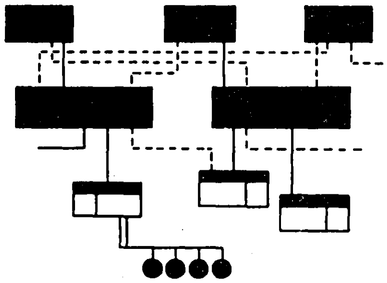
GE-625/635 GECOS-III Introduction and System Tables

Information Systems

Information Systems Equipment

SOFTWARE MAINTENANCE DOCUMENT



GENERAL ELECTRIC

CPB-1488

CPB-1488

GE-625 635/GECOS-III Introduction and System Tables

SOFTWARE MAINTENANCE DOCUMENT

May 1968

INFORMATION SYSTEMS

GENERAL ELECTRIC

PREFACE

This manual introduces the GE-625/635 Comprehensive Operating Supervisor (GECOS-III). It includes a description of the organization of GECOS-III and the system tables.

Software maintenance documents are as follows:

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 Fault Processing and Service MME's, CPB-1493

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

GE-625/635 GECOS-III Termination and System Output, CPB-1469

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, B-90, Processor Equipment Department, General Electric Company, 13430 North Black Canyon Highway, Phoenix, Arizona 85029.

(1M 5-68)

CONTENTS

ILLUSTRATIONS

Page

1. GECOS-III GENERAL DESCRIPTION

The GE-625/635 Comprehensive Operating Supervisor (GECOS-III) is a powerful operating system for GE-625/635 large-scale computer systems. It efficiently integrates the user's requirements for on-line batch processing, remote batch, and time-sharing into one system, using a common data base. Also, through GECOS-III, large-scale computer users can avoid the problems of multiple systems with incompatible programs and files.

The GECOS-III centralized file system is organized around a permanent on-line data base, with security protection, which may be referenced and modified by users.

Any file in the system may be read, written, appended to, or executed. However, the owner of the file may specify who may have each of these four types of permissions. Interrogators may use only those files for which they have permission, and only in the manner for which they have permission.

Files may be accessed in the batch processing, remote batch, and/or time-sharing modes.

MULTIPROGRAMMING, MULTIPROCESSING

Jobs are entered in GECOS-III from on-line peripherals, remote terminals, and/or time-sharing terminals. The GECOS-III system can handle simultaneous operations from all three sources through multiprogramming and multiprocessing.

Multiprogramming is the concurrent operation of many jobs to gain greater throughput. While jobs or programs are entered in the system and then examined for processing requirements, others may be undergoing peripheral and core storage allocation. As many jobs as system resources will permit can be in various stages of execution. Others can be undergoing termination procedures and still others having their final results produced.

At the same time, additional jobs can be queued in storage awaiting their turn. High-priority jobs in the queue can be expedited through a job swap (i.e., temporarily removing from core storage a job in execution).

The throughput can be further increased through multiprocessing, joining two or more processors in the system. All processors have direct access to all core storage; each can act on any job, performing computations and requesting input/ output operations.

BATCH PROCESSING AND REMOTE BATCH

Efficient batch processing is a basic function of GECOS-III. Remote input/output capabilities can be added to the system by including one or more DATANET-30* communications processors in the configuration. Compact GE-115 computers enter remote batch programs and data into the GE-625/635. The GE-115 also prints or punches the remote program output at high speed. Any batch job that can be entered at the central system can be entered via remote processing.

TIME-SHARING

GECOS-III is designed for installations needing time-sharing without disruption of their batch-processing commitments. The hardware dedicated to time-sharing is dynamically variable, offering an operating spectrum from total time-sharing through total batch processing.

General Electric time-sharing features BASIC, a conversational language developed at Dartmouth College. A beginner can use BASIC after only a few hours of study, yet experienced programmers have found it a very powerful tool.

There is a deep integration of the GECOS-III file system and GE-625/635 time-sharing. Through this centralized file system, user programs in the time-sharing system and the batch system can communicate with each other.

This capability allows a large batch job to generate or update a file (perhaps using remote input data) and have the updated file available for inquiry by time-sharing users.

With GECOS-III, time-sharing users can even generate a job to be run in the batch mode.

ON-LINE PERIPHERAL TESTING

GECOS-III includes a comprehensive on-line peripheral test system, OPTS-600. Through special interfaces, the OPTS-600 can test any peripheral operation. Through the control console, operations personnel may request that any device be tested. Also, when any device error is noted, the operator can request a test of that device as soon as it is free.

Error information is accumulated by the operating system and the on-line peripheral test system. This allows continual measurement of peripheral device performance. Problems can be spotted and corrected before they become emergencies.

*DATANET, Registered trademark of the General Electric Company

SLAVE PROGRAMS

Each program is assigned contiguous core storage space whose origin and range are protected by the base address register (BAR) except when a privileged slave is executing in master mode. Associated with each program in core is one or more slave service areas (SSA). These are 1024-word blocks of core assigned adjacent to and immediately below the origin of the core space assigned to each program. The SSA immediately adjacent to the slave origin is divided into two equal blocks. The upper block (the block closest to the slave origin) contains all system information about the program including the Peripheral Assignment Tables (PAT) for the slave. These tables map logical file space into physical space on a device. A table for queuing I/O requests made by the slave is also found here. When a program requests I/O, a record of the request is made in one of these tables.

All other information about the program (except its program number and location in core) is kept in the upper block SSA. For this reason it is quite simple to swap a program out of memory when the occasion demands, since the upper block is swapped out with the program.

The lower block of the first SSA is used to contain for execution any system function modules required by the program. For instance, the MME GESNAP processor operates from within the SSA. All GECOS modules which execute from the SSA are called SSA modules. These modules need not be reentrant. Stacks for storage of the Instruction Counter and Index (IC and I) of the program and for register storage are also found in the lower SSA block.

Most slave programs are also assigned a second SSA, used to accumulate SYSOUT data prior to actually writing it into the SYSOUT file. The data is collected into the second SSA in order to reduce the number of physical file writes required. This, in turn, cuts down on time consuming seek operations on disc files. However, the time-sharing executive program uses its second SSA to hold more PAT's because a large number of files may be needed by the program at the same time.

The limited space of the program execution area of the first SSA poses difficulties. Some functions simply cannot be performed in the space available there. For this reason, a push-down file is assigned to every program in execution. If an SSA module executing in the SSA should call another SSA module, the caller is pushed down (copied) into the push-down file. When the called module exits, the caller is popped up (reloaded) into the SSA. Thus, the SSA area is like a paged (or virtual) memory. Calls between SSA modules may be made to any reasonable depth. If a program is swapped out of core, its own push-down file is used to hold it. In some cases, of course, the size of the push-down file must be extended to accommodate the program.

Thus, the SSA precisely limits the core demands that must be met in order to service any program. All GECOS-III service functions are either in the HCM or operate within the fixed area of an SSA block.

JOB FLOW

Jobs may enter the system in a number of ways: via a card reader, a magnetic tape, or through a remote terminal. In all cases they must be processed by a version of GEIN which is a privileged slave program. Multiple copies of the GEIN processor may be present in memory and executing simultaneously. For instance, batch jobs may be processed simultaneously from all configured system card readers.

At the same time, as many Input Media Conversion (IMCV) tapes as desired may be running. A separate soft core program containing a copy of tape GEIN is loaded for each IMCV or IMCV9 (9-track IMCV tape) message typed by the operator.

Remote GEIN may also be running. In this case, a special input collector, called RGIN accepts input from all remote stations and sorts it into files by job. As soon as any job input is complete, RGIN has a copy of Disc/Drum GEIN loaded as a slave program and gives it one of the files it has made to process.

No matter what the source, GEIN processes each input job file, and produces two output files, called the J* and *J files for each job. These files are held in the system without modification until the job has terminated.

The J* file is written in the file space assigned for the job's SYSOUT. It consists of a truncated image of all control cards in the job input file except for $ALTER cards. Thus, it is a synopsis of the requirements for running each job. The *J file carries all other data for the job. The *J file space definition is carried in the J* file. It consists of all the system data files, such as I*, S*, G*, etc. Thus, in case of a system failure, the *J file space for a job that was in execution at the time of the failure can be recovered by analysis of the J* file.

In order to correlate the control cards on J* with the system data files on *J, pointers are carried on the J* file. The pointers give the block number on *J where the data files are to be found.

GEIN separates the job into these files by carrying lists of all known control cards along with indications of special conditions. For instance, some control cards must appear as the first card on the data file they introduce. In other cases, they must not.

When GEIN has finished processing a job, it sends the space definition of the J* file to the Peripheral Allocator (ALOC). This is done by putting a message into the peripheral allocation queue. It should be noted that a job is defined completely by the interior name of the device and the link number on that device where its J* appears. Within the system, the interior "name" of any device is the address of the system configuration table (SCT) entry for the device. Some devices have exterior names also, like ST1.

The SCT and link number of the job are also recorded in the communication region tables for that job in case restart is necessary.

When multiple copies of GEIN are executing, the order of initial consideration of jobs by the Peripheral Allocator is simply the order that they appear in its queue. That is, there is no need for on-line GEIN to terminate before a remote job can be given to ALOC.

The J* file prepared by GEIN becomes the first part of the printed execution report after the job terminates. The control cards placed on this file are printed by GEOT. Thus, there is a visible record of the deck set-up after the run.

The Peripheral Allocator processes all jobs from the various copies of GEIN. This processing is done in two steps. First, the J* file is read. An encoded and very much shortened version of it is made. If the job clearly cannot be run for any reason, it is deleted. Otherwise, the first part of J*, as encoded, is read onto space reserved for *J. This file is usually on a faster device than J*, so time is saved by copying the control information onto *J.

An urgency or priority for the first activity of the job is calculated from the resources needed by the job, and the job is entered into various control tables of the Peripheral Allocator. At some time an attempt will be made to allocate all peripheral devices needed by the job. If this is successful, an image of the SSA of the job is constructed by the Peripheral Allocator on the *J file. The SSA image contains all PAT's set up for the file, as well as SYSOUT limits, and various other data.

The Peripheral Allocator then passes the job to the Core Allocator (GEPOP) through the Core Allocator's queue. The message sent is the SCT and link number of *J from which the SSA image is to be read. It also gives the amount of core desired, the number of SSA's, and a flag for the 9SA (7094 Simulator) job.

One of the most sensitive system parameters is the rule for passing jobs from the Peripheral Allocator to the Core Allocator. If jobs aren't sent often enough, core is not used. If too many jobs are sent, peripheral space is wasted. The balance between these functions is maintained through the core damper (.CRCSW), which informs the Peripheral Allocator when the Core Allocator has enough programs, so that no more will be sent to it.

When the Core Allocator gets a new job from the Peripheral Allocator, it places it into its tables using an urgency supplied by the Peripheral Allocator. An attempt is made to find a space in core in which the program fits. If none is found, all jobs with zero urgency (inactive jobs) are swapped out of core. Then another attempt is made. If this is again unsuccessful, either a core compaction or a swap out may be done. However, experience has shown that too much core compaction or swapping actually slows the system, so these are damped down. Neither a core compaction nor a swap is attempted until the urgency of the job not in core has reached certain levels. If core allocation fails for the most urgent job, its urgency is incremented.

When space is found, the Core Allocator zeros it, and copies a bootstrap program into it. It also sets up the common data needed in the SSA for all programs. It then puts the new program in the Dispatcher queue. Control passes then to the bootstrap program. This program reads the image of the SSA into core, sets up the job start message, and goes to the MME GECALL used to load the proper software. The program is now in execution.

While the program runs, it may call the various MME functions which will operate under the program number of the program itself, like subroutines. If GEPR or Move/Swap should be needed, the Dispatcher will preserve the state of the program, and send control to GEPR or Move/Swap. These modules then operate as if they were the program itself. When their functions have been completed, they will return control to the point at which the program was interrupted.

When a program terminates, the termination procedure again acts as if it were the program. The termination procedure uses the *J file to determine what the disposition of all peripherals is to be. The PAT's for all files that are to be saved are collected and written back out onto *J. The other peripherals are released. The termination procedure finally releases the program push-down file and tells the Core Allocator that it is finished. The Core Allocator then informs the Peripheral Allocator that the activity is done and releases the core. The Peripheral Allocator will then try to allocate the next activity and everything begins over again.

An important efficiency in the GECOS-III terminator sequence is that successive activities of the same type are compressed. For example suppose there are a number of FORTRAN activities in a job. When the first terminates, the terminator will see that the following activity is of the same type and will start it as a continuation of the activity, rather than go through the deallocation procedures. This saves considerable system overhead.

If an activity should abort, the Peripheral Allocator releases all system resources used by the aborted activity. Abort subactivities, if any, will be initiated. Subsequent compilation activities will be executed; however, all subsequent execution activities will be deleted until a $BREAK card is detected. The $BREAK card restores full activity execution.

At the end of a job, SYSOUT puts the SCT and J* in the GEOT queue and releases the *J file for the job. GEOT will then print and punch the output from the job as soon as possible.

SUMMARY

This overview represents only a brief introduction to GECOS-III. The following chapters in this manual describe the organization and implementation of GECOS-III.

2. GECOS-III ORGANIZATION

Physically, the GECOS-III operating system is broken into approximately 200 different modules; logically, the system has five related component parts:

(1) The Startup program

(2) The system executive (Hard Core Monitor)

(3) Several system programs (Privileged Slave Programs)

(4) Subroutines and service functions (SSA Modules)

(5) A set of system macros

SYSTEM STARTUP

The Startup program, although considered to be part of the GECOS-III system, is in itself a miniature operating system providing for the initialization of all system file devices. It edits various files into a file system, so that a copy of the operating system and its associated files may be continuously available. It also loads the system executive and tables needed by the system, and loads the resident system executive and turns control over to it.

