RESIDENT MAINTENANCE
SOFTWARE SUBSYSTEM DESCRIPTION
3A PROCESSOR EXTENDED OPERATING SYSTEM

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

1. GENERAL

1.01 This section provides a description of the resident maintenance task MAINT (PR-4C607; release G2A) of the Extended Operating System (EOS). Also included in this section is a brief description of the input/output messages associated with the main store (MAS) audit.

1.02 Whenever this section is reissued, the reason for reissue will be listed in this paragraph.

1.03 The following sections provide background information for this section:

SECTION	TITLE
254-340-001	Extended Operating System (EOS) Overview, Software Subsystem Description, 3A Processor Extended Operating System
254-340-031	Processor/Process Management, Interrupt Handling and Timer Management, Software Subsystem Description, 3A Processor Extended Operating System
254-340-080	Maintenance Overview, Software Subsystem Description, 3A Processor Extended Operating System
254-340-082	System Utilities, Software Subsystem Description, 3A Processor Extended Operating System
254-340-086	Initialization And Recovery, Software Subsystem Description, 3A Processor Extended Operating System
254-340-106	Extended Operating System Macros and Glossary

1.04 The following assembly units contain code that relate to this section:

- Operating system tables (OSTABS, PR-4C120 or LOSTAB, PR-4C147) provide various parameters which define system hardware and software configuration as specified by the application.
- Maintenance Data Layouts (MAIDTA) PR-4C606 contain the miscellaneous data layouts for maintenance.
- Maintenance Task (MAINT) PR-4C607 supervises the duplex operation of the processor.
- Base Level Maintenance Monitor (CBLM) PR-4C617 controls a number of loop sequence programs utilized by EOS.
- Paging Monitor (CPAGM) PR-4C619 controls passing functions for nonresident programs.
- Common System Data Layout (CTSD) PR-4C621 contains layout information for temporary store locations.
- Common System Utility Program (CUTIL) PR-4C622 provides a set of general read and monitor functions.

1.05 MAINT, along with the other resident maintenance assembly units, is used to ensure the integrity and reliability of the system. The assembly units that provide resident maintenance are listed in Table A.

1.06 To aid in understanding the operation of MAINT, the data layouts used by MAINT are listed in Table B, and the main routines used by MAINT are listed in Table C.

2. OVERVIEW

A. Task MAINT

2.01 The purpose of MAINT is to supervise the resident maintenance of simplex or duplex processor operation. MAINT performs its functions in an endless loop. The functions are normally scheduled to be performed every 100 to 300 milliseconds as established by the application in OSTABS.

2.02 The maintenance functions are divided into three priority levels: high, medium, and low. They are performed on a periodic schedule in accordance with the Maintenance Priority Structure (paragraphs 2.04 and 2.05). Thus, MAINT runs at three priority levels during its execution. The following functions are performed by MAINT:

(a) High Priority Functions

(1) Clearing the loop control flag

(2) Monitoring the state of the off-line 3A CC

(3) Resetting the stuck interrupt counters

(4) Restoring stuck interrupts 3, 5, and 7

(5) Resetting the program timer

(6) Performing utility functions

(7) Satisfying macro requests

(8) Checking software sanity and changing priority (to medium).

(b) Medium Priority Functions

(1) Informing the application as to changes of the SSP

(2) Performing nonmaintenance functions

(3) Updating MAS

(4) Checking software sanity and changing priority (to low).

(c) Low Priority Function

(1) Auditing the off-line MAS.

2.03 MAINT also contains event routines that remove interrupts 3, 5, and 7 from service. These routines are not called by MAINT internally and are only performed when flagged from another program.

Maintenance Priority Structure

2.04 The EOS tasks are run on a priority basis with the highest priority task running first. There are 256 available priority levels in EOS, 255 (highest) through 0 (lowest). As each task is created, a priority is assigned to it. The priorities may be changed during system operation.

2.05 MAINT runs at three priority levels which are designated by the application.

(a) The High Priority Functions level is specified by the parameter MAIPRIOR in OSTABS. A typical value for MAIPRIOR is 248 (255 for applications that use LOSTAB instead of OSTABS). High Priority Functions must be performed on a rigid schedule to ensure system integrity.

