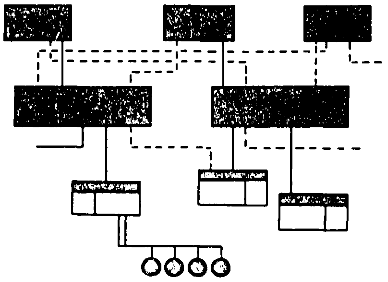
GE-625/635 GECOS-III Fault Processing And Service MME's

Betty J. Coleman

Information Systems

Application Manual

SOFTWARE MAINTENANCE DOCUMENT



GENERAL ELECTRIC

CPB-1493

GE-625/635 GECOS-III Fault Processing And Service MME's

SOFTWARE MAINTENANCE DOCUMENT

May 1968

INFORMATION SYSTEMS

GENERAL ELECTRIC

PREFACE

This manual describes the implementation of fault processing and service MME's for the GE-625/635 Comprehensive Operating Supervisor (GECOS).

Additional software maintenance documents are as follows.

GE-625/635 GECOS-III Introduction and System Tables, CPB-1488

GE-625/635 GECOS-III Startup, CPB-1489

GE-625/635 GECOS-III System Input, CPB-1490

GE-625/635 GECOS-III Dispatcher and Peripheral Allocation, CPB-1491

GE-625/635 GECOS-III Rollcall, Core Allocation,
Operator Interface, CPB-1492

GE-625/635 GECOS-III I/O Supervision, CPB-1494

GE-625/635 GECOS-III Error Processing, CPB-1495

GE-625/635 GECOS-III Termination and System Output, CPB-1496

GE-625/635 GECOS-III File System Maintenance CPB-1497

GE-625/635 GECOS-III Utility Routines, CPB-1498

GE-625/635 GECOS-III Comprehensive Index and Glossary, CPB-1499

GE-625/635 GECOS-III Flowcharts, CPB-1500

GE-625/635 GECOS-III Time-Sharing System, CPB-1501

This manual was produced using the General Electric Remote Access Editing System (RAES). RAES is a time-shared disc-resident storage and retrieval system with text-editing and manuscript formatting capabilities. The contents of the manual were entered into RAES from a remote terminal keyboard, edited using the system editing language, and formatted by RAES on reproduction masters.

The index was produced using a computer-assisted remote access indexing system. This system produces an index using source strings delimited at manuscript input time.

Suggestions and criticisms relative to form, content, purpose, or use of this manual are invited. Comments may be sent on the Document Review Sheet in the back of this manual or may be addressed directly to Documentation Standards and Publications, B-90, Computer Equipment Department, General Electric Company, 13430 North Black Canyon Highway, Phoenix, Arizona 85029.

(1M 4-68)

CONTENTS

1. INTRODUCTION TO FAULT PROCESSING

Fault processing includes processing of all MME's and recovery action in all other faults. For all faults (including MME's) the following actions are immediately undertaken by the Fault Processor.

1. Registers are stored in .SREG.

2. Instruction counter and indicators (IC and I) at fault are stored in .SSTAK.

3. A 1 is subtracted from the value in .SSTAK and bit 35 is set in the IC and I entry to indicate a register store has occurred.

4. The Fault Processor takes appropriate action depending on fault type

For MME's the following actions take place.

1. The MME address is examined to see if the address is legal.

2. The MME is tested to see if it is legal.

3. If the MME is legal, a .GOTO will be made to the MME processing routines.

Each MME processing routine may be categorized as one of three general types:

1. Some of the simple MME's are executed within the Fault Processor itself and control is returned to the caller.

2. Other commonly used MME's are in hard core. These are treated as an extension of the program itself.

3. Some MME's are called into the Slave Service Area (SSA) for execution.

Since a .GOTO is executed for each MME, the same MME can be called from within GECOS by a .CALL to the same module and entry. Because of the manner in which the registers and IC and I are stored in the Fault Processor, the .SSTAK entry will be the same with respect to the return address whether the module was called by a MME or by a .CALL. The only difference is the storage of registers in .SREG in the case of the MME. The processor determines this from bit 35 of .SSTAK.

Some MME's have core-contained calling sequences. The calling sequence may be obtained by getting the IC and I value from .SSTAK. Note that if a MME has a core-contained calling sequence and it does not operate in Hard Core Monitor (HCM) then it cannot be .CALLed from the SSA because the calling sequence has been overlayed.

When a MME processor returns arguments in registers to its caller then it must interrogate bit 35 in the .SSTAK to determine if it was .CALLed or MMEed to. If it was .CALLed, then it may simply load the appropriate registers and .EXIT. The caller must restore A- and Q-registers after return from the other hand, if the MME processor was entered with a MME, then it must change the register values in .SREGS before exiting.

Macros have been defined for all MME's. All MME's begin with the characters "GE" The macros are essentially the same except the "GE" is changed to .G. For instance the macro used for GETIME is .GTIME.

A fault can occur when no program is in execution or when the system is processing an interrupt. This is a disaster and the system must die at this point. If a fault occurs at any other point, then it is possible to continue. Even if the fault is in a system routine, it may be caused by unusual conditions within the slave program itself. Thus, if a dump of the job is taken, it may be possible to debug the problem while the system itself continues to run. The following discussion outlines a strategy for each fault:

Fault Type	Action
Startup Fault	Ignore.
Execute Fault	System dies immediately; master mode dump taken.
Operation Not Complete	If no program, system dies immediately; otherwise, abort job.
Lockup	Abort the program in execution, if any; otherwise, program dies immediately (except time-sharing program).
Divide Check	If no program in execution, system dies; if fault in slave area, treat as slave fault; if fault in HCM or SSA, abort the program.
Overflow	If in slave area, treat as slave abort; otherwise, ignore.
Parity	Scan memory to find parity: 1. If none found, system dies. 2. If parity in HCM or GEPOP, system dies. 3. Otherwise, abort the program.
Command	System dies.
Memory Fault	If a program is in execution, abort the program; otherwise, system dies.
Derail	If a program is in execution but fault is not in slave area, abort; otherwise, give to program. If no program is in execution, system dies.

Fault Type	Action
Fault Tag	If a program is in execution, abort it; otherwise system dies.
Zero Op Code	If a program is in execution, abort it; otherwise system dies.
Connect	Ignore this fault. Used by the system for interprocessor communication, and it is given to blast noncontrol processors out of DIS status.
Timer Runout	Treat as forced relinquish.
Shutdown	Ignore.