Eight major sections control the loading, initialization and utility functions required for system Startup and are as follows:

(1) Environmental Control

(2) Hardware Configuration Control

(3) File Edit Control (optional section)

(4) Software Configuration Control

(5) Program Patch Control (optional section)

(6) Module Card Load Control (optional section)

(7) Module File Load Control

(8) GECOS Entry

Environmental Control

The environmental control section (STRT) interfaces directly with the manually induced bootstrap sequence and establishes the hardware environment in which the Startup program will operate. It initiates the reading of the Startup control card deck.

Hardware Configuration Control

The hardware configuration control section ($CONFIG) processes those control cards which describe the physical system that GECOS-III is to control and builds many primary system configuration and device control tables.

File Edit Control

The file edit control section ($EDIT) is optionally executed when requested by an $EDIT control card. Through this section, system file catalogs may be created, system files may be loaded, space for random system files may be defined, and existing alternate system files may be selected for use during GECOS operation. Some system file maintenance may be performed, including the dumping and copying of selected files.

Software Configuration Control

The software configuration control section ($FILES) is always executed during Startup, and is used to process that part of the Startup deck in which the software configuration to be supervised by GECOS is described. Five major system files, (the system itself, the system save file, system output, subprogram library and accounting files) are assigned by this section. Directories for these files are built for use by GECOS.

Program Patch Control

The program patch control section ($PATCH) is optionally used when temporary patches are to be made to GECOS-III or software modules. The patches are taken from the Startup deck and built into an internal patch table where they are held until the modules are brought into core storage. The patches are made to the module in core, thus enabling test and checkout of minor module corrections without altering the original version of the module.

Card Load Control

The module card load control section ($LOAD) is optional and only executed when the $LOAD control card is encountered in the Startup deck. This section permits the loading of complete HCM modules into core where they will be used in place of any existing module version in the system library files.

File Load Control

The module file load control section (XBGEX) is always executed and is used to load into memory all of the GECOS-III HCM modules not loaded by the module card load section and to prepare the GECOS-III memory maps. Entries are made in a module directory and the modules loaded are allowed to initialize themselves. Uniquely, one privileged slave program is also loaded for system rollcall use.

GECOS Entry Control

Operational transition from the Startup program to GECOS-III is accomplished by the GECOS entry section (XBDTF). Final setting of hardware vectors, primary and secondary mailboxes, interrupt and channel masks is completed before the core area occupied by the Startup program is cleared. Final transfer of control is made to the Dispatcher module (.MDISP) via processor fault action and operational control is assumed by GECOS-III.

Figure 1 depicts the layout of core storage after Startup has been cleared and control given over to GECOS-III.

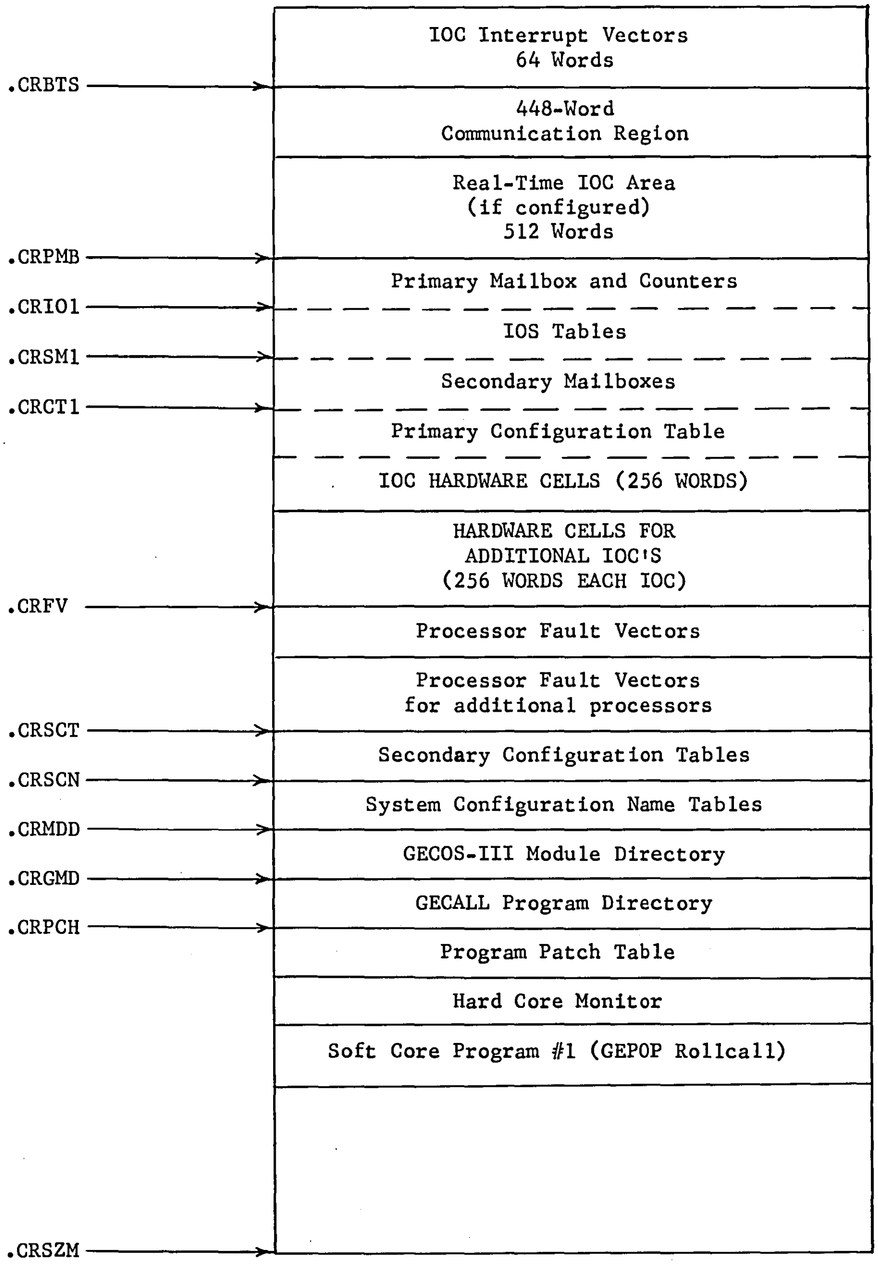


Figure 1. GECOS-III Memory Layout After Startup

FILE SYSTEM

Organization

The operating system is organized around a permanent file system. Permanent files may be cataloged into this system and protected against unauthorized access by a system of permission labels and passwords. GECOS-III itself is made up of several files from the file system. Other files within the file system are required by GECOS-III for storage space. Each file is defined by specifying the catalog name and file name of the file. By convention, the catalog name chosen for the system files is GECOS3; however, other catalog names may be used. The space occupied by each system file must be contiguous.

GECOS-III modules and all programs obtainable through an ordinary MME GECALL must be contained on not more than eight files. These files are indicated on the $SYSTEM control card and in the $FILES section of the Startup deck. If a duplicate module or GECALL name is found in more than one of these files, the duplicate will be ignored. Thus, it is easy to "patch" a system by editing updated parts into new files. These are then defined ahead of the standard file on the $SYSTEM card.

SYSOUT space is also described by naming as many as twelve files. These files may be assigned to separate devices but all must be on devices of the same type. The files should also be of the same length. For system efficiency, it is desirable to spread the files across as many channels or devices on a channel as possible.

Besides the system and SYSOUT files, accounting, library and save files are required by the system. The accounting file may be assigned to a magnetic tape unit or a shared device. The library file may be a part of the file system or assigned to a magnetic tape; the save file must be a part of the file system and is used for system recovery in the event of a system failure. This file should be on the device having the greatest speed. It is also of definite fixed length, six links.

File Space

A system of file catalogs is set up on all file-sharing devices in GECOS-III. A fixed area on each shared device is dedicated to contain sections of the file catalog and a bit table which defines the currently available space on that device. When permanent files are created, this table is updated. However, it is considered too time consuming to keep the available space table up to date to reflect all temporary file activity. During Startup, each table is scanned and an in-core link table generated for each shared device. All temporary file allocation is done using these tables.

The in-core link table designates in separate entries the 64 longest strings of links found in the device at Startup. When file space is requested, the link table is searched to find strings of the desired length and the table entry either deleted or adjusted to a shorter length. It may happen that, due to extreme fragmentation of the strings in the in-core table, no space will be found in the table to enter a string that is being released. In this case, the entry for the shortest string is replaced with one for the string that is being released.

When file space is allocated, no distinction is made between linked (serial access) and random access files. An attempt is always made to allocate space contiguously. Neither random nor linked files need be contiguous; however, the I/O supervisor will make any discontiguities invisible to the user. Single reads or writes can span discontiguities because a linking command will automatically be reissued to continue transmission across them.

File space assignment information for all temporary files is carried in the file PAT entry in the SSA. In order to conserve room, the information is in terms of strings of contiguous links. It should be noted that special formats are available to describe blocks which are really part of another file. This form is used mainly for describing the program execution file.

File space information for permanent files is usually carried in the file PAT entry also, but special provision is made to accommodate file space definitions too lengthy to fit into the SSA. In the PAT entry for each permanent file, the address of the file catalog is stored so that the other parts of the file space description may be obtained if necessary.

SYSTEM EXECUTIVE

The system executive, known as the Hard Core Monitor (HCM), processes all interrupts and faults and dispatches processors to the programs needing them. When the system executive is in control, no "program" is in control. At all other times, the system is under "program control;" That is, all functions are being performed by, or on behalf of, a program. It is the system executive which is resident in core storage after being loaded by the Startup program.

The HCM consists of a group of GECOS-III executive modules and the most frequently used system service modules. The modules are as follows:

.MDISP	Dispatcher/System Trace	
.MFALT	Fault Processor Executive	
.MIOS	I/O Supervisor	
.MDS20	Disc I/O	Channel Modules (Loading is determined by Startup Hardware Configuration Section)
.MPRIO	Printer	
.MTYPE	Typewriter	
.MDR20	Drum I/O	
.MGPIO	Card Reader	
.MCPIO	Card Punch	
.MDNET	DATANET-30	
.MRACE	Race	
.MPTAP	Paper Tape I/O	
.MMTAP	Magnetic Tape	
.MDUMP	System Disaster Dump	
.MSYOT	System Output Collector	
.MROUT	Remote Output Processor	
.MBRT1	Termination Executive	
.MGEPR	Error Processing Executive	
.MPOPM	Soft Core Program Required For System Rollcall, Core Allocation And Operator Interface	

The HCM is variable in length, depending upon system configuration. Core not used by the HCM is occupied by soft core (system) programs and slave programs. Program overlaying is never permitted within the HCM. Since all programs must originate at an address which is an even multiple of 1024 (because of the Base Address Register), there is a variable length block between the last module of HCM and the next even modulo 1024 address. This area is used to contain miscellaneous tables which are too big to fit into the communication region. Chief among these are the space definition tables for all shared devices and the system's trace table. The trace table holds a record of the last system events for which the trace is enabled. There are some fifty different system events which may be traced; any combination may be selected at system Startup time.

Dispatcher

It is the function of the Dispatcher to sequentially assign the use of all processors in the system to all programs that are currently resident in core. This module and the hardware processor interrupt logic have been designed to function together as the command and control center of the GE-625/635 GECOS System.

Specific features of the .MDISP module provide for processing the MME GEROAD (Roadblock) and MME GERELC (Relinquish) requests and the timer runout (Forced Relinquish) interrupt. The Dispatcher also collects the processor use time for the program releasing use of a processor to be later gathered for the accounting report. In addition, .MDISP processes the MME GEENDC (End Courtesy Call) request.

A work queue accessible to other privileged slave programs is provided in .MDISP. In this way, system work may be scheduled by the Dispatcher. The programs to be executed at courtesy call (I/O interrupt) level are given priority over those to be executed at main program level.

Processors are redispatched to programs after every interrupt has been processed.

The Dispatcher also provides information for the TRACE option, as well as transfers control for swapping and moving slave programs. It also has the ability to determine that a requested GECOS service module is in core or to supervise its loading, if required.

I/O Supervisor

The .MIOS module is divided into several functional parts, notably: the general I/O request handler, the I/O interrupt handler, the internal service subprograms, and the external service subprograms. An initialization section is required for use at Startup to initialize tables and vectors and to list for Startup the channel modules that must be loaded as part of the HCM, based on the current system configuration.

Channel Modules

In GECOS-III, modules are provided to drive each type of I/O device configured in the system. The modules handle all device-dependent operations such as interrupts, I/O request, error processing and channel queue select strategy.

Channel modules reside in core as part of the HCM and are part of the IOS/GEPR interface. Each channel module can handle as many channels as are configured with like devices, thereby eliminating the need for separate modules for each channel and greatly reducing the core requirements for GECOS-III. The format of a channel module is the same as any other GECOS-III module, thus preserving the homogeneity of the system.

The channel configuration tables built by Startup provide a four-word entry for each channel configured on each IOC. The fourth word in each entry points to an absolute address in the channel module that will drive all devices on that channel. The entry also provides for indirect address modification to access a desired entry point in the channel module.

All channel modules are constructed so that the first four entry points in each module are transfer vectors to parts of the module where four basic but device-oriented operations are performed. They are as follows:

Entry Point	Transfer To
1	Interrupt Handler Routine
2	I/O Request Handler
3	Select Routine
4	Error Processing Routine

In addition to the standard entry points, all mass storage device channel modules contain negative entry points used to provide storage for pertinent I/O information as well as transfer vectors to an appropriate Seek Address calculation routine.

The channel modules for mass storage devices contain device-dependent constants and device-dependent seek calculation routines. It is necessary that any processing done by master routines on mass storage devices use the information contained therein to achieve device independence.

A mass storage channel module is organized so that negative entry points contain constants and the transfers to seek calculation routines. The positive entry points (0-3) are used by IOS/GEPR. Transfer to a channel module entry point is accomplished by register indirect address modification.

