the model has specified a minimum number of aircraft required to accomplish the mission. If abort consequences cause the number of in-flight aircraft to fall below this minimum, the aircraft replacement logic will attempt to find sufficient aircraft so that the mission may be continued. The logic in the routine uses aircraft endurance data, flight time since launch, and flight time to finish the mission, in order to compute whether or not the remaining mission aircraft have sufficient endurance to loiter until a replacement can be obtained.

If insufficient time exists, all remaining mission aircraft will abort to base. If time permits, a demand will be placed on the aircraft selection logic to obtain a replacement aircraft. The original mission aircraft will interrupt the normal mission sequence until their loiter time is expended or a replacement aircraft is able to join them. When loiter time is

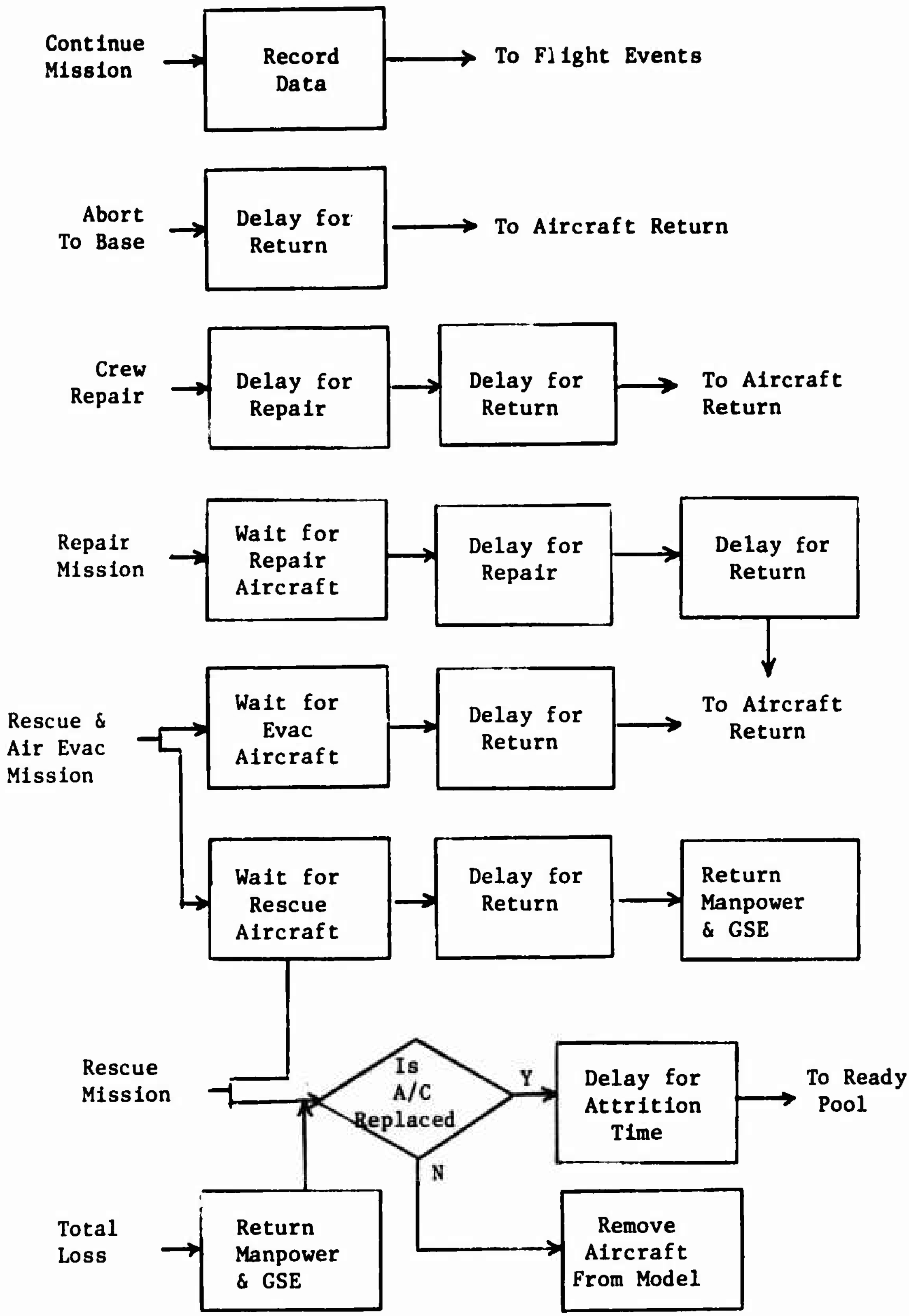

Figure 19. Red X/Blue X Subroutine.

expended, all aircraft abort to base. If a replacement aircraft is found, the mission is restarted from the point of interruption.

A flow diagram of aircraft replacement logic is presented in Figure 20.

Mission Manpower Routine

The Mission Manpower Routine is used to establish requirements and obtain any manpower and GSE necessary to accomplish mission segments, SMA's or reconfiguration events. The routine is logically divided into three subroutines: Manpower Determination, Mission Manpower and SMA Manpower. The narrative presented in this section will cover all three of these subroutines.

Logic is entered for each mission event. If a manpower package is required, the availability of each MOS in the package is determined. When all required skills are not available, the event is delayed and MOS queueing is recorded. Once it has been established that the skills specified are simultaneously available, GSE requirements are interrogated. A queueing situation with GSE causes delay of the event. This logic section will delay each event until it has been determined that all input specified manpower and GSE are simultaneously available.

When the required assets are available, the task time is computed. If the task involves a flight event, the manpower and GSE will fly with the aircraft.

Mission events that are not an SMA or reconfiguration will retain the manpower and GSE for the entire task time. Note that this may involve retaining manpower assets after their normal shift time has expired. When the event delay has occurred, manpower is returned and GSE transactions are transferred to the GSE Return Routine, where each is tested to determine if an equipment failure has occurred. The next mission event is then able to take place.

A slightly different utilization technique is used if the event is an SMA or reconfiguration. For these events, manpower is not retained past the end of a normal working shift. At each shift change, men currently working the event are released and new manpower must be obtained from the oncoming shift to complete the task.

Figure 21 is a flow diagram of Mission Manpower Routine logic.

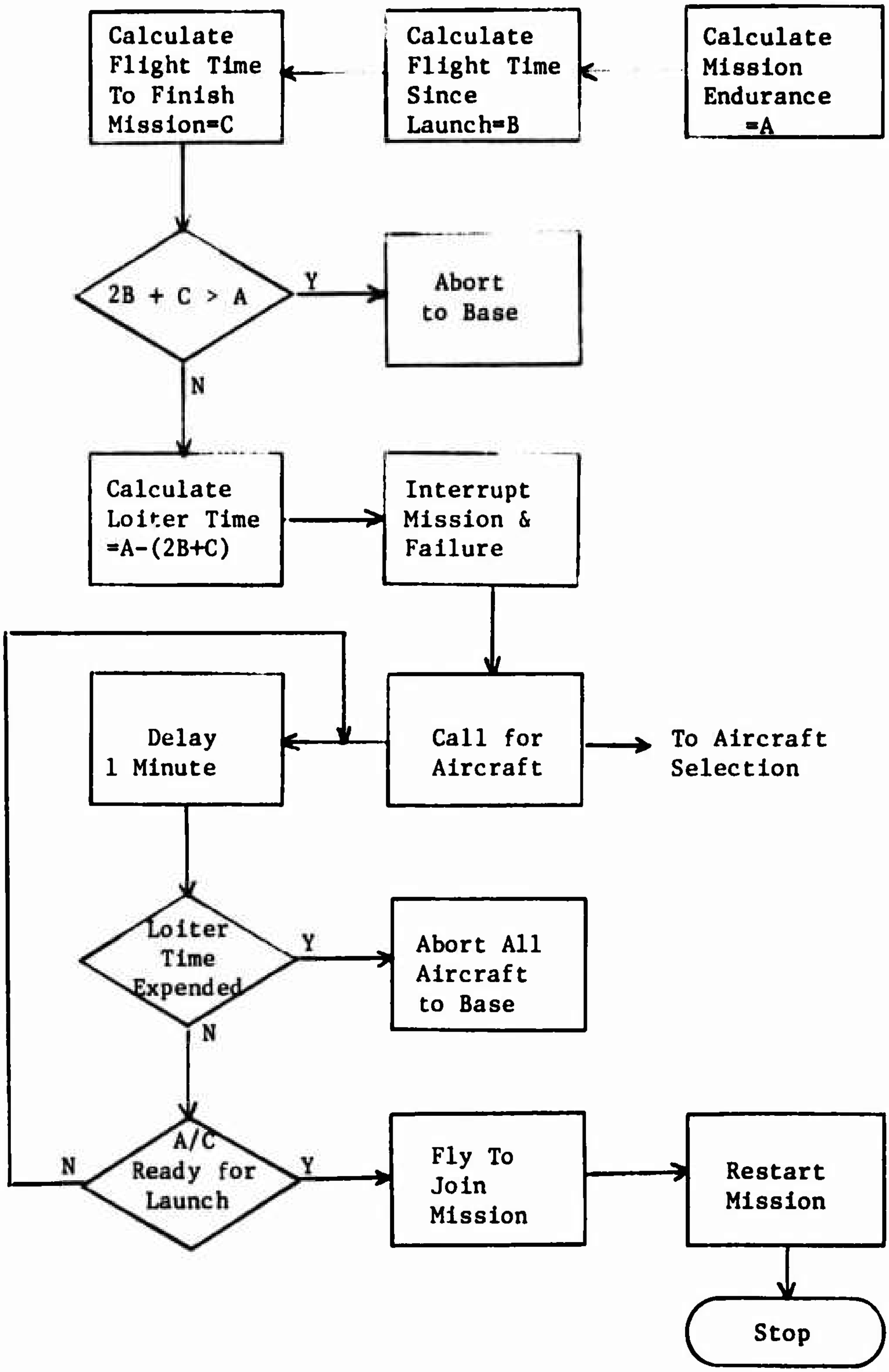

Figure 20. Aircraft Replacement Subroutine.

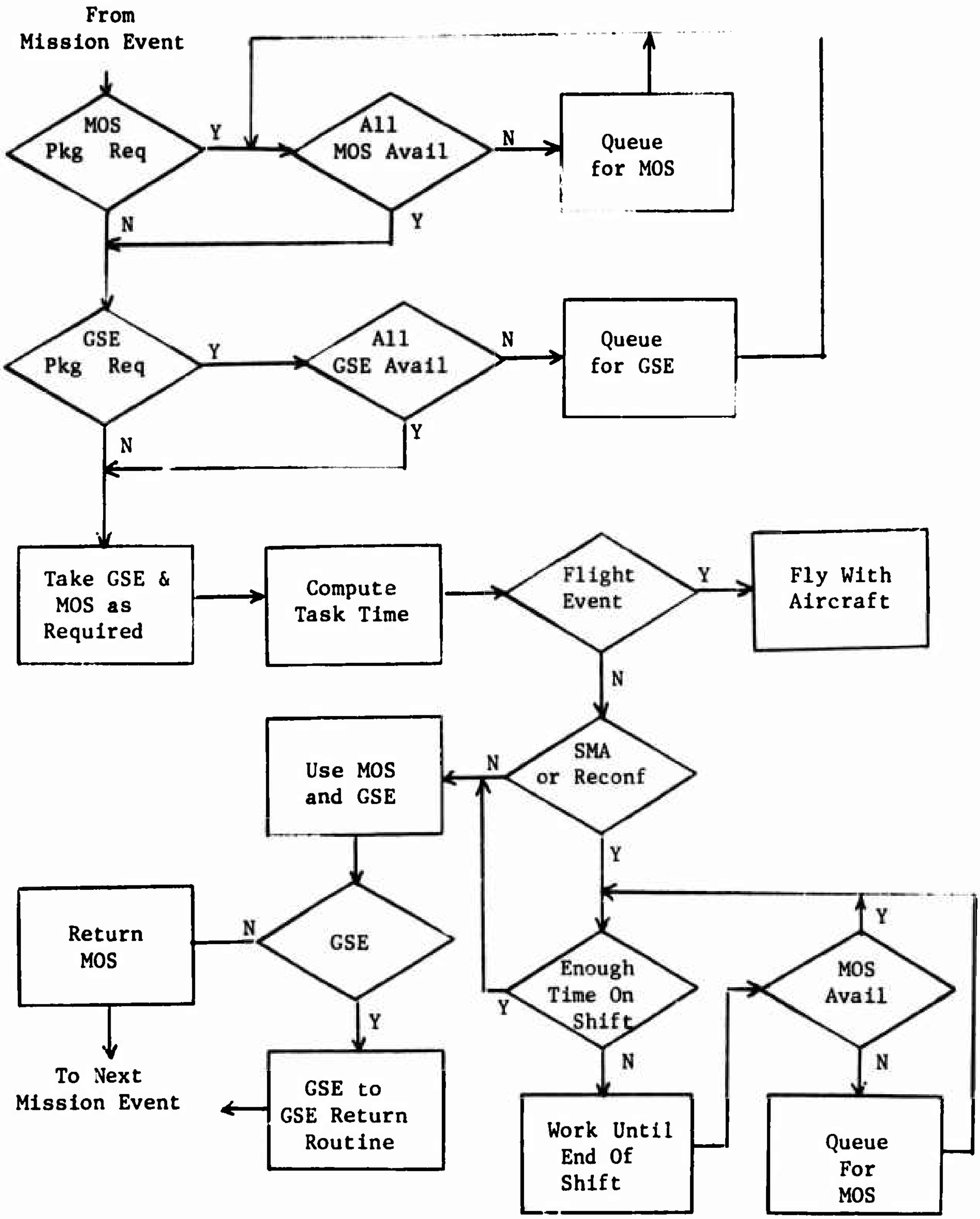

Figure 21. Mission Manpower Routine.

Aircraft Accounting Routine

Three subroutines combine to make up the Aircraft Accounting Routine. Each routine serves the purpose of maintaining data relative to aircraft flying operations. These data include failure detection data, flight hours, mission calls and launches, mission manpower and GSE assets, and many other operations-oriented statistics. The three subroutines are Mission Accounting, Failure Detection and Aircraft Return. Each subroutine is described in the following narratives.

Mission Accounting Subroutine

The Mission Accounting Subroutine maintains flight statistics for each aircraft sortie flown in the model. Aircraft enter the logic at the completion of each mission segment. After recording flight information, a test is made to determine if the segment just accomplished is the one specified on input card "X" as the mission completion segment. Additional data is recorded for each complete mission accomplished. If the completed mission was part of a continuous mission, the number of successful flights required by input card type "2" is reduced.

These are then made to establish the proper destination for the aircraft. If no more segments are required, the aircraft transfers to aircraft return logic. A ground event causes the aircraft to transfer to ground event logic, while a flight segment causes transfer to the Flight Events Subroutine.

Figure 22 is a flow diagram of Mission Accounting Subroutine logic.

Failure Detection Subroutine

Logic in this subroutine is entered for each detection event specified on input card type "E". In general, these are mission and scheduled inspection oriented. If the aircraft in question has no undetected failures, it is immediately returned to the main model logic.

When undetected failures exist, each is tested against its probability of detection data; and if a detection is indicated, the maintenance action is routed to the maintenance action logging logic. When the Monte Carlo process indicates that the failure remains undetected, it is refiled on the undetected maintenance action chain. This process is repeated until all undetected

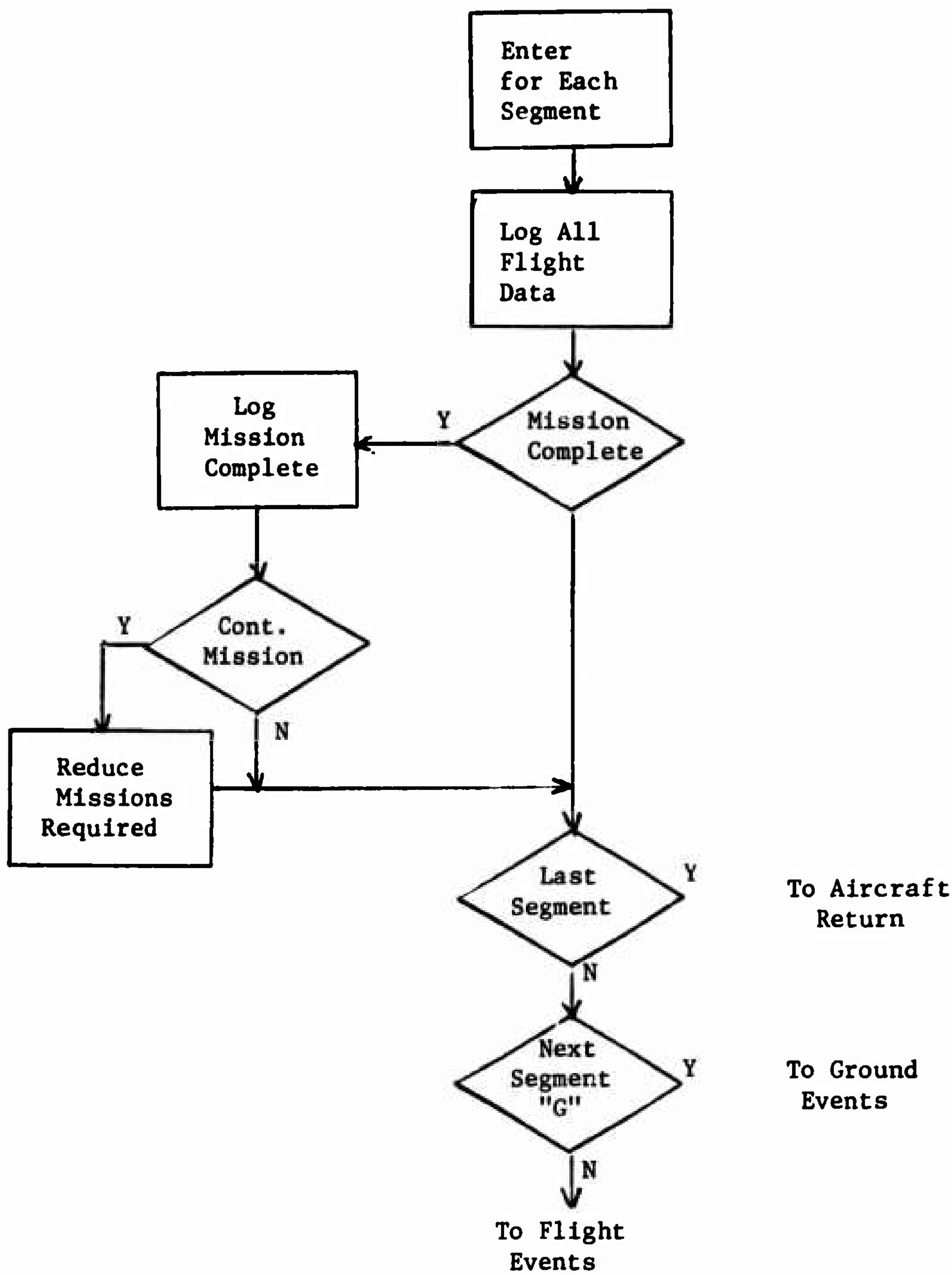

Figure 22. Mission Accounting Subroutine.

failures have been tested. At this time, the aircraft will return to the main model logic.

Figure 23 presents a flow diagram of the Failure Detection Subroutine logic.

Aircraft Return Subroutine

The purpose of this Aircraft Return Subroutine is to interrogate the status of each aircraft leaving mission events logic in order to properly record failure data and route the aircraft to the appropriate model logic.

Each new failure detected during mission events is recorded. Any SMA's caused by the mission, such as a hard landing, are also recorded.

The aircraft status is then interrogated, and it is transferred to unscheduled maintenance, scheduled maintenance, SMA logic, or the ready pool. If other than a ready pool destination is indicated and the aircraft was assigned to a continuous mission, a call for a replacement aircraft is made prior to inducting the aircraft into its maintenance event.

Figure 24 is a flow diagram of Aircraft Return Subroutine logic.

Supply Routine

The Supply Routine is entered each time a remove and replace maintenance action occurs on the aircraft. Its major purpose is to determine if the required parts are available in supply. Supply information may be provided in two forms: probabilistic and deterministic. In a probabilistic scenario, parts availability is specified as the probability of parts availability. This input information is exercised in a Monte Carlo process to establish parts availability.

In the deterministic scenario, the input data specifies an initial supply quantity for each part and the point at which reorder is made. The model maintains a count of the shelf level for each part and places the order at the appropriate time.

Any given simulation experiment may contain both probabilistic and deterministic parts. The Supply Routine separates and handles the two supply scenarios simultaneously. To establish probabilistic parts availability, it is first determined if the operational scenario is in a surge

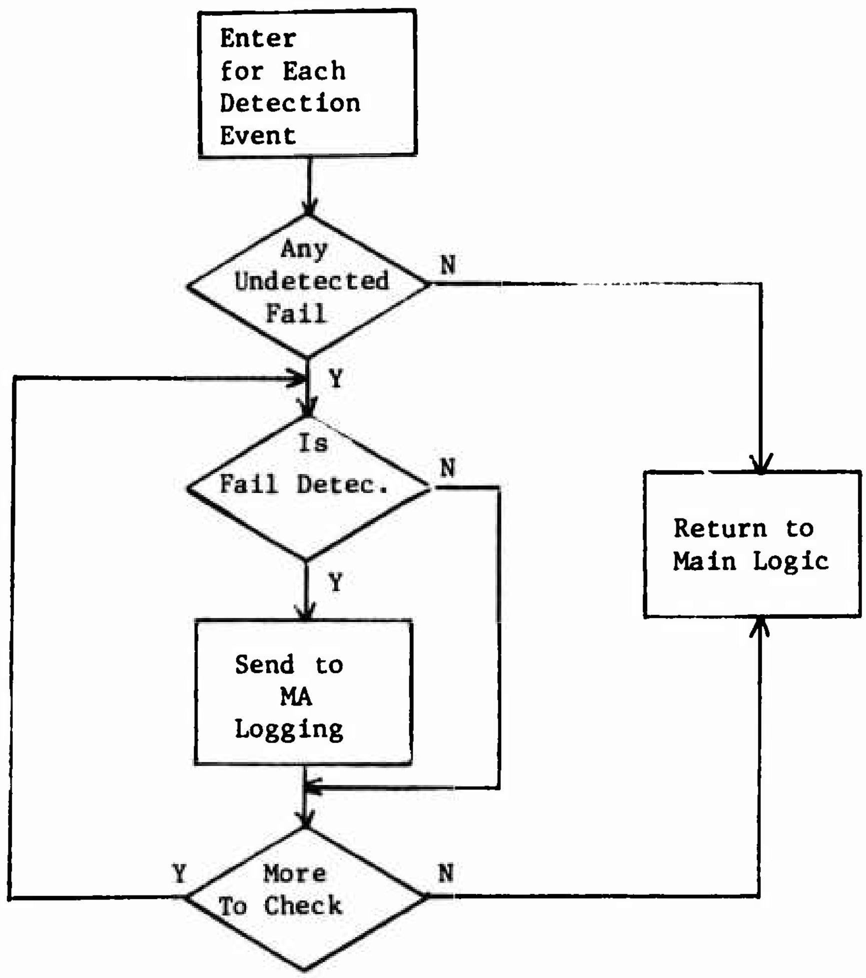

Figure 23. Failure Detection Subroutine.

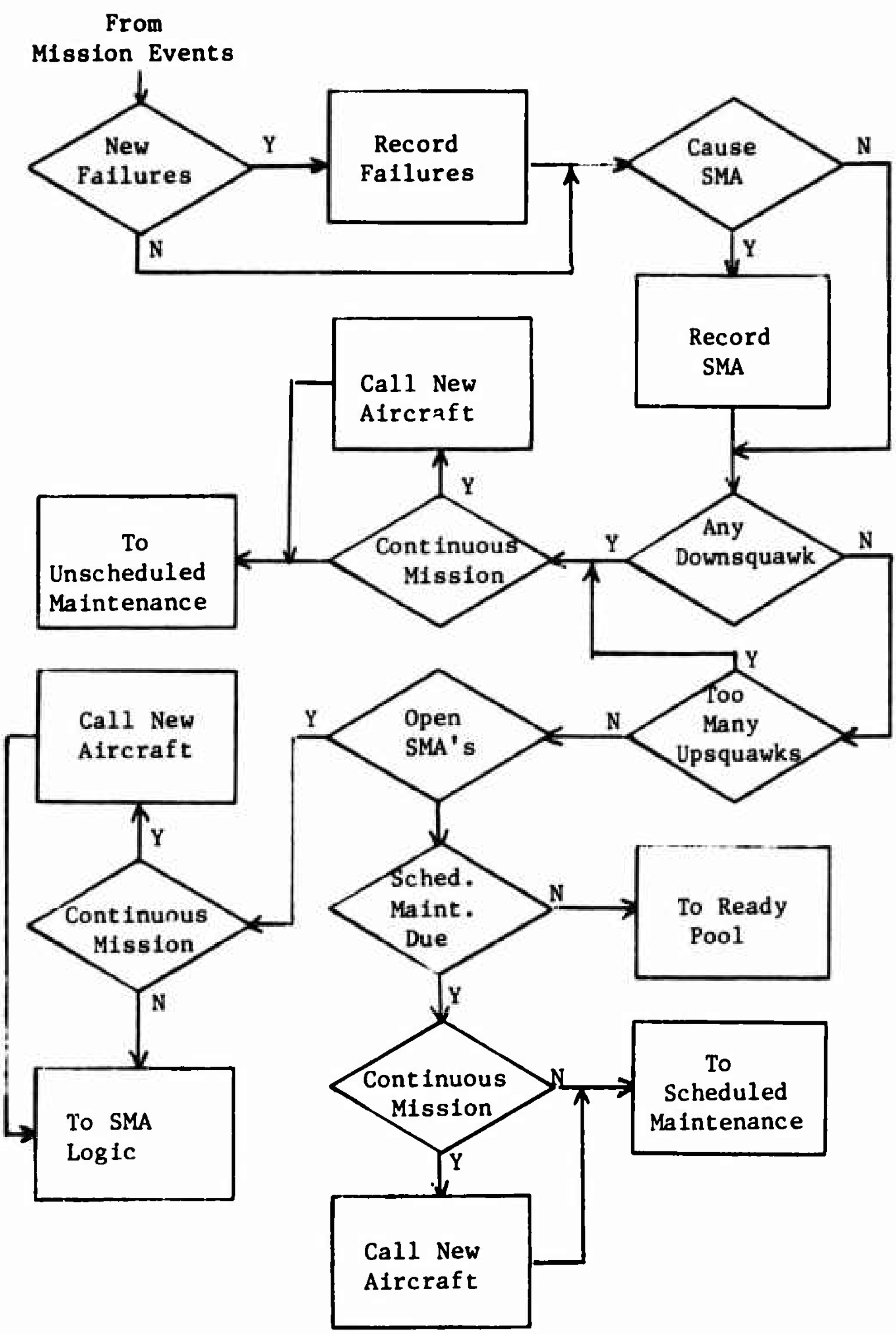

Figure 24. Aircraft Return Subroutine.

condition. Surge parts availability may be specified separately by input data. If the Monte Carlo process determines that the parts are not available, the probabilistic NORS Routine is entered. If parts are available, a transaction is sent to the Off-Equipment Repair Subroutine, and the primary maintenance action proceeds to the GSE Acquisition Subroutine.