(b) The Medium Priority Functions level is specified by MAIPRIM in OSTABS (typically 10). These functions are somewhat deferrable but cannot be delayed for long periods of time.

(c) The priority level for the Low Priority Function (MAS audit) is zero. This is not critical and can be deferred for long periods of time.

B. Main Store Audit

2.06 The MAS audit runs automatically as a Low Priority Function and on completion of a memory update. In addition, the audit can be initiated manually from the TTY whenever necessary. The audit function—

- Locates and corrects memory faults
- Outputs information about the faults to the TTY
- Maintains a list of recent faults.

The input/output messages associated with the MAS audit are described in part 4 of this section.

3. FUNCTIONAL DESCRIPTION OF MAINT

CREATION

3.01 During system initialization, the EOS kernel builds the task descriptor MAINT based on data defined by the TASK macro in OSTABS. The task is created in the READY state with a priority of MAIPRIOR. The kernel then inserts MAINT on the READY LIST. When MAINT becomes the

highest priority task, the dispatcher moves MAINT to the RUN state.

PSEUDO-INITIALIZATION ROUTINE

3.02 When the task MAINT moves to the RUN state, control enters MAINT at the pseudo-initialization routine (entry point MAINETRY). The pseudo-initialization routine performs, or causes to be performed, the following functions:

(1) Sets up the message buffer to be used by MAINT for all EOS messages (BUFF in MAIDTA).

(2) Declares and enables the event routines used for event processing.

(3) Initializes the software sanity counters to the values specified in OSTABS (see paragraph 3.03).

(4) Checks for the out-of-service (OOS) option of the INITCC macro. If OOS is specified, MAINT calls the MSFREQ subroutine in CBLM MSFREQ requests a multiscan function.

(5) Clears the first word of the System Status Register (SYSTATE) in CTSD if the system is in the simplex operation mode. This permits TTY messages.

(6) Ensures that the second word of SYSTATE is zero. This word contains information on bad memory locations in MAS.

(7) Sends a message to the application requesting that the MAS lamp of the SSP be turned off, if it is on.

(8) Opens the nonresident paging files via the OPENPAG subroutine in CPAGM.

(9) "Wakes up" the task (if any) that is using the off-line CC (see paragraph 3.04).

3.03 The software sanity counters are used to detect looping of a task between specified priority levels. The counter CTR_B in MAIDTA is initialized to the value of CTR_B_MX in OSTABS. CTR_B is used to detect looping of a task between priority levels MAIPRIOR and MAIPRIM. The counter CTR_C in MAIDTA is initialized to the value of CTR_C_MX in OSTABS. CTR_C is used to detect looping in a task between priority levels MAIPRIM and 0 (zero). The OSTABS parameters CTR_B_MX and CTR_C_MX are set by the application to specify the maximum values of CTR_B and CTR_C, respectively.

3.04 "Waking up" a task that is using the off-line CC, removes that task from the hung state (ie, waiting for an event that will not occur). MAINT first checks OFLCCRQ in MAIDTA. If it is set (logical one) then a task has requested use of the off-line CC. The name of the hung task (specified by OFLCCSN in MAIDTA) and the event number that hung the task (specified by OFLCCEV in MAIDTA) are passed to the MAINT subroutine SETFLG when it is called. SETFLG sets the specified event flag for the named task.

3.05 The functions of the pseudo-initialization routine are performed during each system initialization at priority MAIPRIOR.

LOOP CONTROL

3.06 After termination of the pseudo-initialization routine, control enters an endless loop (entry point STRTLOOP). At the beginning of the loop, MAINT is running at priority MAIPRIOR. The loop consists of the High Priority Functions followed by the Medium, then the Low Priority Functions. The High Priority Functions must be performed on a rigid schedule, even at the sacrifice of Medium and Low Priority Functions (in some cases). MAI_TIME in TIMEAU schedules the High Priority Functions of the loop based on the parameter MAIRATE in OSTABS.

3.07 MAIRATE is defined as Q/10, with Q being the time interval in milliseconds (typically 200) that should occur between reinitiating the High Priority Functions. MAIRATE specifies that rate in terms of 10ms timer interrupts (typically 20).