System Output Collector

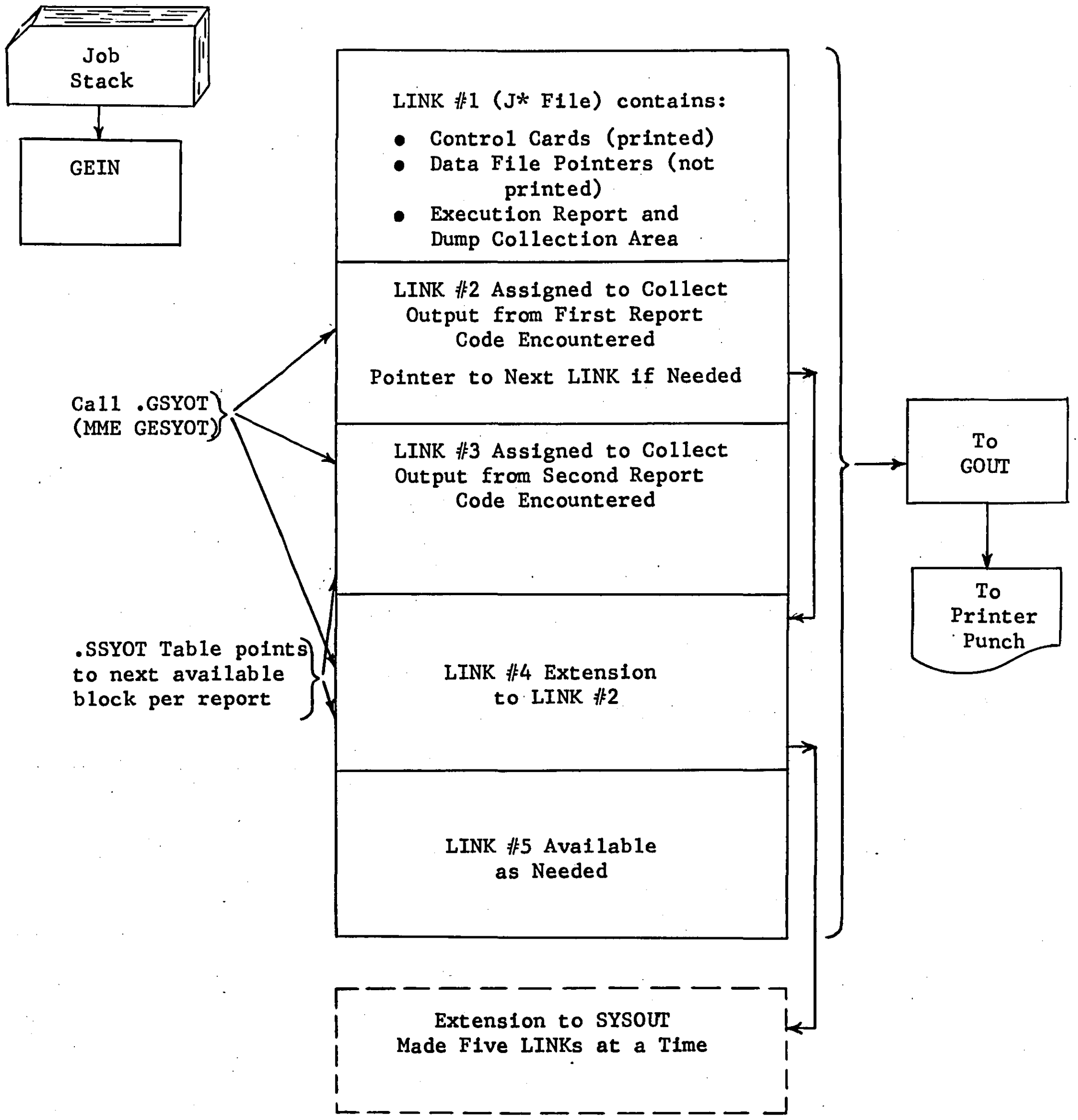
The files on which the system output is to be collected for a job are set aside by the file system out of the pool of links available for system data files. The SYSOUT files are always given an increment of five links at a time. In the first of the five links, GEIN creates a section known as the J* file. This file is used to contain all job control cards (which will be printed at a later time as part of the job execution report) and pointers to other temporary system files assigned to the job.

The SYSOUT Collector (.MSYOT) is a permanent resident of the HCM and processes all MME GESYOT requests made by slave and soft core programs. It interfaces directly with the slave service area of the program requesting output to the SYSOUT file, where it builds DCW strings and an I/O request with courtesy call. The I/O request is threaded into the queue of requests to be honored by the I/O Supervisor (.MIOS). When given control via a MME GEINOS, the I/O Supervisor checks the I/O queue.

Status information is returned to the .MSYOT module where it is used to update the SYSOUT status data in the requesting slave's SSA in preparation for the next MME GESYOT request.

When the job execution terminates, it is the function of .MSYOT to wrap up the last activity and process an execution report message. It enters an end-of-job flag record into the SYSOUT Collector file and then requests that the job be turned over to the SYSOUT Disperser (.MGEOT). Since GEOT is a soft core program, it is the function of the Dispatcher to assure that GEOT is in core and to give it control. Information to be passed to the GEOT program is signaled via the input queue to GEOT (.CRSYT) which may be accessed by the soft core program while it is executing in privileged master mode.

Figure 2 shows the structure of a SYSOUT file and Figure 3 diagrams the SYSOUT collection.


Job
Stack
GEIN
LINK #1 (J* File) contains:
• Control Cards (printed)
• Data File Pointers (not printed)
• Execution Report and Dump Collection Area
LINK #2 Assigned to Collect Output from First Report Code Encountered
Pointer to Next LINK if Needed
Call .GSYOT
(MME GESYOT)
LINK #3 Assigned to Collect Output from Second Report Code Encountered
.SSYOT Table points to next available block per report
LINK #4 Extension to LINK #2
LINK #5 Available as Needed
To GOUT
To Printer Punch
Extension to SYSOUT Made Five LINKs at a Time

(Five LINKs Initially Assigned to a Job)

Figure 2. SYSOUT File Structure

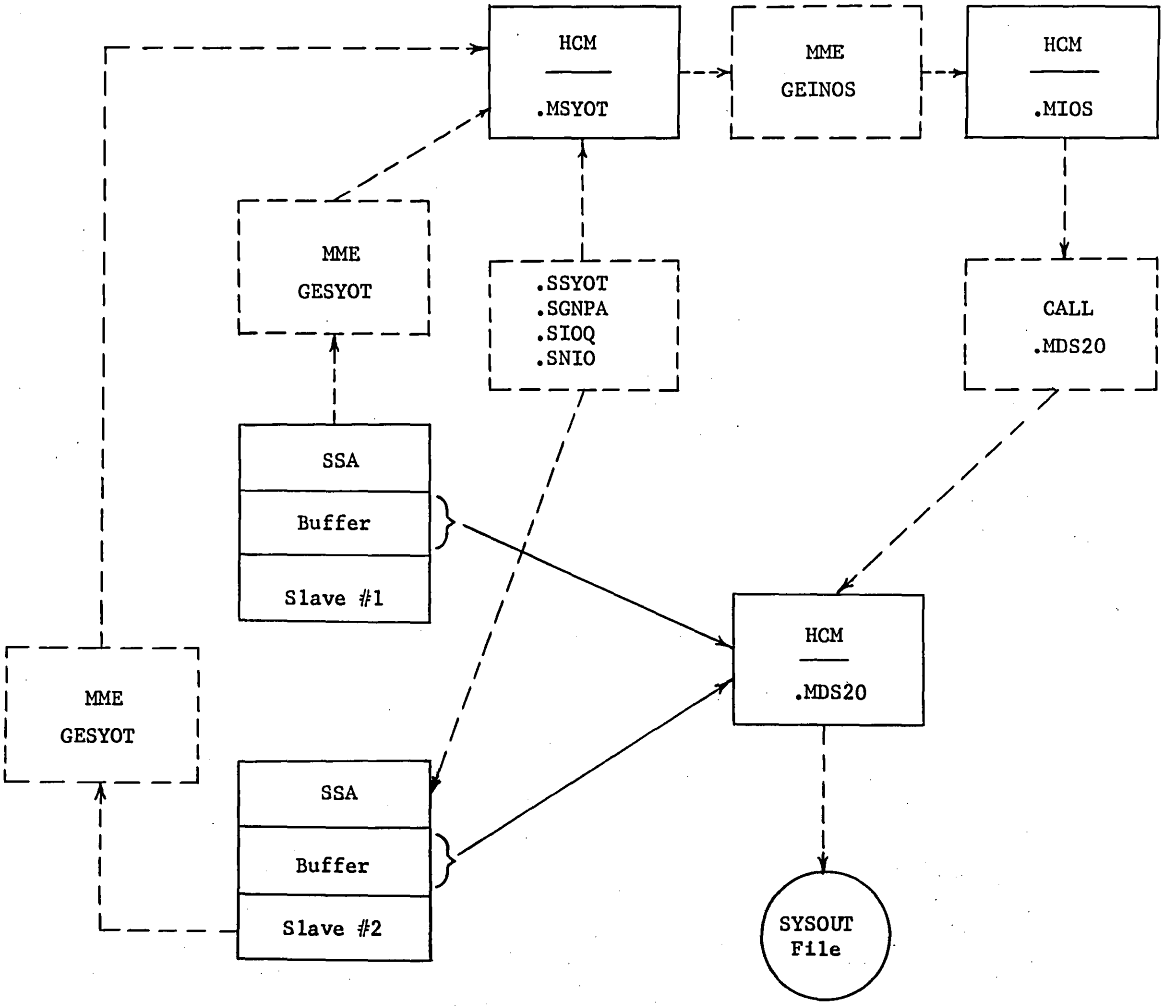


Figure 3. SYSOUT Collection

Termination

Terminating the execution of an activity requires the combined operations of the .MBRT1 module in the HCM, a slave service area module (.MBRT2) and a slave overlay module (.MBRT6). The hard core termination module oversees the servicing of MME GEBORT/GEFINI requests as well as those aborts detected through system faults, interrupts, and other GECOS detected errors.

Control is given to the SSA .MBRT2 module so that a direct interaction with the slave and its SSA may be accomplished during termination. It prepares the activity termination messages for operator and for SYSOUT as well as performing the optional functions of wrapup, dump and abort/utility operations.

The unique slave overlay module, .MBRT6, is loaded directly into the slave program area where it completes the activities in the PAT and SSA areas, deallocates temporary files, communicates dismount messages to tape operators, and releases the core storage occupied by the activity. If an end-of-activity is being terminated, the compacted PAT and SSA areas are copied back to the J* file where they may later be used for the next activity. If an end-of-job, the module communicates with the .MSYOT SYSOUT Collector to start the sequence of events which leads to the printing and punching of the job execution report and system output.

Accounting

A necessary adjunct to execution of an activity is the accounting for resources used so that appropriate charges may be distributed to the user. At activity allocation time, pertinent data are recorded and stored in the *J control records. During execution of the activity, processor time, core residence time, and peripheral file (channel busy) time are accrued in the slave SSA. At activity termination, these data are used to produce an activity accounting record which is written on the installation error and accounting file, and to provide information on the execution report which may be necessary to the user. The execution report is designed to be as helpful as possible in aiding a user to determine the cause of any malfunction or error, and the accounting data produced can be of considerable use in debugging a job deck.

Error Processing

There is a special relationship between the exception processing (GEPR) and the SSA. The GEPR component contains mostly SSA modules. Only the GEPR executive and the console typewriter GEPR routines are in the HCM. GEPR SSA modules generally must be only 242 words long, while regular SSA modules can be 478 words long. This limit permits GEPR to be used to clear errors on I/O done by other SSA modules. All status words and DCW's associated with I/O in an SSA module must be located in the upper half of the top SSA block. When a GEPR module is brought into core, only the first 256 words of the SSA are pushed down. Thus, the data needed by GEPR is still present in the upper SSA block.

There is one exception to this rule. MME GESNAP writes 320-word blocks to the SYSOUT file from the last 320 words of the SSA. Therefore, GEPR modules used to recover from errors on file-sharing devices must be correspondingly shorter.

SYSTEM PROGRAMS

In general, soft core programs are user-submitted jobs, but include a small number of system programs which are needed to process the user programs through the system. All system programs are executed as privileged slave programs and include:

System Input Collector (GEIN)
Peripheral Allocator (ALOC)
Core Allocator (GEPOP)
Remote GEIN Collector (GEROUT)
Time-Sharing Executive (TSS)
System Output Disperser (GEOT)
Test and Diagnostic System (OPTS)

All programs are, by definition, independent of one another. No program can directly affect the core space or file space of another.

System programs are not in synchronous operation. Thus, it is necessary to provide communication paths between all system programs. For instance, GEIN must tell the Peripheral Allocator that it has finished reading a job. Each system program (except GEIN) has an input queue for this purpose. Messages for a particular program are placed in its input queue by any one of the other programs. GEIN has no queue because there are no messages from other parts of the system that need go to GEIN.

Each program within GECOS-III has a program number which is assigned to the job when it enters the system and remains with it until the job has been finished. There are a total of 63 program numbers, 1 through 63. Zero is not a program number. Some program numbers are permanently assigned to the system programs mentioned above and shown below:

Program Number	Program Function	Program Number Symbol
1	Core Allocator (.MPOPM)	.PNPOP
2	Peripheral Allocator (.MALC1)	.PNALC
3	System Output Disperser (.MGEOT)	.PNSYT
4	Remote GEIN (.MRGIN)	.PNRGN
5	Time-Sharing Executive (.MTIMS)	.PNTSS
6	Test and Diagnostic Executive (.MTDSY)	.PNTAD
7	Held open for possible user installation assignment.	.PNOPN
8 9 . . . . 62 63	Slave programs, including GEIN programs, receiving from several devices concurrently. GEIN program numbers are assigned using the highest available numbers. Other slave programs are assigned the lowest available numbers.	