In a deterministic supply scenario, parts availability is established by checking the count of parts in stock. If parts are not available, a transaction transfers to the deterministic NORS Routine. When parts are found, the shelf count is reduced by one and a test is made to determine if the reorder point has been reached. The main transaction sends a copy to off-equipment repair and then transfers to the GSE Acquisition Subroutine.

If parts are reordered, the time delay involved is computed from the input specified distribution function. Following the delay, the parts transfer to the Incoming Parts Subroutine for distribution to aircraft requiring that part.

When the Incoming Parts Subroutine does not immediately use the total reorder quantity, transactions will return to the Supply Routine, where any remaining parts are restocked on the shelf and an additional reorder is generated if required.

Data for the Supply Routine is extracted from input card type "H". Figure 25 is a flow diagram of the Supply Routine logic.

Incoming Parts Subroutine

The purpose of this subroutine is to distribute incoming parts in a deterministic supply scenario to any aircraft which may require the part. If no aircraft are awaiting the part, the total reorder quantity is restocked by transferring to the Supply Routine.

If one or more aircraft require the part, the maintenance actions from the aircraft are removed from a holding chain and made active in the model. Refer to Figure 26 to follow the discussion presented.

After all maintenance actions have been activated, a test is made to determine if the incoming order contained sufficient parts for all aircraft involved. When this is true, the parts are immediately distributed and a count is maintained of the remaining parts in the order. Each time an aircraft receives a part, its readiness status is interrogated and adjusted as

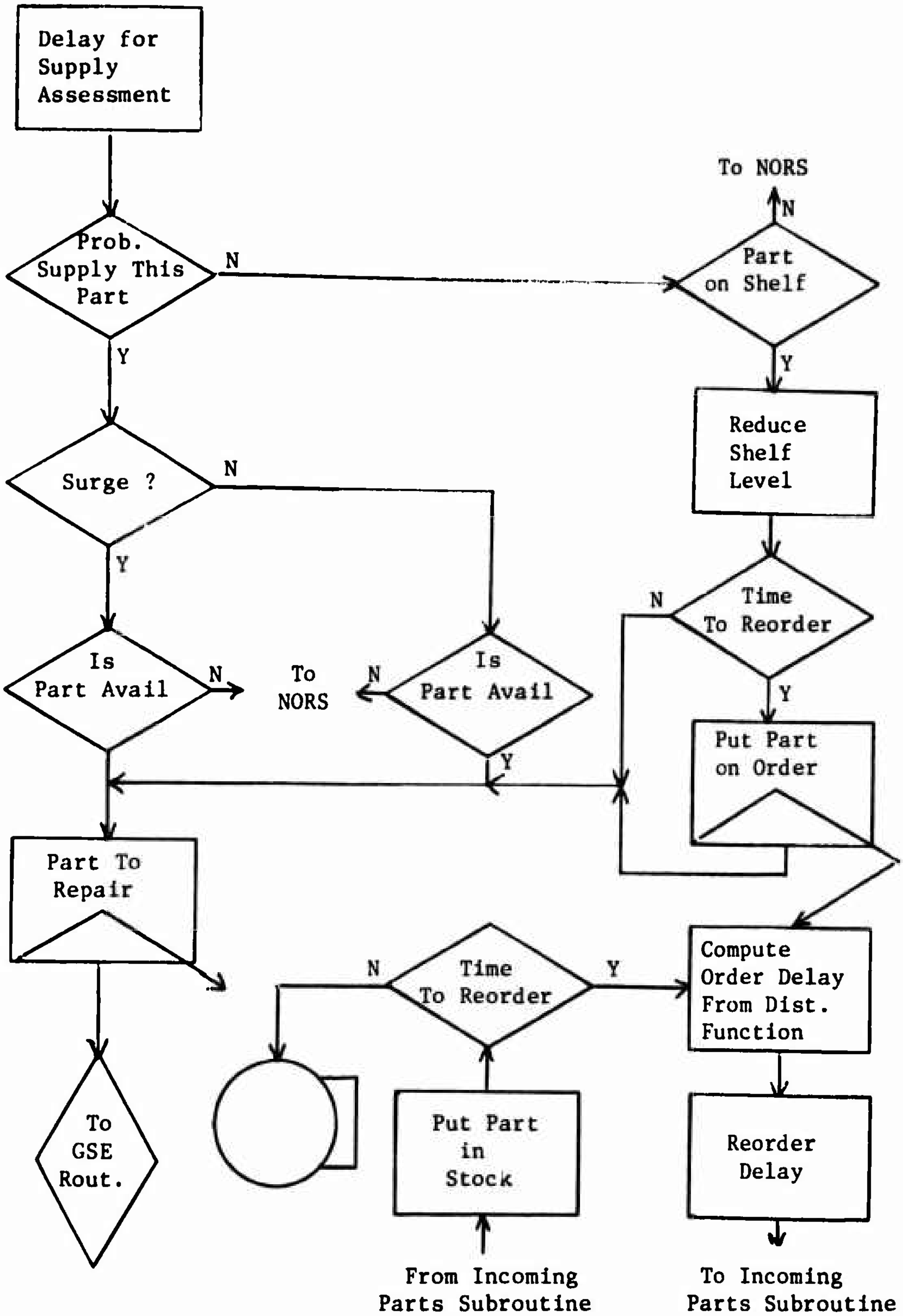

Figure 25. Supply Routine.

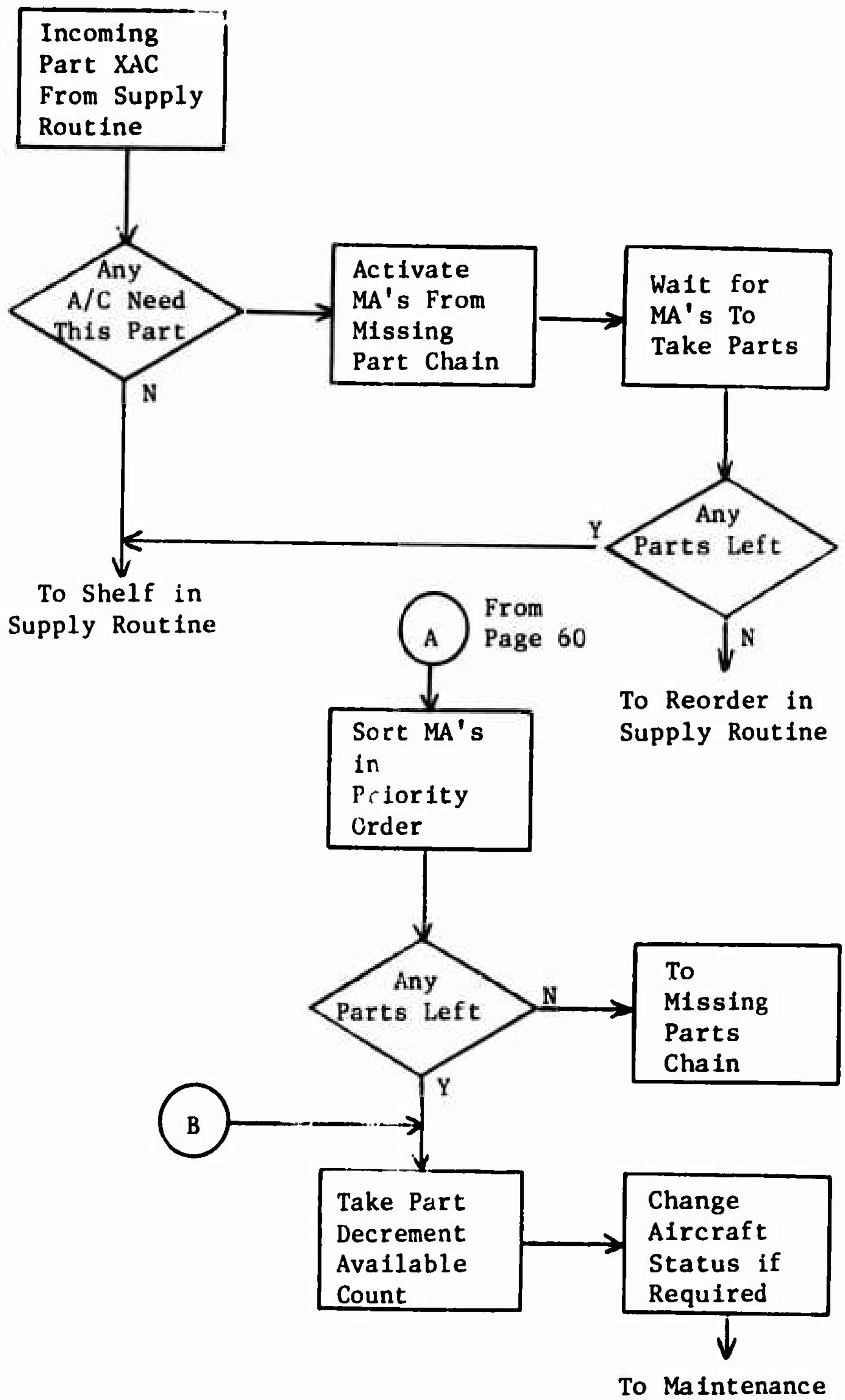

Figure 26. Incoming Parts Subroutine.

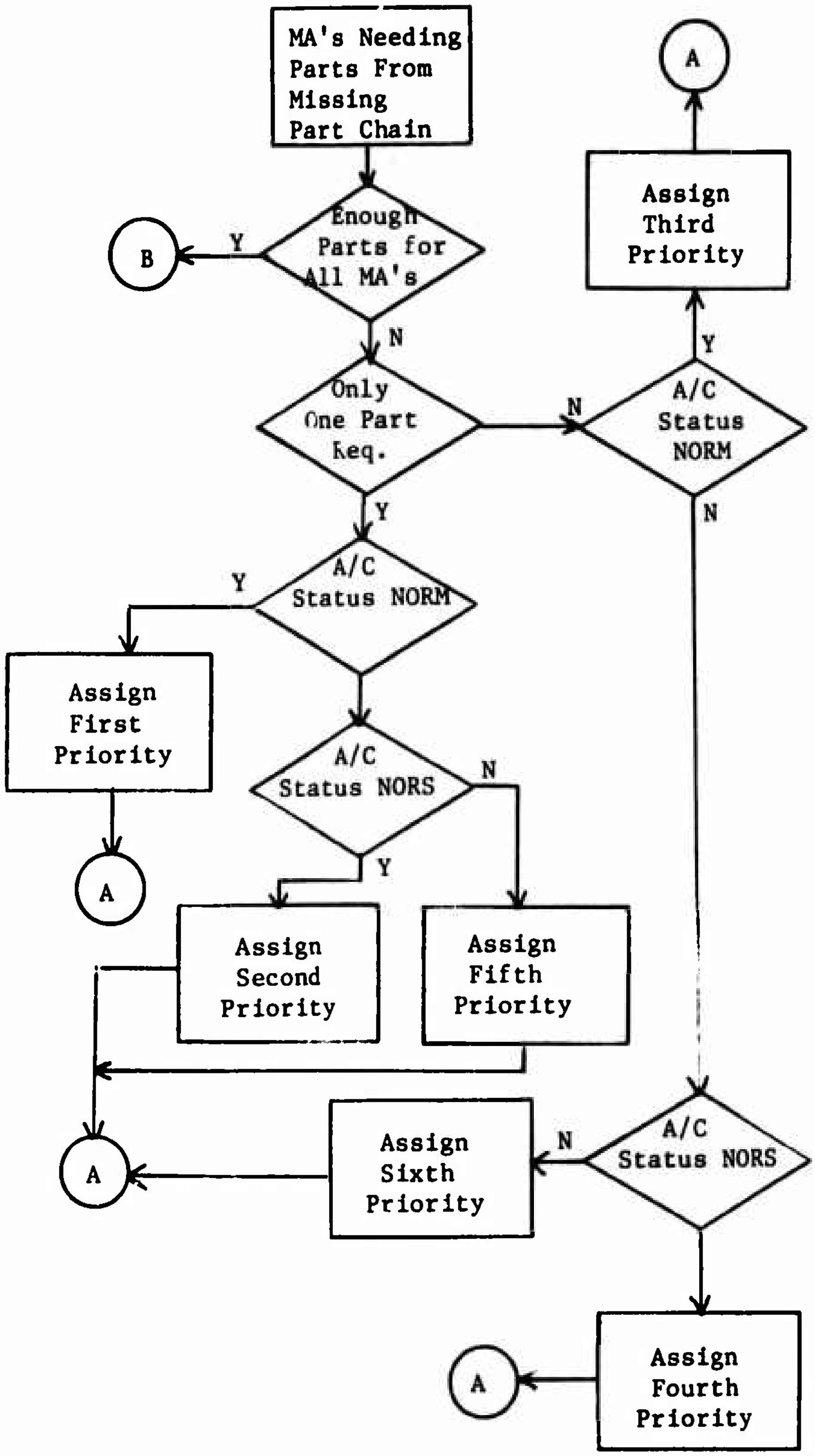

Figure 26. Continued

required. The maintenance action receiving the part transfers to the Aircraft Repair Routine, where unscheduled maintenance is accomplished.

If the immediate requirement for parts exceeds the quantity in the order, it is necessary to establish a priority system. Priorities are established as follows:

Priority 1: Aircraft in a NORM status requiring only one part.

Priority 2: Aircraft in a NORS status requiring only one part.

Priority 3: Aircraft in a NORM status requiring more than one part.

Priority 4: Aircraft in a NORS status requiring more than one part.

Priority 5: Ready aircraft with one upsquawk requiring parts.

Priority 6: Ready aircraft with more than one upsquawk requiring parts.

Each maintenance action is assigned one of these six priorities, and parts are distributed in the priority order established. If all requirements are not satisfied, the maintenance actions not obtaining parts are relinked to the missing parts chain. The incoming order transaction is returned to the Supply Routine, where a new part order will be placed.

This subroutine does not require the use of any user-specified input data.

<u>Probabilistic Supply NORS Subroutine</u>

This logic is entered when a maintenance action for a subsystem defined with probabilistic supply is unable to obtain a needed part. A test is made to determine if input data permits cannibalization. These data are extracted from card type "H". If cannibalization is permitted, the cannibalization subroutine is entered. When cannibalization is not permitted, the maintenance actions are separated into two classes: upsquawks and downsquawks.

A downsquawk will change the aircraft status to NORS. At this time any maintenance actions that are being delayed by parallel maintenance restrictions are reexamined to see if the blocking condition has been removed. A part order delay is computed from the distribution specified on input card "H" and the computed delay introduced in the model. When the part order delay has expired, aircraft status is interrogated and adjusted if required. The maintenance action transfers to the Aircraft Repair Routine, where unscheduled maintenance is completed.

When the maintenance action requiring the part is an upsquawk, it first causes maintenance actions delayed by parallel maintenance restrictions to be reevaluated. Upsquawks fall into two categories. If it is a must upsquawk, that is, one in work because the maximum upsquawk limit has been exceeded, it will attempt to activate additional upsquawks. In either case, aircraft status is interrogated and revised according to its existing maintenance posture. A part order delay is introduced, after which the upsquawk will be placed in the correct maintenance status so that it will be inducted into unscheduled maintenance at the appropriate time.

Figure 27 presents a flow diagram of the probabilistic supply NORS subroutine.

<u>Deterministic Supply NORS Subroutine</u>

The function of this subroutine is similar to that previously described for probabilistic supply. Maintenance actions enter when a subsystem operating in the deterministic supply mode cannot find a needed part. Input data from card "H" are examined to determine if the reorder quantity for this specific part is equal to one. When this is true, a transaction is sent to the Supply Routine, where the part is placed on order.

All transactions cause the examination of maintenance being delayed because of parallel maintenance restrictions to determine if the blocking condition has been removed. Upsquawk maintenance actions are handled differently from downing maintenance actions. If the upsquawk is a must work upsquawk, it attempts to activate an additional maintenance action. All upsquawks interrogate the aircraft status and revise it as required. The maintenance action is then linked onto a missing parts chain, where it will remain until it is activated by the Incoming Parts Subroutine.

Downing maintenance actions use input data from card "H" to establish if cannibalization is permitted.

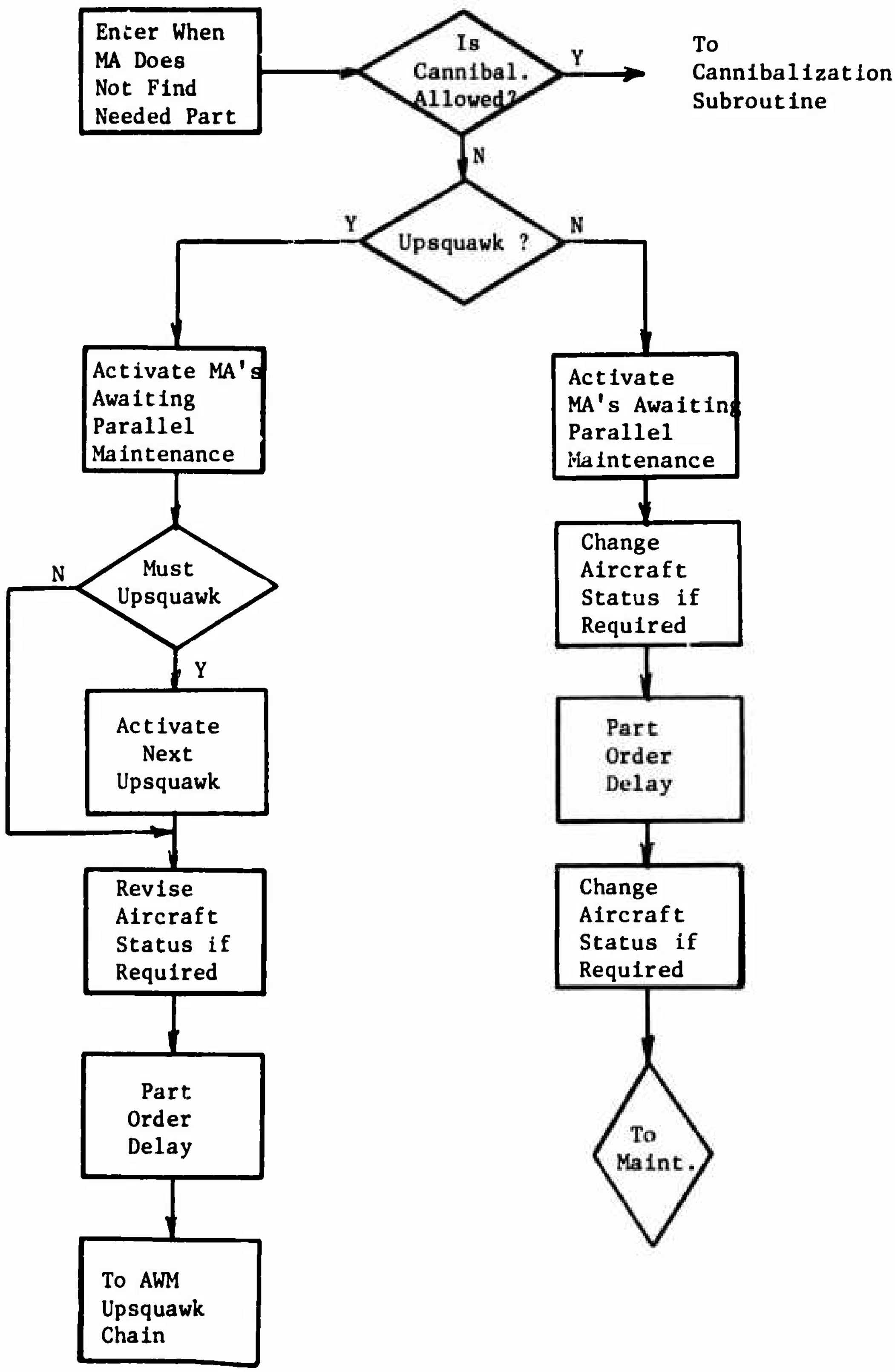

Figure 27. Probabilistic Supply NORS Subroutine.

Cannibalization is accomplished by transferring to the Cannibalization Subroutine. If this is not permitted, the aircraft status is appropriately revised and the maintenance action is linked onto the missing parts chain.

A flow diagram of the Deterministic Supply NORS Subroutine is presented in Figure 28.

Cannibalization Subroutine

Cannibalization is attempted when a needed part is not available in supply and input data permits cannibalization. Only NORS aircraft are cannibalized. If NORS aircraft are not available, the transaction will return to the NORS Routine to adjust the status of the aircraft needing the part.

Aircraft are selected as cannibalization candidates in the order of their estimated NORS time. Long time aircraft are cannibalized first. Each aircraft selected is interrogated to determine if the required part is already missing. If this is true, another aircraft is selected and the process repeated until no NORS aircraft remain or the required part is found available.

When the part is cannibalized from a NORS aircraft, it is recorded as missing so that incoming parts will be correctly allocated. The maintenance action which caused the cannibalization transfers to the Aircraft Repair Routine where unscheduled maintenance is accomplished.

A flow diagram of the Cannibalization Subroutine is presented in Figure 29.

Scheduled Maintenance Routine

This routine controls the performance of scheduled maintenance events. It is entered by an aircraft which has been found to need scheduled maintenance in the Scheduled Maintenance Induction Subroutine. Entering aircraft have their status updated and are placed in a NORMS status. If the event is an SMA, the aircraft leaves the routine and enters SMA logic. For non-SMA events, the event time is computed from the input data provided on card type "L". Also obtained from this card are manpower and GSE requirements. The routine establishes the necessary maintenance assets, and the aircraft is delayed until all assets have been obtained and their appropriate usage time expended.

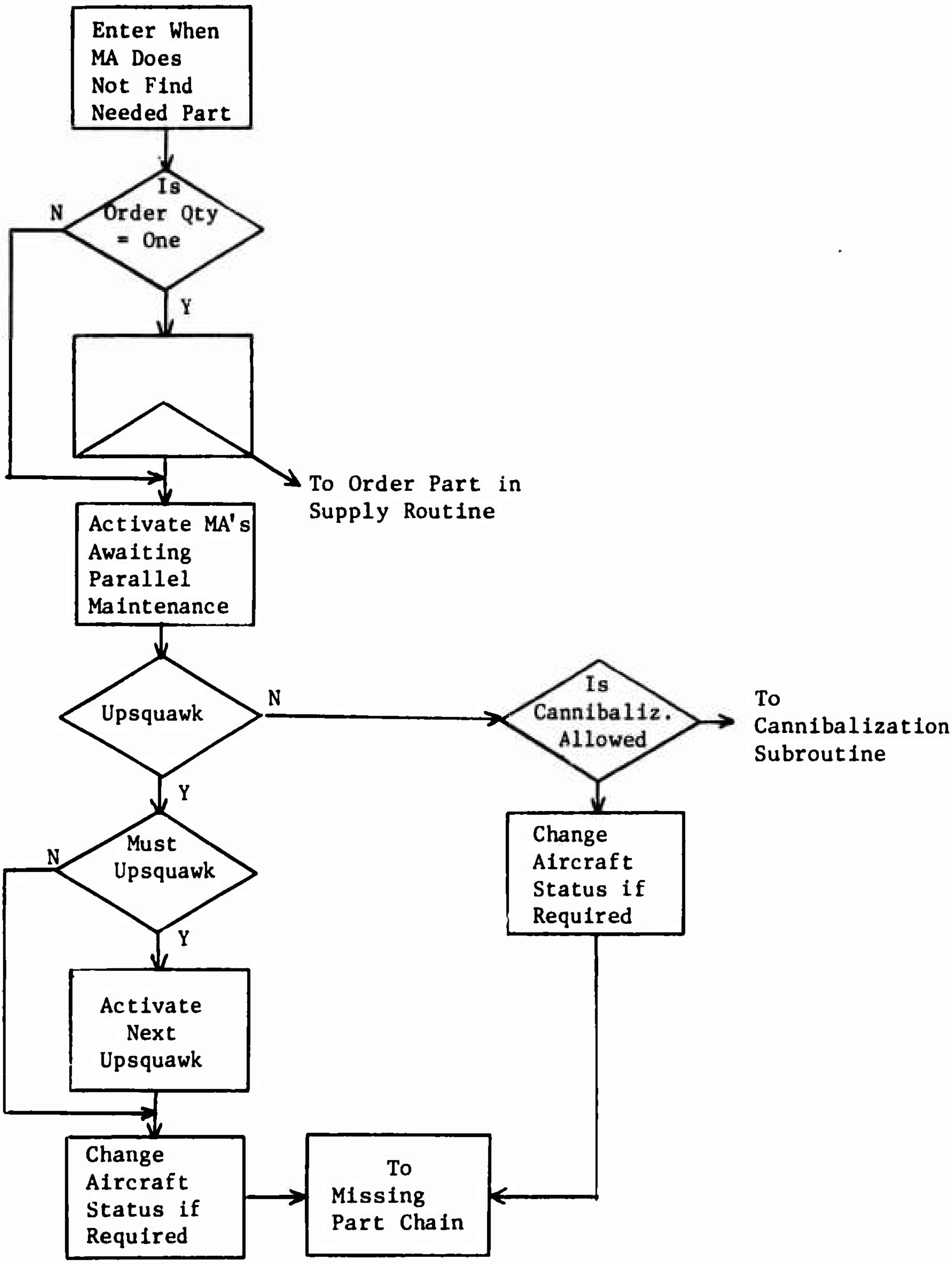

Figure 28. Deterministic Supply NORS Subroutine.