3.08 A counter, 10MSCNT in TIMEAU, is initially set to MAIRATE. The 10MSCNT is decremented by one on every occurrence of the 10 ms timer interrupt. When the counter reaches zero, MAI_TIME resets 10MSCNT to MAIRATE, changes the priority of MAINT to MAIPRIOR, and sets the loop control flag (LOOPCTL in MAIDTA) to logical one.

3.09 At several points during the Medium and Low Priority Functions, MAINT checks the LOOPCTL flag. If it is set (logical one), MAINT then returns to entry point STRTLOOP and begins the High Priority Functions. If LOOPCTL is not set, then MAINT continues with the Medium or Low Priority Functions.

3.10 Since few tasks have a higher priority than MAINT when it is running at MAIPRIOR, it is unlikely that MAINT will be suspended during the High Priority Functions. MAINT could very likely be suspended at priorities MAIPRIM or zero by a higher priority task. If MAINT is in the suspended state when MAI_TIME ups the priority of MAINT to MAIPRIOR, MAINT should suspend the current task immediately. Depending on the point where MAINT was suspended, the Medium or Low Priority Functions may execute at MAIPRIOR up to the point that MAINT checks LOOPCTL (see Fig. 1).

HIGH PRIORITY FUNCTIONS

3.11 Resident maintenance functions that are critical to system operation and integrity are performed at priority MAIPRIOR. The following paragraphs describe the high priority functions.

3.12 The first function of MAINT is to clear LOOPCTL. This permits Medium and Low Priority Functions to be executed, time permitting.

A. Monitoring the State of the Off-Line 3A CC

3.13 MAINT checks the system mode (MAI_CRL in MAIDTA). If duplex operation is specified, then MAINT calls the System State Detector (SSD) monitor routine (entry point STATD) in CBLM. After the state of the off-line 3A CC is evaluated by the SSD monitor, control is returned. MAINT then updates the SSP with the new status of the off-line 3A CC.

B. Resetting the Stuck Interrupt Counters

3.14 MAINT sets the 16 stuck interrupt counters to their maximum value per time interval (MAIRATE). These maximum values are defined by the application within the Stuck Interrupt Table (STKINTTB) in OSTABS. MAINT copies the 16-word STKINTTB into the 16-word layout: FIXED.PROCESSOR O.STUCK. Each word corresponds to a specific interrupt. Every time an interrupt occurs, its associated word is decremented by one. When a specific word reaches zero (within a single MAIRATE interval, typically 200 ms), the associated interrupt is assumed to be stuck and is marked OOS. MAINT performs this function once per loop.

C. Restoration of Stuck Interrupts 3, 5, and 7

3.15 MAINT restores (ie, places back into service) 3 of the 16 interrupts. These interrupts are the Matcher interrupt (3), the Error interrupt (5), and the Other CC interrupt (7). Each interrupt is handled separately, first 3, then 5, followed by 7. During every loop through the High Priority Functions, MAINT checks the interrupt restoral counters, designated as ICOUNT3, ICOUNT5, and ICOUNT7 in MAIDTA. Since all three are handled in the same manner, a general description of stuck interrupt restoration is contained in the following paragraphs. The letter x is used to represent the appropriate interrupt (3, 5, or 7) being processed. MAINT tests for three values of the ICOUNTx parameter. These values are RICOUNT, one and zero.

3.16 RICOUNT (typically 9000) in OSTABS specifies the number of loops through the High Priority Functions that should occur before MAINT restores the interrupt. Thus, RICOUNT specifies the amount of time that stuck interrupts remain OOS in terms of the loop of MAINT. If ICOUNTx is set to RICOUNT, then interrupt x has just been removed from service by the appropriate removal routine. (See paragraphs 3.41 through 3.43 for details on removal.) MAINT then outputs a message to the maintenance TTY and decrements the ICOUNTx parameter by one. The TTY message is in the form:

RMV INT x

Note: The RMV INT message will cause a minor alarm.

After handling of the RICOUNT condition of an interrupt, control proceeds to the next interrupt restoration routine or the Program Timer reset function.