Since there are, at most, 58 slave program numbers available, GECOS-III can accommodate a maximum of 58 user jobs at any time. It should be noted, however that program number 5 is used to handle all concurrent time-sharing users regardless of the number of individuals using it. On small systems, it is possible that a lack of secondary storage may limit the total program numbers to less than 58.

It is necessary, at times, for system programs to reference the GECOS-III communication region. These programs, therefore, are allowed to execute a special master mode entry (MME .EMM) which allows the program to operate in the master mode. These programs are termed privileged slaves or soft core programs. Ordinary slave programs are aborted if they attempt to execute the MME .EMM.

System Input

GECOS-III allocates space and loads the system input program (GEIN) as a privileged slave program to process a waiting job. GEIN then reads input into the system from a system device:

Card reader
Shared devices (disc, drum, etc.)
IMCV tape

The input device becomes an allocated peripheral of the GEIN slave program.

It should be noted that multiple GEIN programs can be operating concurrently in core, reading input from several devices.

The GEIN program performs two functions: it separates the input into jobs and the input for each job into (1) data used by the job and (2) data used to control or describe the job. Every job (distinquished by an $SNUMB card) results in the creation of two files.

One file contains the control cards describing the job and one-word pointer records pointing to subfiles on the data file.

The second file, data file, contains several subfiles consisting of the data used by the job.

The records of these files are in the standard system format (320 words, with logical records of the variable type).

For example, consider the following job:

```
$    SNUMB
$    IDENT
$    GMAP (comdeck)
$    UPDATE (alter deck)
$    EXECUTE
$    LIMITS
$    INCODE (data cards)
$    ENDJOB
```

The control card file consists of the following logical records:

```
$    SNUMB
$    IDENT
$    GMAP
     Pointer to comdeck data subfile
$    UPDATE
     Pointer to alter deck data subfile
$    EXECUTE
$    LIMITS
$    INCODE
     Pointer to data card data subfile
$    ENDJOB
```

It should be noted that the noncontrol cards, comdeck, alter, and data, constitute the data file. In addition, transliteration will have been performed on the cards of the data file according to the content of the leading $INCODE card. The arrangement of the subfile records within the data file might look as follows:

record 1 through i comdeck
records i+1 through j alter deck
record j+1 through k data cards

When complete, the total job files will be made available to the Allocator.

As GEIN builds the control card file, it inserts the appropriate report, media, and slew codes into each logical record so that the file can be processed by SYSOUT.

The control card file thus becomes the beginning of the execution report for each job. If GEIN encounters errors in reading the job (checksum errors or illegal characters on the decimal cards), appropriate comment records will be inserted in the control subfile. No additional data subfiles will be constructed for that job. When complete, the job file will be given to SYSOUT for printing and the job will not become a candidate for allocation.

GEIN continues to build two files per job until it encounters an end-of-file condition. Then it terminates, the input device is released, and the core storage space becomes available for other use.

Peripheral Allocation

Peripheral allocation is the primary job scheduling function of GECOS-III. The Peripheral Allocator (ALOC) determines the order in which activities are assigned the necessary peripheral resources (both explicit and implicit) and presented to the Core Allocator for execution. The Peripheral Allocator initiates the necessary operator instructions associated with the assignment of peripheral devices and, in most cases, verifies that requested action has been taken before presenting the job for core allocation.

Conditional execution of activities of a job is also a function of the Peripheral Allocator. The condition may be a function of activity abortion or definitive conditional statements included by the job submitter.

Core Allocation

Core allocation is a soft core privileged slave program (GEPOP) always resident in memory. It has several additional functions besides core allocation.

GEPOP first takes a roll call of the system when GECOS-III starts. All peripheral devices that are off-line or in abnormal status are released. Any catalogs or files left allocated are deallocated. A GESPEC'ed RESS with a courtesy call is sent to all card readers. When the card reader is readied, the courtesy call to GEPOP will cause a copy of GEIN to be brought to core to read the cards.

Likewise, a GESPEC'ed WRITE THEN READ is sent to all four on-line consoles to type ??? and accept operator requests. Courtesy calls in GEPOP generate and type the responses to most operator requests.

GEPOP also acts as a queue for typing job start/stop messages. This allows programs to have these messages typed without requiring that they stay in core to type them.

Only the control processor may execute GEPOP. The GEPOP IC+I and register stacks in its SSA are used by the interrupt processor as needed. Program GEPOP, after the system roll call is taken, is 1024 words long and has a single SSA.

System Output Disperser

The System Output Disperser (GEOT) is uniquely constructed in that it is made up of an executive or main level and as many subprograms as required to drive the system output to as many as eight output printers and punches. It is the function of the main level of GEOT to receive a job, find and assign a free device, acquire core storage and load into it an available subprogram and assign the job to the subprogram. The subprogram contains two buffers and all necessary read and write sequences to read a data block from the SYSOUT file, to unpack it and set up in the output buffer the physical records to be passed to the output printer or punch.

In addition to the subprograms, GEOT also provides for the use of SSA overlay modules, GOU1, GOU2, and GOU3. As a job is received from the SYSOUT file, information is taken from the J* file (an integral part of the SYSOUT file) and used by GOU1 to produce the banner page or card at the beginning of an execution report. When the last record of the SYSOUT file has been processed, GOU1 also produces a terminal banner. GOU2 is used to produce the replies to remote status requests. GOU3 is used to produce headers for the individual reports produced for a job.

Figure 4 traces the sequence of events which affect the production of the system output. This output is produced only after an entire job has terminated. No option is provided for production of output after termination of an activity.

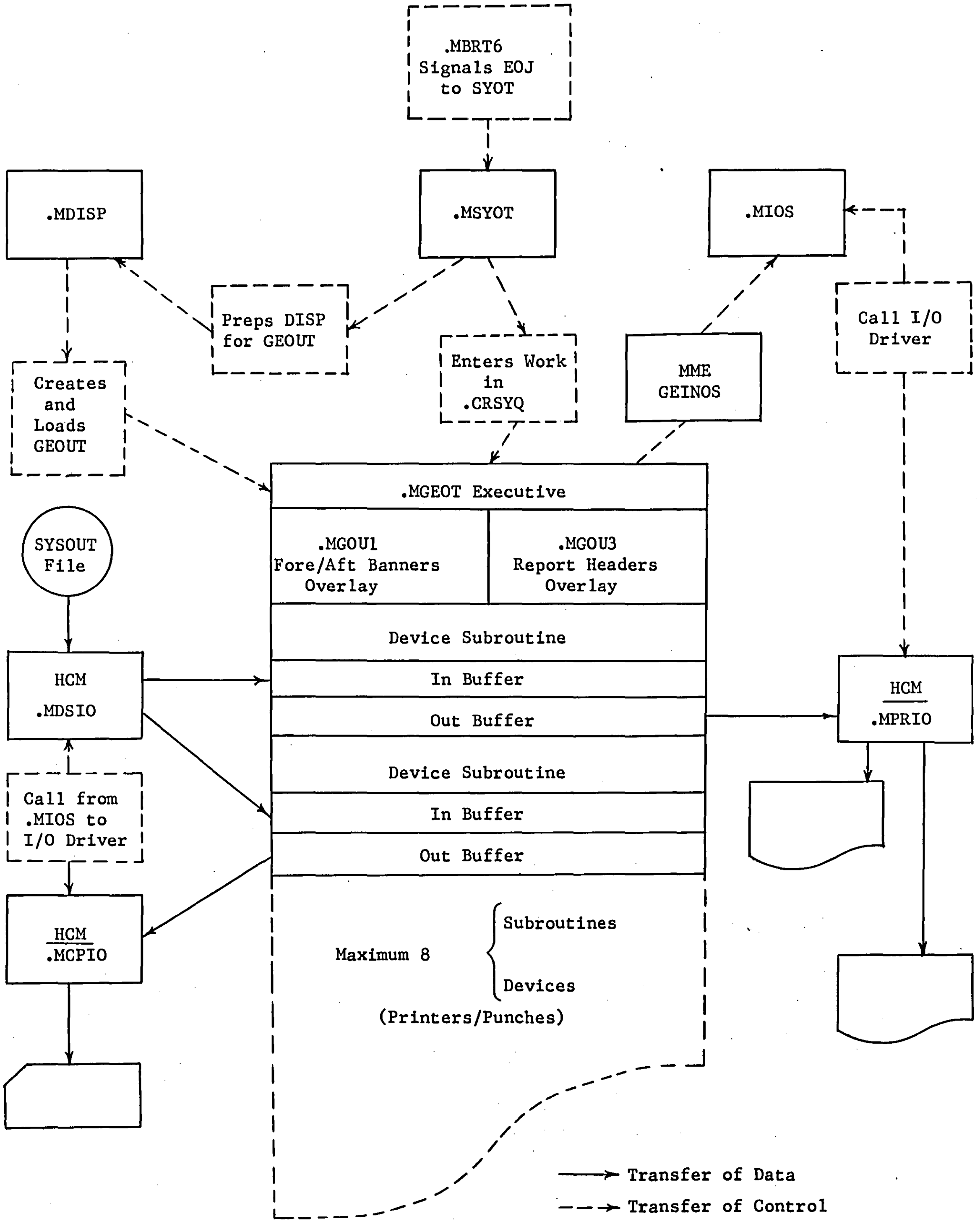


Figure 4. SYSOUT Dispersion

Remote System Input/Output

Remote System I/O (GEROUT) is the central part of the remote access system within GECOS-III, serving as a remote terminal interface and coordinator for the following GECOS programs.

- Remote SYSOUT (.MROUT)
- Remote GEIN (.MRGIN)
- Direct Access GECOS Programs
- GEPOP (.MPOPS)
- Time-Sharing Executive (.MTIMS)

All remote I/O requests for these programs are initiated via a MME GEROUT. Communication with GEROUT may be for the purpose of:

(1) Initiating I/O into or out of the requesting program's core.

(2) Passing information to or obtaining information from GEROUT.

In the first case, the calling sequence to GEROUT contains all the parameters required to obtain an I/O queue entry from the requesting program's SSA and initialize the queue entry in behalf of the requesting program. GEROUT has access to the status of all lines and the CIU; therefore, it may or may not initiate the I/O request depending upon existing conditions.

GEROUT has the responsibility of rejecting I/O requests which are inconsistent with the specified data flow and required sequence of events between the requesting GECOS program and the DATANET-30.

Again some calls to GEROUT are for the purpose of obtaining information on the status of a particular line or any line. Other calls to GEROUT relay information to GEROUT in order to update line status with respect to the calling program.

The functions performed by GEROUT are described in the following paragraphs.

REMOTE SYSOUT

Since remote SYSOUT accepts only one request per DATANET-30 for an output buffer, it is necessary for GEROUT to maintain a queue of all lines or terminals awaiting service by remote SYSOUT. The required queuing and status information is maintained by the remote termination interrupt processor and by GEROUT in the remote line status table. During the time that remote SYSOUT is servicing several remote lines (termination of each write operation by remote SYSOUT), the DATANET-30 triggers a courtesy call to remote SYSOUT to prepare output for a waiting output line. Queuing of this courtesy call to the Dispatcher is done only when another remote line is ready for output and remote SYSOUT is not currently in execution. GEROUT also answers remote job status requests.

REMOTE GEIN

As with remote SYSOUT, remote GEIN processes one input buffer at a time from the DATANET-30 in such a way that it permits n terminals to send the job source input simultaneously. Remote GEIN requires two types of calls to GEROUT.

(1) Remote GEIN rejects requests for new job input from line XX. This rejection tells GEROUT to transmit a terminate input message to the DATANET-30 and reset the GE-625/635 remote line status to an idle status.

(2) "Input accepted;" ready for more job source input from any remote terminal which has been previously accepted by remote GEIN.

GEPOP

For the purpose of remote access, GEPOP services certain requests from terminals which are associated with remotely originated GECOS programs. Specifically, GEPOP processes remote Abort Job/Activity requests. Responses to remote requests processed or forwarded by GEPOP are processed via GEROUT.

TIME-SHARING

GEROUT is the interface between the DATANET-30 and the time-sharing executive. All remote I/O requests are processed by the remote access system.

DIRECT ACCESS GECOS PROGRAMS

The direct access mode is compatible with GERTS-II.

These programs include Direct Access Output, Direct Access Output/Input, Remote Inquiry, and Terminal Identification.

Time-Sharing System

The Time-Sharing System (TSS) (executive module .MTIMS) is a privileged slave program. It is loaded in response to an operator request on the system console. The operator can control the amount of memory used by TSS, can warn all users of imminent termination of TSS and can terminate its operation.

TSS manages its own memory space and all user I/O. The status of the time-sharing system may be obtained at any time by a console request to TSS.

Test And Diagnostic System

The Test and Diagnostic Program (.MTDSY) allows the OPTS on-line peripheral test package to run as a GECOS privileged slave program. Operations may, from the console, call out a standard test on any specific peripheral, or a test may be called in on any error condition requiring operator intervention. When a peripheral is in test, the results may be directed to a console, or may be written into the accounting file for later analysis.

Field engineering personnel may program special tests for the peripherals using the GEPOP language. This language allows the specification of I/O.

SUBROUTINES AND SERVICES

The subroutines and service functions in GECOS-III make up the bulk of the GECOS-III modules. These include the processing modules for all MME* functions. Each MME processor operates as an extension of the program that called it. In contrast to GECOS-II, there are no master mode programs to perform MME functions like GECALL, GESNAP, etc. All are executed under the program number of the requesting user program and occupy part of the core area provided to each slave program as a service area. As a result, these modules are known as Slave Service Area (SSA) modules. This organizational philosophy greatly simplifies the logic of each MME function processor. An even greater advantage is that several programs can contain a MME processor at one time. Programs do not have to wait their turn while a master mode program processes the MME's one at a time.