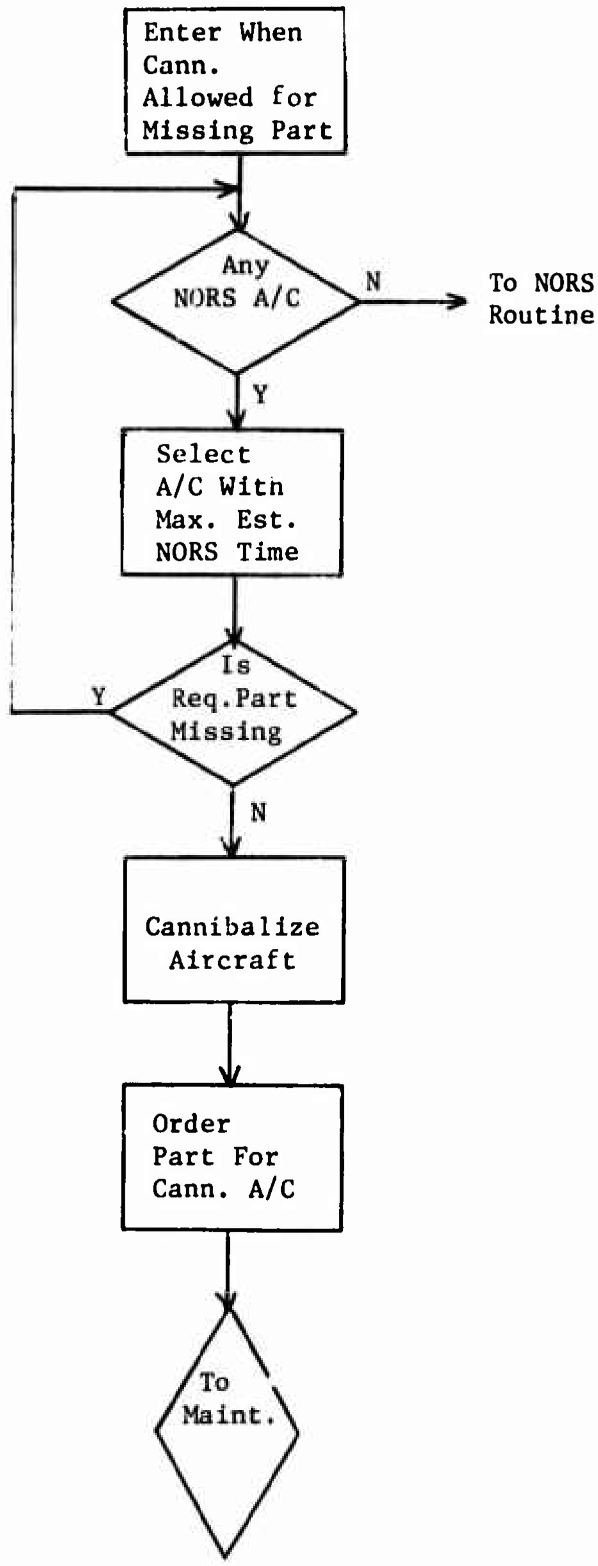

Figure 29. Cannibalization Subroutine.

At the completion of all tasks for the scheduled event, the maintenance schedule is updated. This establishes the time in flying hours or calendar days when the aircraft will be reinducted. Data from card type "M" is used to determine if a TBO is due. It is also established whether this TBO is permitted at this time. When TBO's are to be worked, the aircraft is again delayed until the removal has been accomplished. TBO removals are performed in the Aircraft Repair Routine.

Following TBO events, the aircraft transfers to the Failure Detection Subroutine, where data from card type "E" is used to determine if any new maintenance actions are detected. The number of upsquawks outstanding on the aircraft is then compared to the upper limit specified on card type "B". If the limit is exceeded, the aircraft is inducted into scheduled maintenance.

Test hop requirements are also checked. The aircraft proceeds either to test hop logic or has its status revised and reenters the ready pool.

A flow diagram for the Scheduled Maintenance Routine is presented in Figure 30.

Scheduled Maintenance Initialization Subroutine

The Scheduled Maintenance Initialization Subroutine is utilized once during each simulation experiment. Its purpose is to establish the timing requirements for all calendar and flight hour oriented scheduled maintenance events. Each aircraft in the model is examined for up to twenty scheduled maintenance events. When an event has been defined on card type "L", its time of occurrence in the model is computed from the initialized aircraft flying hours and the input specified inspection cycle. This process is repeated for each aircraft and each event specified.

A flow diagram of this process is presented in Figure 31.

Scheduled Maintenance Induction Subroutine

The Scheduled Maintenance Induction Subroutine is used to determine if an aircraft should be inducted into a scheduled maintenance event. This decision is made by a logical examination of input specified induction tolerance and the status of aircraft and maintenance assets in the simulation scenario.

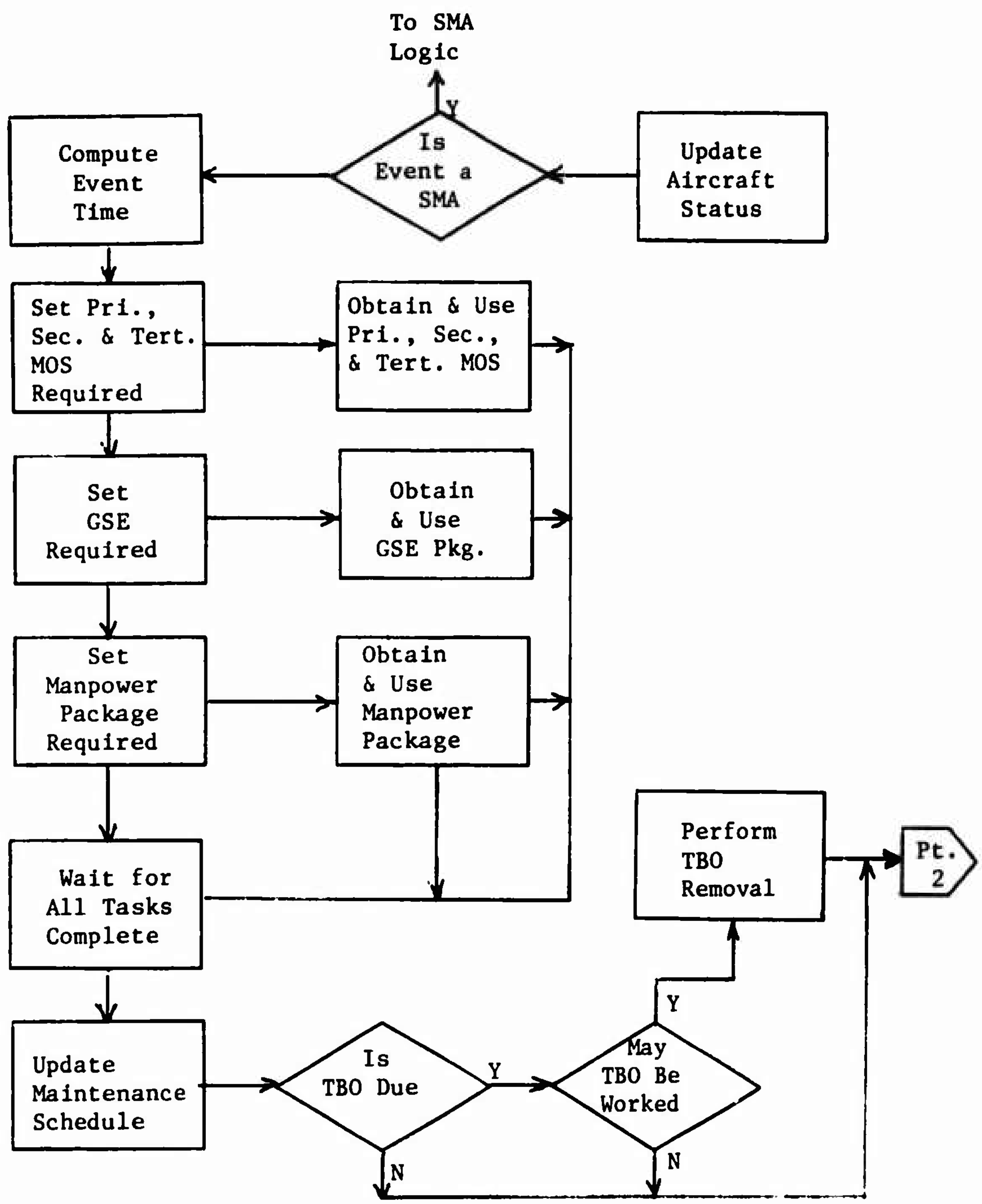

Figure 30. Scheduled Maintenance Routine .

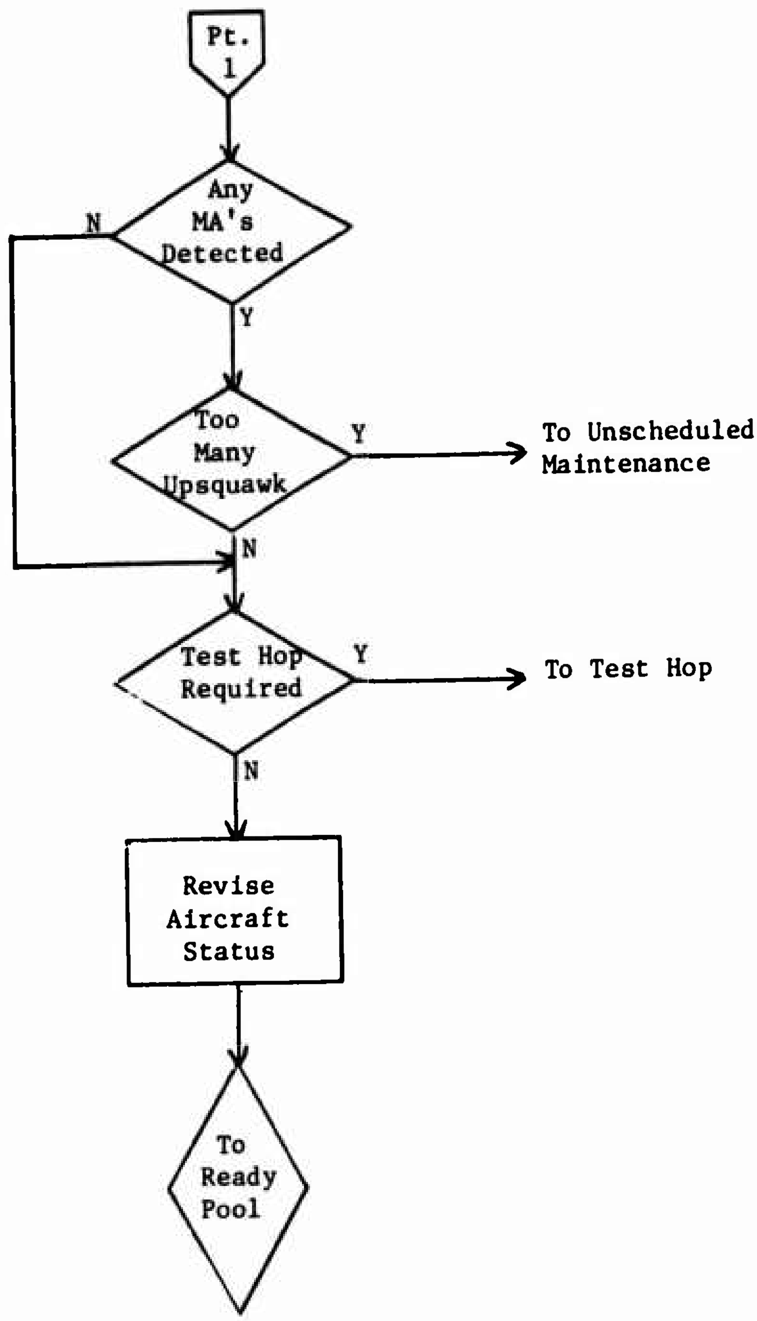

Figure 30. Continued

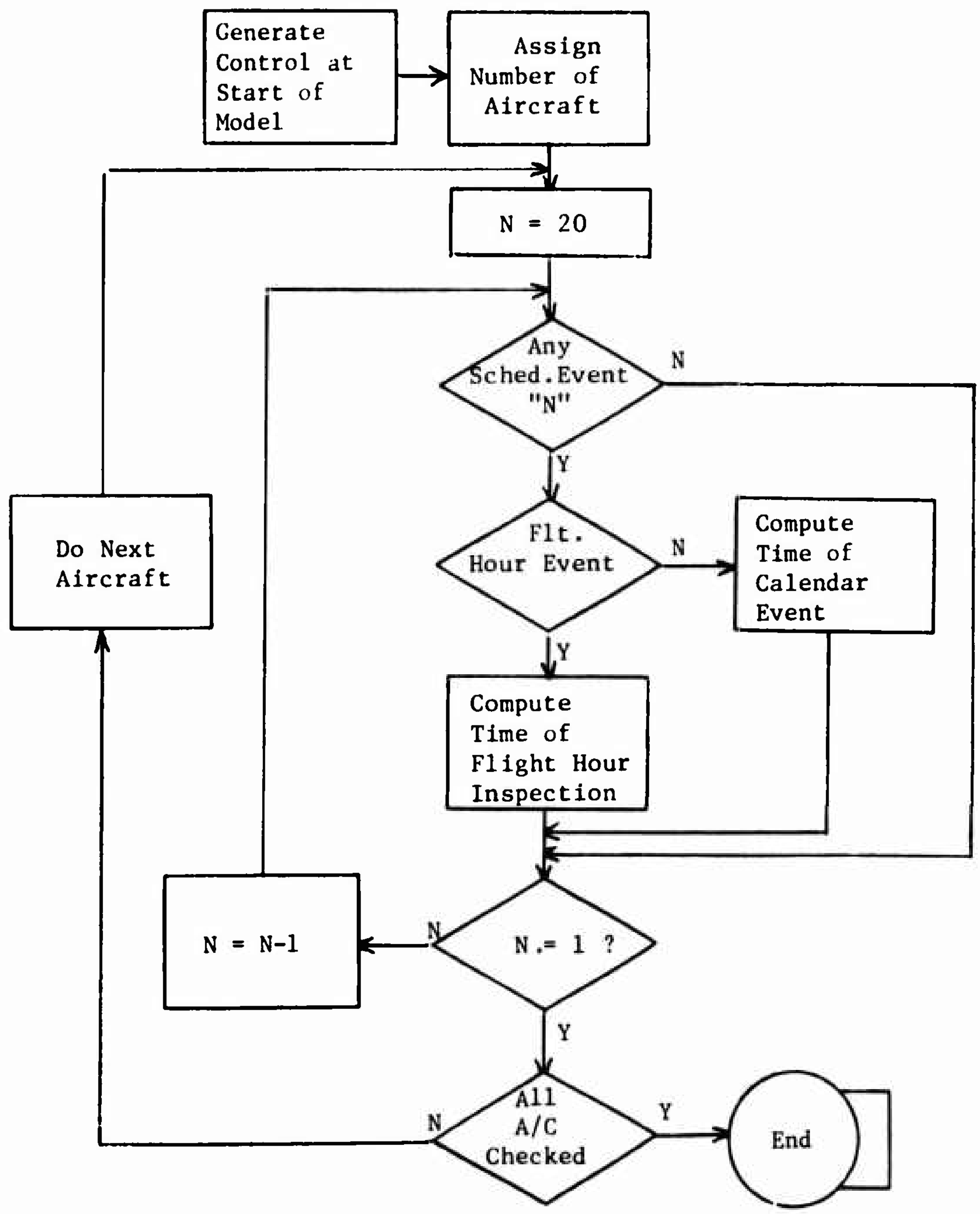

Figure 31. Scheduled Maintenance Initialization Subroutine.

The subroutine is entered by control transactions from either aircraft return or scheduled maintenance control. Transactions from aircraft return are tested to determine if the requirement for the event in question is already established and also if the event is a subevent of one already established. If neither of the above is true, no further action is taken.

All requirements are tested to determine if the induction tolerance for the event has been exceeded. When this is true, an attempt is made to immediately induct the aircraft into scheduled maintenance.

If induction tolerance has not been exceeded, a series of tests relative to model status is used to decide if the performance of the scheduled maintenance should be delayed. When no other aircraft are currently in a scheduled maintenance status, an attempt is made to immediately induct the aircraft. Another test is made relative to the input specified limit on aircraft in each scheduled event. If the limit has been reached, the requirement for this inspection is recorded on a pending event file.

If event limits are not exceeded, an estimate of the time to accomplish the required maintenance is made, and this value is compared to the time remaining until the next scheduled launch. If sufficient time exists to accomplish the task, an attempt is made to induct the aircraft.

When insufficient time exists the total number of mission and standby aircraft required for the next two launches is computed. If there are currently enough ready aircraft to meet these requirements, the aircraft is inducted into the scheduled event. When aircraft assets are low, the scheduled maintenance is delayed and the requirement recorded on the pending event file.

Figure 32 is a flow diagram of the Scheduled Maintenance Induction Subroutine.

Scheduled Maintenance Planning Subroutine

The purpose of this subroutine is to provide scheduled maintenance planning on a daily basis and to maintain each aircraft's position in calendar maintenance cycles. All pending scheduled events that have been recorded by the Scheduled Maintenance Induction Subroutine are reactivated to determine if aircraft should be inducted.

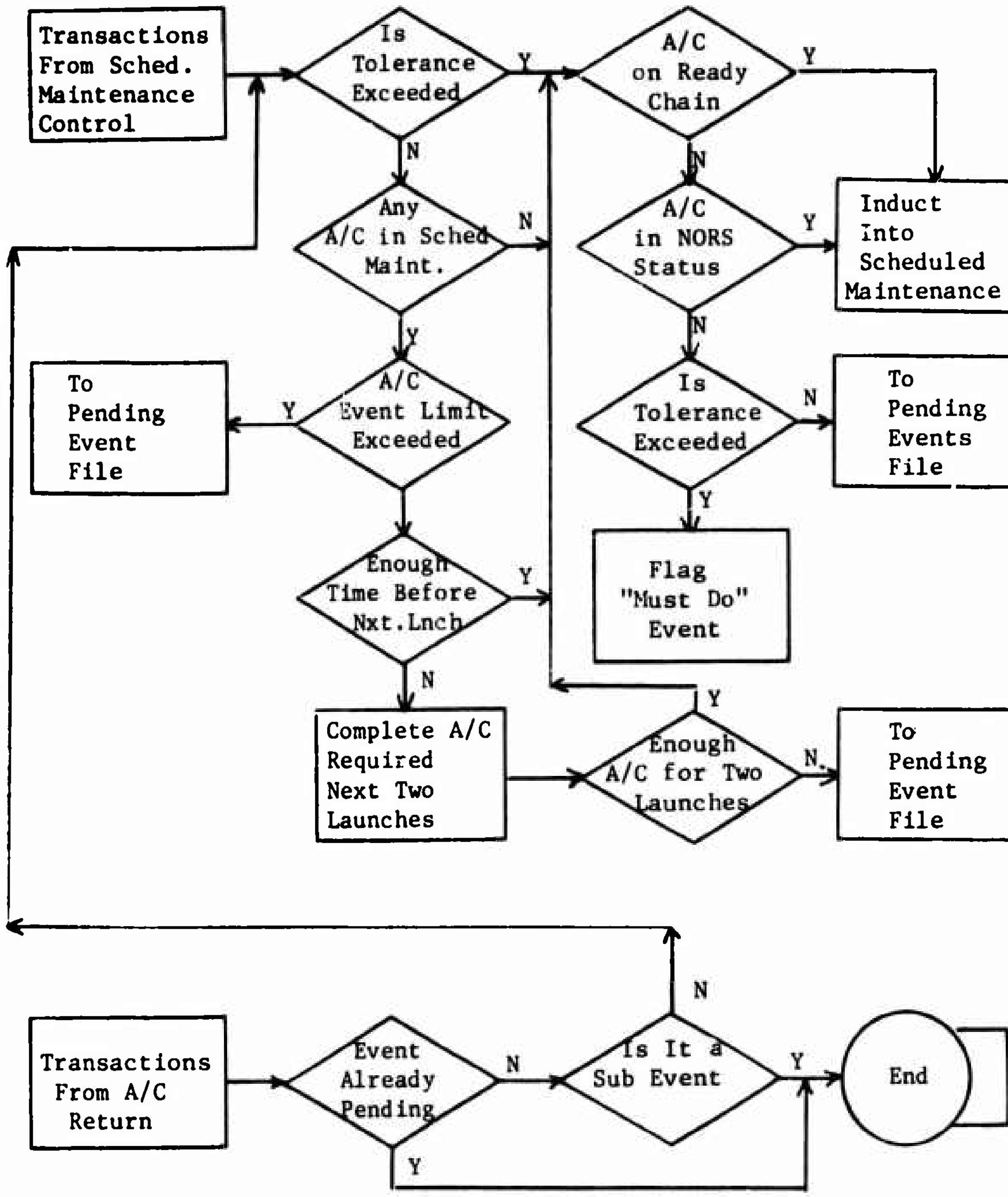

Figure 32. Scheduled Maintenance Induction Subroutine.

If no calendar maintenance has been defined in the simulation experiment, the transaction will be terminated. When calendar maintenance is specified, the calendar cycle day parameter is incremented on each aircraft. Each aircraft is then examined to determine if it falls within the induction tolerance specified. If calendar maintenance is due, the Scheduled Maintenance Induction Subroutine is entered to determine if aircraft should be inducted.

A flow diagram of the Scheduled Maintenance Planning Subroutine is presented in Figure 33.

<u>Daily Inspection Control Subroutine</u>

This subroutine utilizes input data from card type "M" to control the induction of aircraft into daily inspections. There are two portions to the logic in this subroutine. The first section establishes the time of day when each aircraft is examined to determine if a daily inspection should be performed. This time may be specified in three ways: (1) time of day control; (2) time before first flight control; and (3) time after last flight control. For each control method specified, the appropriate delay until dailies are due is computed. After the computed delay has expired, all ready aircraft are interrogated to determine if a daily must be performed. Following this, an additional delay until midnight is introduced and the entire cycle repeated.

The second portion of daily control logic looks at individual aircraft to establish if inspection is required. Each aircraft is marked at the time of the daily inspection. Entering aircraft are tested to determine if any flights have taken place since the previous daily. Input specified daily intervals for flying and nonflying periods are then tested to determine if each aircraft requires an inspection. Aircraft are either inducted for dailies or returned to the ready pool.

Aircraft also enter this subroutine after a daily inspection has been accomplished. The daily time mark and aircraft flight flag are reset. Additional maintenance actions may have been detected during the daily inspection. This is determined in the Failure Detection Subroutine. The number of open maintenance actions is tested against the input specified limits, and the aircraft is either inducted into unscheduled maintenance or returned to the ready pool.

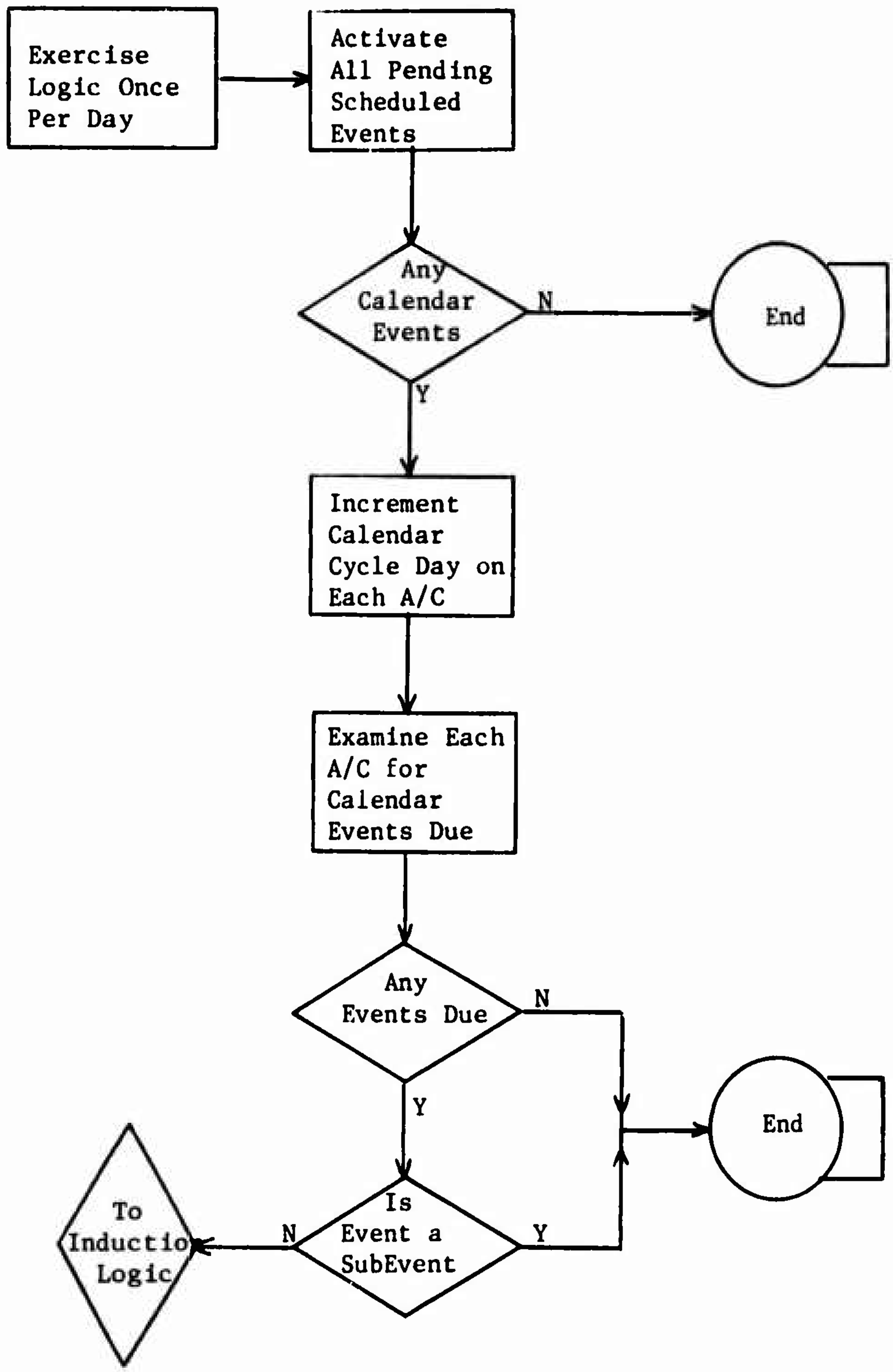

Figure 33. Scheduled Maintenance Planning Subroutine.