2. FAULT PROCESSOR MODULES

The Fault Processor consists of two major modules:

.MFALT	Fault Processor
.MFLT1	Slave Service Area Fault Processor

FAULT PROCESSOR MODULE (.MFALT)

This module processes all processor faults and some of the simple Master Mode Entries (MME's). All slave mode entries to the Fault Processor are made through .MFALT, which transfers to .MFLT1 for the processing of certain slave service area MME's. The primary routines of .MFALT are as follows:

FPRC		Secondary Fault Vector
FLT		Recognize Fault Type and Process It
FSB	(EP1)	Take Dump for System Abort
FGAD	(EP2)	Get Address for File Code
FGCON	(EP3)	Put Information in FCB
FGFIL	(EP4)	Switch Primary and Secondary Logical Units
FGLAP	(EP5)	Provide Processor Time Prior to Request
FGRET	(EP6)	Reset Program Switch Word Bits
FGSET	(EP7)	Set Program Switch Word Bits
FGTIM	(EP8)	Provide Date and Time of Day
FGLP	(EP9)	Divide Execution Time Into Seconds
FEMM	(EP10)	Let Slave Program Enter Master Mode
FNDE	(EP11)	Find PAT From File Code
GMRL	(EP12)	Deallocate Memory
BOOT	(EP13)	Call .MDUMP at EP2
.IFALT		Fault Processor Initialization

SECONDARY FAULT VECTOR

FPRC (.MFALT) is entered from the primary fault vector when a fault occurs. The secondary fault vector contains the fault processing vectors. The secondary fault vector finds the fault type, saves the fault type, saves the registers for the faulting processor, saves the base address register, places the processor number in X7, and places the fault type in the Q-register. This action is repeated for as many times as there are processors.

PRECALLING SEQUENCE

None.

CALLING SEQUENCE

FPRC is entered from the primary fault vector.

```
STC1        FICI fault origin
TRA         FLT,$
```

OPERATING SYSTEM INTERACTION

None.

ROUTINE RETURNS

QU Constant to change fault type to a value
   of -1 to 11.
X7 Processor number
Transfer is to the FLT routine

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FPRC is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FPRC occupies approximately 24 core storage locations for each processor in the system.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FPRC, .MFALT module.

FLT
.MFALT

RECOGNIZE FAULT TYPE AND PROCESS IT

FLT (.MFALT) is entered from the secondary fault vector. This routine determines the fault type and transfers to the proper routine. If the fault is an execute fault, control is transferred to FSBM (system disaster routine). If the fault is not an execute fault, further processing is determined by whether or not a program is in execution.

If a program is not in execution, the fault is checked for a MME fault. If it is MME and not .EMM, control is transferred to FSYB. If .EMM, the master mode bit is set in the IC and I of the faulting processor. If it is not a MME fault, a check is made to determine if the fault is timer run-out, parity, or overflow. Timer runout and parity faults are processed by the appropriate routines. Overflow will be ignored. Any other fault will cause transfer to FSBM.

If a program is in execution when the fault occurs, FLT recovers the appropriate registers, and the fault code is examined for the fault type. If it is a MME fault, it is processed by FMME (MME Processing Routine). If the fault is timer runout or parity, control is transferred to the appropriate routines. If fault is other than the above, the A- and Q-registers are saved and a trace entry will be made as required. The A- and Q-registers are restored and the address of the appropriate fault processing routine is established. The routine is entered from one of the entry points - HCM, SSA, or Slave.

PRECALLING SEQUENCE

Prior to entering FLT, FICI,7 holds the IC and I at fault, FICI+4,7 holds IC and I at the secondary vector, and QU holds factor to be added to FICI+4,7 to get fault type. The .CRPRG,7 holds the program number of the program execution. FREGS contains the registers of the faulting processor. FBAR contains the BAR of the faulting processor, .CRLAL contains the LAL for the program in execution, and .STATE,5 contains the disaster bit. Location .CRTRC+2 contains the trace information.

CALLING SEQUENCE

FLT is called from FPRC (.MFALT).

```
8           16
TRA         FLT
```

OPERATING SYSTEM INTERACTION

None.

ROUTINE RETURNS

FLT processes all faults and MME GETIME, GELAPS, GESETS, and GERETS. Transfer is made to the appropriate MME processing routines for all other MME's.

Return is to the exit routine, FGLA3.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FLT is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FLT occupies approximately 340 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FLT, .MFALT module.

FSB (EP1)
.MFALT

TAKE DUMP FOR SYSTEM ABORT

FSB (EP1 of .MFALT) causes a system dump when requested by the operator from the system console.

[illegible] DUMP command (from [illegible])

PRECALLING SEQUENCE

Prior to entering FSB, .SSTAK+1,5 must contain the IC and I.

CALLING SEQUENCE

FSB is called from the .MPOP7 module.

```
       8          16
       .CALL      .MFALT,1
```

OPERATING SYSTEM INTERACTION

This routine stores the IC and I of the fault in the A-register, sets fault type equal to 13, and stores processor registers before transferring to FSYB.

No .STEMP storage is used.

No gates are used.

ROUTINE RETURNS

The A-register contains the IC and I of the faulting processor.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FSB is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FSB occupies approximately 9 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FSB (EP1), .MFALT module.

FGAD (EP2)
.MFALT

GET ADDRESS FOR FILE CODE

FGAD (EP2 of .MFALT) is used to obtain physical addresses for use in operator messages. This routine searches the slave Peripheral Assignment Table (PAT) for the requesting program to locate an entry containing the specified file code. If a match is found for the file code in the Q-register, the necessary information is gathered from the Configuration Table (SCT) and the physical file address and the relative PAT pointer are returned to the user. If no match is found in the PAT, both the A- and Q-registers contain zeros. The A-register distinguishes between an unsuccessful search and a successful one in which the IOC channel and device address are legitimately all zero.

PRECALLING SEQUENCE

Slave Mode (MME GEFADD Fault)

Prior to entering FGAD, the registers listed must contain the data indicated.

QL	File code

The .MFALT module sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FGAD Call)

Prior to a master mode program entering FGAD, the registers listed must contain the data indicated.

QL	File code
X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode

```
8           16
MME         GEFADD
Return
```

Master Mode

```
8           16
.CALL       .MFALT,2
Return
```

or

```
8           16
.CALL       .MFLT1,1
Return
```

OPERATING SYSTEM INTERACTION

Prior to entering FGAD, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. On entering FGAD, the A- and Q-registers are restored and a .GOTO .MFLT1,1 is executed. The MME is processed by FGAD (EP1 of .MFALT).