3.17 If the value of ICOUNTx is one, then MAINT will restore that interrupt via a call to the RST_INT routine in INTRPT. MAINT then outputs a message to the maintenance TTY and decrements

ICOUNTx by one. The TTY message is in the form:

RST INT x

After handling the "one" condition of an interrupt restoral counter, control proceeds to the next interrupt routine or the Program Timer reset function.

3.18 If the value of ICOUNTx is zero, then no action is required, and control proceeds to the next interrupt routine or the Program Timer reset function. This case corresponds to a properly functioning interrupt.

3.19 If the value of ICOUNTx is other than RICOUNT, one or zero, MAINT will only decrement that ICOUNTx by one and proceed to the next interrupt routine or to the Program Timer reset function.

3.20 Hardware problems in the circuitry of interrupts 3, 5, or 7 will result in that interrupt being repetitively removed and restored until the problem is cleared. Thus, a small value of RICOUNT could result in a flood of interrupt messages to the TTY. The typical time interval between removal and restoration of interrupts 3, 5, or 7 (when necessary) is 15 minutes, but this is dependent on the values of RICOUNT and MAIRATE.

D. Resetting the Program Timer

3.21 After restoration of stuck interrupts 3, 5, or 7, MAINT resets the Program Timer via a call to the RESETPT subroutine of CBLM. RESETPT presets the program timer to time out in the hundreds-of-milliseconds range.

E. Performing Utility Functions

3.22 MAINT performs common system utility functions via a call to the CUTILBAS routine of CUTIL. The utility functions are described in Section 254-340-082.

F. Satisfying Macro Requests

3.23 After the utility functions are performed, MAINT branches to the OFLCCBL routine in MAISVC. OFLCCBL performs background maintenance for requests to the off-line CC that were generated by maintenance macros.

G. Software Sanity Check and Priority Change

3.24 After satisfying the macro requests, MAINT decrements the CTR_B counter by one. If CTR_B is zero, then some task with a priority between MAIPRIOR and MAIPRIM is looping. In this case, MAINT will issue the INTCC macro to initialize the CC at level LCRIT. If CTR_B is not equal to zero, MAINT proceeds to the next High Priority Function.

3.25 The last High Priority Function that MAINT performs is changing its priority to MAIPRIM. This is accomplished via the PRIORITY macro.

MEDIUM PRIORITY FUNCTIONS

3.26 The following paragraphs describe the functions that MAINT performs at MAIPRIM. These functions are not critical, but they cannot be delayed for long periods of time.

3.27 MAINT first checks LOOPCTL to determine if it is time to return to the High Priority Functions. If so, control branches to STRTLOOP; if not, MAINT then sets the software sanity counter CTR_B to its maximum value, CTR_B_MX, since no task is looping.

A. Detecting SSP Changes

3.28 MAINT performs several functions related to the SSP if the system is operating in the duplex mode or if the SSP is enabled. MAINT informs the application of the occurrence of any of the following:

- Depression of the EXEC key
- Major changes of the off-line CC
- Uncorrectable memory errors
- Depression of the ALARM RELEASE key
- Depression of the ALARM TRFR key.

3.29 MAINT only informs the application of the previously named events. It is the responsibility of the application to reset keys, turn off SSP lamps, activate and transfer alarms, etc. In the

case of the EXEC key depression, MAINT will inform the EOS diagnostic monitor (DGCTSK) in addition to any specified application diagnostic monitor.

B. Nonmaintenance Functions

3.30 MAINT performs several functions that are not strictly related to maintenance. The first is releasing the Paging Buffer for use by EOS when no multiscan function is in progress. The second is the processing of ESS-type maintenance messages from the TTY. MAINT processes five messages via the PROCESS_INPUT_ESS_MESSAGE subroutine. These messages are listed in Table D.

3.31 Next, MAINT may update the common system clock from the EOS real-time clock. This updating only occurs once per second (approximately every five passes through the Medium Priority Functions).

3.32 After (possibly) updating the EOS clock, MAINT branches to the Multiscan Function Controller (MSFC) in CBLM. Upon return, MAINT checks the LOOPCTL to determine if it is time to start the High Priority Functions again. If LOOPCTL is set (logical one), control branches to STRTLOOP. If LOOPCTL is not set, MAINT continues with the Medium Priority Functions.