Figure 34 presents a flow diagram of the Daily Inspection Control Subroutine.

Aircraft Repair Routine

The primary function of this routine is to work off all unscheduled maintenance actions. TBO removals and off equipment repair are also handled. Prior to entering, each MA has obtained, as required, a part, a GSE package and a primary MOS. From this point in the model, GSE will proceed to accomplish its usage independent of manpower.

The primary MOS will work until his portion of MTTR has elapsed or shift time expires. The secondary MOS will be called to the job so that under a no-queue condition he will finish at the end of the specified repair time. The tertiary MOS will be called at a random interval between start and finish of MTTR.

The three MOS types will work independently. If multiple men of the same MOS are specified, they must all be available for work to proceed.

Completion of the MA occurs when all MOS's have completed their work and the MA is not in incorrect diagnosis or incorrect removal/repair. Card types "G" and "H" supply data to this routine.

Figure 35 presents a flow diagram of routine logic.

Incorrect Diagnosis Subroutine

The purpose of this subroutine is to assess the probability of incorrect diagnosis and select an incorrect element for repair action.

Each maintenance action on any element may have specified a probability of incorrect diagnosis. The subroutine tests this probability and selects another element in the same subsystem to be repaired when an incorrect diagnosis is indicated.

Input data for the probability of incorrect diagnosis are extracted from card type "G". Figure 36 presents a flow diagram of this subroutine.

Parallel Maintenance Subroutine

This subroutine examines each unscheduled maintenance action for possible interference from MA's presently in work.

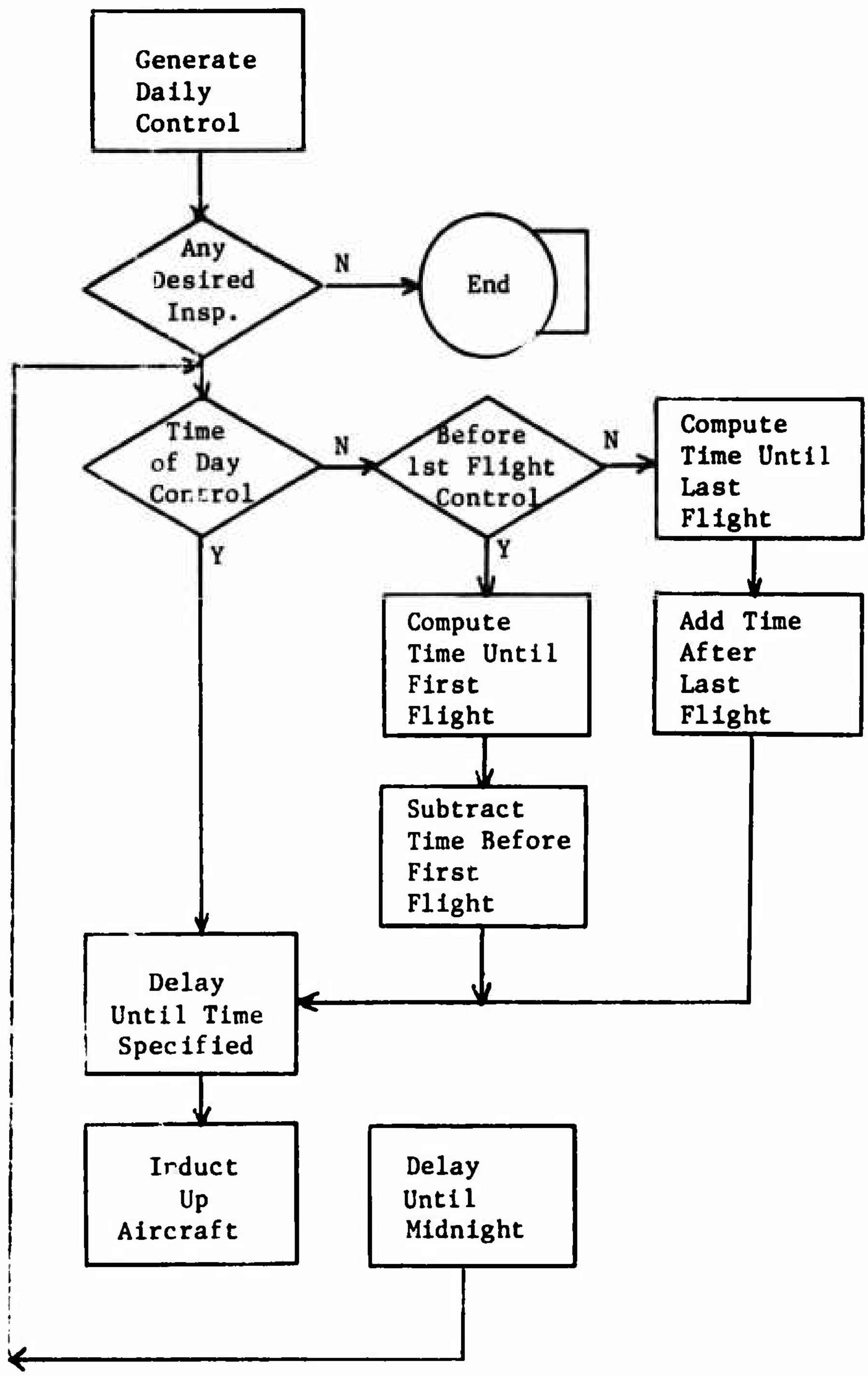

Figure 34. Daily Inspection Control Subroutine.

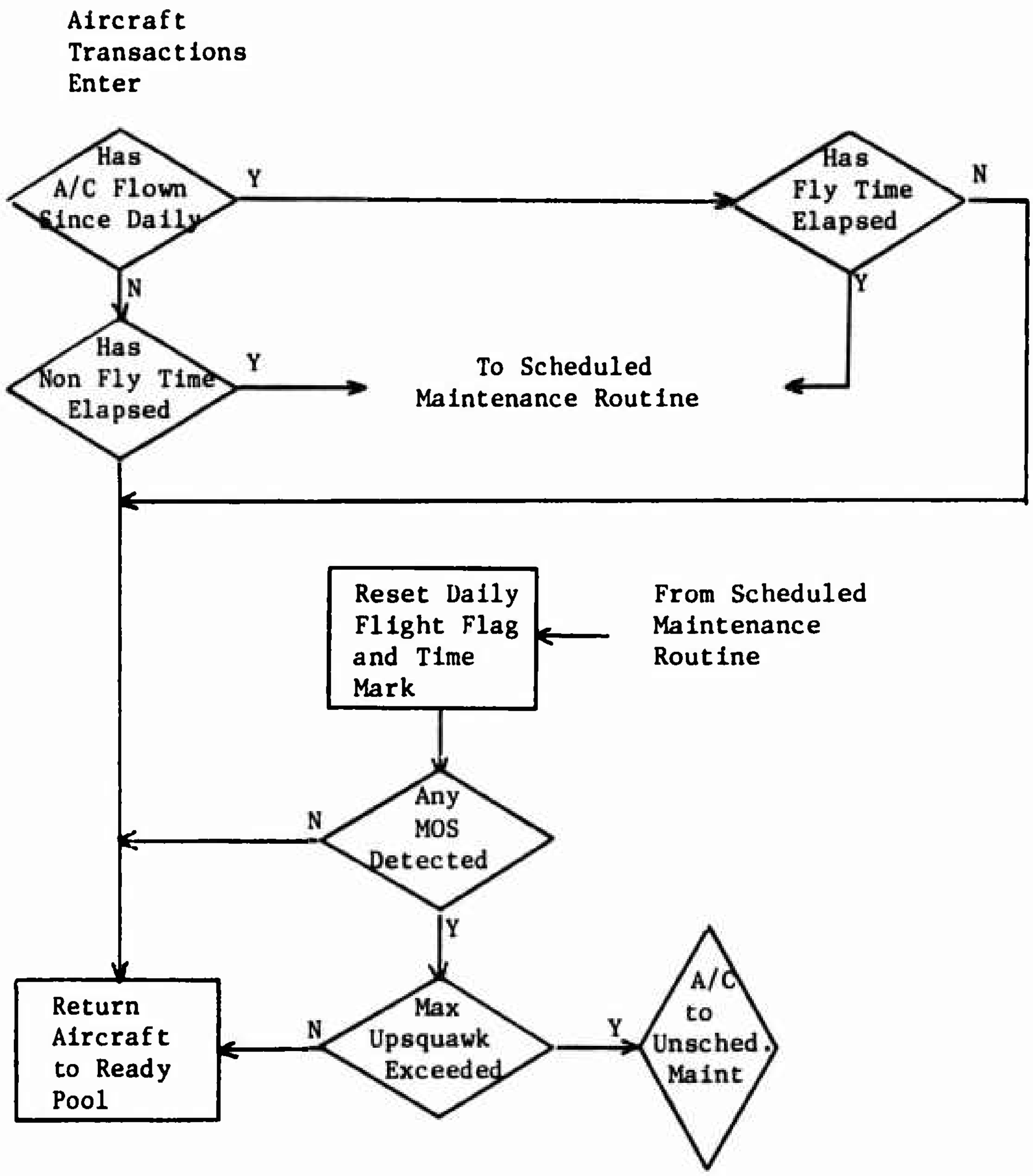

Figure 34. Continued

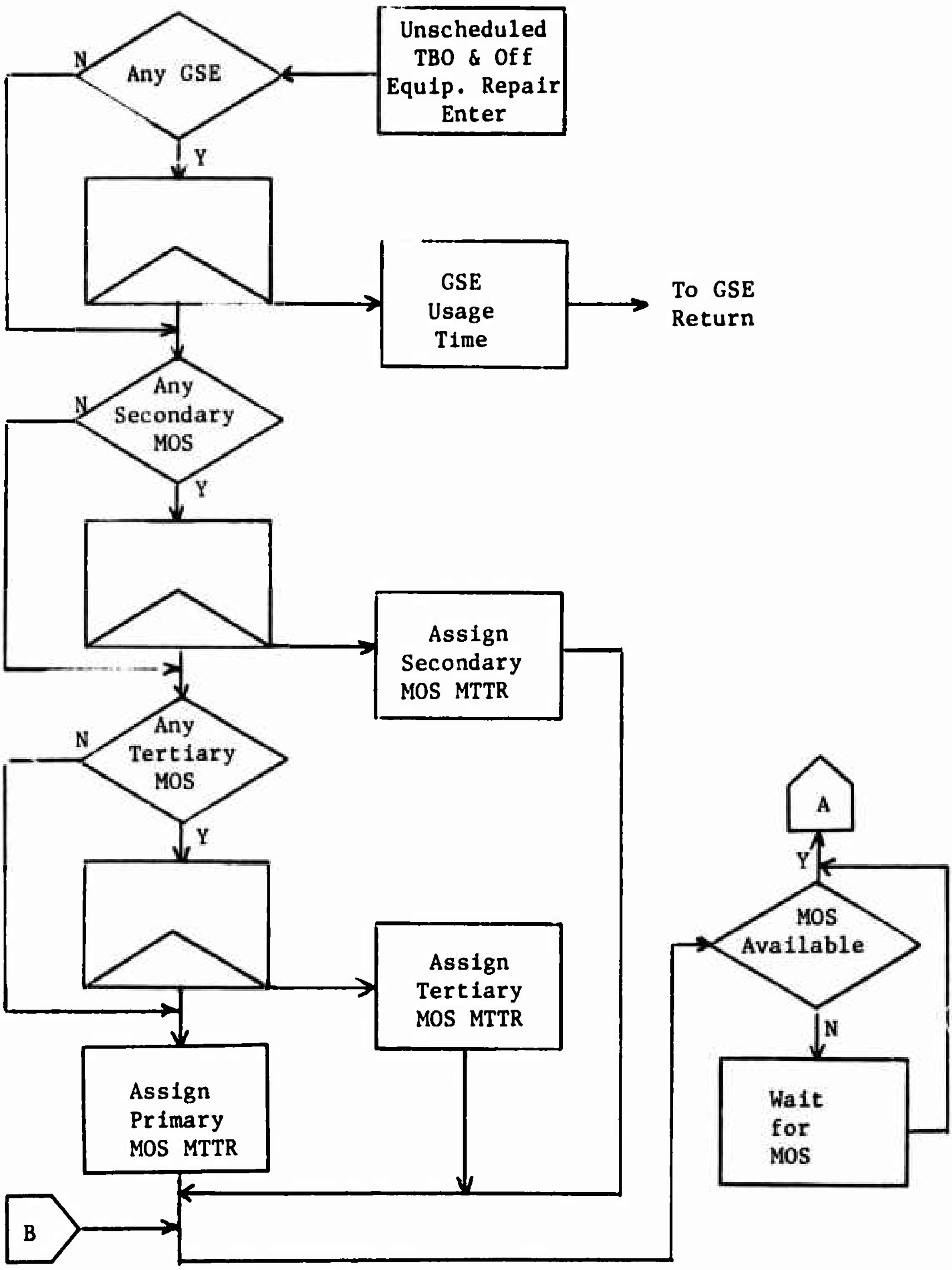

Figure 35. Aircraft Repair Routine.

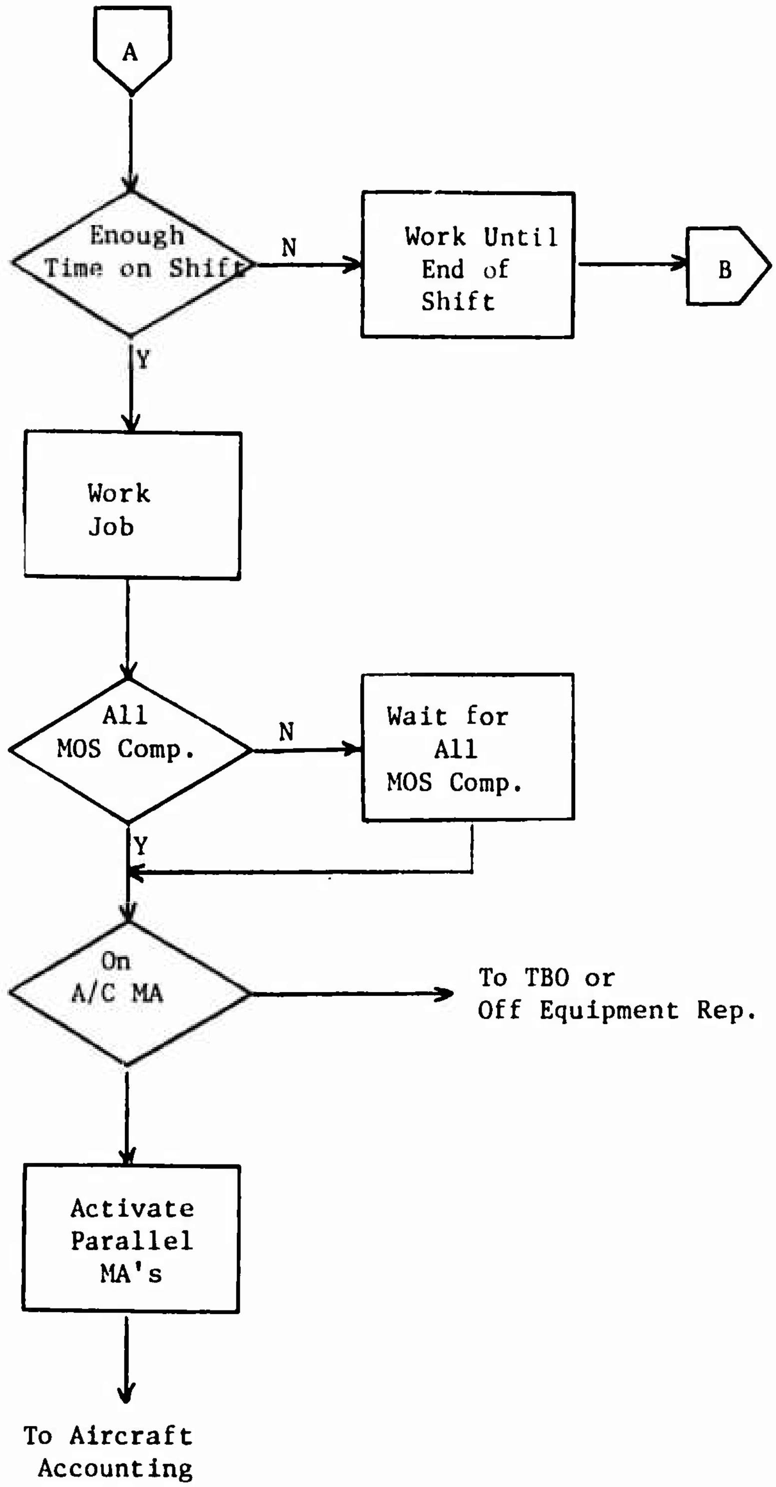

Figure 35. Continued

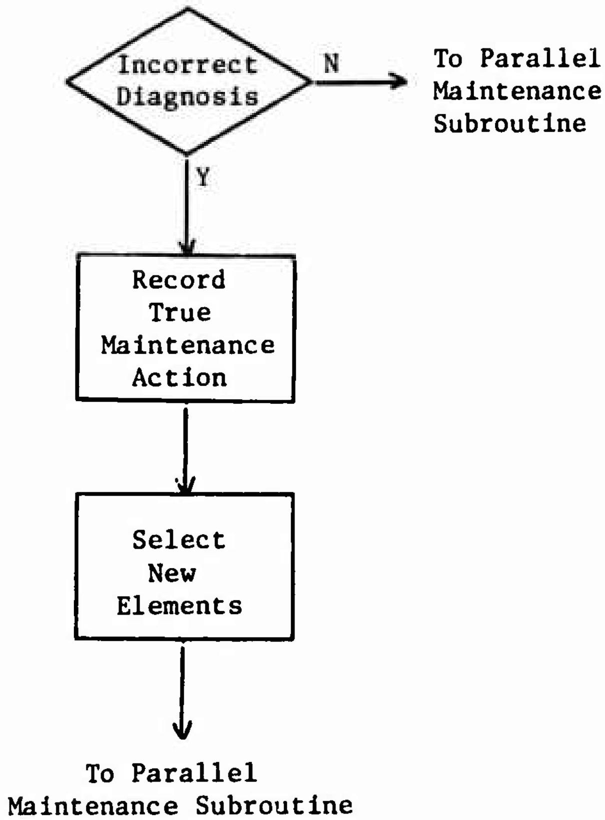

Figure 36. Incorrect Diagnosis Subroutine.

Parallel maintenance may be restricted in three ways: a specified maximum on the entire aircraft, a maximum on any subsystem, and a probability of parallel maintenance between subsystems. These three may be specified individually or in combination. This subroutine examines each maintenance action for each of these three limiting conditions before permitting the maintenance to be accomplished.

Card types "I" and "J" supply input to this subroutine. A logic diagram is presented in Figure 37.

MOS Assessment Subroutine

The purpose of this subroutine is to assign primary, secondary and tertiary MOS requirements to each unscheduled maintenance action.

Up to three types of MOS may be required to accomplish an unscheduled maintenance action. This routine assigns these requirements by type and quantity required to each unscheduled maintenance action. The administrative delay specified on input is also encountered in this subroutine.

Card types "G" and "H" supply input to this subroutine. A logic diagram is presented in Figure 38.

GSE Acquisition Subroutine

The purpose of this subroutine is to evaluate the availability of all GSE and the primary MOS before beginning repair on a maintenance action.

Each maintenance action may specify a GSE package. Each package may specify up to 99 pieces of GSE. This subroutine evaluates the availability of all required GSE and the primary MOS before beginning work on the MA. GSE is not reserved; i.e., until all required pieces of GSE are available, none are taken. Also, GSE is not taken until the primary MOS specified is simultaneously available.

When manpower and GSE requirements are simultaneously met, the time to acquire and set up the GSE is computed and compared to the time of shift change for the MOS. If setup time is less, the task is completed. If not, that portion of the job that can be accomplished up to shift change is worked. The oncoming shift is then evaluated to find the necessary manpower, and the GSE acquisition and hookup time is completed.

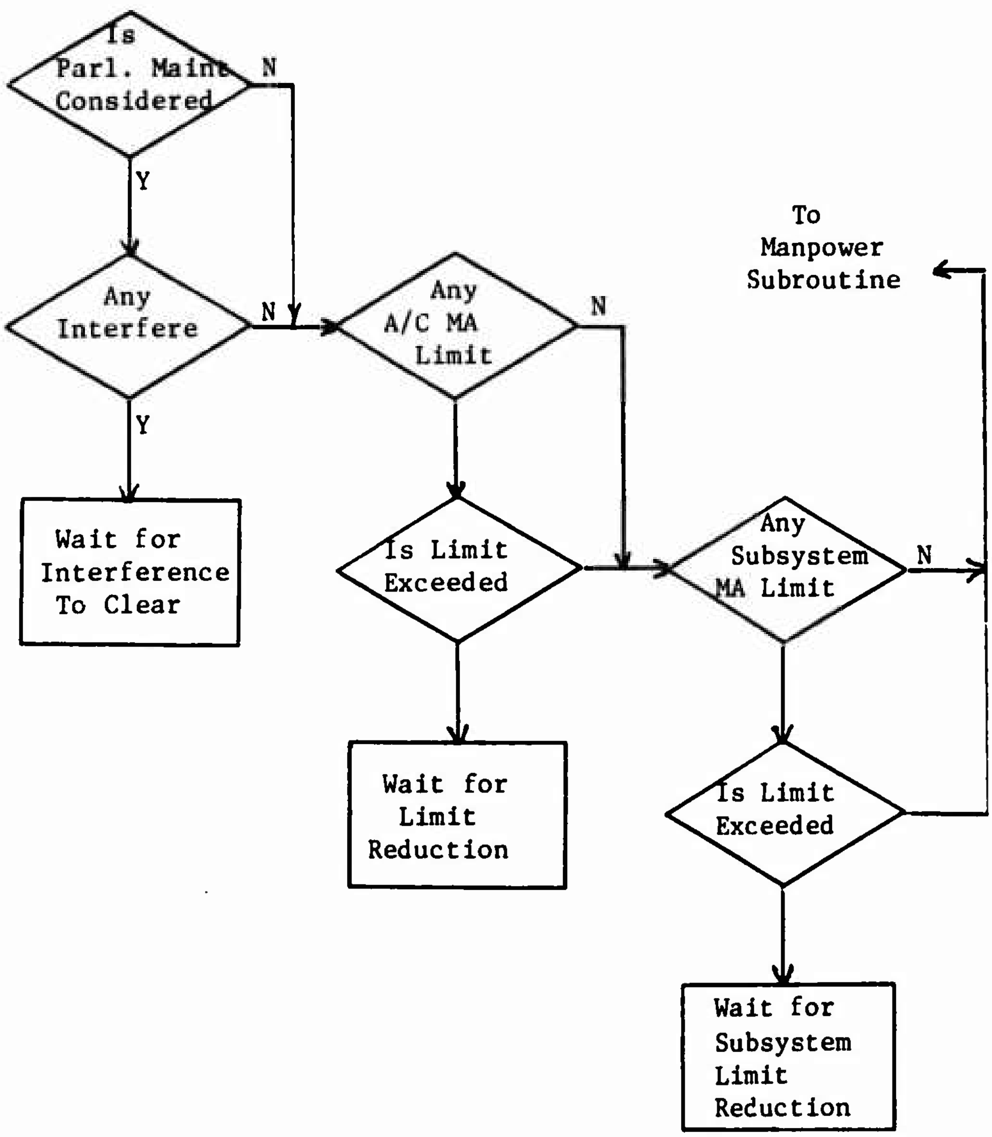

Figure 37. Parallel Maintenance Subroutine.

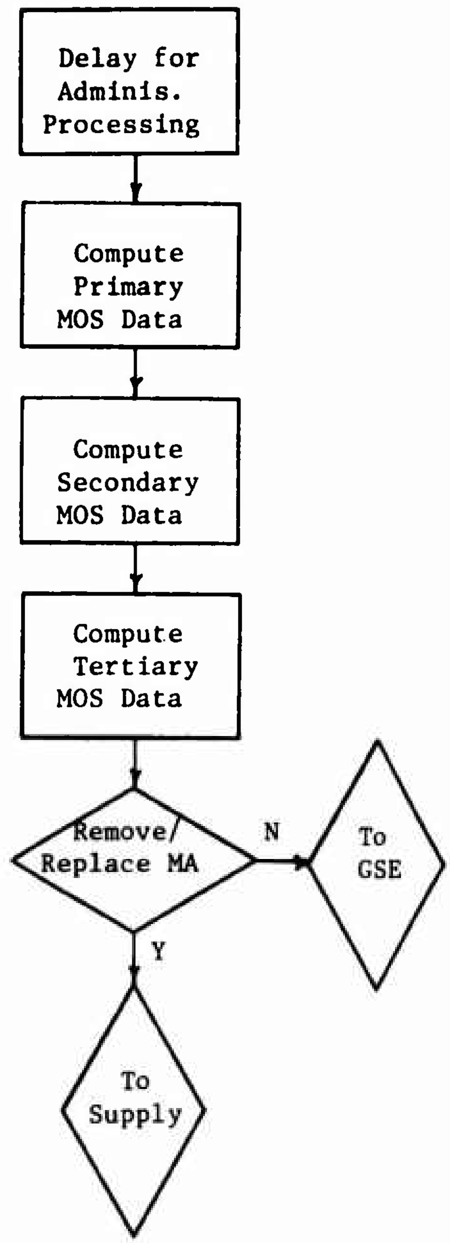

Figure 38. MOS Assessment Subroutine.

Note that in the above sequence, if a queueing situation is encountered at shift change, the GSE will be returned and the entire acquisition process repeated. Also, hookup time is computed as the maximum individual hookup time for any GSE in the package specified.

Card types "T", "G" and "H" supply input data for this subroutine. A flow diagram of the logic process is presented in Figure 39.

GSE Return/Repair Subroutine

This subroutine will accomplish the function of replacing GSE in stock after use and to test for GSE failure. A time delay for GSE repair is computed when failure occurs.