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GEFADD. Registers are not restored if the request was made by a FGAD call. Return is made to the location following the MME in FGAD call. When a match is found, the following is returned to the user:

Q	(bits 6-11)	Device Address
	(bits 12,13)	IOC
	(bits 14-17)	Channel Address
A	(bits 12-17)	PAT Pointer

If a match is not found, the A- and Q-registers contain zero.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGAD is reentrant and written in floatable (SCT) code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGAD occupies approximately 4 core storage locations.

Other Routines Used

Get Address For File Code FGAD (EP1 of .MFLT1)

Flowchart

See CPB-1500 for the flowchart of FGAD (EP2), .MFALT module.

FCON (EP3)
.MFALT

PUT INFORMATION IN FCB

FCON (EP3 of .MFALT) is provided for the exclusive use of GEFRC. It is used to place specific information in the file control block (FCB) of the file(s) identified by the Q-register as follows:

1. First and last link number of the allocated random disc or drum files are stored in word 7 of the FCB ?? Locsym - 7 ?
2. Link/random bit (provided for the disc or drum) bit 24, Locsym+0
3. SYSOUT indicator bit 25, Locsym+0
4. Type of device bits 26 - 29, Locsym+0
5. Physical device address Locsym-1, bits 18 - 29
6. File serial number (magnetic tape) Locsym-7, bits 0 - 29
7. Reel sequence number (magnetic tape) Locsym-8, bits 30 - 35
8. File present indicator Locsym-5, bit 18
9. C or D disposition indicators (magnetic tape) locsym - 5, bit 29

When additional FCB's are linked to the original FCB (via LOCSYM -1 of the FCB), the above information is supplied for all files in the chain. The chain is ended by LOCSYM -1 being zero or pointing to the beginning of the chain.

Slave prefix location 17_8 is used to point to beginning of chain open files.

PRECALLING SEQUENCE

Slave Mode (MME GEFCON Fault)

Prior to entering FCON, the registers listed must contain the data indicated.

QU Location of the first FCB

If bit 0 in Q-register contains a 1, then this is the "special" request to take the tape reel number from the FCB and insert it in the PAT.

The .MFALT module sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FCON Call)

Prior to a master mode program entering FCON, the registers listed must contain the data indicated:

QU	Location of first FCB
X5	LAL of program in execution
X6	Program number
X7	Processor number

If bit 0 in Q-register contains a 1, then this is the "special" request to take the tape reel number from the FCB and insert it in the PAT.

CALLING SEQUENCE

Slave Mode (MME GEFCON Fault)

```
8           16
MME         GEFCON
Return
```

Master Mode (.FCON Call)

```
8           16
.CALL       .MFLT2,3
Return
```

or

```
8           16
.CALL       .MFALT,2 ?
Return         3
```

~~2~~ 3 ? .MFALT, 3 per first line, page 14

OPERATING SYSTEM INTERACTION

Prior to entering FCON, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. On entry, FCON recovers the A- and Q-registers and executes a .GOTO .MFLT1,2 for processing the routine.

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was a MME GEFCON. Registers are not restored if the request was made by a FCON call. Return is made to the location following the MME or FCON call. The A- and Q-registers are restored.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FCON is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FCON occupies approximately 4 core storage locations.

Other Routines Used

Put Information in FCB FCON (EP2 of .MFLT1)

Flowchart

See CPB-1500 for the flowchart of FCON (EP3), .MFALT module.

SWITCH PRIMARY AND SECONDARY LOGICAL UNITS

FGFIL (EP4 of .MFALT) is used to switch primary and secondary logical units (physical devices). The secondary logical unit for this file code becomes the primary logical unit and, in turn, the primary logical unit becomes the secondary logical unit. If no secondary unit has been assigned, the peripheral device assigned to the primary logical unit is assumed. The physical device address of the resultant primary logical unit is returned in the Q-register. This entry is for magnetic tapes only.

PRECALLING SEQUENCE

Slave Mode (MME GEFILS Fault)

Prior to a slave mode program entering FGFIL, the registers listed below must contain the data indicated.

QL	File Code

The .MFALT module sets registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FGFIL call)

Prior to a master mode program entering FGFIL, the registers listed must contain the data indicated:

QL	File code
X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode (MME GEFILS Fault)

```
8          16
MME        GEFILS
Return
```

Master Mode (.FGFIL Call)

```
8          16
.CALL      .MFLT1,3
Return
```

or

```
8          16
.CALL      .MFALT,4
Return
```

FGFIL (EP4)
.MFALT

OPERATING SYSTEM INTERACTION

Prior to entering FGFIL, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, FGFIL recovers the A- and Q-registers and executes a .GOTO MFLT1,3 for processing the routine.

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GEFILS. Registers are not restored if the request was an FGFIL call. The Q-register returns the following:

Bits	6-11	Device address
	12-13	IOC address
	14-17	Channel address

new primary device addre

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGFIL is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FGFIL occupies approximately 4 core storage locations.

Other Routines Used

Switch Primary and Secondary Logical Units FLSW (EP3 of .MFLT1)

Flowchart

See CPB-1500 for the flowchart of FGFIL (EP4), .MFALT module.

FGLAP (EP5)
.MFALT

PROVIDE PROCESSOR TIME PRIOR TO REQUEST

FGLAP (EP5 of .MFALT) provides the requesting program with the total amount of processor time it has expended up to the time of the request.

PRECALLING SEQUENCE

Slave Mode (MME GELAPS Fault)

Prior to a slave mode program entering FGLAP, .MFALT sets registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FGLAP Call)

Prior to a master mode program entering FGLAP, the registers listed must contain the data indicated:

X5	LAL of program
X6	Program number of program in execution
X7	Processor number

CALLING SEQUENCE

Slave Mode (MME GELAPS Fault)

```
8           16
MME         GELAPS
Return
```

Master Mode (.FGLAP Call)

```
8           16
.CALL       .MFALT,5
Return
```

OPERATING SYSTEM INTERACTION

Prior to entry into FGLAP, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, the A-register is restored.

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GELAPS. Registers are not restored if the request was a .FGLAP call. FGLAP adds the amount of processor time used by the program (.SPRT) to the difference between the current timer setting (.CRCCK) and the last timer setting (.CRLCK). The elapsed time is returned, right justified, in the Q-register to the requesting program. Return is to the location following the MME or FGLAP call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGLAP is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGLAP occupies approximately 19 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FGLAP (EP5), .MFALT module.