Besides the MME processors, there are four other classes of service functions. These include the Move/Swap functions, the program termination sequence, and the I/O exception processing routines (GEPR). Finally, there is a set of internal subroutines used by any of the above functions. In GECOS-III, a program prepares itself for core compaction or swapping out of core by putting the Move/Swap module in control. Likewise, it terminates itself by calling in the termination modules. GEPR operates as a subroutine of the program which needs exception processing. Because of this, the system is able to recover concurrently from errors in different programs.

SYSTEM MACROS

The GECOS-III system macros are used to define all common symbols for each module of the system when it is assembled. The macros may be grouped into three classes. Class I consists solely of the .ENTRY macro which must be used by every module in the system. It establishes the basic linkage and defines all common symbols. Class II is the set of special purpose macros. Class III contains the macros which are called only by other macros.

These macros are more fully described in Chapter 3.

*MME - Hardware command "Master Mode Entry," the normal method by which a user program communicates with the operating system.

3. GECOS-III PROGRAMMING CONVENTIONS

The conventions established in GECOS-III to provide modularity and continuity of its implementation are described in this chapter.

SYMBOL CONVENTIONS

All symbols generated by GECOS-III are of the form .YZZZZ with few exceptions. The value of Y shows the type of the symbol according to the following table.

.CXXXX Location in the communication region

.MXXX Name of a GECOS-III module

..XXX Entry word of a module

.IXXX Initializing entry of a module

.S Location within the SSA

.N A symbol whose value gives the number of some system parameter.

.L A symbol whose value gives the length of a system table.

.P A symbol whose value is the program number of one of the dedicated GECOS-III program numbers.

The single character symbols 'L' and 'M' are used within the GECOS-III macros. They should not be used elsewhere.

.STEMP

Many of the MME processors in GECOS-III require some scratch storage. Some of these processors are part of HCM and are pure procedures. Therefore, an eight word scratch area has been set aside for the exclusive use of the MME processors.

Whenever a program begins a courtesy call, these scratch locations are saved in .STEMP+10 through .STEMP+19. They are restored at the end of the courtesy call. GEPR modules may not make use of this area; nor do the .QUEUE and .LINK routines.

FLOATABILITY

All GECOS-III modules are written in floatable code. That is, they may be executed from any location in memory without relocation at load time. This is done by using IC (Instruction Counter) modification on all references to locations within the module. Thus, to transfer to the location SYM,

```
     TRA  SYM-*,IC
```

is written. GMAP allows a shorthand notation for this:

```
     TRA  SYM,$
```

Here the ',$' replaces '-*,IC'. It should be noted that full word literals may be floated by appending ,$ to them as follows:

```
     LDA  =0525252525252,$
```

Here the literal will be referenced with IC modification.

Hard core modules of GECOS-III must be floatable, even though they do not move through memory, because the system loader does not allow relocation. However, program modules, like GEIN, may have relocation in sections that operate in the slave mode since the relocation is done by GELOAD when the system edit into GECALL format is done. In those cases where an indexed address into a floatable table must be generated, the following is recommended:

```
     EAA    TABLE,$
     STA    .SSTAK,5*
     LDX0   index value
     ADLX0  .SSTAK+1,5*
```

Now X0 contains the absolute address of TABLE plus the index value.

DEDICATED INDEX REGISTERS

When GECOS is in master mode, the contents of X5, X6, and X7 are dedicated as follows and must not be changed by any program.

X5 Lower Address Limit (LAL) of program
X6 Program number
X7 Processor number

The LAL of the program is the absolute address of slave word 0 for the program currently in execution. The processor number is the number of the processor executing the program. (The control processor is processor 0.)

When the Interrupt Handler (.MIOS) or the Dispatcher (.MDISP) is executed, processing is not being done on behalf of a specific program. Therefore, X5, X6, and X7 may contain other information. In these cases, the registers are not dedicated. In the Interrupt Handler when the macros .CALL, .EXIT, or .GOTO are executed, X5 and X6 are loaded with the LAL and number of the GECOS program GEPOP (program number .PNPOP). The GEPOP program can be executed only by the control processor. Thus, during interrupt processing, which also can only be done by the control processor, the Interrupt Handler assumes the identity of GEPOP, which must be out of execution.

Within a module, data cannot be stored by processor number unless all instructions performed during the interval in which the data is saved are interrupt inhibited. Data, therefore, is saved in .STEMP or in the stack (.SSTAK); this method is discussed in detail in the following.

Since X5 points to the LAL of the program, all locations within the slave service area (SSA) as well as locations within the slave program can be referenced through X5 modification. Thus, the values assigned to all SSA symbolic locations by GECOS macros are the differences between the SSA location and Word 0 of the slave. That is, the last location in the SSA has the value -1, while the first has the value 1024. Addresses specified as a difference between slave zero and another location are called offsets. For example, if the offset to a location is loaded into X4, it is referenced with an effective address of .CRBA4,7*.

LENGTH OF MODULES

There is a fixed amount of space in the SSA for modules. A module must not overflow the assigned space. The origin of each module is at location .SNTRY. An ordinary module may extend to location .SSA+511. A GEPR module, except for disc and drum error processing routines, may extend to .SSA+225. Disc and drum processors in GEPR may extend only to .SSA+191.

When any SSA module does I/O, all locations associated with the I/O transfer must be in the area above .SSA+256. All status return words and DCW's also must be in this area so that GEPR modules, which push down the first 256 words of the SSA when they are loaded, will find all data needed to reissue the I/O still in the second half of the SSA and not covered up by the GEPR module itself. If the I/O does not follow this convention, GEPR override must be given in the I/O. Since GESNAP to SYSOUT writes a 320-word block, rather than 256, the disc and drum routines must be correspondingly shorter.

REGISTER STORAGE

There are two possible areas in each program for safe-store of the program registers. The first area is in the slave prefix at locations 32 and 40 decimal. When a program is interrupted by an I/O interrupt, its registers are stored at 32, whether it is in main level or courtesy call. When a courtesy call begins, the registers at main level are moved to cell 40. At end courtesy call they are switched back. During courtesy call, the state word of the program is saved in cell 47, so that roadblock or relinquish conditions in effect at main level can be reestablished after the courtesy call.

The second register storage area is a stack at .SREG through .SREG+32. This stack is controlled by a tally word at .SSA+1 and NOP's at .SREGS through .SREGS+2. To save registers in this stack

```
     SREG   .SREGS,5*
```

is executed. To recover register,

```
     LREG   .SREGS+1,5*
```

If a module stores registers in this stack, it may overflow into the area reserved for SSA modules. In general, modules should never use SREG in master mode. When the registers are again loaded, using an LREG, another processor may be dispatched. Thus, the processor number in X7 will be lost.

When the Dispatcher gives control to a program and that program's IC and I shows master mode, the correct values for X5, X6, and X7 are inserted into the register stack or in the slave prefix. Within a module, the register storage stack should not be used. It is only sufficiently long to allow a program's state to be saved in order to begin a Swap, a GEPR, and/or an Abort. Registers are also stored in this stack when a program faults.

When the main level of a program is interrupted for GEPR, Swap, or Abort , the registers of the program are moved into this stack, if they are not there already; and the state word is stored in the eighth word, so that the state may be reestablished when main level goes back into control.

In all .CALL's, .EXIT's, and .GOTO's only the indicators and registers A and Q are disturbed. These may always be recovered with an LDAQ .STEMP+8,5.

Within GECOS-III modules linkage, no provision is made to save the registers. If the caller of a module wishes to preserve his registers through the .CALL, he generally must save them himself. However, the module calls .QUEUE, .LINK and .GSNAP do, in fact, save and restore all registers except A and Q. These may be recovered with

```
     LDAQ  .STEMP+8,5
```

when the return is made.

IC AND I STORAGE

The current IC and I value for each program is kept in a stack located at .SSA+2. This stack is controlled by a tally word at .SSA and NOP's at .SSTAK through .SSTAK+2. These NOP's have ID, DI, and I address modification, respectively. The stack tally pointer always points one cell beyond the last entry in the stack. To add an entry,

```
     STC1   .SSTAK,5*
```

is executed.

To return to the top stack entry

```
     RET   .SSTAK+1,5*
```

is executed.

The last three bits of the IC and I entry are encoded to indicate if the SSA was pushed down, or if the registers were stored in .SREGS when the IC and I entry was made. Bit 35, if one, means the registers were stored. Ones in bits 33 and 34 indicate SSA push-down.

When a module must save data across a .CALL, it may preserve the data by storing it into the stack, and then recovering it. For example, to save X4

```
     STX4   .SSTAK,5*

     .CALL

     LDX4   .SSTAK+1,5*
```

MASTER MODE ENTRIES

No Master Mode entries (MME) are allowed from within GECOS-III HCM or SSA modules because there is insufficient space in the register stack to accommodate the number of recursive MME's that could occur. Every MME has a similar GECOS-III macro which does the same thing. The macro name is obtained from the MME by replacing the symbols "GE" which prefix the MME address by the symbol ".G". For instance,

```
     MME   GEINOS
```

is accomplished inside GECOS-III by the macro call

```
     .GINOS
```

Not only does the macro avoid the possible overflow of the register stack, but it also requires less processor time.

A special MME,

```
     MME    .EMM
```

is used by system programs like GEOT to gain master mode control. This is done when the system program must reference data in the GECOS-III communication region. If an ordinary slave program attempts to execute this MME, it will be aborted.

When a slave program is in master mode, it will not be moved or swapped out of core. All code executed must be floatable. However, any absolute address generated, as for instance from a TSX instruction, will remain correct because the job will never move until it returns to slave mode. The MME .EMM loads X5, X6, and X7; and they are kept properly updated at each dispatch. When the program wishes to return to slave mode, it executes

```
     TSS    *+1
```

INITIALIZATION

When the Startup deck loads the GECOS-III hard core modules, it can transfer control into a module to allow it to do certain initialization. The initialization may be any combination of these functions:

(1) Setting up interrupt vectors

(2) Setting up fault vectors

(3) Modification of the module based on machine configuration.

(4) Presentation to Startup of a list of additional GECOS modules that must be loaded by Startup.

(5) Absolutizing addresses within the module when floatable addressing is not possible.

In order to do these functions, the X2 and X3 are set by Startup to point to the range of location zero and the address of the first fault vector, respectively.

The module is entered with a

```
     TSX1    .Ixxxx
```

When the module has initialized itself, it sets the A and Q registers (AR and QR). The value in QR is the next available address where the next module may be loaded. The AR may contain the address of a list of additional module numbers that must be loaded. The lower half of AR gives the number of such modules. If there are no additional modules, AR must be zero. The module returns control to Startup by a

```
     TRA    0,1
```

CLASS I MACROS

Name: .ENTRY

Purpose: To define all symbols known to GECOS-III and establish the entry point linkage to the module.

Macro Call: .ENTRY MODNO,END,GATE,(E1,E2,...En),ETC.,TYPE, INIT

Where:

MODNO is the name of a GECOS-III module without the .M prefix. For instance, for the Dispatcher module, .MDISP, this argument should be DISP. This name should also appear within the .MDDEF macro described below.

END is the symbolic location of the last assembled instruction of the module. This argument need be given only for GEPR and SSA modules. It is used to generate the checksum of a module when it is loaded into core.

GATE indicates only one copy of the module will be allowed to execute at a time. It is used only when a hard core module is not reentrant, or when an SSA module references nongated data.

E1,...En are the symbolic entry points of the module. E1 is the location to which control will pass on a .CALL or .GOTO to entry point one of the module, and so forth.

TYPE is a symbol which identifies the type of GECOS module. This symbol may take one of the following values:

- REQ for modules always required in HCM
- HCM for modules that may be part of the HCM but are not needed for every system configuration.
- SSA for slave service area modules
- GPR for GEPR modules

INIT.If this symbol appears for an HCM or REQ module, it means that the module when loaded must be given control by Startup to initialize itself. When INIT appears, the user must have a symbol of the form .Ixxxx where xxxx is the value in MODNO. This symbol identifies the initializing entry point of the module. The .ENTRY macro will produce a SYMDEF psuedo op for this symbol.

Expansion:

The in-line expansion of this macro is variable. Basically it consists of a great many SETS, BOOLS and EQU's which define all common symbols in the system, plus the linkage for the module. This linkage is outlined below. These SYMDEF's may be produced.

(a) If the module is not the Startup module, .MINIT, then

```
          SYMDEF..xxxx
```

is generated.

(b) If the sixth argument is INIT, then

```
          SYMDEF .Ixxxx
```

is generated.

(c) The entry word is produced, and labeled:

```
 ..xxxx  VFD    18/.Mxxx,12/YY,3/Z,3/W
```

The entry word format is described below.

(d) Transfers to all entry points follow:

```
          TRA  E1,$
          TRA  E2,$

           .
           .
           .

          TRA  En,$
```

(e) Finally, two words of dating information follow:

```
          BCI  1,mmddyy  DATE OF MACROS
          DATE           DATE OF ASSEMBLY
```

The meaning of the fields in the entry word are as follows:

YY is the number of words in an SSA module. It is calculated from the argument END. For HCM modules, YY is the difference between the address of .Ixxxx and the entry word.

Z is the type of module:

Z = 0 for GEPR modules
Z = 1 for SSA modules
Z = 2 for HCM modules

W is bits 33-35 with the following meanings:
33=1, Module flag
34=1, Busy
35=1, Not reentrant

If the module is a program, then the ENTRY word is generated as

```
          ..XXXX NULL
```

CLASS II MACROS

Name: .GATE

Purpose: To generate the gate tally for a data gate.