After each GSE package completes its percentage of the MTTR on a maintenance action or ground event, it enters this subroutine. The package contents is examined to determine the longest return time specified, and an appropriate delay is introduced.

Each piece of GSE is then tested against the probability of failure. If no failure results, the equipment is made available for the next usage. When a failure is indicated, the repair time is computed using the specified mean, variance and distribution function. After the computed delay, the GSE is made available for the next usage.

Card type "T" provides data to this subroutine. A flow diagram of the routine is presented in Figure 40.

Off-Equipment Repair Subroutine

The purpose of this logic is to establish the maintenance level, manpower and GSE assets required to accomplish off-equipment repair. The subroutine is entered whenever a remove and replace maintenance action occurs on the aircraft. An initial test is made to determine if any off-equipment repair data has been provided for the entering part. If none is found, the maintenance action is recorded as a scrappage. When data are available, the routine will determine at which echelon the maintenance is to occur and whether the action will be a scrap or repair.

If all specified echelons have been checked and neither a scrap nor repair outcome is obtained, the maintenance

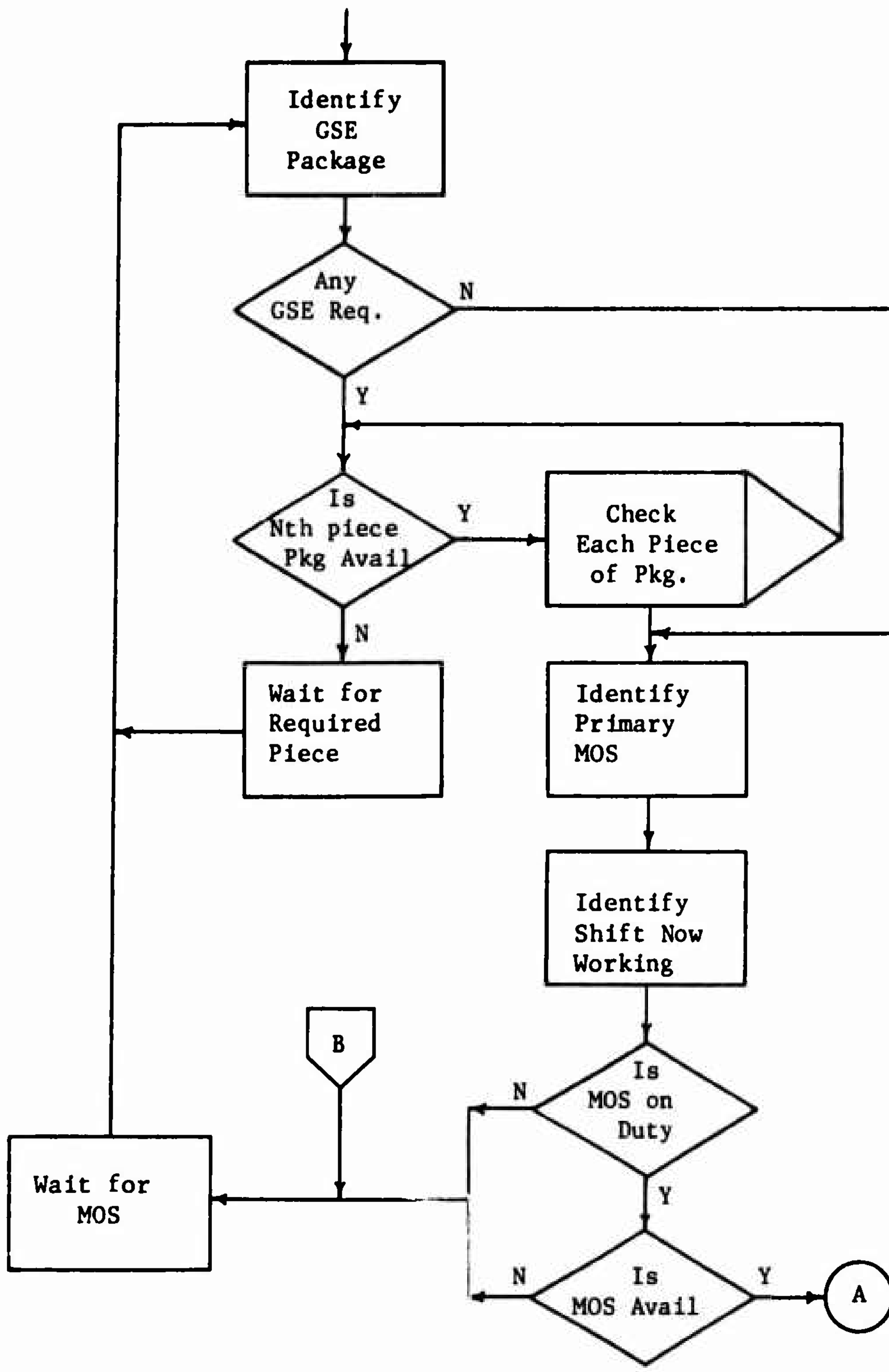

Figure 39. GSE Acquisition Subroutine .

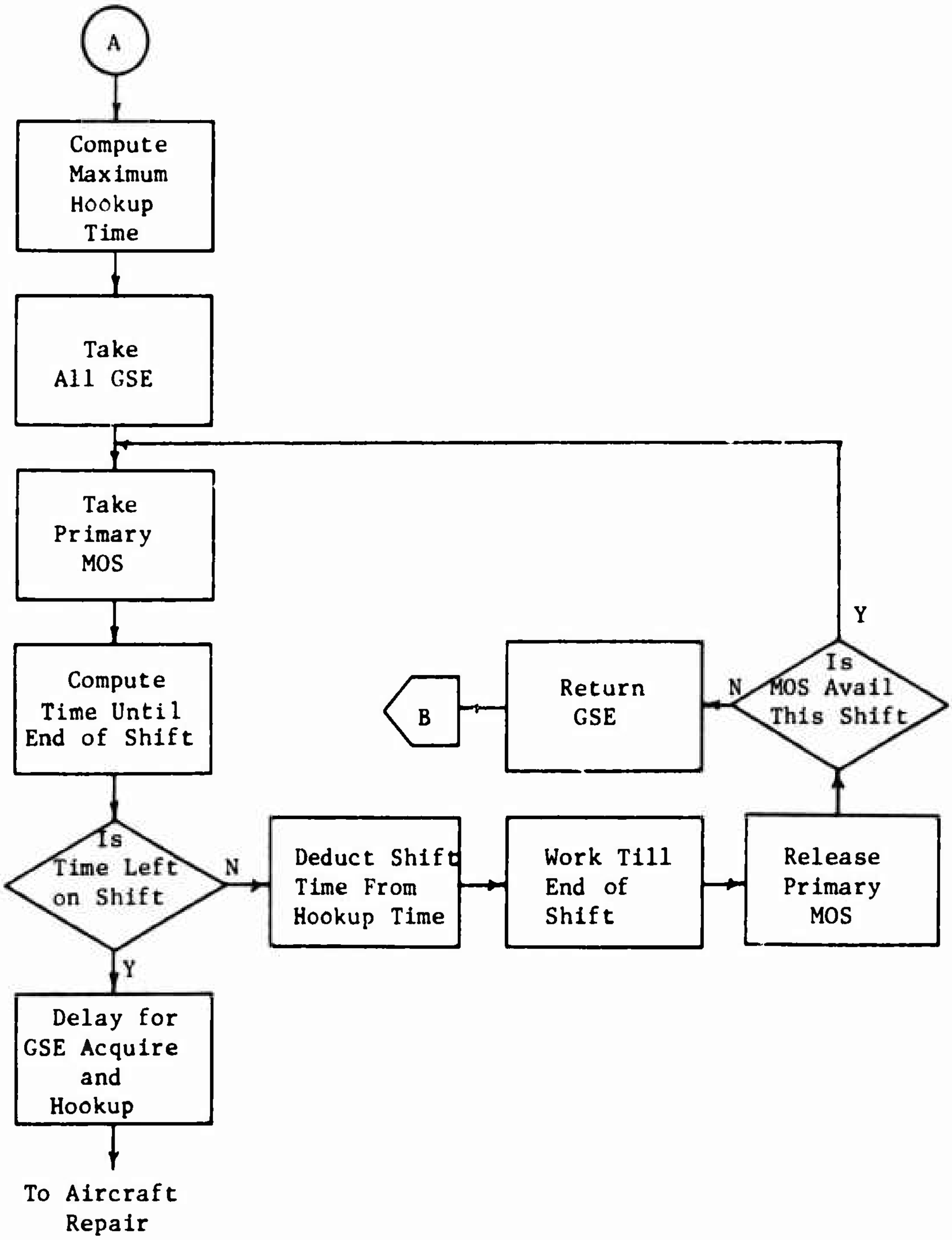

Figure 39. Continued

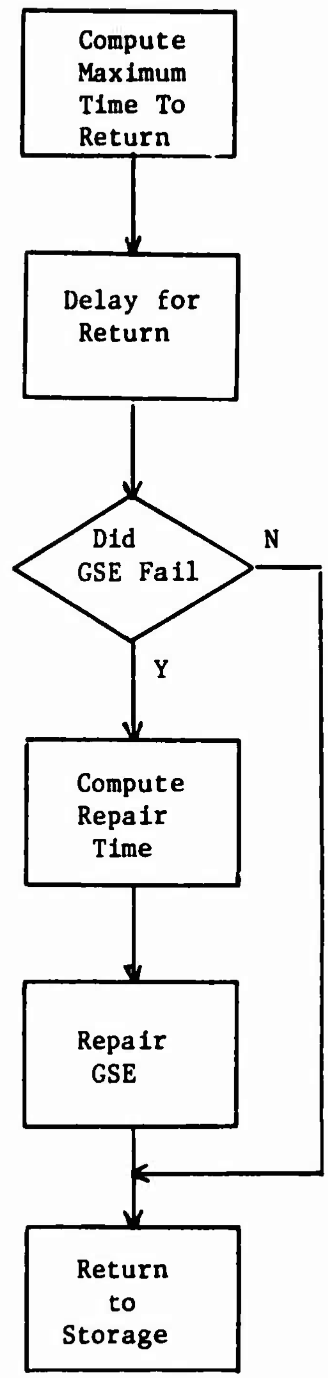

Figure 40. GSE Return/Repair Subroutine.

action is recorded as a NRTS (not repairable this station). Scrappage outcomes are recorded for each echelon.

If a repair action is indicated, the transaction undergoes an input specified administrative delay. Manpower, GSE and repair time are assigned. The GSE Acquisition Subroutine is then entered and the repair work accomplished.

The off-equipment repair subroutine is reentered when maintenance has been completed. Tests are made for incorrect repair and repeat repairs at the same echelon. The maintenance action will be recorded as an accomplished repair, be re-repaired at the same echelon, or transferred to the next higher maintenance level.

Figure 41 is a flow diagram of the off-equipment repair subroutine.

Aircraft Maintenance Accounting Routine

The purpose of this routine is to examine the status of aircraft as each maintenance action is completed, in order to determine if a change in status is required. Maintenance actions enter this routine from aircraft repair. If the maintenance action causes an SMA, the required SMA number is recorded. Test hop requirements are also recorded.

Maintenance actions are divided into upsquawks and downsquawks and the aircraft maintenance status board is adjusted. If additional downsquawk maintenance exists, no change is made in the aircraft status. If the action being processed is the last downsquawk, SMA requirements are tested. When open SMA's exist and no other maintenance is in process, the aircraft will be inducted to accomplish the required SMA. If other maintenance is in process, the aircraft status will not change and induction into SMA logic is delayed.

When no open SMA's are found, the aircraft status board is interrogated for NORS maintenance actions. If open NORS is found, the aircraft status is changed to NORS. In the absence of NORS MA's, a check is made for any other in-work maintenance. Without other active MA's, the aircraft will either return to the ready pool in an up status or transfer to test hop logic, depending on previously recorded test hop requirements.

A completed upsquawk is processed somewhat differently. The first test determines if the number of remaining open upsquawks is less than the input specified limit. If the

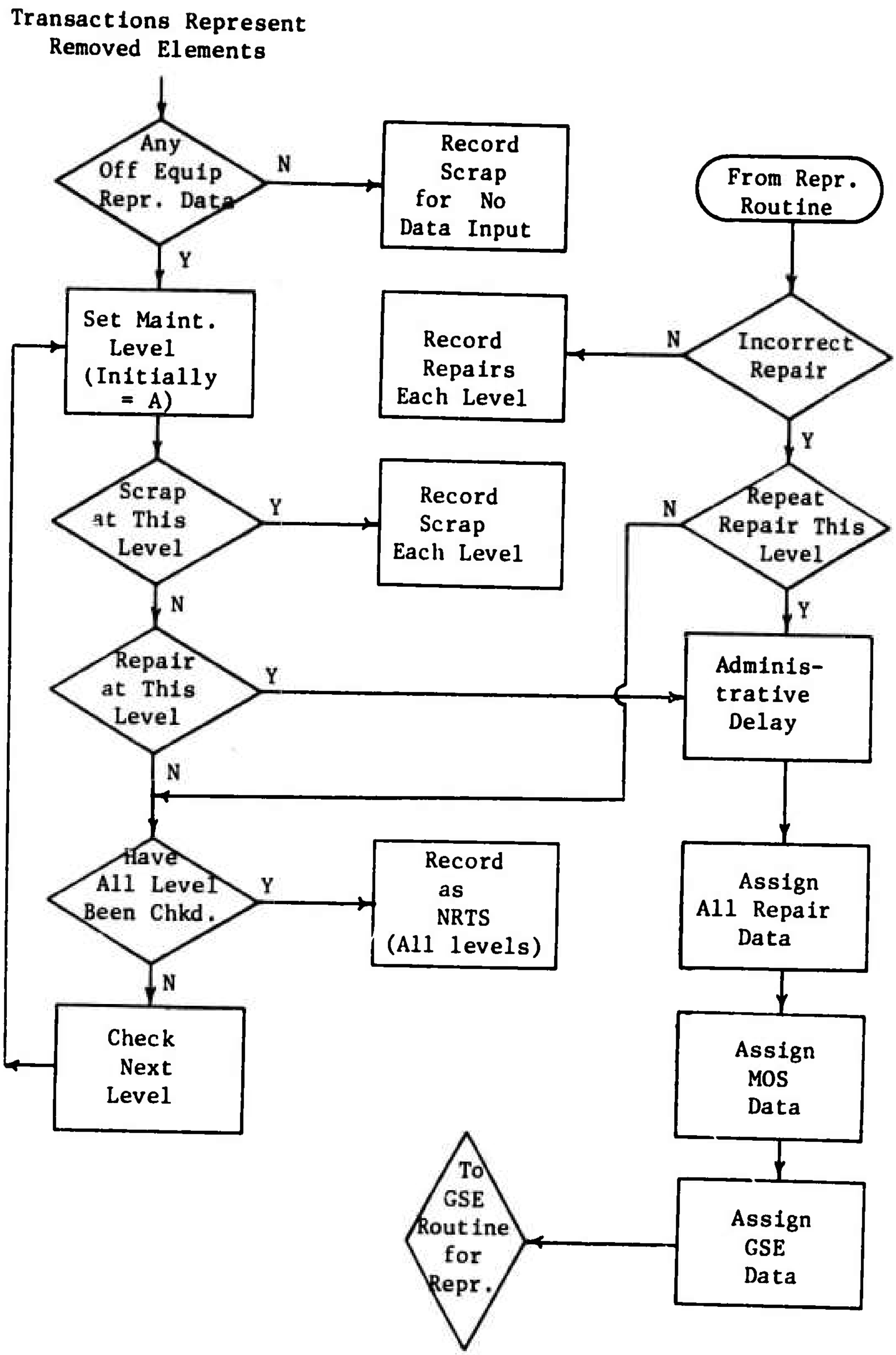

Figure 41. Off-Equipment Repair Subroutine.

limit is still exceeded, no change in aircraft status will occur. When fewer than maximum upsquawks are found, a test is made to determine if open downsquawk maintenance exists. If this is true, no change in aircraft status is made. Without open downsquawks the aircraft status board is interrogated for any other in-process maintenance and open SMA's. If other maintenance or open SMA's exist, no change in aircraft status is made. Other in-process maintenance without SMA's indicates that all downing maintenance has been cleared and the aircraft status is changed to up. The aircraft, however, remains in maintenance until the remainder of the upsquawks have been cleared.

If no in-process maintenance was found, the transaction will look for open SMA or test hop requirements. The aircraft will either be inducted for SMA, fly a test hop, or be placed in an up status and transferred to the ready chain.

Figure 42 is a flow diagram of the aircraft maintenance accounting routine.

Maintenance Action Logging Subroutine

The purpose of this subroutine is to accept each maintenance action as it is detected and provide preliminary data relative to the required maintenance. Upsquawks and downsquawks enter at separate locations and are marked accordingly. Each maintenance action is recorded in the aircraft maintenance status matrix. The maintenance action is also identified as a removal or repair in place, and the repair time is computed.

Upsquawks and downsquawks are again separated and placed on separate chains until they are activated by the Maintenance Control Subroutine.

Figure 43 is a flow diagram of the Maintenance Action Logging Subroutine.

Maintenance Control Subroutine

Aircraft enter this logic when various other model logics determine that unscheduled maintenance is required. All open downsquawks are immediately activated so that the required maintenance can be accomplished. The longest estimated repair time for this group of maintenance actions is determined and recorded in the aircraft maintenance status matrix. Aircraft status is changed to NORMU.

If open upsquawks exist, a test is made to determine

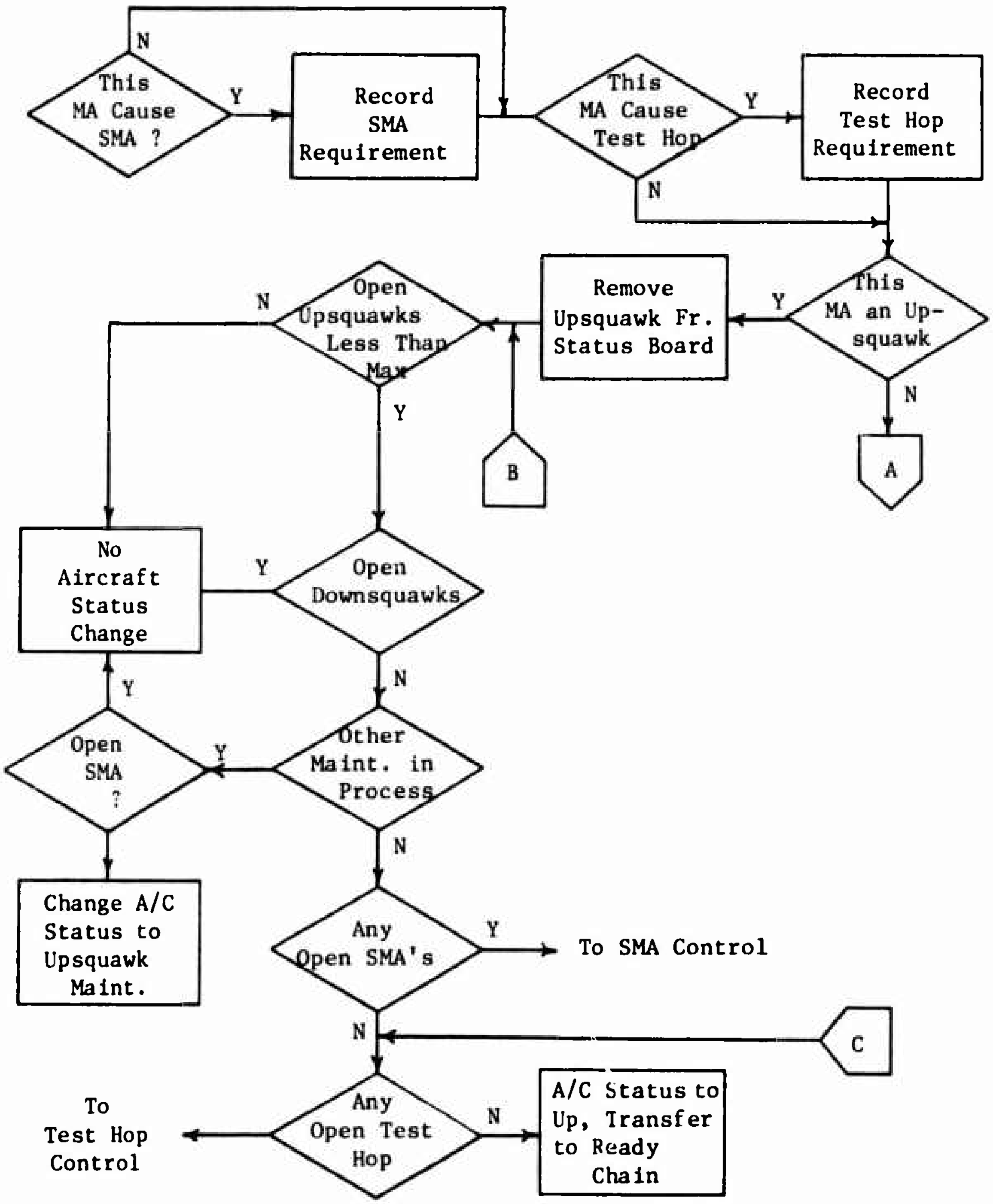

Figure 42. Aircraft Maintenance Accounting Routine.

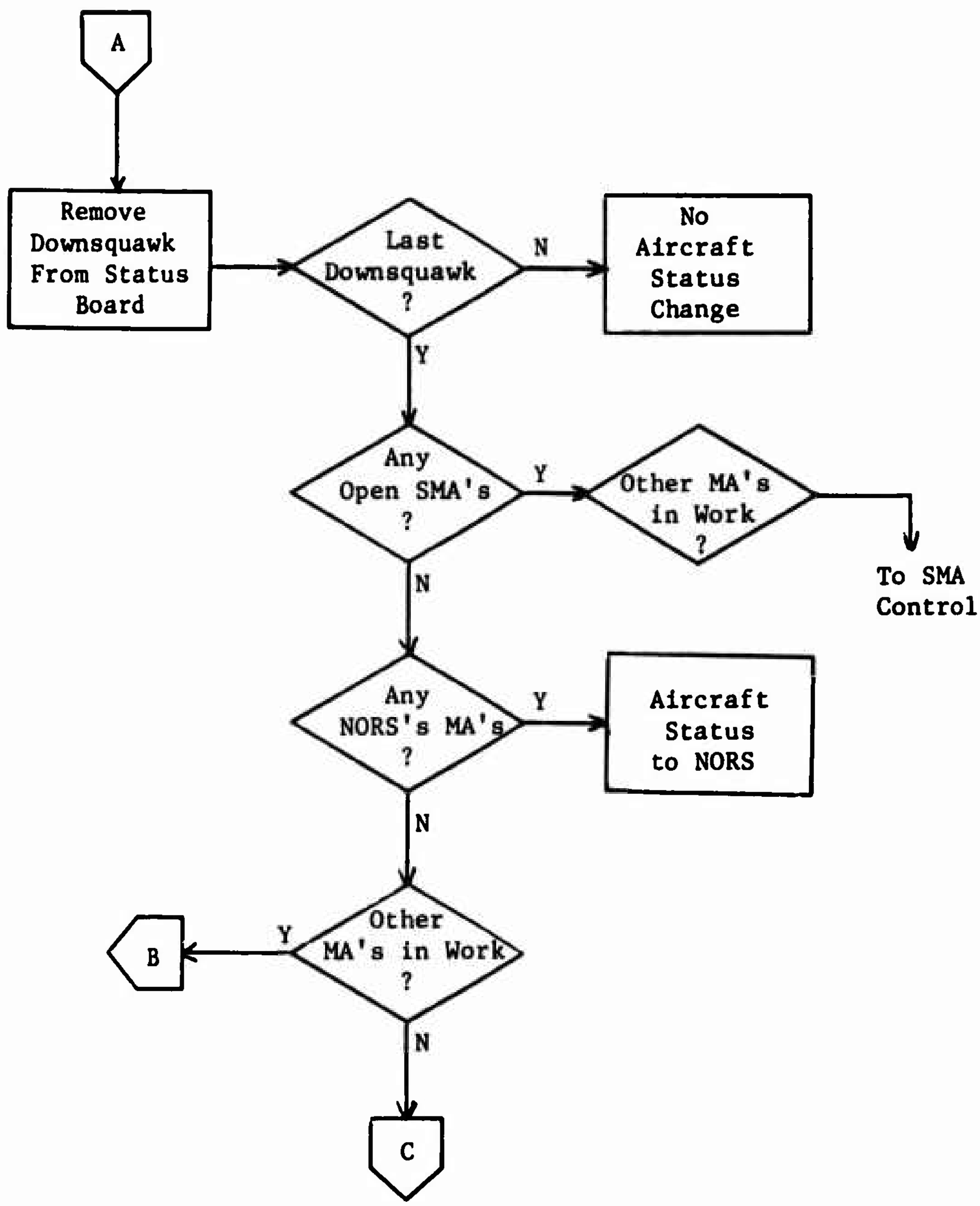

Figure 42 Continued

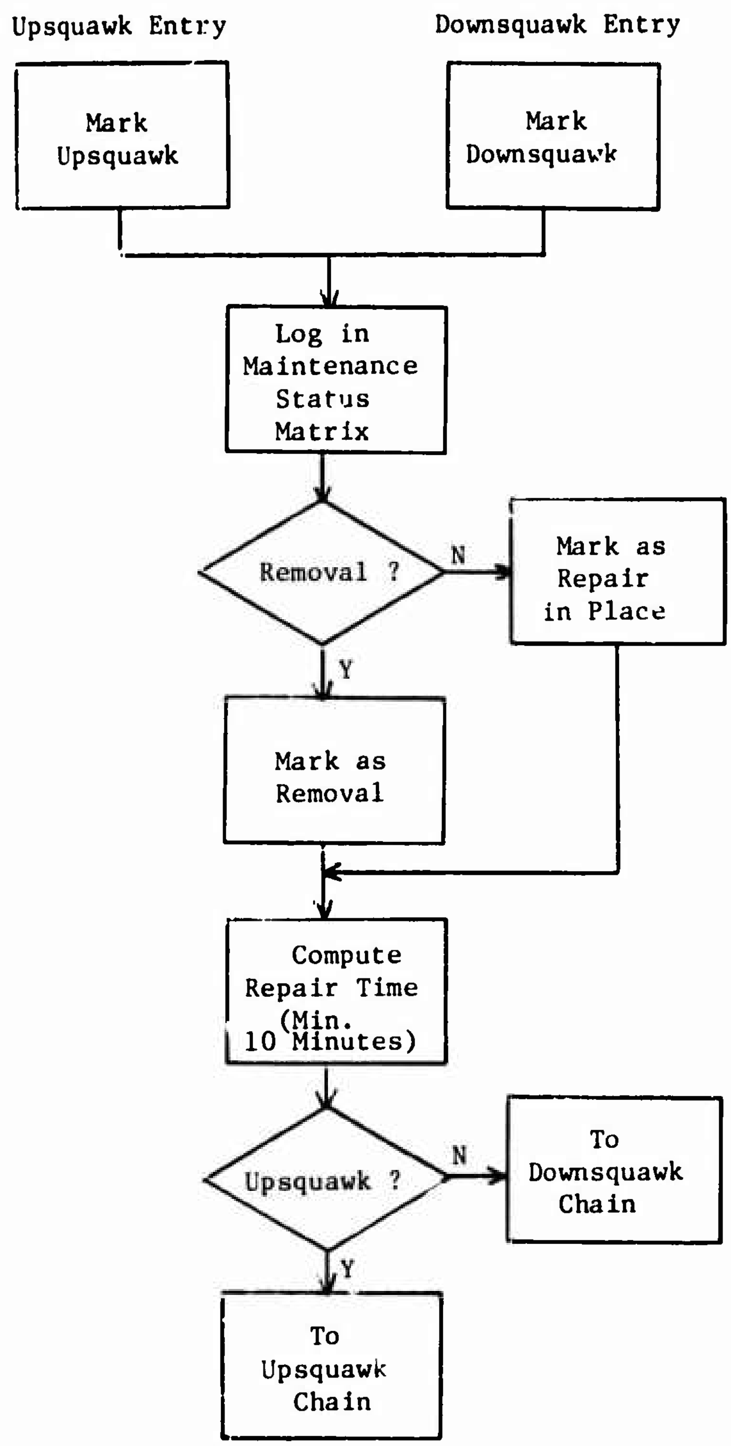

Figure 43. Maintenance Action Logging Subroutine.

if the input specified limit has been exceeded. When this is true, input data will be used to compute what percentage of these upsquawks are to be activated.