RESET PROGRAM SWITCH WORD BITS

FGRET (EP6 of .MFALT) is used to reset bits in the Program Switch Word according to corresponding bits in the Q-register. For each bit position of the Q-register that contains a 1, the corresponding bit of the Program Switch Word is set to zero. For each bit position of the Q-register that contains a zero, the corresponding bit of the Program Switch Word is not changed. The resultant setting of the Program Switch Word is returned in the Q-register.

PRECALLING SEQUENCE

Slave Mode (MME GERETS Fault)

Prior to a slave mode program entering FGRET, the registers listed below must contain the values indicated.

QR 36-bit mask for the Program Switch Word setting

The .MFALT module saves slave registers and sets registers listed below to the value listed.

X5 LAL of program
X6 Program number
X7 Processor number

Master Mode (.FGRET Call)

Prior to a master mode program entering FGRETS, the registers listed must contain the data indicated.

QR 36-bit mask for the Program Switch Word
X5 LAL of program
X6 Program number
X7 Processor number

CALLING SEQUENCE

Slave Mode (MME GERETS Fault)

```
8       16
MME     GERETS
Return
```

Master Mode (.FGRET Call)

```
8       16
.CALL   .MFALT,6
Return
```

OPERATING SYSTEM INTERACTION

Prior to entry into FGRET, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, the A- and Q-registers are restored.

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GERETS. Registers are not restored if the request was a FGRET call. The resultant Program Switch Word is returned in the Q-register. Return is to the location following the MME or FGRET call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGRET is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGRET occupies approximately 4 core storage locations.

Other Routines Used

Provide Processor Time Prior to Request FGLAP (EP5 of .MFALT).space Specifically, the subroutine FGLA3 in FGLAP tests the register-stored bit of the IC and I and, if set, restores the A- and Q-registers and then does a .EXIT. If the bit is not set, it merely does a .EXIT.

Flowchart

See CPB-1500 for the flowchart of FGRET (EP6), .MFALT module.

SET PROGRAM SWITCH WORD BITS

FGSET (EP7 of .MFALT) is used to set bits in the Program Switch Word according to corresponding bits in the Q-register. For each corresponding bit in the Q-register that contains a 1, the corresponding bit in the switch word is set to a 1. For each corresponding bit in the Q-register that contains a zero, the corresponding bit in the switch word is unchanged. The resultant switch word is returned in the Q-register.

PRECALLING SEQUENCE

Slave Mode (MME GESETS Fault)

Prior to a slave program entering FGSET, the registers listed below must contain the data indicated.

QR	36-bit mask for setting the Program Switch Word

The .MFALT module sets registers listed below to the value indicated and saves slave registers.

X5	LAL for program
X6	Program number
X7	Processor number

Master Mode (.FGSET Call)

Prior to a master mode program entering FGSET, the registers listed below must contain the data indicated:

QR	36-bit mask for setting the Program Switch Word
X5	LAL of program
X6	Processor number
X7	Processor number

CALLING SEQUENCE

Slave Mode (MME GESETS Fault)

```
8        16
MME      GESETS
Return
```

Master Mode (.FGSET Call)

```
8        16
.CALL    .MFALT,7
Return
```

OPERATING SYSTEM INTERACTION

Prior to entering FGSET, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. Upon entry, the registers are restored.

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GESETS. Registers are not restored if the request was made by a FGSET call. The Q-register returns the resultant Program Switch Word. Return is made to the location following the MME or FGSET call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGSET is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGSET occupies approximately 4 core storage locations.

Other Routines Used

Provide Processor Time Prior to Request FGLAP (EP5 of .MFALT)

Specifically, the subroutine FGLA3 in FGLAP tests the register stored bit in the IC and I. If set, the A- and Q-registers are restored and .EXIT is done. If not set, then only .EXIT is done.

Flowchart

See CPB-1500 for the flowchart of FGSET (EP7), .MFALT module.

FGTIM (EP8)
.MFALT

PROVIDE DATE AND TIME OF DAY

FGTIM (EP8 of .MFALT) provides the date and time of day to the requesting program.

PRECALLING SEQUENCE

Slave Mode (MME GETIME Fault)

Prior to a slave mode program entering FGTIM, .MFALT sets registers listed below to the value indicated and saves slave registers:

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FGTIM Call)

Prior to a master mode program entering FGTIM, the registers listed must contain the data indicated:

X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode (MME GETIME Fault)

```
8          16
MME        GETIME
Return
```

Master Mode (.FGTIM Call)

```
8          16
.CALL      .MFALT,8
Return
```

OPERATING SYSTEM INTERACTION

The date and time (in time pulses past midnight) are loaded into the A- and Q-registers .CRDAT. Time since the last "boot" is added to the time-of-day. The routine transfers to a subroutine (FGLA1) in FGLAP (EP5 of .MFALT) where this total is added to the difference between the current timer setting (.CRCCK) and the last timer setting (.CRLCK).

No gates are used.

ROUTINE RETURNS

Return is made to the requesting program with the date (in BCD) in the A-register and the time of day right-justified in the Q-register. Registers are restored if the request was made by a MME GETIME. Registers are not restored if the request was made by a FGTIM call. Return is to the location following the MME or FGTIM call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGTIM is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FGTIM occupies approximately 3 core storage locations.

Other Routines Used

Provide Processor Time Prior to Request FGLAP (EP5 of .MFALT)

Specifically, the subroutines FGLA1 and FGLA3 in FGLAP, MFALT are used. See Operating System Interface for the use of FGLA1. FGLA3 is used to test the register-stored bit in the IC and I. If set, the A- and Q-registers are restored and .EXIT is done. If not set, then only a .EXIT is done.

Flowchart

See CPB-1500 for the flowchart of FGTIM (EP8), .MFALT module.

FGLP (EP9)
.MFALT

DIVIDE EXECUTION TIME INTO SECONDS

FGLP (EP9 of .MFALT) provides slave program loop protection by dividing the execution time limits into segments.

PRECALLING SEQUENCE

Slave Mode (MME GELOOP Fault)

Prior to a slave mode program entering FGLP, the registers listed below must contain the data indicated.

QR	Desired loop time (in seconds), right-justified

The .MFALT module sets registers listed below to the value indicated and saves slave registers.

X5	LAL for program
X6	Program number
X7	Processor number

Master Mode (.FGLP Call)

Prior to a master mode program entering FGLP, the registers listed must contain the data indicated.

QR	Requested loop time (in seconds), right-justified
X5	LAL for program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode (MME GELOOP Fault)

```
8                 16
MME              GELOOP
Return
```

Master Mode (.FGLP Call)

```
8                 16
.CALL            .MFALT,9
Return
```

or

```
8                 16
.CALL            .MFLT1,5
Return
```

Desired time interval should not be zero, and may not exceed the remaining processor use time.