3.33 MAINT then checks to determine if any CC diagnostics are active via a call to MSFCHK in CBLM. If any are active, MAINT repeats the functions in paragraph 3.32 and the call to MSFCHK. This permits the CC diagnostics to be run at MAIPRIM instead of zero in order to complete them more quickly. The looping through the call to MSFC will continue until the diagnostics finish or TIMEAU sets LOOPCTL. When the diagnostics finish or if no diagnostics are active, MAINT proceeds to the next function.

C. Memory Update

3.34 MAINT next calls AUUPDMAS of CBLM to perform the memory update and audit. After return from AUUPDMAS, MAINT checks the LOOPCTL in MAIDTA. If it is set, control branches to STRTLOOP and begins the High Priority Functions. If LOOPCTL is cleared, MAINT continues to the next function.

3.35 MAINT checks the AUMASCTL flag of CTSD to determine if the memory update is active. If so, MAINT repeats the functions in paragraph 3.34 and the AUMASCTL check. Like the CC diagnostics, the update will run at MAIPRIM to permit faster completion.

D. Software Sanity Check and Priority Change

3.36 After completion of the MAS update (if present), MAINT decrements the CTR_C sanity counter by one. If CTR_C is zero, then a task with priority between MAIPRIM and zero is looping. In this case MAINT will issue the INTCC macro to initialize the CC at LCRIT. If CTR_C is not equal to zero, MAINT proceeds to the next function.

3.37 The last Medium Priority Function of MAINT is that of changing to priority level zero via the PRIORITY macro. There is a good chance MAINT will be suspended by a higher priority task after dropping priority down to zero.

LOW PRIORITY FUNCTION

3.38 The first function MAINT performs at priority zero involves checking the LOOPCTL flag to determine whether MAIRATE has expired. If so, control is passed to entry point STRTLOOP; if not, MAINT then sets CTR_C to its maximum value, CRT_C_MX.

3.39 MAINT then calls AUUPDMAS in CBLM to perform the MAS update and audit. MAINT will continue checking LOOPCTL and alternately calling AUUPDMAS until LOOPCTL is set by TIMEAU. If the system is in the simplex mode, the MAS audit is not necessary and the repetitive call to AUUPDMAS is used to waste time until LOOPCTL is set. Application audits may run on a time-shared basis during this "waste time" loop.

REMOVAL OF STUCK INTERRUPTS (3, 5, AND 7)

3.40 MAINT also contains three routines that remove interrupts 3, 5, and 7 from service when they are stuck. Stuck is the condition when an interrupt occurs more times than its assigned maximum value. Since all three interrupts are removed in the same manner, a general description of stuck interrupt removal is contained in the following paragraphs. The letter x is used to represent the particular interrupt 3, 5, or 7.

3.41 When interrupt x is detected as stuck, control is transferred to entry point STKINx. The STKINx routine calls the RMV_INT routine of INTRPT. RMV_INT removes the interrupt x from service (the number of the interrupt is passed to RMV_INT on the call). Upon return, the routine sets the ICOUNTx parameter to RICOUNT to signal removal of that interrupt. The routine then terminates.

3.42 These routines remove an interrupt from service, and are not used internally by MAINT. The routines described in paragraphs 3.15 through 3.20 restore these interrupts.

4. MAIN STORE AUDIT MESSAGES

A. Input

4.01 There are four TTY messages that can be used to control the MAS audit. These commands are listed in Table E and described in the following paragraphs.

4.02 The AU:MAS command initiates the MAS audit. This command is effective only if the off-line CC is ***not*** in the OOS state. The optional form AU:MAS;UCL is used if the off-line CC is in the OOS state. This unconditional audit tests only for parity correctness in the on-line MAS. Completion of the audit is signified by the AU MAS COMPL message.

4.03 The INH:MASAU command will inhibit the MAS audit. This condition will exist until the next automatically initiated update or the ALW:MASAU message is input. The inhibit is useful in stopping excessive output messages during a manually initiated audit.

4.04 The ALW:MASAU command allows the MAS audit in cases where it has been stopped by the INH:MASAU command or by other maintenance functions.