If an upsquawk limit has not been provided, a test is made to establish if a limit has been placed on the number of in-process maintenance actions. When no limit has been specified, three upsquawks whose estimated repair time is less than the maximum downsquawk repair time are activated for repair.

If a limit for in-process maintenance has been specified, sufficient upsquawks whose estimated repair time do not exceed the downsquawk repair time will be activated to bring the total in-process maintenance up to the limit specified. The aircraft then enters the Unscheduled Maintenance Routine.

Figure 44 is a flow diagram of the Maintenance Control Subroutine.

SMA Control Subroutine

This logic serves two purposes. It records pending SMA's and activates these when other routines determine it is timely. Pending SMA's are linked on a chain representing an open SMA file. They remain here until activated by a transaction indicating a requirement for inducting the aircraft into the SMA event.

As each SMA is activated, it transfers its identifying number to the proper aircraft. It then causes the aircraft to be transferred to the Ground Event Routine, where the SMA will be accomplished. The open SMA count in the aircraft status matrix is decremented and the transaction terminates.

Figure 45 is a flow diagram of the SMA Control Subroutine.

Test Hop Control Subroutine

This logic is entered when it has been determined that a test hop must be flown. A count is maintained of successive test hops so that failure rates may be adjusted in accordance with input data. The test hop mission number is assigned to the aircraft, and the maintenance status matrix is adjusted to reduce the test hop requirement to zero.

Input data is interrogated to determine if test hops

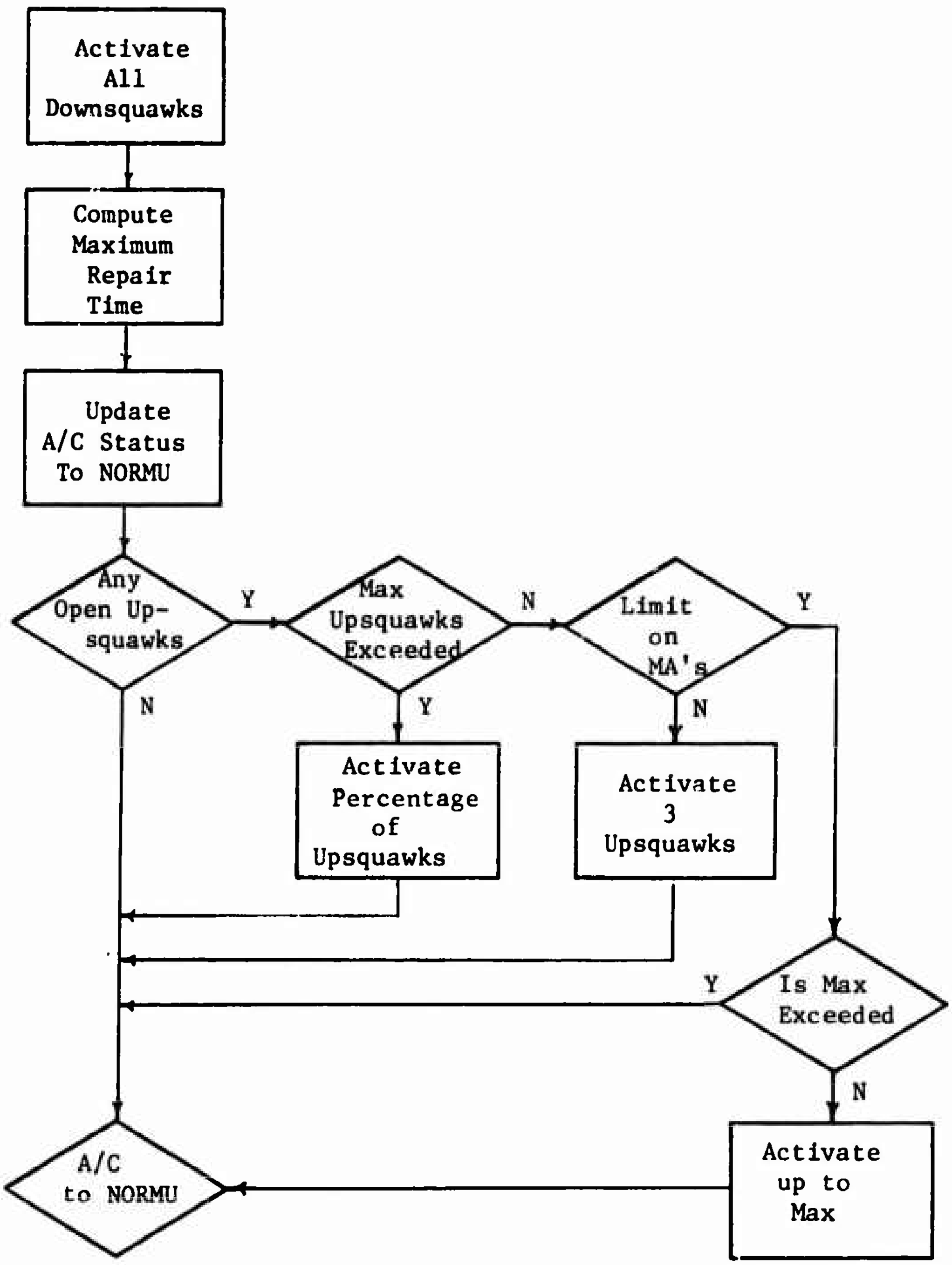

Figure 44. Maintenance Control Subroutine.

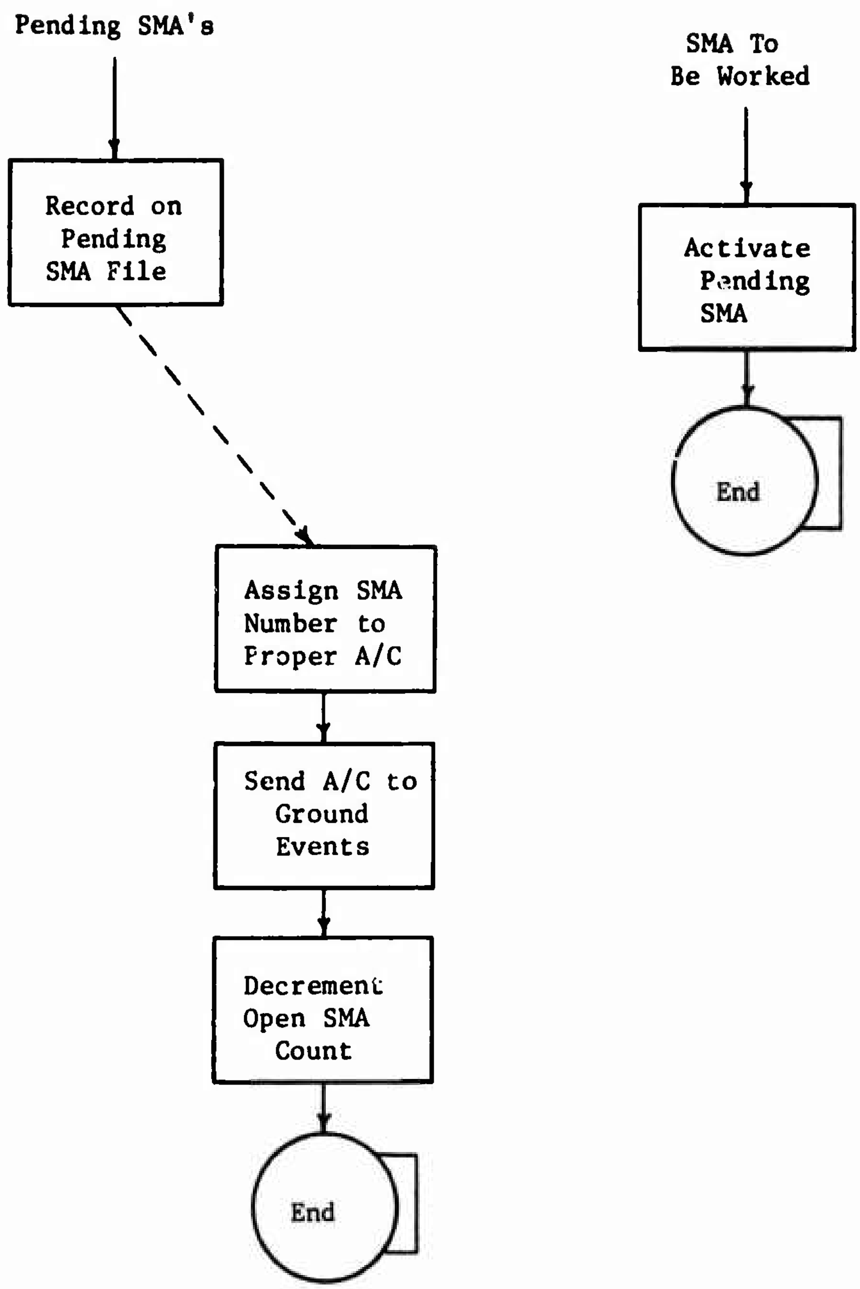

Figure 45. SMA Control Subroutine.

are permitted at the present time of day. The aircraft will either be immediately inducted or be delayed until the earliest test hop time specified by input.

Figure 46 is a flow diagram of the Test Hop Control Subroutine.

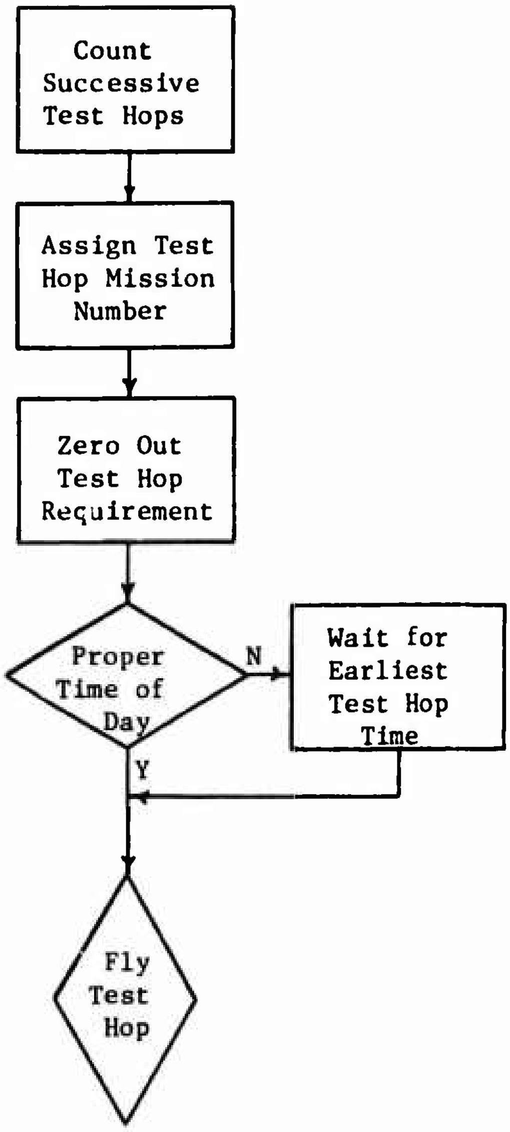

Figure 46. Test Hop Control Subroutine.

SSDDIP

"SSDDIP", an acronym for Simulation Scenario Description and Data Input Program, edits the input data supplied. It appraises the analyst of errors prior to converting the data into usable formats for the ARMS simulation program. Data input is obtained from the complete package of loading forms prepared by the analyst.

GENERAL OPERATION

There are three phases to SSDDIP:

A. Card Edit
B. Cross Edit
C. Arithmetic Program.

The Card Edit phase examines the data provided on each of the 34 different formats. In this phase, the data on each form is independently examined for applicability, errors, omissions, and consistency with other data on the same form. If errors are noted, a printout is made of the data being examined and in which area the error is detected. Also, an abbreviated message is given describing the type of error noted. The data is then appropriately stored and recorded for later reference. Should duplicate data be found, it will be detected and both sets of data will be printed out for the analyst to review. A more detailed description of the items checked is presented later. It is appropriate to note that the data input to Card Edit, as well as the data which it creates, must be retained for use in subsequent phases. These data are stored on a uniquely named data set for use by all subsequent phases. Output from subsequent phases is entered into the same data set. Therefore, the possibility of mixing data from different simulation experiments is eliminated.

A message is printed saying that the job has successfully passed the Card Edit phase when no errors are detected.

The Cross Edit phase recalls the data stored during the previous phase and investigates its compatibility and sufficiency. For example, if a particular MOS is specified for a repair, this phase checks to insure that the MOS is scheduled to work at some time during the week. Also, checks are made to insure that a sufficient quantity will be working at some time to meet the maximum required by any single task. The absence of required supporting data is also noted.

At the conclusion of the checks for the properly interrelated data, a separate routine reexamines all data input to ascertain if any superfluous data existed. Warning messages are given to advise the analyst of the existence of unused data so that he may judge its validity. Thus, two types of "errors" are detected in this phase: the first is of a fatal nature requiring corrective action before the job can be processed further, and the second is a warning which may or may not require corrective action. As with Card Edit, a message is printed out at the end of a successful Cross Edit phase, permitting the job to proceed.

The third or last phase of SSDDIP converts the data stored from the previous two phases into the proper format and sequence for use in the ARMS simulation model. Also accomplished within this third arithmetic phase is the identification of the sizes associated with many of the variables; e.g., the maximum quantity of aircraft involved, the total number of WUC's - elements, building blocks, etc.

Each of the formatted data cards is assigned a sequence number to automatically insure its proper location in the ARMS model program. Formatted data can alternatively be stored on magnetic tape or other data storage devices for later merging into the ARMS model. The option is also provided the analyst to obtain a printed listing of these data. The storage of these data on tape is intended to be on the same reel used to store the data from the preceding two phases. Thus, all data associated with the initial input data is stored together. As will be seen in the descriptions for other programs in this model, their outputs will also be stored on this same file, providing one package of related data.

The sections which follow will describe in greater detail that which is accomplished in the three phases of SSDDIP.

PROGRAM NARRATIVE

This section will provide descriptions of what is accomplished within Card Edit, Cross Edit and the Arithmetic phases of the SSDDIP program.

The Card Edit phase, Figure 47, reads each input card. Data is checked for completeness and to see that allowable ranges are not exceeded. Duplicate data are detected. Errors apprising the user of discrepancies are printed as an aid in correcting inputs. All decoded data is stored on magnetic tape for future use.

The Cross Edit phase, Figure 48, tests for the completeness of data among the various card types. For example, if a particular skill is specified for a repair, the Cross Edit

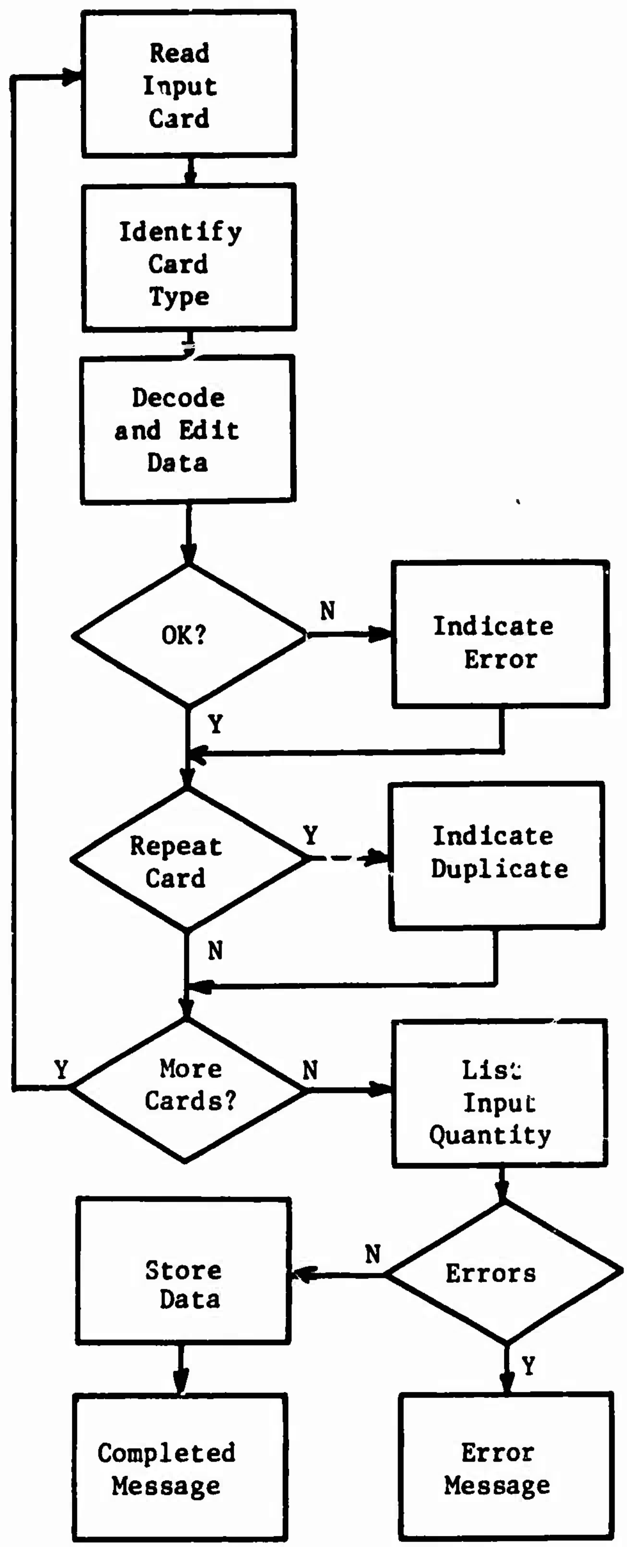

Figure 47. Logic Diagram, Card Edit Program.

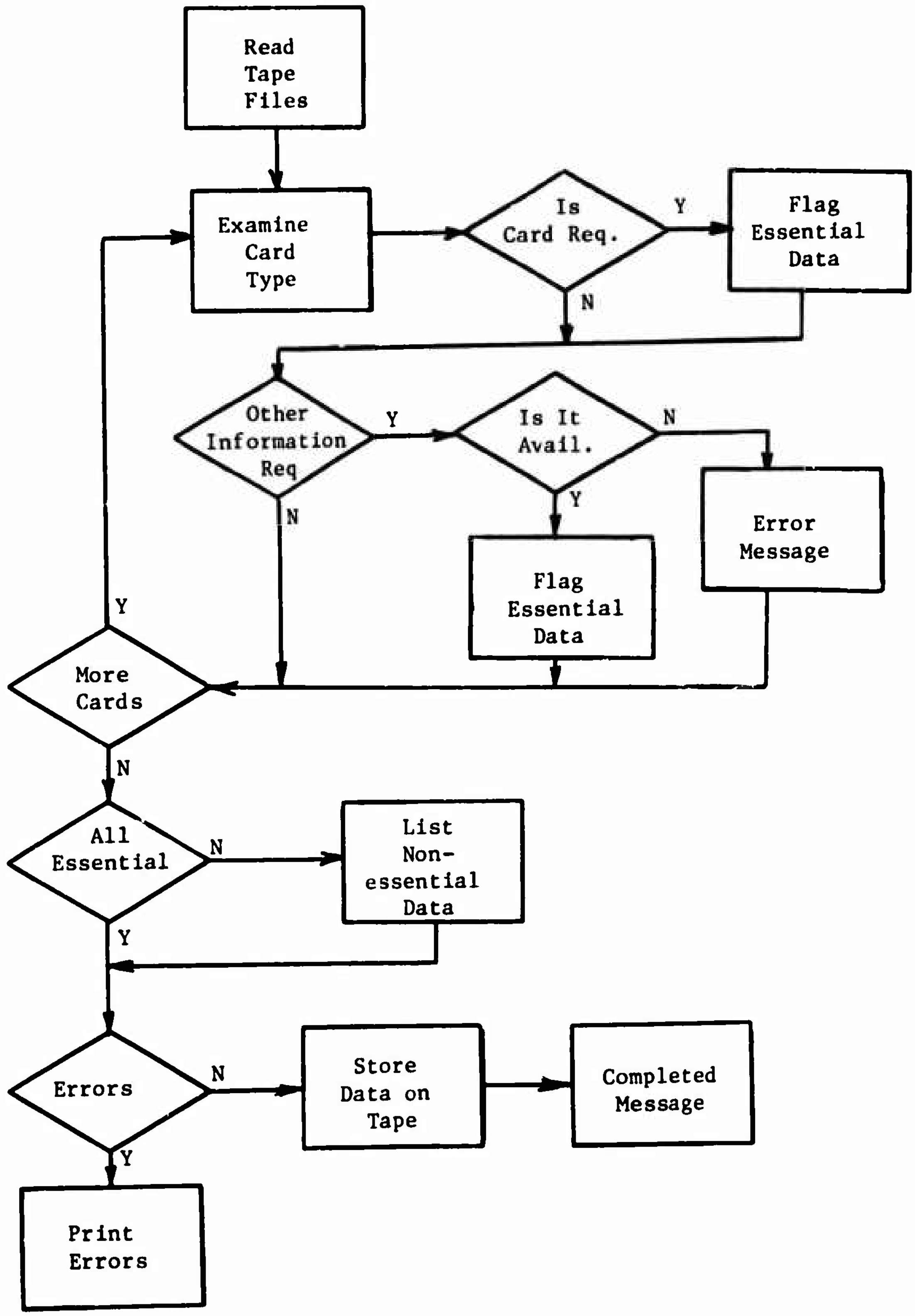

Figure 48. Logic Diagram, Cross Edit Program.

program checks that he is available and is scheduled to work at some point in the model.

Cross Edit will also detect and flag superfluous data. An MOS assigned and working a shift who is never required to perform a task is flagged so the analyst may double-check his input.

The Arithmetic phase computes and formats all edited data. It creates the matrix definition cards for GPSS based on the specific scenario simulated. This feature allows the computer core required to adapt to the size needed by a specific experiment. All data are sorted, sequenced and formatted so that they may be inserted in space reserved by the ARMS program. A flow diagram of arithmetic phase logic is provided in Figure 49.

Detailed narratives of each SSDDIP phase are provided in the following paragraphs.

Card Edit

SSDDIP is programmed utilizing FORTRAN language. The selected approach used the individual character/symbol reading technique (A-mode for those familiar with FORTRAN) wherein each symbol, including blanks, has a unique numerical equivalence. In the reading of the input data, these symbols/characters must be decoded or packed into groups having numeric significance. If an alpha character appears in a zone where numeric characters should have appeared, an obvious error exists. Separate subroutines have been created to accomplish this and are referred to in the decoding of each of the 34 different loading formats.

To provide for storage of the input information, two different data storage sets are used. The first is nothing more than the sequential storage of the data included on each of the many loading forms, regardless of the order in which they appear. The second is sized to accommodate a maximum of 1000 uniquely identified elements and is used for the systematic storage of the sequence number from the first set that relates it to a specific item. Other data that is essential to the methodology used is also retained in other areas of this same second storage set.