To cancel the time interval, a value representing −1 (Q all ones must be given to GELOOP, else the program will abort when the selected interval runs out.

FGLP (EP9)
.MFALT

OPERATING SYSTEM INTERACTION

Prior to entry into FGLP, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. Upon entry, the A- and Q-registers are restored, and a .GOTO MFLT1,5 is executed for processing the routine.

No gates are used.

ROUTINE RETURNS

None.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGLP is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGLP occupies approximately 4 core storage locations.

Other Routines Used

Divide Execution Time Into Seconds FGLP (EP5 of .MFLT1)

Flowchart

See CPB-1500 for the flowchart of FGLP (EP9), .MFALT module.

LET SLAVE PROGRAM ENTER MASTER MODE

FEMM (EP10 of .MFALT) is restricted to GECOS system functions. A user attempt to execute this MME results in an abort of the user program. Slave programs that allow this MME are termed privileged slaves, [illegible]

PRECALLING SEQUENCE

Prior to entering FEMM, the registers listed must contain the data indicated.

X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

FEMM may be called by any master mode or privileged slave programs.

```
8          16
MME        .EMM
Return
```

OPERATING SYSTEM INTERACTION

This routine first examines bit 18 of the .STATE word to determine if a MME .EMM is allowed. If bit 18 is off, the MME is not allowed. The abort code is set to 23 (octal) in the Q-register, and a .GOTO .MBRT1,3 is executed.

If the MME is allowed (bit 18 of .STATE is on), the IC and I is recovered from .SSTAK and the I portion is examined to determine if the program is in master mode. If bit 28 is on (program in master mode), the IC and I is put back into .SSTAK and a .EXIT is executed.

If bit 28 of the IC and I is off (program in slave mode), bit 28 is turned on and the LAL is added to the IC portion of the IC and I. The IC and I is put back in .SSTAK and a .EXIT is executed.

No .STEMP storage is used.

No gates are used.

ROUTINE RETURNS

Return is to the location following the MME .EMM.

How long can the program be in master mode?

POSTCALLING SEQUENCE

None.

How does the program exit from master mode operation?

SUPPORTING INFORMATION

Programming Method

FEMM is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FEMM occupies approximately 15 core storage locations.

Other Routines Used

None.

Flowcharts

See CPB-1500 for the flowchart of FEMM (EP10), .MFALT module.

FIND PAT FROM FILE CODE

FNDE (EP11 of .MFALT) finds Peripheral Assignment Table(s) (PAT) from the file code supplied.

PRECALLING SEQUENCE

Prior to entering FNDE, the registers listed must contain the data indicated.

QR File code

CALLING SEQUENCE

FNDE is called from any module requiring this service.

```
8          16
.CALL      .MFALT,11
FLD        0,DU            No hits found
.EXIT
```

or

```
8          16
.CALL      .MFALT,11
.EXIT      1               Hit found
```

OPERATING SYSTEM INTERACTION

Prior to entering FNDE, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. Upon entry, these registers are restored.

No gates are used.

ROUTINE RETURNS

If a file is found, X4 contains the offset to a pointer word and the return skips one instruction. If a file is not found, the A-register is set to zero and return is to the location following the call.

X0 and X1 are destroyed by this routine.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FNDE is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FNDE occupies approximately 15 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FNDE (EP11), .MFALT module.

DEALLOCATE MEMORY

GMRL (EP12 of .MFALT) is used to deallocate a specified amount of memory from a requesting program's total assigned memory. Actual memory is deallocated in multiples of 1024-word blocks.

PRECALLING SEQUENCE

Slave Mode (MME GEMREL Fault)

Prior to a slave mode program entering GMRL, the registers listed below must contain the information indicated.

AU	Return address
QU	Number of words of upper memory
QL	Number of words of lower memory

The .MFALT module saves slave registers and sets the following registers to the value indicated.

X5	LAL for program
X6	Program number
X7	Processor number

Master Mode (.GMRL Call)

Prior to a master mode program entering .GMRL, the registers listed must contain the value indicated.

AU	Return address
QU	Number of words of upper memory
QL	Number of words of lower memory
X5	LAL for program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode

```
8         16

MME       GEMREL
```

Return is made to the address specified in the A-register.

Master Mode

```
8         16
.CALL     .MFALT,12
```

or

```
8         16
.CALL     .MFLT1,4
```

Return is made to the address specified in the A-register.

GMRL (EP12)
.MFALT

OPERATING SYSTEM INTERACTION

Prior to entering GMRL, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, the registers are restored, and a .GOTO .MFLT1,4 is executed for processing the routine.

ROUTINE RETURNS

None.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

GMRL is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

GMRL occupies approximately 4 core storage locations.

Other Routines Used

Deallocate Memory GMRL (EP4 of .MFLT1)

Flowchart

See CPB-1500 for the flowchart of GMRL (EP12), .MFALT module.

CALL .MDUMP AT EP2

BOOT (EP13 of .MFALT) stops all processors and takes a dump when a system disaster occurs. The processors are all put in DIS, their registers stored in FREGS, their IC and I in FICI. The reason code for the abort is found in FICI4 at the processor number entry for the processor found in FSYB2. After the processors have been stopped, the control processor waits 30 seconds and writes a portion of memory onto the system storage device. It then reads the module .MFLT1 from GECOS. This module surveys the System Configuration Table and takes a dump of memory.

Operator Proceed to BOOT
via Console

PRECALLING SEQUENCE

None.

CALLING SEQUENCE

BOOT is called from the .MPOP7 module

```
8              16
.CALL          .MFALT,13
.GOTO          .MDUMP,1
```

or from FLT (.MFALT) or FSB (EP1 of .MFALT)

```
8              16
TRA            FSYB
.GOTO          .MDUMP,1
```

OPERATING SYSTEM INTERACTION

This routine is used when a memory dump is requested. It shuts .CRGAT to allow only one processor to fault out. Transfer is to .MDUMP,1 for the dump routine.

ROUTINE RETURNS

None.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

BOOT is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

BOOT occupies approximately 55 core storage locations.

Other Routines Used

System Dump SDMP (EP1 of .MDUMP)

Flowchart

See CPB-1500 for the flowchart of BOOT (EP13), .MFALT module.

FAULT PROCESSOR INITIALIZATION

Startup gives control to this routine after it has been loaded. .IFALT (.MFALT) is responsible for "absolutizing" coding in the body of the program and in the fault vector images.