B. Output

4.05 There are six TTY output messages associated with the MAS audit. One of these, the AU MAS COMPL, will not occur during an automatic audit. During an automatic audit, the REPT messages for both on-line and off-line are supressed after their first occurrence. Messages are never suppressed during a manually requested audit. The output messages are listed in Table F.

4.06 The AU MAS COMPL signifies completion of the audit. This message is output for manually requested audits only.

4.07 The RCOVRY MAS ERR message indicates that the MAS audit has successfully corrected a problem. The source of the problem and its location are identified. The fields of the RCOVRY MAS ERR message as listed in Table G. Single bit memory failures located by CU diagnostics use this output message (except that the first word is DGN instead of RCOVRY). The pack designated by either message should be replaced **immediately** and tagged to identify the failing DIP.

CAUTION: Either message signifies potential service-affecting conditions and is accompanied by a major alarm.

4.08 The REPT MAS COMP message is output if a word in complement-corrected form is detected. All complement-corrected words are listed during a manually initiated audit.

4.09 The REPT MAS TRBL message is output if an attempt to complement-correct a word has been unsuccessful.

CAUTION: This message signifies potential service-affecting conditions and is accompanied by a major alarm.

The audit will continue to attempt correction of the location during subsequent audit cycles. If correction is successful, the REPT MAS COMP message is output. The REPT MAS TRBL message can appear during an off-line MAS update indicating that the update of the specified location was ***not*** performed.

4.10 The REPT MAS MISMATCH message occurs whenever a logical difference between corresponding on-line and off-line MAS locations is detected and the system is in the double store read mode. This condition causes the double store read to be disabled. This message may indicate a double parity memory problem.

CAUTION: This message signifies potential service-affecting conditions and is accompanied by a major alarm.

5. GLOSSARY

5.01 The following terms and acronyms are provided to aid in understanding the definitive words used in this section.

3A CC—The 3A Central Control

Application—A set of functional system programs which use the services of EOS

Assembly Unit—A collection of code that is assembled or compiled as one entity

CSECT—Control Section; a relocatable block of code

Duplex—The mode when both 3A CCs are operational, one being active, the other standby

EOS—The Extended Operating System

Event—A specified state change

Hung Task—A task that is waiting for an event that will not occur

Interrupt—A hardware-generated signal that causes the task currently running to be suspended and another function that is required immediately to be performed

Kernel—The basic software functions of EOS as compared to those that run as tasks

LCRIT—The initialization level which causes all tasks to be initialized

MAS—Main Store

Multiscan Function—A function which is performed in short segments interrupted by real-time functions and which requires a number of scans (cycles) of the base level loop to complete

Off-Line—The state when a processor is available for use but is not active

On-Line—The state when a processor is active, performing it primary functions

OOS—Out-of-Service

Paging Buffer—An area of MAS that is used to load nonresident programs

PR-Number—A unique number assigned to an assembly unit listing which identifies the system to which the listing pertains

Program Timer—the hardware sanity counter; used to detect hardware faults and looping of software above MAIPRIOR

Pseudo-Initialization Routine—a standard type of task initialization provided by the EOS for those tasks which do not require a real initialization routine

Restore—place back into service

Simplex—the mode when only one 3A CC is operational, the other being ignored or OOS

SSP—System Status Panel

Stuck Interrupt—an interrupt that occurs a greater number of times than the specified limitations

Suspended—the state of a task in which it is essentially put to sleep and another task is allocated the processor

Task—an execution module scheduled to run asynchronously at a specified priority.

ASSEMBLY UNIT TIMEAU

EVERY Q INTERVAL THE ROUTINE MAI_TIME SETS THE PRIORITY OF MAINT TO MAIPRIOR (TYPICALLY 248) AND SETS THE LOOPCTL FLAG OF MAIDTA TO LOGICAL ONE.

THE DISPATCHER

IF, AT ANYTIME, A TASK ON THE READY LIST HAS A HIGHER PRIORITY THAN THE CURRENT PRIORITY OF MAINT, THEN MAINT IS SUSPENDED AND THE HIGHER PRIORITY TASK IS RAN. WHEN MAINT BECOMES THE HIGHEST PRIORITY TASK ON THE READY LIST AGAIN, IT IS RESUMED AT THE POINT IT WAS SUSPENDED.