To read and interpret each group of input data representing one of many formats, a subroutine identified as CARDID, "Card Identification", is used. This routine identifies whether the character in the 77th position of the loading forms is other than alphanumeric. The alphanumeric character identifies which of the 34 different formats is

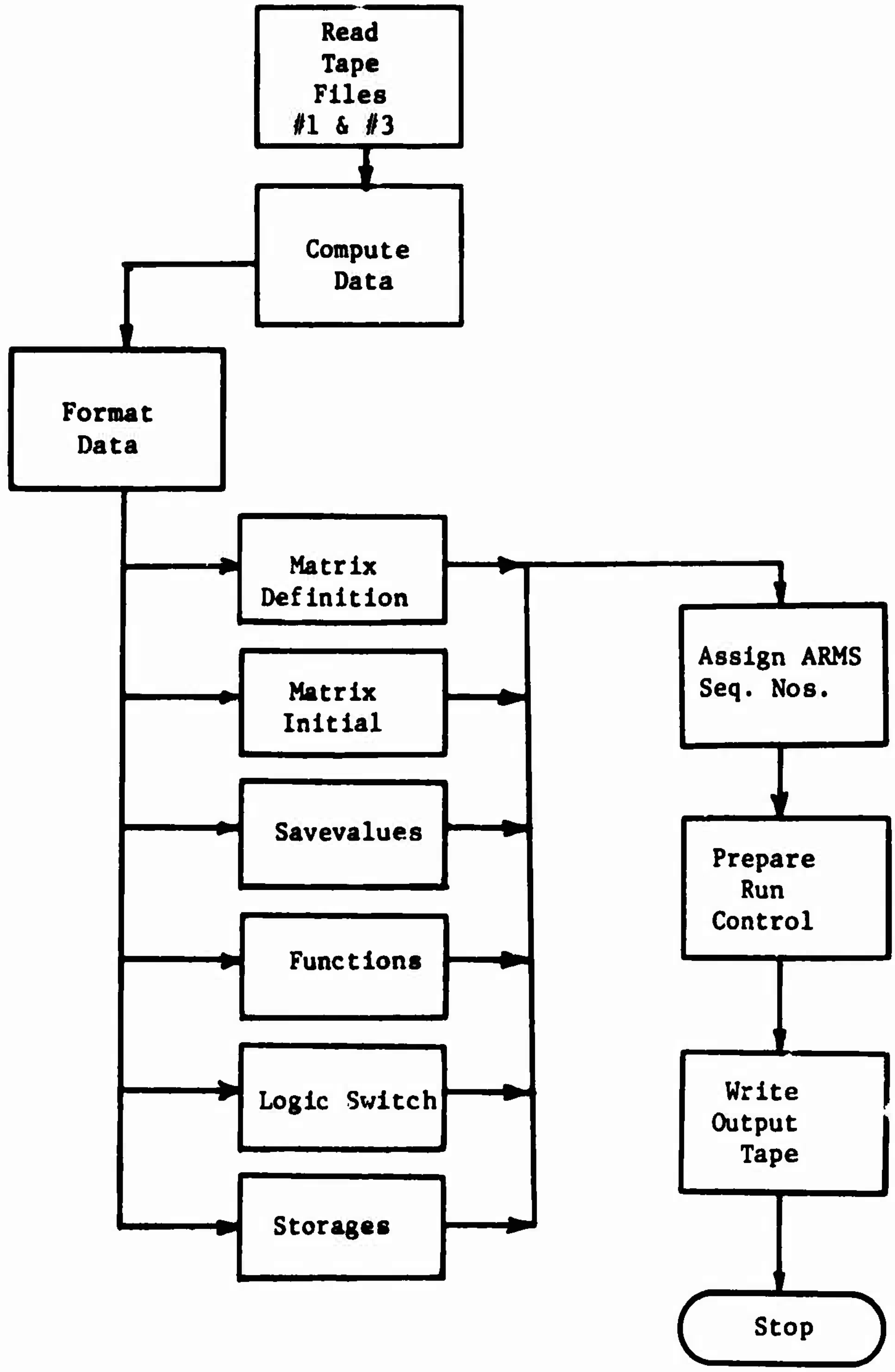

Figure 49. Logic Diagram, Arithmetic Program.

involved. At this point a subroutine identified as FILEF is entered to establish this data group's sequence number and to locate a direct access file for its storage. The next step is the calling of one of the applicable subroutines, as individually discussed below, to decode the particular loading format. Following the completion of the decoding, the data input is stored in the sequential direct-access file as well as the storage set before proceeding to the next group. When all data groups have been examined, a listing of the quantity of each type is given.

The individual cards are decoded and tested as indicated in the following. All numeric entries must be right-handed, adjusted with appropriate trailing zeroes if applicable. Also, column 77 on each card type must have the appropriate alphanumeric identity entered.

To aid in reviewing the following, a complete set of the 34 different loading forms is included in Volume II.

<u>Card "A"</u>

The decoding of the "A" card is done using subroutine SIMCON. The data is decoded and tested as follows:

Column 1.

Only a "1", "0" or blank permitted; "1" permitted only when columns 2 through 5 are blank.

Columns 2-5.

Any contiguous numeric value that is an integer multiple of 24, or a blank. See also column 1. above.

Columns 6-9.

Requires a value that is an integer multiple of 24. This value must also be equal to or greater than the product of the sum of the values in columns 10-13 and 14-17 and the sum of column 19 plus 1.

Columns 10-13, 14-17.

Optional entry; if used, both values must be greater than zero and integer multiples of 24.

Column 18.

Optional. Permissible entries, in addition to a blank, are a "0" or a "1". If "1", then columns 10-17 must have entries greater than zero.

Column 19.

If a "1" appears in column 18, this column must have an integer value greater than zero; otherwise, it must be either a blank or a zero.

Columns 20-22.

Optional; if used, it must be a multiple of 24 and also wholely divisible into the value in columns 6-9.

Columns 23 and 24.

One and only one of these must have a "1". The other may either have a "0" or be left blank.

Columns 25-76.

This is the simulation run name and its contents are optional.

Card "B"

The AIRCRAft General data card is decoded and analyzed using subroutine AIRCRA as follows:

Columns 1-2.

Any value greater than zero must be found in this area.

Columns 3-4, 5-6, 7-8 and 9-10.

The analyst may choose any combination of one to four values for insertion into these four groups; their total must equal the value given in columns 1-2. When 1-2 is 99, these areas must be left blank.

Columns 11-12, 13-14, 15-16 and 17-18.

If columns 1-2 value is not equal to 99, and a nonzero value appears in any of four groups in columns 3-10, a nonzero value must also appear in the corresponding configuration group in this area. If columns 1-2 value is 99, then values must appear for those configuration groups within this area when the applicable configuration is specified as being required on any of the "U" cards.

Columns 19-20, 21-23 and 24-25.

The delay time to obtain a float aircraft contains these three groups of data. They are decoded by means of a separate subroutine called REPTYM (repair time). This routine is also used in the deciphering of other cards. It decodes the distribution number, mean time and variance values as specified in this area. If a distribution function number is indicated (columns 19-20 in this example), it may be any number between 1 and 30 inclusive, except for 9 and 10, which are not valid at this time. Certain other tests insure that when function numbers of 7 or less are specified, a mean value and variance each greater than zero must also be indicated; that for a function number 8, only a mean value is indicated; and for functions 11-30, a mean of "0" should be specified and the variance either a "0" or blank. An acceptable alternate to these is a mean time, with or without a variance, and without a function number. For this card "B", if all three areas are blank, this will imply no attrition replacement.

Columns 26-28.

A value greater than zero must appear.

Columns 29-31.

Any positive value is acceptable; however, a blank is also acceptable and implies the same value as in columns 26-28.

Columns 32-33, 34-35, 36-37, 54-55 and 56-57.

Each of these groups of two is to be a value which is understood to be a percent, where 99 implies 100%. These, and similar areas on other cards, are decoded (and converted) by use of subroutine PERCNT. A blank is decoded as zero.

Columns 38-41, 42-45, 46-49 and 50-53.

The four groups of "time of day" data appearing in columns 38-53 are each decoded using subroutine called TIMEDA. That subroutine insures that if an entry exists, it is neither 0000 nor 2400. This avoids the possible confusion as to whether it is intended to be the midnight at the beginning or end of a particular day. Editing insures that no entry outside the limits of of 0001 and 2359 exist, nor that the last two digits are greater than 59. While entries must be found for the first two values, the absence of any entry for the third or fourth is interpreted as the same value as the first or second, respectively. A further test is made to insure that the second and fourth values are greater (later in the day) than the first and third, respectively.

Columns 58, 59-62, 63-65, 66-67, 68-70, 71-73, 74 and 75.

Columns 58, 74 and 75 require a yes-no answer, where a "1" implies a yes and a zero or blank is a no. Any other character results in an error message. These are decoded using subroutine YESNO or YESNOS. Columns 59-73 contain five groups of numerical data that are decoded using the basic DKODE subroutine. Data in this area together with that in columns 58, 74 and 75 are subjected to several interrelated tests. First, if there is any data in columns 63-65, then there must also be data in columns 59-62. These data must be an integer multiple of the former. The same is true for data in columns 66-67 and 63-65. If an entry appears in columns 71-73, a test is made to insure that it is not greater than columns 68-70. Another test insures the absence of any other entries in columns 59-75, if a value appears in column 58, and vice versa, with an additional test that if column 58 is not entered there must be a yes answer to either columns 74 or 75, but not both.

Card "C"

The aircraft reconfiguration card is decoded and tested by use of subroutine RECONF and other supporting routines. It is to be noted that there are four zones of configuration data on this card, each of which is decoded in the same manner. Accordingly, the column references below will simultaneously refer to all four groupings.

Columns 1-7, 20-26, 39-45 and 58-64.

The first two columns in each of these zones may contain a function number, the next three columns a mean time, and the last two a variance, all of which are decoded using subroutine REPTYM. Card "B", columns 19-25 explanation is applicable.

Columns 8-9, 27-28, 46-47 and 65-66.

If a number appeared in the corresponding preceding group for either the function number or the mean time, then a value must appear in this two-digit zone.

Columns 10-13, 29-32, 48-51 and 67-70.

Each of these four groups contains the GSE package number and percentage of mean remove time, two columns each. These are decoded and tested by means of subroutine GSEPKG. That subroutine decodes the first two spaces for the package number and the next two for the percentage value via the PERCNT subroutine (wherein 99 becomes 100).

It also tests to assure that the package number indicated is not greater than 25 and that if a package number is specified, the percentage was greater than zero.

Columns 14-19, 33-38, 52-57 and 71-76.

The SMA numbers indicated in each of the three groups of two spaces within the above areas are decoded and tested to assure that if any values (SMA numbers) appear, they are within the limits of 10 to 29, inclusive.

Card "D"

The decoding and testing of the data on these cards use subroutine BASELE. There are 9801 acceptablee element number combinations. Many "D" cards can exist. The program can accommodate up to 1000 such elements. Within this limit, any combination of element numbers can be used.

Columns 1-2.

An entry in these columns is necessary, and may be any value from 0 (or 00) to 98, inclusive.

Columns 3-4.

Optional or any value 0 through 99. A blank is interpreted as zero.

Columns 5-13.

Optional - any alphanumeric data.

Columns 14-18.

Mandatory entry with any desired value greater than zero. However, if column 1-2 is "00", this value must be between 1 and 100, inclusive. Also, the sum of all "00" subsystem "D" cards must equal 100.

Column 19.

If columns 1-2 equal "00", then this must be zero or blank. If columns 1-2 do not equal "00", then this must be any integer value 0 to 9.

Columns 20-23, 24-27.

If columns 1-2 equal "00", then these should either be 100 or blank (which assumes 100). If columns 1-2 do not equal "00", then these must either have a nonzero value or be blank (which assumes 100).

Column 28.

If columns 1-2 equal "00", this must be a "1". If columns 1-2 do not equal "00", this may be either "1", "0", or blank.

Column 29.

Optional "0", "1" or blank; may not be a "1" if column 28 is.

Columns 30-31.

May be any value 0 to 99 (99 = 100%) or blank (equals 0%); however, when either columns 28 or 29 contains a "1", then this must be 1 to 99.

Columns 32-37.

These six columns may each be a "1", "0" or blank with the following restrictions:

1. If columns 1-2 equal "00", columns 32-36 must be zero or blank and column 37 must have a "1".
2. If columns 1-2 do not equal "0", column 37 must be zero or blank.
3. If column 32 is a "1", then columns 33-36 must all be "0" or blank.

Columns 38-76.

No restrictions.

<u>Card "E"</u>

Card "E" is decoded and tested using subroutine DISCOV. The following tests on data requirements are made.

Columns 1-4.

Same as Card "D".

Columns 5-76.

These 36 two-column entries are optional. When entries are made, they should be 1 to 99 (where 99 implies 100% probability). Whenever a specific scheduled maintenance event, mission or SMA area contains an entry, the analyst should insure there is a corresponding card "L", "M" or "X" for it.

Card "F"

These Red X consequence cards are decoded and tested using subroutine REDXCO. It should be noted that each "F" card can define the Red X data for up to three different elements and/or subsystems by use of the three different areas on the card. The card must be filled, starting in the first area.

Columns 1-2, 27-28, 53-54.

If this area is used (see below), it must be a value of 0 to 98.

Columns 3-4, 29-31, 55-56.

Entries are optional. A "00" or blank, however, implies data is applicable to all elements that exist within that subsystem, unless separately defined on other cards of this type.

Columns 5-20, 31-46 and 57-72 (except as noted next field).

Entries in these three groups of two-column data are optional. However, if a subsystem is specified implying this group is to be used, there must be an entry in at least one of these seven zones. Furthermore, the sum of all entries within a group must equal 100%. Since a 99 implies 100%, a single entry must be a 99; and if there are two entries, neither one may be a 99.

Columns 15, 18, 21, 41, 44, 47, 67 and 73.

If a probability is entered in the blocks immediately preceding these columns, a nonzero integer must appear in these to denote which mission is desired.

Columns 22, 48 and 74.

If another aircraft of the same or of a different mission as that in the previous column is desired, the same mission number or the different mission number is to be indicated.

Card "G"

These cards are decoded using subroutine REPINP and require the following data:

Columns 1-4.

See explanation for columns 1-4 of card "F".

Columns 5-6, 7-8, 9-10 and 44-45.

Entries in these three areas are optional. A 99 is interpreted as 100%. Note that if columns 7-8 are "0" or blank, an "H" card is not required. Also, if columns 7-8 are 99, then columns 9-43 should be blank, or have "0" entries for the individual zones.

Columns 11-12.

May be blank or contain any value between 1 and 30 except 9 or 10.

Columns 13-16.

Must have a value greater than zero if columns 11-12 are blank or between 1 and 8.

Columns 17-19.

Must have a value greater than zero if columns 11-12 are between 1 and 7, and zero or blank if greater than 7. If columns 11-12 are blank and columns 13-16 have a nonzero value, an entry is optional.

Columns 20-21.

Optional; a blank or desired numeric value.

Columns 22-25.

Entries should be as explained under columns 10-13 for Card "C".

Columns 26-27, 32-33 and 38-39.

Columns 26-27 must have a nonzero numeric entry except when columns 7-8 are 99. Use of columns 32-33 and columns 38-39 is optional.

Columns 28-29, 32-33 and 40-49.

If an entry is made in the preceding columns, a nonzero value must appear.

Columns 30-31, 34-35 and 42-43.

Same requirements as above, except 99 is interpreted as 100%.

Columns 46-76.

Optional - any characters.

Card "H"

These cards use subroutine REMRPL for decoding. Note, however, that an element whose "G" card indicates a zero probability of remove and replace requires no "H" card unless the element is a TBO item and a Card "N" exists.

Columns 1-4.

Same as for "G" cards.

Columns 5-13.

The same rules as for columns 11-19 of card "G" apply. If card is used, a valid entry is mandatory.

Columns 14-15, 22-24.

Optional, but if an entry is made it must be numeric. Note that the delay is in tenths of hours.

Columns 16-17, 62-63.

Optional, but if used it must be numeric.

Columns 18-21.

Entries, if made, should be in accordance with explanation for columns 10-13 of card "C".

Column 25.

Only a blank, a "0" or a "1" permitted.

Columns 26-43.

Instructions same as columns on Card "G".

Columns 44-45.

If used, a numeric entry 0 to 99 must be made.

Columns 46-47.

Same requirements as 44-45; however, if left blank and 44-45 is not, then same value as for 44-45 is assumed.

Columns 48-49.

If an entry is made in columns 44-45, this zone must be blank. Conversely, if 44-45 is blank, a nonzero numeric entry is required.

Columns 50-51.

If an entry is made in columns 48-49, it is mandatory that an entry be made.

Columns 52-57.

Optional; however, if entries are made they must comply with instructions for columns 5-13 above.

Columns 58-59.

Optional, but if entry is made it must be a value between 10 and 29.

Columns 60-61.

Optional, unless an entry is made in columns 58-59, which requires a nonzero entry here.

Columns 62-63.

If a test hop is desired, its probability of occurrence must be entered here.

Columns 64-76.

Optional.

<u>Card "I"</u>

This data card is decoded using subroutine PMAINT. A maximum of 18 "I" cards is permitted, and tests are made to insure they do not duplicate the same subsystem specified in columns 1-2.

Columns 1-2.

The applicable subsystem 1 - 98 must be entered.

Columns 3-4, 7-8, 11-12, 15-16,...,71-72.

While essential that columns 3-4 must have an entry, all others are optional. Entries made may not be 0 or 99.

Columns 5-6, 9-10, 13-14, 17-18,...,73-74.

If an entry is made in the two columns preceding the group in question, this group must also have nonzero entry.

Columns 75-76.

Not used.

Card "J"

These cards are decoded and tested using subroutine QTYPMA.

Columns 1-2.

If column 75 contains a "1", there must be a value 1 to 99 in these columns. Entries in this area on cards not having a "1" in column 75 are ignored.

Columns 3-4, 7-8, 11-12, 15-16,...,71-72.

Requirement same as for card "I".

Columns 5-6, 9-10, 13-14, 17-18,...,73-74.

If an entry is made in the two columns preceding the group in question, this group must have a numeric entry.

Column 75.

It is mandatory that an entry 1-9 be made in this column. While any number in this range will be accepted in Card Edit, subsequent programs will check to insure that if N cards were input, they were numbered 1-N.

Column 76.

Not used.

Card "K"

Subroutine OFFEQR is used to decode and test the data on these cards. "K" cards are optional, but are meaningful only when the specified element has a related remove/ replace "H" card.

Columns 1-4.

Same requirements as "H" cards.

Columns 5-8.

While any number or combination of maintenance levels may be applicable to a given element, separate "K" cards are required for each level. Only one of these columns may have a "1" on any given card.

Columns 9-10, 11-12, 13-14, 15-16.

With the exception of columns 11-12, entry into these four groups is optional. Columns 11-12 must be a value greater than 0, and the sum of columns 9-10 and 11-12 must not be greater than 100. If it equals 100% and if columns 15-16 are a 99 (100%), then there should be no "K" card for any subsequent levels.

Columns 17-20.

Optional, but if used see columns 10-13 of card "C" for requirements.

Columns 21-22.

Blank or any value between 1 and 30 except 9 or 10.

Columns 23-26.

Must have a value greater than zero if columns 21-22 are blank or between 1-8, inclusive.

Columns 27-29.

Must have a value greater than zero if columns 21-22 are between 1-7, inclusive, and zero or blank if greater than 7. If columns 21-22 are blank and columns 23-26 have a nonzero entry, an entry herein is optional.

Columns 30-31.

Optional.

Columns 32-33, 38-39, 44-45.

Columns 32-33 must have a nonzero entry. Use of the other two areas is optional, but if used they must contain a nonzero entry. A zero is the equivalent of a nonentry.

Columns 34-35, 40-41, 46-47.

If an entry appears in the preceding two columns, a nonzero entry is required. Otherwise, it must be zero or blank.

Columns 36-37, 42-43, 48-49.

Same as above except 99 implies 100%.

Columns 50-76.

Optional - any characters.

Card "L"

There are a maximum of 20 scheduled maintenance events, excluding dailies, which may be specified for the model. Each can be specified on a flight time or calendar time basis. Flight time inspections must be numbered in order based on increasing time. Calendar time inspections are also to be in ascending order of calendar intervals following. These cards are decoded and tested using subroutine SKEDME.

Columns 1-2.

Must be a number corresponding to the event number 1-20.

Column 3.

Optional. "1" equals yes; "0" or blank is no.

Columns 4-7, 8-9, 10-13.

Must be nonzero entries if the event is flight hour based. Otherwise, must be blank or zero.

Columns 14-16, 17-18.

Must be nonzero entries for calendar events. Otherwise, must be blank or zero.

Columns 19-20.

Blank, or any value between 1 and 30 except 9 or 10. Must be blank or zero if columns 53-54 contain a nonzero entry.

Columns 21-25.

Must have a nonzero entry if columns 19-20 are blank or less than 9 and columns 53-54 are blank or zero.

Columns 26-28.

Must have a value greater than zero if columns 19-20 are between 1 and 7. Must be zero or blank if greater than 7. However, if columns 19-20 are blank and columns 21-25 have a nonzero entry, an entry is optional.

Columns 29-32.

Optional, but if used, editing follows columns 10-13 of card "C".

Columns 33-34.

A blank or nonzero entry is required.

Columns 35-36, 41-42 and 47-48.

Optional; however, if used, columns 35-36 must contain a nonzero entry. A zero is the equivalent of a nonentry.

Columns 37-38, 43-44 and 49-50.

If an entry appears in preceding two columns, a nonzero entry is required.

Columns 39-40, 45-46, 51-52.

Same as above except 99 implies 100%.

Columns 53-54.

When used, entries must be a numeric value of 10-29.

Columns 55-76.

Optional.

<u>Card "M"</u>

This card is decoded and tested using the DAILY subroutine.

Columns 1-3, 4-6.

Nonzero numeric entries are required in both of these groups.

Columns 7-10, 11-12, 13-14.

A nonzero entry must appear in one of these groups, and the other two must be blank or zero. Columns 7-10 must comply with instructions as for columns 38-41 of card "B".

Columns 15-16, 17-19, 20-21.

Same requirements as for columns 19-25 of card "B".

Columns 22-23, 24-25.

Optional, but if used, refer to columns 10-13 of card "C" for requirements.

Columns 26-27.

See comments regarding columns 33-34 on card "L".

Columns 28-29, 34-35, 40-41, 46-47.

A nonzero numeric entry is required when an entry is made. A zero is the equivalent of a nonentry.

Columns 30-31, 36-37, 42-43, 48-49.

If an entry appears in the preceding two columns, a nonzero numeric entry is required.

Columns 32-33, 38-39, 44-45, 50-51.

Same as above except 99 implies 100%.

Columns 52-76.

Optional.

Card "N"

These "TBO" cards are optional, and each card can define two separate elements; however, if only one is listed, it should be in the first grouping. Data on these cards is decoded using subroutine TBOS.

Columns 1-4, 39-42.

The subsystem/element numbers are to be listed. Note, however, that neither subsystem "00" nor "99" is a valid entry.

Columns 5-8, 43-46.

If the interval between overhaul is based on aircraft flight hours, a nonzero numeric entry is required. If the interval is based on calendar time, this area must be blank or zero.

Columns 9-10, 14-15, 47-48, 52-53.

If a nonzero exists in the preceding two columns, a nonzero entry is necessary.

Columns 11-13, 49-51.

If the interval between overhaul is based on calendar days, a nonzero numeric entry is required.

Columns 16-17, 54-55.

Optional, but if used it must be numeric (99 = 100%).

Columns 18-37, 56-75.

Use of each column in these groups is optional; however, the only valid entries are a "1" or a "0", and a "1" must appear in at least one column of the applicable group.

Columns 38 and 76.

Optional. Only valid entries are "1" and "0".

Card "P"

These cards are tested and decoded using subroutine MOSLEV. Each card can accommodate up to three different MOS's. It is necessary that on each card MOS entries begin in the first data group.

Columns 1-2, 26-27, 52-53.

Use of the first group is mandatory, but optional on the other two. Entries must be any nonzero numeric value.

Columns 3, 28, 54.

If an entry is made in the preceding column, the desired maintenance level letter must be entered.

Columns 4-12, 29-37, 55-63.

If entries are made in preceding columns of the group, the MOS code designation should be entered.

Columns 13-25, 38-50, 64-76.

If an MOS is specified in the associated grouping, a name or other identification must appear somewhere within these areas.

Card "Q"

These cards are decoded and tested using subroutine WORKHR. The quantity of cards is dependent on the number of shifts, the similarity of the working schedules, the number of maintenance levels, and the use of regular and surge operating periods.

Columns 1-4.

At least one of these columns must have a "1" entered to indicate the applicability of the data to a particular maintenance level.

Column 5.

The applicable shift number 1, 2 or 3 must be entered.

Columns 6, 7, 8.

A "1" must appear in only one of these three columns.

Columns 9-12, 18-21, 27-30, 36-39, 45-48, 54-57, 63-66.

The working hours (based on a 24-hour clock) are to be entered in the particular columns. Neither 0000 nor 2400 may be used. If the shift does not work on a particular day, no entry should be made. See also instructions for columns 38-41 on card "E".

Columns 14-16, 23-25, 32-34, 41-43, 49-51, 58-60, 67-69.

If a starting time is entered for the same day, a nonzero duration should be entered in these columns.

Columns 70-76.

Not used.

Card "R"

The manpower shift designation cards utilize subroutine MANPOW for decoding and editing. Up to five different MOS's can be specified simultaneously on one card if the period(s), shifts, quantity and days are the same. It is to be noted that the MOS manning level specified for a shift on Monday through Friday must be the same. Also, Saturday shift level must be the same as for Sunday. The quantities for each shift may be different. Tests will insure that this is followed.

Columns 1-2, 3-4, 5-6, 7-8, 9-10.

A nonzero numeric entry must appear in columns 1-2. Use of any of the remaining four areas is optional, but if used, they must also be nonzero entries.

Columns 11, 12, 13.

One of these areas must have a "1".

Columns 14-16, 23-25, 32-34, 41-43, 50-52.

Use of these is optional, but when used, a numeric value must be inserted and all five entries must be identical.

Columns 17-19, 26-28, 35-37, 44-46, 53-56.

Same as for above. These five values need not equal the above five values.

Columns 20-22, 29-31, 38-40, 47-49, 56-58.

See previous comment.

Columns 59-61, 68-70.

An entry is optional, but if made, it must be numeric and both entries identical.

Columns 62-64, 71-73.

Same as above. These two values need not equal the above two values.

Columns 65-67, 74-76.

See previous comment.

Note - Columns 14-76.

A nonvalue entry must appear somewhere within this group of data.

Card "S"

These cards are tested and decoded using subroutine MANPKG. Use of these cards is optional, but there must be one for each package as specified on other cards.

Columns 1-2.

A nonzero numeric value is required.

Columns 3-4, 5-6, 7-8.

A nonzero numeric value must appear in each of these three areas. A 99 in columns 7-8 indicates 100%.

Columns 9-14, 15-20, 21-26, 27-32, 39-44, 45-50, 51-56, 57-62.

Use of these nine groups is optional. If any are used, they should be in sequence with each of the three zones within each group having nonzero entries.

Columns 63-76.

The name of the package should appear beginning in column 63.

Cart "T"

These GSE/Test Equipment cards provide the data for any equipment or packages specified by other cards. Note also that test packages are also indirectly created by these cards since all equipment cards must be examined in total to ascertain whether a particular package exists and, if so, what it contains. These cards are decoded and tested by use of subroutine TESTEQ.

Columns 1-2, 3-4.