The primary fault vector images are then moved to an area designated by startup.

The initialization routine then returns an address (dependent upon the number of processors) where startup can load the next module.

PRECALLING SEQUENCE

X3 Location startup wants Fault Processor to put primary fault vector images for n processors, where n = 1, 2, 3, or 4.

CALLING SEQUENCE

```
8            16
TSX1         .IFALT
TRA          0,1         return to startup
```

OPERATING SYSTEM INTERACTION

None.

ROUTINE RETURNS

On return to startup, the following registers contain the information indicated:

AR Zero, indicating no alternate load list
QU Address where next module can be loaded

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Information

.IFALT is reentrant and written in floatable code.

Interrupts are inhibited.

.IFALT
.MFALT

Storage

After initialization and return to startup, the memory occupied by .IFALT is overlayed with other modules.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of .IFALT, .MFALT module.

SLAVE SERVICE AREA FAULT PROCESSOR MODULE (.MFLT1)

This Slave Service Area module contains the following routines:

FGAD	(EP1)	Get Address For File Code
FCON	(EP2)	Put Information in FCB
FLSW	(EP3)	Switch Primary And Secondary Logical Units
GMRL	(EP4)	Deallocate Memory
FGLP	(EP5)	Divide Execution Time Into Seconds
FPRIO	(EP6)	Give I/O Action Priority
FLBAR	(EP7)	Reset BAR to Smaller Area
FWAKE	(EP8)	Allow a Program to Delay
GMRLM	(EP9)	Memory Release For Time-Sharing

FGAD (EP1)
.MFLT1

GET ADDRESS FOR FILE CODE

FGAD (EP1 of .MFLT1) is used to obtain physical addresses for use in operator messages. This routine searches the slave Peripheral Assignment Table (PAT) for the requesting program, looking for an entry containing the specified file code.

If a match is found for the file code supplied in the Q-register, the necessary information is gathered from the SCT and the physical file address in bits 6-17 of the Q-register and the relative PAT pointer in bits 12-17 of the A-register are returned to the user.

If no match was found in the PAT, both A-register and Q-register contain zero. The A-register distinguishes between an unsuccessful search and a successful one in which the IOC channel and device address are legitimately all zero.

PRECALLING SEQUENCE

Slave Mode (MME GEFADD Fault)

Prior to entering FGAD, the registers listed below must contain the data indicated.

QL	File code

The .MFALT module sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FGAD Call)

Prior to a master mode program entering FGAD, the registers listed must contain the data indicated.

QL	File code
X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode

```
8           16
MME         GEFADD
Return
```

Master Mode

```
8         16
.CALL     .MFLT1,1
Return
```

or

```
8         16
.CALL     .MFALT,2
Return
```

OPERATING SYSTEM INTERACTION

Prior to entering FGAD, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, .STEMP+9 is recovered in the A-register and a TSX1 FNDE is executed to find a match for the file code. If no hit is found, a TRA to the .EXIT routine FGLA3 is executed. If a hit is found, AU will equal the offset to the PAT pointer and X4 is the pointer to the body of the PAT. The .CRSCT gate is opened and the routine transfers to the exit routine.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GEFADD. Registers are not restored if the request was made by a FGAD call. Return is made to the location following the MME or FGAD call. When a match is found, the following is returned to the user:

QR	(bits 6-11)	Device address
	(bits 12-13)	IOC
	(bits 14-17)	Channel address
AR	(bits 12-17)	PAT pointer

Return is to the location following the MME or call.

When no match is found, the A- and Q-registers contain zeros.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGAD is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGAD occupies approximately 60 core storage locations.

FGAD (EP1)
.MFLT1

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FGAD (EP1), .MFLT1 module.

PUT INFORMATION IN FCB

FCON (EP2 of .MFLT1) is provided for the exclusive use of GEFRC. It is used to place specific information in the file control block (FCB) of the file(s) identified by the Q-register as follows:

- First and last link numbers of the allocated random disc or drum files are stored in word 7 of the FCB.
- Link/random bit (provided for disc or drum)
- SYSOUT indicator
- Type of device
- Physical device address
- File serial number (when file is magnetic tape)
- Reel sequence number (when file is magnetic tape)
- File present indicator
- C or D disposition indicators (for magnetic tape)

When additional FCB's are linked to the original FCB (via LOCSYM -1 of the FCB), the above information is supplied for all files in the chain. The chain is ended by LOCSYM -1 being zero or pointing to the beginning of the chain.

There is also a "special" entry for obtaining a tape reel number from the FCB and inserting it into the PAT. This will only be done for the original FCB.

PRECALLING SEQUENCE

Slave Mode (MME GEFCON Fault)

Prior to entering FCON, the registers listed must contain the data indicated.

QU	Location of the first FCB

If bit 0 of the Q-register contains a 1, then this is the "special" request to take the tape reel number from the FCB and insert it in the PAT.

The .MFALT module sets slave registers listed below to the value listed and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FCON Call)

Prior to a master mode program entering FCON, the registers listed must contain the data indicated.

QU	Location of first FCB
X5	LAL of program
X6	Program number
X7	Processor number

If bit 0 of the Q-register contains a 1, then this is the "special" request to take the tape reel number from the FCB and insert it in the PAT.

FCON (EP2)
.MFLT1

CALLING SEQUENCE

Slave Mode

```
8                16
MME              GEFCON
Return
```

Master Mode

```
8                16
.CALL            .MFLT1,2
Return
```

or

```
8                16
.CALL            .MFALT,3
Return
```

OPERATING SYSTEM INTERACTION

Prior to entering FCON, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, the relative BAR setting (.SALIM) is added to a value to obtain the highest relative address allowed. This value is stored in .STEMP+5. The relative address of the first FCB is recovered from .STEMP+9.

No gates are used.

ROUTINE RETURNS

Return is made to the location following the MME or call. The A- and Q-registers are restored.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FCON is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FCON occupies approximately 147 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FCON (EP2), .MFLT1 module.

FLSW (EP3)
.MFLT1

SWITCH PRIMARY AND SECONDARY LOGICAL UNITS

FLSW (EP3 of .MFLT1) is used to switch primary and secondary logical units (physical devices). The secondary logical unit for this file code becomes the primary logical unit and, in turn, the primary logical unit becomes the secondary logical unit. If no secondary unit has been assigned, the peripheral device assigned to the primary logical unit is assumed. The physical device address of the resultant primary logical unit is returned in the Q-register. This entry is for magnetic tapes only.

PRECALLING SEQUENCE

Slave Mode (MME GEFILS Fault)

Prior to a slave mode program entering FLSW, the registers listed must contain the data indicated.