Macro Call:

```
.GATE     OPCODE, (ADDRESS)
```

Where

OPCODE is any legal GE-625/635 operation code

ADDRESS is any GE-625/635 address

Expansion:

(a) If there are arguments

```
OPCODE    ADDRESS

TALLYD    .CRGAT,0,2
```

(b) If there are no arguments

```
TRA       2,IC

TALLYD    .CRGAT,0,2
```

The first form is the one usually used. When coding, the second form is used only for installing gate tallys in-line.

Name: .SHUT

Purpose: To shut a gate

Macro Call:

```
.SHUT     ADDRESS, TAG
```

Expansion:

(a) If the TAG argument is absent

```
XED       ADDRESS+1,AD
NOP       ADDRESS+1,SD
XED       .CRGAT+10
```

(b) If the TAG argument is given

```
EAX0      ADDRESS, TAG
STX0      2,IC
XED       **,AD
NOP       **,SD
XED       CRGAT+10
```

The first form is the usual one used. The second form is used to shut a gate where tally word is in floatable code.

Name: .OPEN

Purpose: To open a gate

Call: `.OPEN      ADDRESS,TAG`

Expansion:

(a) If there is no TAG argument

```
NOP       ADDRESS+1,SD
XED       .CROGT
```

(b) If the TAG argument is present

```
EAX0      ADDRESS,TAG
STX0      1,IC
NOP       **,SD
XED       .CROGT
```

Form (a) above is the most commonly used. The other form is used to open a gate whose tally word is in floatable code.

Name: .PGATE

Purpose: To install a gate tally in line. This is called a procedure gate.

Call: `.PGATE`

Expansion:

```
TRA       2,IC
TALLYD    .CRGAT,0,2
```

Name: .SHUTP

Purpose: To shut a procedure gate.

Call: `.SHUTP   ADDRESS`

Expansion: `.SHUT     ADDRESS+1-*,IC`

Name: .OPENP

Purpose: To open a procedure gate.

Call: `.OPENP    ADDRESS`

Expansion: `.OPEN     ADDRESS+1-*,IC`

Name: .SHUTS

Purpose: To shut a gate when no program is in execution. This occurs only in the Dispatcher and Interrupt Handler.

Call:

```
.SHUTS     ADDRESS, TAG
```

Expansion:

```
EAX5       .CRGAT-.SGATE
```

Name .OPENS

Purpose: To open a gate when no program is in execution. This occurs only in the Dispatcher and the Interrupt Handler.

Call:

```
.OPENS     ADDRESS, TAG
```

Expansion:

```
EAX5       .CRGAT-.SGATE

.OPEN      ADDRESS,TAG
```

Name: .CALL

Purpose: To send control to a GECOS-III module so that an .EXIT will return control to the location following the .CALL expansion.

Call:

```
.CALL      MODNAM,EP
```

Where

MODNAM is a symbol of the form .MXXXX and is a name known to the GECOS-III system.

EP is the number of the entry point in the module.

Expansion:

(a) If argument EP is not given

```
XED        .CRCALL
ZERO       MODNAM,1
```

(b) If argument EP is given

```
XED        .CRCALL
ZERO       MODNAM,EP
```

Name: .GOTO

Purpose: To transfer control to a GECOS-III module when control is not to return to the point after the .GOTO.

Call:

```
.GOTO     MODNAM,EP
```

Expansion:

(a) If argument EP is given

```
XED       .CRGTO
ZERO      MODNAM,EP
ZERO      .SMDNO,0
```

(b) If argument EP is not given

```
XED       .CRGTO
ZERO      MODNAM,1
ZERO      .SMDNO,0
```

Name: .EXIT

Purpose: To return control to the point after the last .CALL.

Call:

```
.EXIT     SKIP
```

Where SKIP is the number of locations to skip following the .CALL expansion.

Expansion:

(a) If SKIP is not given

```
XED       .CREXT
ZERO      .SMDNO,1
```

(b) If SKIP is given

```
XED       .CREXT
ZERO      .SMDNO, SKIP+1
```

Name: .LINK

Purpose: To link an I/O request into the end of the request queue; no arguments are defined.

Expansion:

```
.CALL     .MIOS,1
```

Name: .LINKF

Purpose: To link an I/O request into the front of the request queue.

Expansion:

```
.CALL     .MIOS,2
```

Name: .LINKR

Purpose: To relink an I/O request from GEPR.

Expansion:

```
.CALL     .MIOS,3
```

Name: .QUEUE

Purpose: To get table space for an I/O request in the SSA.

Expansion:

```
.CALL     .MIOS,4
```

Name: .GEINOS

Purpose: MME GEINOS from within GECOS-III.

Expansion:

```
.CALL     .MIOS,5
```

Name: .GSPEC

Purpose: MME GESPEC from within GECOS-III.

Expansion:

```
.CALL     .MIOS,6
```

Name: .DISP

Purpose: To send control to the Dispatcher when the current program's state has been saved.

Expansion:

```
.GOTO     .MDISP,1
```

Name: .GROAD

Purpose: MME GEOAD from within GECOS-III.

Expansion:

```
SREG      .CRIRS,7*
.CALL     .MDISP,2
```

Name: .FRELC

Purpose: To relinquish control when the program's state is already saved in the stack.

Expansion:

```
.GOTO     .MDISP,3
```

Name: .GRELC

Purpose: MME GERLC from within GECOS-III.

Expansion:

```
SREG  .CRIRS,7*
.CALL .MDISP,4
```

Name: .GENDC

Purpose: MME GEENDC within GECOS-III.

Expansion:

```
.CALL     .MDISP,5
```

Name: .GFCON

Purpose: MME GEFCON from within GECOS-III.

Expansion:

```
.CALL     .MFLT1,2
```

Name: .GFILS

Purpose: MME GEFILS from within GECOS-III.

Expansion:

```
.CALL     .MFLT1,3
```

Name: .GLAPS

Purpose: MME GELAPS from within GECOS-III.

Expansion:

```
.CALL     .MFALT,5
```

Name: .GRETS

Purpose: MME GERETS from within GECOS-III.

Expansion:

```
.CALL     .MFALT,6
```

Name: .GSETS

Purpose: MME GESETS from within GECOS-III.

Expansion:

```
.CALL     .MFALT,7
```

Name: .GTIME

Purpose: MME GETIME from within GECOS-III.

Expansion:

```
.CALL     .MFALT,8
```

Name: .GBORT

Purpose: MME GEBORT from within GECOS-III.

Expansion:

```
SREG      SREGS,5
.CALL     .MBRT1,3
```

Name: .GCALL

Purpose: MME GECALL from within GECOS-III.

Expansion:

```
.CALL     .MCAL1,1
```

Name: .GCHECK

Purpose: .MME GECHECK from within GECOS-III.

Expansion:

```
.CALL     .MCHK1,1
```

Name: .GFINI

Purpose: MME GEFINI from within GECOS-III.

Expansion:

```
.CALL     .MBRT1,2
```

Name: .GMORE

Purpose: MME GEMORE from within GECOS-III.

Expansion:

```
.CALL     .MMORE,1
```

Name: .GRELS

Purpose: MME GERELS from within GECOS-III.

Expansion:

```
.CALL     .MRELS,1
```

Name: .GROLL

Purpose: MME GEROLL from within GECOS-III.

Expansion: .CALL .MROL1,1

Name: .GROUT

Purpose: MME GEROUT from within GECOS-III.

Expansion: .CALL .MROUT,1

Name: .GRSTR

Purpose: MME GERSTR from within GECOS-III.

Expansion: .CALL .MRES1

Name: .GSAVE

Purpose: MME GESAVE from within GECOS-III.

Expansion: .CALL .MSAV1

Name: .GSNAP

Purpose: MME GESNAP from within GECOS-III.

Expansion: .CALL .MSNP1

Name: .GSYOT

Purpose: MME GESYOT from within GECOS-III.

Expansion: .CALL .MSYOT,1

Name: .GITYM

Purpose: To type console messages in a standard manner.

Expansion: .CALL .MIOS,7

Name: .MDDEF

Purpose: To produce various lists of module information.

Call: .MDDEF ARGUMENT

Expansion: If ARGUMENT is blank, this expands into a series of SET psuedo ops defining all symbolic module names. If ARGUMENT is "LIST" this expands into a series of BCI 1, psuedo ops defining the BCI character string name of each module.

If ARGUMENT is "TV" this expands into a series of VFD's defining the type of each module.

```
VFD  3/4,3/0,6/0,3/L,3/M,18/0
```

The values of L and M are given as a function of module type below:

TYPE	L	M
HCM	0	2
REQ	2	2
SSA	0	1
GPR	0	0
HALF	0	0
PRG	0	4

Name: .SSADF

Purpose: To define all symbols in the SSA.

Expansion: This macro expands into a series of SET psuedo ops that give value to all symbols of the form .SXXXX.

Name: .CRDEF

Purpose: To define all symbols in the communication region of GECOS-III and all symbols associated with the IOC mailbox area.

Expansion: This macro expands into a series of SET pseudo ops defining all .CRXXX symbols.

Name: .SETPN

Purpose: To define all dedicated program numbers.

Expansion: This macro expands into a series of SET pseudo ops defining all .PNXXX symbols.

CLASS III MACROS

Name: .SBSS1

Purpose: To simulate a BSS pseudo op using the present value of symbol L as the current location counter.

Call: .SBSS1 NAME,EVEN,NOWORDS

Expansion: Using the current values of L, the argument NAME will be SET to the current value of L. If EVEN is given the value "O" or "E", L will be adjusted to odd or even before the SET is generated. Then L will be incremented by NOWORDS. If NOWORDS is blank, L will be advanced by one.

Name: .SBSS2

Purpose: To simulate a BSS; but decrementing the location counter symbol L prior to defining the value of the first argument.

Call: .SBSS2 NAME,EVEN,NOWORDS

Expansion: NOWORDS is decremented from value of symbol L. If NOWORDS is blank, one is decremented. If EVEN is given the value of "O" or "E", L is made odd or even respectively by another decrement. Finally a SET pseudo op is used to equate NAME to the updated value of L.

Name: .MDSET

Purpose: To simplify the expansion of .MDDEF macro, by allowing each module name and type to be an argument to this interior macro.

Call: .MDSET NAME,TYPE,#1,LABEL

Where

NAME is the BCI module name without the .M prefix. In order for any module name to be known to the system, it must appear in a call to this macro.

TYPE is the module type. It may have the following values:

HCM	Hard core module
REQ	Module in HCM, and must be present
SSA	SSA module
GPR	GEPR module
HALF	GEPR module
PRG	A program, not a module.

#1 must be present; it is replaced by the argument in the .MDDEF call generated by that macro.

LABEL is any BCI string beginning with 4 blanks and enclosed in parentheses. It is used to produce a descriptive label for the module.

Expansion: This macro is the macro which actually expands the .MDDEF macro. For an explanation of the expansion see .MDDEF.

COMMUNICATION REGION

The modular implementation used in GECOS-III requires a convenient way in which separate modules and programs may communicate with each other or obtain data pertaining to system configuration, status, etc. This has been accomplished by the establishment of a GECOS communication region in the low end of core memory. Tables, parameters, gates, system counters, and table pointers are defined by the GECOS-III macros using symbols having .CR as the first three characters. Where the tables are small or fixed in size, the value associated with the .CR symbol refers to the first word of the table; where the tables are large, or variable in size depending on Startup parameters, the value associated with the .CR symbol is the location of a pointer to the first word of a table elsewhere in hard core.

Symbols Used In The Communication Region

The gates, tables and table pointers included in the communication region are described below.

The .CR9SA contains the processor number cabled to SSA in bits 33-35.

The .CRACF table (2 words) holds the SCT address of the accounting file in the upper of the first word. The second word is a gate tally.

The .CRACK table (3 words) holds alarm clock information. The first word holds the time value at which nearest alarm is to ring, and the program number for that alarm. The time is in milliseconds since Startup and is in bits 0-29. The program number is in bits 30-35. Word two is a gate tally. Word three points to a table of program numbers. Each word holds the time, if any, at which the alarm is to ring. This time is in milliseconds in bits 0-29. The urgency with which the program is to be enabled is found in bits 30-35. If no alarm is set for a particular program, its entry in the table pointed at by the third word is: bits 0-29, all one's and bits 30-35, all zeroes.

The .CRALB word contains the alternate library identifier for low use routines. Format is as for .CRLST.

The .CRBA4 table contains base address values. X4 modification is called out in the lower. Thus, when a location is known only by a variable offset from the base address, it may be referenced as follows:

```
        LDX4    OFFSET
        LDA     .CRBA4,7
```

The .CRBAS table, by processor, contains the base address of the program in execution on that processor in the upper. The lower is zero.

The .CRBGT table (2 words) contains a pointer to the SYSOUT blink table and to a gate tally.

The .CRBTS word, when executed, causes a system disaster dump.

The .CRCAL table (3 words) is used to execute .CALL. The first two words are:

```
        STC1    STAK+2,5*
        TRA     CALL
```

The absolute address of CALL is calculated by the Dispatcher at system startup time and placed here. The third cell is used as entry point into the .CALL routine by the MME processor in order to save several instruction times in fault processing.

The .CRCCK table, by processor, is a scratch pad used to record the current timer setting when time of day is calculated.

The .CRCCS table (4 words) controls the queue of all courtesy calls waiting to be paid. Word one is a pointer to the first I/O queue entry containing a courtesy call address. In the lower is the last queue in the list. The second word is a gate tally. Word three is a switch used to test if any courtesy calls are to be paid. If so, it is a NOP. Otherwise, it is a transfer instruction. Word four is the transfer instruction used in word three when no courtesy call is outstanding.

The .CRCIC table (4 words) holds the port number of each of four possible IOC's on memory controller zero.

The .CRCMC table (4 words) holds the port number of each of four possible processors on memory controller zero.