A nonzero numeric value must appear in each of these columns. It is advisable to assign equipment a numberr in sequence beginning with 1 as a means of reducing core size in the ARMS model.

Columns 5-6.

If equipment failures are to be included, a nonzero value should be included in these columns.

Columns 7-14.

An entry must appear if an entry appears in columns 5-6. Entries for columns 7-8, 9-12 and 13-14 must comply with instructions for columns 11-12, 13-16 and 17-19, respectively, of card "G".

Columns 15-16, 17-18.

These require numeric values when used.

Columns 19-43.

Use of any of these 25 columns is optional, but when any of the columns 1-2 test equipment is to be included within a package, the quantity within that package is to be appropriately indicated. Note that the quantity so indicated can not be greater than that available as indicated in columns 3-4.

Columns 44-76.

The test equipment name should appear in this area, beginning in column 44.

<u>Card "U"</u>

The general mission data cards are tested and decoded using subroutine MISDAT. A maximum of 10 cards can be utilized for missions "0" (test hop) through "9".

Column 1.

Must contain a number 0 through 9. Note - when a "0" appears in column 1, columns 6-23 and 46-74 are not to be filled in.

Columns 2-5.

Must contain a nonzero numeric value. Note that a decimal point is implied such that 100 is decoded as a unity severity factor, whereas 200 or 50 would imply twice or one-half of normal.

Columns 6, 7, 8, 9.

(See column 1 note) At least one configuration must have a priority designated, and priorities should be designated in ascending order beginning with 1.

Columns 10-11, 12-13, 14-15, 16-17.

(See column 1 note) Entries should have values of 1 to 36 inclusive.

Columns 18-19, 20-21, 22-23.

(See column 1 note) Appropriate values must be entered.

Columns 24-43.

Use is optional. Up to five different SMA's can be called for on a probabilistic basis. For each SMA specified, its identifying number (10 through 29 are valid) and a nonzero probability from 1 to 99 must be provided.

Columns 44-45.

Enter desired value.

Columns 46-72.

This area provides the capability to call for other missions. A value must be entered under the desired mission type numbers for the probability of occurrence and the quantity of aircraft desired.

Columns 73-74.

If the "X" card for this same mission number specifies a building block that is defined on its "W" card as being a type "C", then an entry is required in these two columns for a distribution number (see card "3") whose identity is 11 or greater.

<u>Card "V"</u>

These cards are checked and decoded using subroutine OPEQPK.

Columns 1-3.

The package identity number must be entered here. Only numbers 5 through 998 are permitted.

Columns 3-67 (16 groups of two 2-column entries).

This package consists of those elements as are specified in these areas either by their complete identity numbers or by being indirectly identified by inclusion within a total subsystem. In the first case, the subsystem number and element number must be specified; whereas in the latter, only the subsystem number needs to be stated. Note - subsystems 01 through 98 are the only valid range, but a "00" in the element number is acceptable (implying all elements of that subsystem). At the least, the first group must contain an entry.

Columns 68-76.

Optional. The analyst may use this area for reference.

Card "W"

This data card, which uses subroutine BLDBLK for its decoding, is unique in that it has two areas of data which may be used either to define two entirely different building blocks or to allow for defining one block that requires more space than is provided by the first area. However, this expanded capability does not extend to using more than one card to define a block. In that which follows, only the first area will be described. If the second area is used for a different block, it must follow the same instructions except for its displacement of 38 columns.

Columns 1-3.

Entry is required. Any number from 001 to 999 is valid. Use lowest numbers possible to keep program size as small as possible.

Columns 4-5, 6-7.

If used, entries must be numeric and columns 6-7 (GSE package number) must not be greater than 25.

Column 8.

Enter a "1" if to be included.

Columns 9-11, 12-14, 15-17, 18-20, 21-23, 24-26.

Operating package numbers 1 through 999 are valid in these zones. Usage is optional. Do not repeat package numbers.

Column 27.

One of the four indicated letters must appear. If a "C" operating package, 999 must be included in one of the above.

Columns 28-38.

A name is required. Length is unimportant providing column 28 is used for its start.

Columns 39-76.

Same as above 1-38.

Card "X"

These scenario construction cards are for the 10 possible missions (including the mission "0" test hop) and 20 SMA's. They are decoded using subroutine SCENAR. One or more must exist for every mission or SMA specified on other cards.

Columns 1-2.

Mission/SMA numbers 0 through 29 are the only valid entries.

Column 3.

A number must be entered here whose value is 0 through 9 to designate the sequence of cards when there are more than 9 segments in the mission. Start with "0" for the group of 9 segments, a "1" for the card containing the next segments, and so forth in sequence up to a maximum of 9.

Columns 4-6, 7-9, 10-11.

These must be used on each card. Columns 4-6 must have a numeric entry. A "000" designates fulfillment of the mission task. For all other segments, the applicable building block number should be specified. Columns 7-9 and 10-11 must be zero if a "000" appears in 4-6. However, if columns 4-6 have a nonzero entry, then columns 7-9 and 10-11 should also have nonzero entries with one exception. If the building block specified is an "A" type, i.e., an intramission ground event (off-site), it is possible that the time (columns 10-11) will be zero; even though the relative stress is meaningless, one should be entered.

Columns 12-19, 20-27, 28-35, 36-43, 44-51, 52-59, 60-67, 68-75.

Use of these areas is optional, but if used, they must adhere to the same instructions as for columns 4-11, and they must be used in sequence, as the first totally blank area denotes end of mission (and end of task if a previous block "000" did not appear).

Card "Y"

These regularly scheduled mission data cards are decoded using subroutine RSKEDM. Each card can accommodate three different mission schedules, as there are three identical zones provided for data input. Columns 1-24 must be used first.

Columns 1-4 (also 25-28 and 49-52).

See comments to columns 38-41, card "B".

Columns 5, 6, 7 (also 29-31 and 53-55).

A "1" must appear in one of these three columns, with the others blank.

Columns 8-14 (also 32-38 and 56-62).

At least one of the days and columns represented by this group of seven columns must have a "1" in it. Others may be blank or zero.

Columns 15-23 (also 39-47 and 63-71).

At least one of these 9 columns must have a nonzero entry. If more than one entry is made, this will be the equivalent of specifying simultaneously scheduled but totally independent missions.

Card "Z"

These data cards are decoded using subroutine RANMIS. They provide three separate zones (1-22, 25-46, and 49-70) for three different missions. Only one random mission of each mission number can be scheduled for each regular or surge period. Use of the first zone of data on each card is mandatory, whereas use of the second and the third is optional.

Column 1 (also 25 and 49).

The desired mission number 1 through 9 is to be entered herein.

Columns 2-5 (also 26-29 and 50-53).

See comments to columns 38-41, card "B".

Columns 6-8 (also 30-32 and 54-56).

A numeric entry is required.

Columns 9-11 (also 33-35 and 57-59).

A "1" must appear in one of these three columns, with the others blank.

Columns 12-18 (also 36-42 and 60-66).

See comments to columns 8-14, card "Y".

Column 19 (also 43 and 67).

A nonzero entry must be made.

Column 20 (also 44 and 68).

The entry in this column must be equal to or less than the previous column.

Columns 21-22 (also 45-46 and 69-70).

A value greater than zero must be entered herein.

<u>Card "1"</u>

This card utilizes subroutine ALERTA for its decoding. It also has three zones for processing three different sets of input. As before, the first zone (columns 1-23) must be used, with the other two being optional. Also, the same instructions as given for card "Z" apply to the quantity of each mission number that can be specified.

Columns 1, 2-5 (also 24, 25-28, 47, 48-51).

Same comments as for these columns on card "Z".

Columns 6-7, 8-9 (also 29-30, 31-32, 52-53, 54-55).

Nonzero values must be provided.

Columns 10-13 (also 33-36 and 56-59).

One or more of these four columns must have a "1" entry.

Columns 14, 15 (also 37-38, 60-61).

One or both of these two columns (periods) must have a "1" entry.

Columns 16-22 (also 39-45, 62-68).

Same comments as for columns 12 through 18 on card "Z".

Column 23 (also 46 and 69).

If the answer is yes, a "1" must be inserted.

Columns 70-76.

Not used.

Card "2"

Only one continue mission card is permitted, and it is decoded using subroutine CONMIS.

Columns 1-4.

Must have a nonzero value.

Columns 5-8.

Optional, but must be a nonzero entry if used.

Column 9.

A nonzero entry is required when an entry is made in columns 5-8.

Column 10.

A nonzero value mission number must be entered.

Column 11.

A nonzero value must be entered.

Columns 12-15.

A start time is required and must comply with the requirements for columns 38-41 of card "B".

Columns 16-18.

A nonzero value is required.

Columns 19-25.

Same instructions as for columns 8-14 of card "Y".

Columns 26-29, 30-33.

Only one of these two zones must have a nonzero value entered.

Columns 34-58.

Insert a "1" in appropriate columns; all others should be blank or zero.

Columns 59-76.

Optional.

Card "3"

The empirical distribution function cards are decoded using subroutine EMPDIS. While each card can accommodate up to ten sets of coordinate data points, additional cards can be used to permit a maximum of 100 coordinate data points for each of twenty different functions.

Columns 1-2.

An entry is required, and it must be a value between 11 and 30.

Column 3.

See instructions for column 3 of card "X", except 10 pairs of data can be provided on each "3" card instead of 9 as on the "X" cards for a maximum of 100 pairs.

Columns 4-5.

The total number of coordinate pairs to be found on this and subsequent continuation cards is to be entered here on the card having a "0" in column 3. All other cards must be blank.

Columns 6-65. General Requirements.

There must be at least two sets of coordinate data points. Data will be interpolated between points, so the analyst must use caution in the selection of coordinates for rapidly changing functions. It is necessary that coordinate points be provided in a sequence of increasing values for the independent variable. Desirably, it should range from 0% to 100%. If the range of values provided for the independent variable does not include 0% or 100% (99), it will be assumed that the dependent variable remains constant for all values less than the first independent variable. Similarly, the same dependent variable value will be used for all points above the last independent variable.

Columns 6-7, 12-13, 18-19, 24-25, 30-31, 36-37, 42-43, 48-49, 54-55, 60-61.

These are the independent variables and, if used, must be provided in ascending order with valid entries being zero through 99. The first blank entry on any card will indicate that the previously listed pair was the last set of points. Note - if there is an integer multiple of 10 pairs of data points, the absence of the next card implies the same.

Columns 8-11, 14-17, 20-23, 26-29, 32-35, 38-41, 44-47, 50-53, 56-59, 62-65.

This is the dependent variable for the independent one immediately ahead of it. These columns must have a numeric entry whenever an entry exists in the two columns preceding.

Columns 66-76.

Optional. For analyst's reference usage.

Card "4"

An executive subroutine data card must be included even though it is completely blank. This card is decoded using subroutine EXEC. All entries are optional and must be numeric. While a zero may be entered for columns 1-2, such should never be used in any of the other indicated 20 groups of two. Leaving these entries blank implies no limit.

Card "5"

Use of sensitivity change data cards is optional. A maximum of 19 is permitted. They are decoded using subroutine REPCYC.

Columns 1-2.

A nonzero numeric entry is required. If a 99 is inserted, this impacts the entire aircraft, except for those subsystems which are the subject of other "5" cards (see also comments for columns 27, 72, 73).

Columns 3-6, 7-10, 11-14, 15-18, 19-22, 23-26, 28-31, 32-35, 36-39, 40-43, 44-47, 48-51, 52-55, 56-59, 60-63, 64-67, 68-71.

Entries within these columns are optional, with a blank or 100 implying no change.

Columns 27, 72, 73.

Use of these is optional. A "1" in any of these columns on any one of the "5" cards submitted affects the entire aircraft for the subject change.

Cards "6", "7", "8" and "9"

All of these cards are similar and use subroutine NAMES for their decoding. It is necessary that these cards be supplied for each subsystem (other than "00"), configuration, mission (other than "0") and maintenance level that are specified on the other cards.

Columns 1-2 (Card "6" only).

Numeric values from 01 through 98 inclusive are the only valid entries and must be included.

Column 2 (Cards "7", "8" and "9")

Card "7" requires a nonzero integer, whereas one of the letters A, B, C, or D must be specified for cards "8" and "9".

Columns 3-25.

A name beginning in column 3 should be provided.

Columns 26-50, 51-75.

Optional but, if used, must comply with instructions for columns 1-25.

Cross Edit

The data that was input, decoded and arranged in appropriate files during Card Edit is recovered at the beginning of this Cross Edit phase. This provides information from input, decoded data values, and file location of any specific input. The next step sorts all of the basic element data so that they may be examined in ascending order.

In the following steps, cards will be examined to ascertain whether they were needed and available and that all complementary data exists. Inasmuch as an "A" card is essential to the model, it is sought first. If not found, an appropriate error message is given. Assuming it is found, the data on the card is examined to ascertain whether a surge period was indicated.

The next card sought is a "B" card and, as before, if not found, a message is given. Inasmuch as a "B" card may specify a distribution function, if this was given and was a number greater than ten, a card "3" for this function is also sought. In a similar manner, a card "C" is sought; and if available, the appropriate cards "3", "S", "T" and "X" for any specified function numbers, manpower packages, GSE packages and SMA scenarios are checked. Also checked is whether all reconfigurations specified in the model are also provided an endurance time on card "B".

A card "4" is required. Other than checking its availability, no checks are made regarding the maintenance events indicated.

The next steps involve identifying which MOS levels were indicated on the "P" cards and whether corresponding level name cards "9" are supplied. Concurrent with this, a check is also made to ascertain whether the configuration name cards "7"are also available. Having identified which levels are included, the next check involves ascertaining whether the "Q" cards with working hours for both regular and surge periods are available. Also checked is the availability of the "R" cards for the previously identified MOS types. Data to describe the quantities of MOS's assigned on the various days and shifts must be present.

Having examined both the "Q" and "R" cards individually, they are now compared to insure that any time manpower is assigned, working hours have also been provided. Any incompatibilities result in an appropriate error message.

Next, the availability of all elements and various related cards are checked in the order of ascending element numbers. Each element is first checked to see if there is a card "N" identifying it as a TBO item. A TBO causes further checks to see if cards "L" exist for any scheduled maintenance events specified. Each TBO item also requires a basic element "D" card. For non-TBO elements with a "D" card, a cumulative count is made for all subsystem "00" elements to insure that their failure probabilities add up to 100%. If a "D" card is found, it is essential that there also be a "G" card for the repair-in-place data. It is possible, however, that no "G" card exists having this specific element number but that, instead, one exists for the entire subsystem.

Continuing with the same element number, a check is made for the availability of an "E" card. If the "D" card expressed this element as ever being Red X essential, then an "F" card is needed. If an "E" card did exist, a check is made as to the availability of other supporting cards for any scheduled maintenance events, missions and SMA scenarios which may have been specified. Similarly, if an "F" card exists, checks are made for the availability of the mission scenario cards that have been specified.

Each repair-in-place data card, "G", is examined to assure the availability of the appropriate cards "3", "T", "P", and "R" for any specified functions, GSE package numbers, and MOS ID's, and that an adequate quantity was input for those MOS's to meet the demand during at least one shift. If a test hop is specified, the availability of its appropriate card "X" is also investigated. If a probability of remove and replace is given on this "G" card or if this element is also defined as a TBO item, remove and replace data must be provided by a card "H". The availability of this latter card is then checked, and all "3", "T", "P" and "R" cards are checked for the availability of the specific items required by this "H" card. In addition, an "X" card may be required if an SMA has been specified. If the element neither is a TBO item nor has any probability of remove and replace, no check is made for an "H" card. Similarly, if this item were never to have been removed, there would be no reason to examine any corresponding off-equipment repair cards, "K". However, if the item is removed, a check for "K" cards is made. These cards are not required since their absence implies an automatic 100% scrap. If a "K" card is found, checks are made for the appropriate GSE package, function number, MOS and quantity data from other cards. If, in examining a "K" card, it is found that a 100% probability of scrap exists, or that the sum of the probability of scrap and repair at the level indicated is 100%, no other "K" cards at a higher level are investigated.

The parallel maintenance cards, "I" and "J", are next examined. These cards are optional and need not be included in the model. If they do exist, a check is made to assure that at least one element is specified for each of the subsystems identified on them.

If any scheduled maintenance event cards, "L", are found, they are examined to assure the appropriate supporting cards for distribution functions, GSE packages, MOS packages, MOS and their quantities, and SMA scenarios. Scheduled maintenance events are examined in ascending order to assure that the time between events is also in ascending order. Also checked is the availability of a test hop mission scenario if one is specified following a scheduled maintenance event. A similar check is made for the one daily inspection card, "M", although no SMA's or test hops can be included on a daily.

All of the operating equipment package cards, "V", that were input are individually examined and checks made to assure that any element or subsystem specified is available within the model. In a similar manner, all of the building block cards, "W", are examined to assure that the manpower packages, GSE packages, basic equipment package and operating equipment packages are available.

Mission data cards, "U", are examined, and data are checked for the existence of any specified SMA or mission scenarios and combat multiple hit distribution functions. Also specified on these "U" cards is a probability that each aircraft on the subject mission might call any other missions. Accordingly, a check is made that a priority ranking is specified on that mission's "U" card under the heading "when called by other missions". After all of the "U" cards have been checked, they are then collectively examined to assure that the ranking assigned to the different priorities are all different and range from 1 through 36.

The manpower package loading cards, "S", are checked to assure that the specified quantities and types of MOS are available within the model. The only check made on any of the GSE test equipment cards, "T", is to assure that any specified distribution function is available.

Any time a mission was specified and checked to assure that its scenario card, "X", was available, an automatic check is made to insure that a mission data card, "U", and mission name card "8" were also available. All of the "X" cards that are included in the model are examined to insure that the building blocks specified are defined by a card "W". If there is more than one "X" card per mission or SMA, then (a) those available are in sequence, (b) there are no intervening blank entries, and (c) there is no more than one "000" building block. If the "X" cards being examined pertain to a mission, the sum of segment times for all flight and combat events is checked for compatibility with the endurance specified on the "B" cards for the configurations permitted on the "U" cards. Also, if the mission is indicated by an "F" card as being a repair mission, a check is made to insure that one and only one intramission ground event existed. If the "X" card being examined is for an SMA, a check is made to assure that none of the building blocks specified in the various segments were identified as being a flight combat event.

From the foregoing, it is known which subsystems are required within the model, and a check is made to assure that a card "6", which provides the subsystem name, is available for each one. Following this, a check is made of all available cards "3" to insure that those having multiple cards are in sequence and that the total number of pairs found is equal to that indicated on the first card of each group.

While the quantity of the regularly scheduled mission data cards, "Y", is variable, all of those supplied are examined; and for every mission type specified, a check is made to insure the availability of cards "X", "U" and "8", and that a priority for those missions is provided on their "U" cards for a scheduled mission. Similar checks are made of any random mission cards, "Z", and continuous mission cards, "2", that may have been input. For continuous missions, no check is made to insure that the scheduled events indicated exist within the model, as the answer provided to the question listed on card "2" would have no impact on the model if those events are not included.

No cross checks are made for any alert aircraft data cards "1". However, when found, these are noted as being included in the model. While the sensitivity change cards, "5", are similarly treated, a check is made to insure that any subsystems specified are also included in the model.

A message is printed out at the end of Cross Edit to indicate whether errors have been found that require corrective action before proceeding to the next phase. Also printed out are warning messages if cards provided as input are not utilized within the model. It is then up to the analyst to decide whether it was an oversight on his part to not have specified these data on other cards or whether it is acceptable to proceed to the next phase. Prior to terminating, Cross Edit will create an internal element number for all elements specified. This provides a consecutive listing of those elements and helps to keep the core requirements for the various matrices within the ARMS model to a minimum. All data generated within this phase is recorded on the previously created tape as file #3.

<u>Arithmetic Phase</u>

As with the previous phase, the data which had been stored on tape during Card Edit and Cross Edit is recovered at the beginning of this phase. Most of the decoded input data is converted into proper format and assigned to a particular GPSS savevalue matrix element, logic switch, storage or function. These are temporarily stored on a disc file in the sequence in which they are generated. They are retrieved and resequenced prior to program completion.

Various arithmetic steps are used to convert those values which were input as hours into minutes or to convert percentages into actual units, as is the case for the deterministic replacement parts inventory reorder quantity. Also involved is summation of data for total mission times, etc. When sensitivity change cards are included, all affected items are modified in accordance with the applicable multiplying parameters. When distribution functions 4 or 5 are specified, these must be converted into appropriate logarithmic values since these functions are of the log-normal type. In many cases, several items of data are packed into a single value before transmittal to GPSS as a means of reducing ARMS core size.

One of the larger and more complex arithmetic tasks performed within this phase involves a calculation of a mission failure rate and the assignment of values to the various elements of the matrices used by ARMS to determine the mission segment in which a failure occurs and which element failed. This involves a several-step effort. First is a listing and accumulation of the failure rates of all elements that are included within the specified operating packages included in each building block. A cumulative distribution function is created for each column of matrix MH11 (each column being a different building block). This gives the relative probability, given that a failure occurred in that building block, that a specific element within the model failed. Each element within the model is represented by a particular row of this matrix. The next step creates another distribution within matrix MH9. Columns are missions or SMA numbers, and the rows are segments. The values for this matrix are calculated using the applicable building block failure rate, segment stress and segment time in the appropriate manner and prorated over the entire mission.

Stored in other matrices are the accumulated mission failure rates, segment times and building blocks associated with th each segment. These combine to provide all data needed within the ARMS model to determine in-flight failures.

Throughout this phase, the quantity of the various items such as the highest numbered MOS, building block, subsystems, etc. were noted so that appropriate matrix definitions could be made for ARMS. This adaptive feature minimizes core requirements. These, too, are placed in a temporary disc file.

All data generated must be recalled from the temporary files and presented to the ARMS model in the proper sequence. Furthermore, these records must be numbered so they can be merged with the basic ARMS model and inserted into the appropriate spaces that have been reserved for it. The manner in which this data is transmitted is controlled by the analyst. The data can be printed, punched on cards, or placed on magnetic tape or disc. Any combination of these three may also be specified. To get any of these choices, a "1" is placed in columns 2, 4 and/or 6, respectively, on the data input card.

On completion of the above, SSDDIP has been completed and the ARMS model can now be run.

SDOSFI

GENERAL OPERATION

SDOSFI is an acronym for Simulation Data Output Selection Formatting and Identification. It is a computer program written in the FORTRAN language, which will compute and format the output data from the ARMS simulation model. The model user has considerable control over the output presented. Output control was the major reason for preparation of SDOSFI.

Output is presented in two forms: tabular data and plotted data. The user has control of what portion of the tabular data is printed. Much broader control of plotted data is provided. Many plots have multiple definitions allowing the user to select the most appropriate formula for his specific application. As examples, availability can be chosen from four separate formulae, and NORM is available in eight. Plot data may be requested in sequential or cumulative formats. Data for different outputs may even be mixed - some sequential and some cumulative.

SDOSFI is controlled by means of an input form. This form provides the means of selecting and controlling the desired output. It should be pointed out that the SSDDIP run control card, "A", has an impact on the output data. If the user selects a continuous simulation with multiple data intervals, SDOSFI output will be cumulative. That is, interval two output will be the sum of intervals one and two.

If a repeat from original start mode is selected, SDOSFI output will represent completely independent data samples. The cumulative or sequential choice on plotted data is available in either running mode.

Briefly, SDOSFI operates by combining all data transferred from ARMS with the input-specified output requirements. The program sorts, computes, and formats these data and provides the user with the tables and plots requested. A full output would contain the following:

- Level A Maintainability Summary
- Logistics Summary
- Operational Summary
- Combat Damage Summary
- General Reliability Outputs
- Reliability Parameters by Subsystem
- Number of Aircraft Abort Types by Mission Types
- Aircraft Without Detected Failures by Mission Type
- General Maintainability Outputs
- Maintainability Parameters by Subsystem

Type of Maintenance Performed Array
Unscheduled MMH/FH by Mission Type
Inspections Performed Array
General Logistics Outputs
Maintenance Equipment Array
MOS Utilization Array
MOS Man Hours by Type of Maintenance
General Operational Outputs
General Mission Data by Mission Type
General Mission Data by Mission Class
Alert and Standby Output by Mission Type
General Combat Damage Outputs
Combat Damage Aborts by Mission Type
Percent Organizational Availability vs. Time
    (Plot, 4 formulas available)
Percent Organizational NORM vs. Time
    (Plot, 8 formulas available)
Percent Organizational NORS vs. Time
    (Plot, 6 formulas available)
Flight Hours vs. Time (Plot)
Mission Classification vs. Time (Plot)
MMH vs. Time (Plot)
Active Time Aircraft Spent Waiting for Primary MOS
    (Histogram)
Number of Active Hours From Completion of Flight Until
    Entry Into Ready Pool (Histogram)
Preparation Time by Day and Mission (Plot)
General Operational Outputs by Tail Number
Status Summary in Active Hours by Tail Number
Number, MMH and MEMT by Inspection Type and Tail Number
Logistics Parameters by Element Name
Logistics Parameters by Element and Maintenance Level

It can be seen from this listing that summaries and details are available for all major data categories. This allows the operator to tailor his output request to obtain details in areas of specific interest and summary data for others.

The program written for SDOSFI involves a mainline and 13 subroutines. Each of these routines is discussed in detail in Volume II.

Volume II presents definitions of each output provided. A full printout of the available data is also included with the sample problem in Volume II.

CONCLUSIONS

All program objectives have been met. The following conclusions can be made relative to the ARMS development program:

1. ARMS is a flexible analysis tool that can be utilized over a wide range of simulation experiments without reprogramming.

2. Input data errors are minimized by an extensive set of Edit and Cross Edit programs.

3. Sufficient output data are collected to cover a wide range of user requirements.

4. Output selection features enable specific data to be highlighted by suppressing unwanted data.

5. The model can be exercised, input developed and output analyzed without any understanding of the GPSS or FORTRAN languages.

6. The model enables the user to integrate the impact of combat damage with other reliability, maintainability and supportability characteristics.

7. A wide range of mission scenarios may be examined.

8. Aircraft definition can be made at a level commensurate with the degree of detail in the available data. The analyst is free to define the aircraft in as much detail as desired.

9. The sensitivity change card permits a reasonable number of parametrics to be performed without creating a new data source.

10. The model contains an option which allows the analyst to cause the model to test itself for stabilization.

7980-75