QL	File code

The .MFALT module sets registers listed below to the value listed and saves the slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FLSW Call)

Prior to a master mode program entering FLSW, the registers listed must contain the data indicated.

QL	File code
X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode

```
8          16
MME        GEFILS
Return
```

Master Mode

```
8          16
.CALL      .MFLT1,3
Return
```

or

```
8          16
.CALL      .MFALT,4
Return
```

OPERATING SYSTEM INTERACTION

Prior to entry into FLSW, the A- and Q-register are stored in .STEMP+8 and .STEMP+9. After entry, the Q-register is restored. The routine uses the FNDE routine to find a PAT pointer for a file code, and FGLA3 for the exit routine when a PAT is not found. If a PAT is found, the routine switches the designators and transfers to FGAD5 subroutine of FGAD (EP1 of .MFLT1) to get a file address for the file code.

The .CRSCT gate is shut while FGAD5 is formatting the physical device address.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GEFILS. Registers are not restored if the request was made by a FLSW call. The Q-register returns the following:

bits	6-11	Device address
	12-13	IOC address
	14-17	Channel address

Return is made to the location following the MME or FLSW call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FLSW is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FLSW occupies approximately 20 core storage locations.

Other Routines Used

Get Address For File Code FGAD (EP1 of .MFLT1)

Specifically, FGLA3 the subroutine in FGAD is used to test the register stored bit of the IC and I. If the bit is set, and A- and Q-registers are restored. If the bit is not set, a .EXIT is executed. The subroutine FNDE in FGAD is used to find file address for the file code, and the subroutine FGAD5 in FGAD is used to format the physical device address.

Flowchart

See CPB-1500 for the flowchart of FLSW (EP3), .MFLT1 module.

GMRL (EP4)
.MFLT1

DEALLOCATE MEMORY

GMRL (EP4 of .MFLT1) is used to deallocate a specified amount of memory from the total assigned memory of a requesting program. Actual memory is deallocated in multiples of 1024-word blocks.

Slave Mode (MME GEMREL Fault)

Prior to a slave mode program entering GMRL, the registers listed must contain the data indicated.

AU	Return address
QU	Number of words of upper memory
QL	Number of words of lower memory

The .MFALT module sets the registers listed below to the values indicated, and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.GMRL Call)

Prior to a master mode program entering GMRL, the registers listed must contain the value indicated.

AU	Return address
QU	Number of words of upper memory
QL	Number of words of lower memory
X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode

```
8              16
MME            GEMREL
Return
```

Master Mode

```
8              16
.CALL          .MFLT1,4
Return
```

or

```
8              16
.CALL          .MFALT,12
Return
```

OPERATING SYSTEM INTERACTION

Prior to entering GMRL, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry these registers are restored. This routine is roadblocked until all outstanding I/O for the program has been stored.

Gate .CRPOQ is shut during memory release.

ROUTINE RETURNS

If there is no memory to release, then a .EXIT is executed. Return is made to the location specified in the A-register.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

GMRL is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

GMRL occupies approximately 80 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of GMRL (EP4), .MFLT1 module.

FGLP (EP5)
.MFLT1

DIVIDE EXECUTION TIME INTO SECONDS

FGLP (EP5 of .MFLT1) provides slave program loop protection by dividing the execution time limits into segments.

PRECALLING SEQUENCE

Slave Mode (MME GELOOP Fault)

Prior to a slave mode program entering FGLP, the registers listed must contain the data indicated.

QR	Requested loop time (in seconds), right-justified

The .MFALT module sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FGLP Call)

Prior to a master mode program entering FGLP, the registers listed must contain the data indicated.

X5	LAL of program
X6	Program number
X7	Processor number
QR	Requested loop time (in seconds), right-justified

CALLING SEQUENCE

Slave Mode

```
8              16
MME            GELOOP
Return
```

Master Mode

```
8              16
.CALL          .MFLT1,5
Return
```

or

```
8              16
.CALL          .MFALT,9
Return
```

OPERATING SYSTEM INTERACTION

Prior to entering FGLP, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. .SALT contains the total time left after the MME GELOOP.

After entry, the Q-register is restored.

No gates are used.

Once used to specify a part of the total time limits, the MME must continue to be used for the remaining total time limit or an I8 abort is experienced. The user may specify an infinite time or a succeeding MME(s). In this case, the total limits are used.

ROUTINE RETURNS

Return is made to the location following the MME or call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FGLP is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FGLP occupies approximately 15 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FGLP (EP5), .MFLT1 module.

FPRIO (EP6)
.MFLT1

GIVE I/O ACTION PRIORITY

FPRIO (EP6 of .MFLT1) assigns an I/O action priority to tape files.

PRECALLING SEQUENCE

Slave Mode (MME GEPRIO Fault)

Prior to a slave mode program entering FPRIO, the registers listed must contain the data indicated.

QR	File code (right adjusted)

.MFALT sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (.FPRIO Call)

Prior to a master mode program entering FPRIO, the registers listed must contain the data indicated.

QR	File code
X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

Slave Mode

```
8         16
MME       GEPRIO
Return
```

Master Mode

```
8         16
.CALL     .MFLT1,6
Return
```

OPERATING SYSTEM INTERACTION

Prior to entry into FPRIO, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, the registers are restored.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GEPRIO. Registers are not restored if the request was a .FPRIO call. Return is made to the location following the MME or .FPRIO call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FPRIO is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FPRIO occupies approximately 22 core storage locations.

Other Routines Used

Find PAT From File Code FNDE (EP11 of .MFALT)

Find PAT From File Code FGAD (EP1 of .MFLT1)

Specifically, the subroutine FGLA3 in FGAD is used to test the register-stored bit in the IC and I. If the bit is set, the A- and Q-registers are restored. If the bit is not set, a .EXIT is executed.

Flowchart

See CPB-1500 for the flowchart of FPRIO (EP6), .MFLT1 module.

FLBAR (EP7)
.MFLT1

RESET BAR TO SMALLER AREA

FLBAR (EP7 of .MFLT1) allows the normal slave mode program to reset the base address register to a smaller area within his allocated area.

PRECALLING SEQUENCE

Slave Mode (MME GELBAR Fault)

Prior to a slave mode program entering FLBAR, the Q-register must contain the increment of processor time before interrupt. This increment must be less than the amount of time remaining before the timer runs out. The time was assigned when the slave was allocated. If the increment is greater, its value is set to the remaining allocated time.

The A-register must contain a pointer to a pair of locations that contain the following: Allocal AL ??