The .CRCSW switch is a damper switch for the peripheral and Core Allocator. The upper contains the largest number of contiguous 512-word blocks of unused core. The lower contains the number of jobs currently waiting for core. The Peripheral Allocator will not continue to do allocation when enough jobs are waiting for core.

The .CRD30 gate is used to reference DATANET-30 tables.

The .CRDAT table (2 words) contains the date in BCD in the first word. Word two contains the time in 1/64 milliseconds from midnight to Startup. It is used in converting to actual time of day.

The .CRDIT table (16 words) is the device index table which holds SCT, link number, and number of links for each file defined in the .CRMDD table. The entries in .CRMDD are referenced to the .CRDIT table by the 3-bit index field found in .CRMDD. This field gives the relative position of the file description within .CRDIT.

The .CRDRL is a pair of derails in front of the GATING vector to stop any system bug which would cause a gate to be opened too many times.

The .CRDSP table (6 words) consists of a gate at the symbol plus one, and entries showing if any processors are idle. At the symbol is a tally showing the number of busy processors in the upper and the negative of that number in the tally field. Words two through five define the status of processors zero through three respectively. If the processor is idle, the word contains a DIS instruction with the processor number in the upper. Otherwise the word contains a negative address in the upper.

The .CRERF table (2 words) is not used.

The .CRETR table is not used.

The .CREXG table contains the SCT of the device which the operator has requested to be exchanged.

The .CREXT is a pair of cells which send control to the .EXIT routine. Their contents are:

```
STC1   .SSTAK+2,5
TRA    EXIT
```

The absolute address of EXIT is set up by the Dispatcher at Startup time.

The .CRFRS pointer table, by processor, points at the cell .SREGS of the program in execution. If the program faults, the program's registers are stored into the register stack through this pointer.

The .CRFSG is a pair of cells containing bits giving the status of each of the thirty-two parts of the file system. If the bit is one, then that part of the file system is locked. The second word is a gate tally.

The .CRFV word contains the address of the first cell of the fault vector in upper.

The .CRGAT vector (12 words) is pointed to by all gate tallys. A pair is executed by the .SHUT macro depending upon the state of the gate. If the gate is not already shut, the first pair is executed. The second, third or fourth pair is executed depending on how many other processors have already attempted to shut the gate. Words eight and nine, like .CRDRL above, are derails to catch bugs causing a gate to be shut too many times. Finally, words ten and eleven are used to cause a processor to attempt to shut a gate again when it has been opened by another processor.

The .CRGID table contains the software system description and the GECOS number and version used by DUMP for header information.

The .CRGMD table contains in the upper a pointer at the GECALL module table. In the lower is the number of entries in the GECALL tables, times 64.

The .CRGPG table consists of a gate tally for manipulating GEPR at the location of the symbol plus one.

The .CRGTC table (3 words) contains NOP's and a tally word for counting system gates. They are similar in format to .SGATE.

The .CRGTO table (2 words) is used to transfer control into the .GOTO routine. Its contents are:

```
          STC1   SSTAK+2,5*
          TRA    GOTO
```

The absolute address GOTO is set up by the Dispatcher at Startup time.

The .CRHCL is the address of the first cell outside HCM.

The .CRICN table (4 words) contains the numbers 0,256,512, and 768 in the lower. It is used to convert IOC numbers into IOC*256, as is required by the channel indexing.

The .CRIDT table contains the accumulated idle time, by processor, in 1/64th milliseconds.

The .CRINT table, by processor, has a flag showing if that processor is currently processing an interrupt; zero means no, nonzero means yes.

The .CRIO1, .CRIO2, .CRIO3, and .CRIO4 tables appear for each channel. They lie inside the IOC mailbox area in cells 64 -127 and contain the addresses of each of the four words of the physical channel tables for IOC zero, channel zero.

The .CRIOC cell indicates the system IOC model. If this cell contains a one, the IOC's on the system are IOC-B's; if a zero, they are IOC-C's.

The .CRIQC is the address of Initiate Interrupt queue counter for IOC zero.

The .CRIRS pointer table, by processor, points at cell 32_{10} of the program currently in execution. It is used to save the registers of a program when an I/O interrupt occurs.

The .CRJOB table is a set of pointers to the Peripheral Allocator input queue similar to .CRPOQ.

The .CRLAL table holds a gate at .CRLAL, followed by the lower address limit for each program. Note that the gate tally word is not found at the address of the symbol plus one, so that .SHUT .CRLAL-1 is required. The contents of the table indicate whether each job is in core, out of core, or not in the system. If the upper half of the word is nonzero, the job is in core at the address given. In this case the lower half is zero. If the opposite is true, the job is out of core. If both halves are zero, the job is out of the control of the Core Allocator.

The .CRLCK table, by processor, holds the value last stored into the timer register. The units are 1/64 milliseconds times 2^{12}, like the timer register itself.

The .CRLIT cell, when zero, causes IOS to look for lost interrupts.

The .CRLST word gives the SCT and link number of the system library file. If that file is a tape, then the lower half of this cell contains the name of the device holding the library file.

The .CRMCM word contains the memory controller masks for each of four possible memory controllers.

The .CRMDD word has a pointer to the .CRMDD table in the upper. This table contains data needed to .CALL all system modules and to GECALL all programs. The lower contains the number of entries in the .CRMDD table. The word following is a GATE.

The .CRMDT word is not used.

The .CRMSC pointer is described along with .CRPAR.

The .CRMSZ word contains in lower the number of 512-word blocks of core.

The .CRMXO cell gives the maximum number of SYSOUT lines allowed for a job before the Peripheral Allocator will inform operations.

The .CRMXS cell gives the maximum size of one activity allowed before the Peripheral Allocator informs operations.

The .CRMXT cell gives the maximum amount of time that a user may request on his job before the Peripheral Allocator will inform operations.

The .CRNIC word contains in upper the number of IOC's configured.

The .CRNPC word contains in upper the number of processors configured.

The .CROGT cells are used to send control into the gate opening routine. The contents of the pair are:

```
        STC1   GATE,7
        XED    OPGAT
```

The absolute addresses of GATE and OPGAT are calculated and stored by the Dispatcher at system Startup time.

The .CROVH table, by processor, contains the accumulated overhead time in 1/64 milliseconds. Overhead time is all time spent in the interrupt processor and Dispatcher.

The .CRPAR, .CRMSC, and .CRREQ locations are indirect pointers to line interface tables contained in .MDNET. They are referenced by a register containing 3 times the DATANET-30 processor index. Each of the cells contain X2 modification such that X2 must contain the line number.

The .CRPAU word contains the urgency of the Peripheral Allocator in bits 18-23.It is used when that program is to be enabled. The second word is the SCT and link number of the file holding the restart data for the system.

The .CRPCH word contains a pointer to a table of system patches entered from OCTAL cards at system Startup time. The address is in the upper half. In the lower half is the number of patches times 2^{10} plus bits to set the TXE option on an RPTX instruction. Each time a program is GECALLED from the system files, or a GECOS-III module is loaded, patches entered in this table are applied to that code. At most, 256 patches may be entered. No relocation is permitted on the address field of the patched instructions.

The .CRPMB is the address of primary mailbox for IOC zero.

The .CRPOP word gives the SCT address and link number of the dump file used for system recovery.

The .CRPOQ table (5 words) defines the status and position of the input queue for GEPOP, program number one. Consequently, word one is an input tally; word two is a gate tally; word three is a removal tally and word four is a tally whose count field is reduced to zero when the queue is full.

The .CRPRF table (4 words) contains one bit in each word. They are successively in positions 32, 33, 34, 35. They are used to test whether a processor may execute a particular program.

The .CRPRG table contains the program number of the program in execution in the upper. The table is ordered by processor number. If one entry is zero, it means that processor is in the Dispatcher, the Interrupt Handler, or is idle.

The .CRPRQ table is a threaded list of all program numbers representing jobs ready for execution. The first word in the table contains the first and last program numbers in the queue. Each entry contains the program number of its successor in the upper, and the program number of its predecessor in the lower. If a job is in execution, it contains its own program numbers in the upper; and the processor number of the executing processor in the lower. A job that is not a candidate for executing has both halves of the word set to zero. This table is gated by .CRDSP.

The .CRQGT table consists only of a gate tally for manipulating I/O queues at the symbol address plus one.

The .CRQST word gives the SCT address and link number of the Q* file used by GECOS-III for system recovery.

The .CRREQ pointer is described along with .CRPAR.

The .CRRGQ table is a set of pointers to the input queue for the first pass of remote GEIN. (See .CRPOQ, above.)

The .CRSCN word contains, in upper, a pointer to the system configuration name table; in lower, the number of entries in the SCN table *64.

The .CRSCT word contains, in upper, a pointer to secondary configuration tables. (Each channel that contains more than one device or a shared file device has a four-word secondary configuration table for each device.) In the lower of .CRSCT is the number of secondary configuration entries 256. The following word is a gate word.

The .CRSEQ word is the control for system-generated SNUMB's.

The .CRSIC table, by processor, points to the absolute address of .SSTAK for the program in execution. It is used to store the IC and I of an interrupted program when an I/O interrupt or fault occurs.

The .CRSID cell has six BCI characters identifying the GECOS version word. It is entered from the Dispatcher module.

The .CRSM1, .CRSM2, .CRSM3, and .CRSM4 locations are the addresses of each of the four words of the secondary mailboxes for IOC zero, channel zero.

The .CRSN1 word points to the first status table. This table, by program number, gives the status of the job in the system.

The .CRSN2 word points to the second status table. This table gives the J* SCT address and link number.

The .CRSNB table contains sequence numbers of all jobs in the system. For all ordinary slave jobs bits 30-36 of this table are 17_8. For system functions, they are always zero. At Startup time, fixed program numbers descending from 63_8 are assigned for all configured card readers. The SCT address for each card reader is placed in the upper half of this table. This table is gated by .CRLAL-1.

The .CRSQC location is the address of the Special Interrupt queue counter for IOC zero.

The .CRSSD word contains the SCT address of master file system device, ST1, in upper. The BCD characters ST1 appear in the lower.

The .CRSTP word is not used.

The .CRSYQ table is a set of pointers to the GEOT program input queue. See .CRPOQ above.

The .CRSYT table is a table of file space to be used by SYSOUT.

The .CRSZL word contains the length of GELOAD in the upper.

The .CRTBL word contains the address at which dump is to be started after HCM.

The .CRTDQ cells control the input queue of the test and diagnostic program.

The .CRTLS cell contains the total number of lost interrupts.

The .CRTOD table, by processor, contains the accumulated time from Startup to the last time the timer was loaded. The units here are 1/64 milliseconds. Note that this is not actually clock time of day.

The .CRTQC location is the address of Terminate Interrupt; queue counter for IOC zero.

The .CRTRC table (4 words) is used to generate system trace entries. Words one and two are used to test for trace and save registers preparatory to making a trace. This is done with an XED .CRTRC. Word three is used to test if the trace is on without saving registers. This is done with an XEC .CRTRC+2. Word four points to a certain cell within the trace routine of use for entering trace data into the trace table.

The .CRTRG table is a tally word used only by the trace routine for storing the system trace entries in word one. Word two is a gate tally, and word three is a refresher tally for word one.

The .CRTSQ table is a set of pointers to the time-sharing program input queue similar to .CRPOQ above.

The .CRTTT cell contains the total numbers of transfer timing errors.

System Counters

For use in evaluation of the overall action and efficiency of GECOS-III, a number of system counters have been established. They are not a necessary part of the operating system per se, but may be used to provide valuable data for insight into the operation of GECOS-III. A system resource monitor has been written for display or selected information, including several system counters, using either a dedicated printer or a DATANET-760.

The system counters, or cells, and their contents are as follows:

.CRTJB	total jobs
.CRTRJ	total remote jobs
.CRTAC	total activities
.CRACT	total activities
.CRTDS	total dispatches.
.CRTCC	total C. C.
.CRTGP	total GEPR
.CRTCN	total connects
.CRTIR	total interrupts
.CRTCL	total .CALLS/GECALLS
.CRTKS	total checksum errors on calls
.CRTMK	total memory compacts
.CRTSP	total swaps
.CRTJM	total jobs moved
.CRTJS	total jobs swapped
.CRTSM	total MME GESYOT
.CRTSN	total lines printed
.CRTSR	total cards punched
.CRTRR	total remote reports
.CRTSW	total writes by GESYOT
.CRTSD	total reads by GESYOT

.CRTWT This cell contains the total number of times any processor went idle.

.CRTCT These cells control the system's trace. For each trace entry type, a bit in order in the first two words is used. If the bit is one, that trace entry is off. Otherwise, it is on. The third word is used by the system to force a dump when a trace is made. Each trace entry is counted in the fourth word. If the third and fourth words are ever the same, the system will immediately dump. The first three words are entered into the system with the $ TRACE control card. The fields on this card are presented in octal.

SLAVE SERVICE AREA SYMBOLS

Certain data, such as peripheral assignment lists, are necessary for each program currently in execution but are not required for the operating system itself. Symbols or tables of this type can expand or disappear entirely, depending on the number of jobs currently in memory. Provision for this is made by assignment of such symbols to the slave service area (SSA) of the associated program.

The SSA is laid out with tables and parameters of a fixed length at the beginning and end of the area, and tables of indefinite length in the middle. The latter consist of the I/O queues and peripheral assignment table (PAT). All SSA symbols are defined by the GECOS-III macros using symbols having .S as the first two characters. The value associated with the symbol is an offset to the first word of the user's slave memory. Thus, all .S--- symbols are negative and lie between -1 and -1024.

The SSA contains two pushdown stacks. The first, .SSTAK, contains the IC and I entry for each .EXIT. The program IC and I entry is stored in .SSTAK when a program faults or is interrupted. The stack of 12 words is managed from a tally word in the first call of the SSA. To add an entry, STC1 .SSTAK,5* is executed. To return to the top stack entry, RET .SSTAK+1,5# is executed.