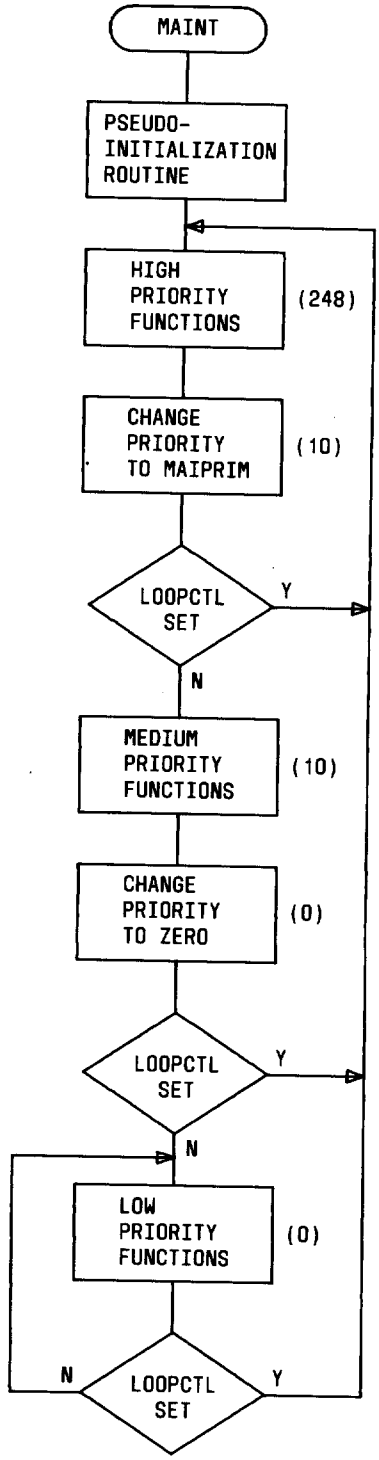

Fig. 1—Functional Flow of MAINT

TABLE A

RESIDENT MAINTENANCE ASSEMBLY UNITS

ASSEMBLY UNIT	ASSEMBLY UNIT FUNCTIONS
CBLM	A common system program which determines loop sequencing in ESS applications. EOS makes use of several CBLM monitors called as subroutines from the maintenance control task MAINT. The monitors used in EOS are: (1) System State Detector (SSD), (2) System Status Panel Controller (SSPC), (3) Multiscan Function Controller (MSFC); (4) Time Monitor (TI).
MAINT	The function of MAINT is to supervise the duplex or simplex operation of the CCs. MAINT consists of an endless loop which runs partially at priority 0 and partially at application specified medium and high levels of priority, identified at system generation times in OSTABS. The rate at which MAINT performs the high priority functions is also specified by the application. The normal rate is 200ms. MAINT also provides administrative control for processor diagnostics.
OSTABS	This program provides the parameters which define the system hardware and software configuration and is specified at system generation time by the application.
MAIDTA	This assembly unit contains the miscellaneous data layouts for the maintenance functions. It also contains various data layout macros which are used to generate the required data tables.
CUTIL	This assembly unit contains a set of general purpose utility functions which provide monitor and load capabilities for registers and store locations.
TTYAPP	This program processes maintenance system output messages. It is indirectly called by MAINT as part of the medium priority functions.
CPAGM	This assembly unit contains the subroutines which control the paging mechanism by which nonresident programs are paged from the Tape Deck Controller to the system paging buffer.