LOCA	BAR	LOCB
LOCA+1	IC	I

where BAR is the desired base address relative to slave zero.

LOCB points to the location where registers have been stored prior to executing MME GELBAR. It is a modulo 8 BSS area of 8 words.

IC and I is the instruction counter relative to the new base where ~~that~~ execution is to be continued and the indicator setting desired.

When any fault, timer runout, or return after having serviced any system interrupt occurs, GECOS returns to word 23 (octal) of the slave program. If word 23 (octal) is zero, the program is terminated.

The information stored in the slave program prefix must be in the following format.

? Why this ??

Word 21 (octal)	Time left in ms/64
Word 22 (octal)	IC and I
Word 23 (octal)	Zero or a transfer to some location

Word 31 (octal) of the slave program prefix, the accumulated fault status, will contain the following information.

bits 0-17 Relative BAR setting used in MME GELBAR.

bit 19 Set if the reason was a fault.

returned info

CPB-1493

bit 20 Will be set to zero if the reason for suspending the user program was an I/O interrupt.

bits 32-35 Will indicate the fault type if a fault occurs.

0	MME
1	Memory
2	Fault Tag
3	Command
4	Derail
5	Lockup
6	Zero operation
7	Operation not complete
8	Overflow
9	Divide check
10	Timer runout

Prior to entry into FLBAR, MFALT sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

Master Mode (FLBAR Call)

Prior to a master mode program entering FLBAR, the same registers listed under slave mode must contain the data indicated.

CALLING SEQUENCE

Slave Mode

```
8       16
SREG    LOCB
MME     GELBAR
Return
```

Master Mode

```
8       16
.CALL   .MFLT1,7
Return
```

OPERATING SYSTEM INTERACTION

Prior to entry into FLBAR, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry the registers are restored.

No gates are used.

ROUTINE RETURNS

Registers are restored if the request was made by a MME GELBAR. Registers are not restored if the request was a .FLBAR call. Return is made to the location relative to the new base that is specified in the calling sequence.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FLBAR is reentrant and written in floatable code.

Interrupts are not inhibited.

Storage

No internal temporary storage is used.

FLBAR occupies approximately 37 core storage locations.

Other Routines Used

Put Information In FCB FCON (EP2 of .MFLT1)

Specifically, the subroutine FGLA3 in FCON examines the register-stored bit in the IC and I to determine the register-stored status. If set, the A- and Q-registers are stored. If not set, then a .EXIT is executed.

Flowchart

See CPB-1500 for the flowchart of FLBAR (EP7), .MFLT1 module.

ALLOW A PROGRAM TO DELAY

FWAKE (EP8 of .MFLT1) allows a normal slave program to delay for some interval before it becomes a candidate again for processor time.

PRECALLING SEQUENCE

Slave Mode (MME GEWAKE Fault)

Prior to a slave program entering FWAKE, the registers listed must contain the data indicated.

QR Time interval (in 1/64 ms) after which program is to be awakened.

The .MFALT module sets the registers listed below to the value indicated and saves slave registers.

X5 LAL of program
X6 Program number
X7 Processor number

Master Mode (.FWAKE Call)

Prior to a master mode program entering FWAKE, the registers listed must contain the data indicated.

QR Time interval (in 1/64 ms) after which this program is to be awakened.
X5 LAL of program
X6 Program number
X7 Processor number

CALLING SEQUENCE

Slave Mode

```
8          16
MME        GEWAKE
Return
```

Master Mode

```
8          16
.CALL      .MFLT1,8
Return
```

OPERATING SYSTEM INTERACTION

Prior to entry into FWAKE, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, these registers are restored. After the time interval has been loaded, the routine does a .CALL .MDISP,13. When control is returned to FWAKE, the A- and Q-registers are recovered from .STEMP+8 and .STEMP+9. The urgency level of the program is changed to zero, and a .GOTO MDISP,1 is executed.

ROUTINE RETURNS

The program will gain control at the cell following the MME or call.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

FWAKE is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

FWAKE occupies approximately 21 core storage locations.

Other Routines Used

None.

Flowchart

See CPB-1500 for the flowchart of FWAKE (EP8), .MFLT1 module.

MEMORY RELEASE FOR TIME-SHARING

GMRLM (EP9 of .MFLT1) is a special routine for time-sharing used to deallocate memory. It functions the same as GMRL, except that the routine is not roadblocked. It is entered from the MME .EMM.

PRECALLING SEQUENCE

Prior to the time-sharing program entering GMRLM, the registers listed must contain the data indicated.

AR	Return address
QU	Number of words of upper memory
QL	Number of words of lower memory

The .MFALT module sets the registers listed below to the value indicated and saves slave registers.

X5	LAL of program
X6	Program number
X7	Processor number

CALLING SEQUENCE

```
8       16
MME     .EMM
.CALL   .MFLT1,9
```

Return is made to the address specified in the A-register.

OPERATING SYSTEM INTERACTION

Prior to entry into GMRLM, the A- and Q-registers are stored in .STEMP+8 and .STEMP+9. After entry, these registers are restored and the routine transfers to GMRL8 of GMRL (EP4), MFLT1.

ROUTINE RETURNS

Registers are not restored.

Return is made to the address specified in the A-register.

POSTCALLING SEQUENCE

None.

SUPPORTING INFORMATION

Programming Method

GMRLM is reentrant and written in floatable code.

Interrupts are inhibited.

Storage

No internal temporary storage is used.

GMRLM occupies approximately 80 core storage locations.

Other Routines Used

Deallocate Memory GMRL (EP4 of .MFLT1)

Flowchart

See CPB-1500 for the flowchart of GMRLM (EP9), .MFLT1 module.

INDEX

DOCUMENT REVIEW SHEET

TITLE: GE-625/635 GECOS-III Fault Processing & Service MME's

CPB #: 1493

FROM:

Name: ____________________

Position: ____________________

Address: ____________________

____

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS: ____________________

Please cut along this line

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on two lines shown on reverse
side, staple, and mail.

STAPLE STAPLE

FOLD

FIRST CLASS
PERMIT, No. 4332
PHOENIX, ARIZONA

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

GENERAL ELECTRIC COMPANY
COMPUTER EQUIPMENT DEPARTMENT
13430 NORTH BLACK CANYON HIGHWAY
PHOENIX, ARIZONA - 85029

ATTENTION: DOCUMENTATION STANDARDS AND PUBLICATIONS B-90

FOLD

INFORMATION SYSTEMS

GENERAL ELECTRIC

LITHO U.S.A.