TABLE B

DATA LAYOUTS USED BY MAINT

NAME	ASSEMBLY UNIT	PURPOSE
AUMASCTL	CTSD	MAS update and audit control word
BUFF	MAIDTA	Buffer for EOS messages
CTR_B	MAIDTA	Software sanity counter (MAIPRIOR to MAIPRIM)
CTR_B_MX	OSTABS	Maximum value for CTR_B
CTR_C	MAIDTA	Software sanity counter (MAIPRIM to zero)
CTR_C_MX	OSTABS	Maximum value for CTR_C
STUCK	OSTABS	Layout for the stuck interrupt counters; part of the FIXED. PROCESSOR 0 layout
ICOUNT3	MAIDTA	Interrupt restoral counter for interrupt 3
ICOUNT5	MAIDTA	Interrupt restoral counter for interrupt 5
ICOUNT7	MAIDTA	Interrupt restoral counter for interrupt 7
LOOPCTL	MAIDTA	Controls the loop of MAINT
MAI_CRL	MAIDTA	Controls maintenance mode of system (ie, Simplex or Duplex)
MAIPRIM	OSTABS	Medium priority level of MAINT
MAIPRIOR	OSTABS	High priority level of MAINT
MAIRATE	OSTABS	Rate of MAINT loop execution
OFLCCEV	MAIDTA	Number of the event requested by the OFLCCSN task
OFLCCRQ	MAIDTA	When set, flags that a task has requested the services of the off-line CC
OFLCCSN	MAIDTA	System name of the task requesting the services of the off-line CC
RICOUNT	OSTABS	Number of loops before interrupt restoration
STKINTTB	OSTABS	Table that specifies the maximum values for the stuck interrupt counters
SYSTATE	CTSD	Current status of system

TABLE C

ROUTINES USED BY MAINT

ROUTINES	ASSEMBLY UNIT	FUNCTION
AUUPDMAS	CBLM	Performs update and audit of the off-line MAS
CUTILBAS	CUTIL	Performs utility functions
MAI_TIME	TIMEAU	Controls the loop of MAINT
MSFC	CBLM	The MFS Controller
MSFCHK	CBLM	Checks for an active MSF
MSFREQ	CBLM	Requests a MSF to be performed
OFLCCBL	MAISVC	Processes requests for the services of the off-line CC that were generated by maintenance macros
OPENPAG	CPAGM	Opens the paging files
PROCESS_ESS_ INPUT_MESSAGE	MAINT	Processes the maintenance TTY messages
RESETPT	CBLM	Resets the on-line CC Program Timer
RMV_INT	INTRPT	Removes an interrupt from service
RST_INT	INTRPT	Restores an interrupt to service
SETFLG	MAINT	Sets the specified event flag of the specified task
STATD	CBLM	The System State Detector Monitor

TABLE D

TTY MESSAGES PROCESSED BY MAINT

TTY MESSAGE	DESCRIPTION
SET:CLK:TIME(HH,MM,SS),DATE(MM,DD,YY)!	Set the clock to the specified time and date; TIME and DATE may be input in reverse order
SET:CLK:TIME(HH,MM,SS)!	Set the clock to the specified time
ALW:PAGEMON:FILE N!	Allows paging requests to be honored; N is 0 or 1 which selects the paging file
INH:PAGMON:FILE N!	Inhibits any paging request; N is 0 or 1 which selects the paging file
MOD:DATCLS:MTC!	Modify data class
ACT:RAID!	Causes RAID to be entered even though no breakpoint has been previously specified

TABLE E

INPUT COMMANDS FOR MAINSTORE AUDIT

TTY COMMAND	PURPOSE
AU:MAS!	Audit MAS
AU:MAS;UCL!	Audit MAS unconditional
INH:MASAU!	Inhibit MAS audit
ALW:MASAU!	Allow MAS audit

TABLE F

OUTPUT MESSAGES FOR MAINSTORE AUDIT

TTY MESSAGES	MEANING
AU MAS COMPL	MAS audit completed
RCOVRY MAS ERR	MAS problem corrected (See TABLE G)
REPT MAS COMP	Report a compliment corrected word
REPT MAS MISMATCH	Report logical difference between on-line and off-line MASs
REPT MAS TRBL	Report uncorrectable location

TABLE G

FIELDS OF THE RCOVRY MAS ERR MESSAGE

RCOVRY MAS ERR cc add good bad bit mas mod pack dip cause		
FIELD	**PURPOSE**	
cc	The CC in control	
add	The address of the fault	
good	The good data that was placed into add	
bad	The bad data that was overwritten	
bit	The faulty bit	
mas	The MAS in which the fault occurred	
mod	The module in which the fault occurred	
pack	The circuit pack in which the fault occurred	
dip	The DIP in which the fault occurred	
cause	The source of the correction:	
	(blank)	Corrected using data from the off-line MAS
	DSR	Double store read
	COMP	Compliment correction
	LIST	From the MAS fault list