The second pushdown stack, .SREGS, holds all registers. This stack is controlled by a tally word in the second word of the SSA.To save registers in this stack, SREG .SREGS,5* is executed. To recover registers LREG .SREGS+1,5# is executed.

Immediately following these two tally words is the 12 entry IC and I stack followed by 458 words to be used by SSA modules. After the module execution area are 32 words for the register stack. Since registers are stored here by a .SREG, only four successive stores can be made.

The definitions of the symbols in the SSA are as follows.

.SSA Contains a tally word pointing to next available entry in IC and I stack.

.SSA+1 Contains a TALLYD word pointing at last currently used register storage address.

.SCKSUM Contains checksum of module read into SSA. If this cell is zero cleared, the module will not be rechecksummed again before entry. This provides a means to allow nonreentrant modules in the SSA to be reentered. Unless .SCKSM = 0, a new copy of the module will be read to core before it is reentered if the checksum does not agree.

.SNTRY Entry word of module. This contains module number, busy bit and gate bit. It also contains the length of the module. The length is used for checksumming the module when it is read to core. The busy bit is used to determine if the SSA must be pushed down, in the event of GEPR. The GATE bit keeps multiple copies of the same module from executing concurrently.

.SREG This is the first cell of a 32 word area which hold registers. An SREG is executed using this area at every fault, or transfer of control into GEPR, Swap/Move Abort. Since recessive faults (such as MME's) can cause a disastrous outflow of this small stack, no MME's are allowed within the system itself. Rather macros, which have the same effect, except for NOT storing registers, must be used in place of the MME's. All of these macros are formed by replacing the "GE" on all MME names with the characters ".G" For example, the equivalent of:

```
                MME  GEINOS
     is
                .GINOS
```

.STATE This word is used to record the status of each job. Requests to Abort, to Move or Swap, or for GEPR are recorded in this word and the Dispatcher sequences control to these functions in response to the appropriate bits set in this word.

.SGEPR I/O entry currently being processed by GEPR in bits 0-17.

.SPROI SCT of file which was an active GEPRIO request in bits 0-17.

.SACT Activity number in binary 18-35.

.SACTY Activity name in BCD in first cell. Name of succeeding activity name in BCD in next cell.

.SMSZ Number of 512 word blocks of memory assigned in bits 27-35.

.SAVG Product of memory size times the elapsed time used up to the last change of memory size, due to either a memory release or a GEMORE.

.SURG Urgency of this job in bits 18-23.

.SSTAK 3 NOP's to control IC and I stacks.

These are as follows:

```
NOP     .SSA, ID
NOP     .SSA, DI
NOP     .SSA, I
```

To make a stack entry, a

```
STC1    .SSTAK, 5*
```

is executed.

To recover a stack entry

```
NOP     .SSTAK+1, 5*
```

is executed. The STAK pointer must always point to the next available cell in the stack. If the stack is backed up to recover the last IC and I entry, then the code must be inhibited until the pointer is moved back to normal position.

For example:

```
INHIB    ON
LDA      .SSTAK+1,5*
NOP      .SSTAK, 5*
INHIB    OFF
```

In an IC and I entry, the last 6 bits are unused by the hardware. They are encoded by software convention as follows:

33 = 1 This IC and I caused the SSA to be pushed down to send control to GEPR. In this case, only the first 256 words are pushed down. Thus, modules in the SSA may do I/O, using GEPR, providing that all data, DCW's, and status return address are in the second half of the SSA.

34 = 1 This IC and I caused the whole 512- word SSA to be pushed down to accommodate a GECOS module other than GEPR.

35 = 1 At the time of this IC and I entry, the program registers were stored in the stack controlled by .SREGS. This can happen only at a fault, or a transfer of control into GEPR, Swap/Move, or Abort.

.SREGS These are three NOP's that control the register storage stack at .SREG; they are as follows:

```
NOP      .SSA+1, SD
NOP      .SSA+1, AD
NOP      .SSA+1, I
```

This stack points at the first cell of the last register store area.

.SGATE This three word entry counts gates that have been closed by this program. The cells are as follows:

```
NOP      .SGATE+2, ID
NOP      .SGATE+2, DI
TALLY    0,0
```

When a gate is shut, the instruction at .SGATE is executed, incrementing the tally address. Conversely, when the gate is reopened, the address is decremented by executing the instruction at .SGATE+1. This is done automatically by the .OPEN and .SHUT macros.

.SALIM The first word of this pair contains the nominal BAR and IOC limits corresponding to actual memory allocation. The second word is used to actually load the BAR. Note that for 9SA jobs, the limits set in the BAR include the memory used by the 9SA; and thus, is different than the nominal limits.

.SSSAL This contains the number of words of SSA allocated to the program in 18-35.

.SPRT Processor time used in 1/64 milliseconds is in the first cell; time of day at the last unswap or memory size change is found in the next cell. The units are 1/64 milliseconds.

.SPTBE Address in offset form of next cell in SSA available for constructing a new PAT in bits 0-17.

.SNIO Number of I/O queue entries allowed for this program irrespective of their current status in 18-35.

.SSYOT Cells used by SYSOUT collector, (see tables).

.SLOAD The first word contains the negative count of the number of attempts left to load, push, or pop the SSA using the routine found in entry point 12 of the Dispatcher.If a module is being loaded, its module number will be found in the second word of this table. If a push or pop is under way, the second word will be zero.

.SCCAL A count of the number of outstanding courtesy calls for this program is in bits 0-17. If the job is disconnected from the system, and is being moved, this cell contains a pointer at the string of CC queues to be reconnected after the move in the lower.

.SRQCT Bits 0-17 contain the number of I/O requests still in process by this program. This includes all I/O not yet begun, or in transmission, or waiting for GEPR or courtesy calls. A roadblock is broken when this count reaches zero. In bits 18-35, the count of all I/O currently in transmission is found. A program cannot be moved when this count is not zero. If the program is disconnected from the Dispatcher, and is being moved, bits 18-35 contain a pointer at the chain of all I/O that must be relinked after the move is completed.

.SICI This is seven words of code that are used to send final control into a program at each dispatch. The code is as follows:

```
LDA      .SSA,1
CANA      1,DL
TZE       3,IC
LREG     .SSA+1,AD
RET      .SSA,I
LREG      32,5
RET      .SSA,I
```

Note that the absolute address of .SSA and .SSA+1 are stored into the first, fourth, fifth and seventh words of .SICI.

.SPUSH This cell contains the relative block number into which the next push down of the SSA will be written. This block number is found in the lower part of the word.

.SSTRT These two words are used to contain the status return words for I/O commands used to load, push or pop the SSA, or to swap the program in or out of core.

.STEMP This is a block of 20 scratch storage words: ten for main level, and ten for courtesy call level of the program. Only :STEMP through call is begun or is finished, the Dispatcher switches the first and last ten cells of .STEMP. At each .CALL, .GOTO, or .EXIT within GECOS, A and Q are stored in .STEMP+8 and .STEMP+9. All other words are available to MME processing routines. Neither IOS nor GEPR uses .STEMP, so it is always available as scratch pad storage for MME processors.

.SMDSK The DCW for the seek address for I/O commands manipulating the program points at this cell. There are five cells following used to hold read or write DCW's associated with the seeks.

.SREMT If a MME GEROUT has occurred in a job, this cell contains the offset to the PAT pointer to the DATANET needed by GEROUT in the upper half, otherwise this cell is zero.

.SSYIO This is an I/O queue entry reserved for GECOS for servicing the job. All I/O used to load, push or pop the SSA is done using this queue; and I/O for swapping jobs also uses it. GEPR also uses this queue to type operator messages, because no further system manipulation is possible when GEPR is thus engaged. It may not be used anywhere else. The .QUEUE routine will not obtain this queue.

.SIOQ This area contains space useable as I/O queues. The number of queues assigned is variable and is given in the lower half of ordinary slaves, the number is presently set to 5. For GEPOP, 20 I/O queues are assigned. Following the last I/O queue position are found the bodies of all PATS assigned by the user program. Following the PAT bodies, there is generally an unused section of the SSA. This is available for the construction of more PAT bodies, if they are needed, during the execution of the program.

.SUID This is 12 BCD characters giving the users identification in the master catalog of the file system.

.SALT Allowable time in 1/64 milliseconds until abort is in the first cell. The second cell holds allowable time till GELOOP abort, if any.

.SWIT User's switch word.

.STQTM This is the time (in 1/64 ms *2) loaded into the timer at each dispatch. Unless the program relinquishes roadblocks or is interrupted within this time, the program suffers a timer run out fault, and is taken out of execution. This time is currently set to 16 milliseconds.

.SGNPA This PAT body is set up by the system to be used for manipulating the control stack (or J* file) of the job. This PAT body is also used by .MME GESYOT for writing SYSOUT data into the SYSOUT files. The third and fourth word of this PAT are heavily encoded with special information used by GESYOT.

.SSAPA This PAT body is used for I/O to load GECOS modules into the SSA.

.SGCPA This PAT is used for all GECALL I/O.

.SGNPT Pointer to .SGNPA.

.SSAPT Pointer to .SSAPA.

.SPDPT Pointer to .SPDPA.

.SGCPT Pointers to .SGCPA.

.STPPA Pointer to first PAT body for console.

.STOTY Pointers to PATS for *T, T/, /T.

.SPATP Pointers to user's PATS begin here, proceeding downward in memory. Below the last pointer and above the highest PAT body is an open area that may be used for building extra PATS during execution of a job.

.START Time of day at the start of this activity recorded in 1/64 milliseconds.

.STPPT First console PAT pointer.

4. GECOS-III SYSTEM TABLES

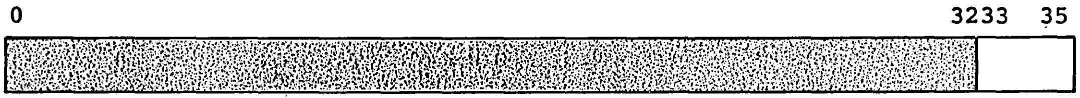
The flexibility of the hardware configuration of the GE-625/635 system together with the wide variety of job mixes occurring among different user installations requires in turn a flexible operating system which can be adopted to a broad spectrum of conditions. To effect such a system, the hardware and software variables are described in various tables.

During system Startup, a preliminary set of tables is used for initialization of modules which may be affected. When preliminary initialization is complete, the operational system tables are established in the GECOS communication region and, where necessary, in other portions of the core resident system (HCM).

These tables are described in this chapter.

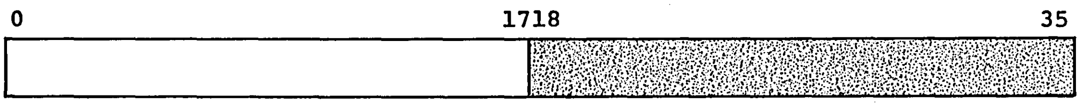
STARTUP TABLES

.CR9SA

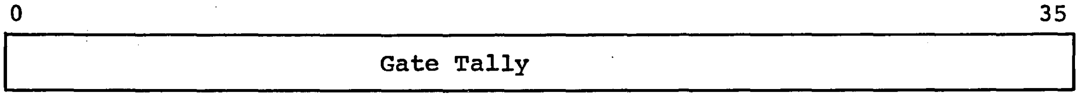


Bits 0-32 Not used
33-35 Number of the processor cabled to the 9SA.

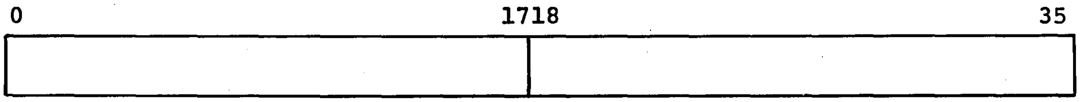
.CRACF



Bits 0-17 Absolute address of the SCT for the accounting file
18-35 Not used



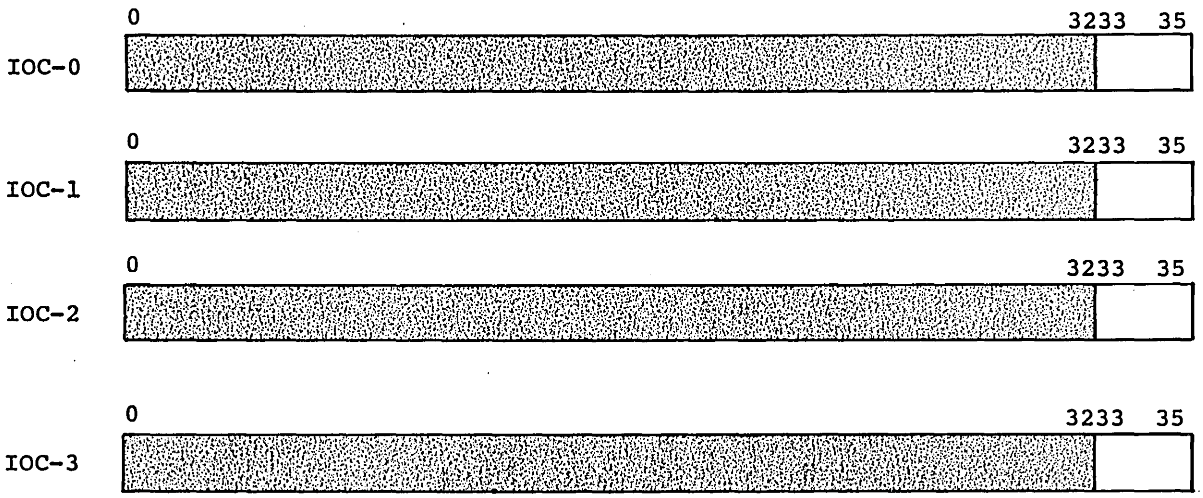
.CRBTS



Bits 0-17 Manual dump entry address in module .MDUMP
18-35 TRA, inhibit on

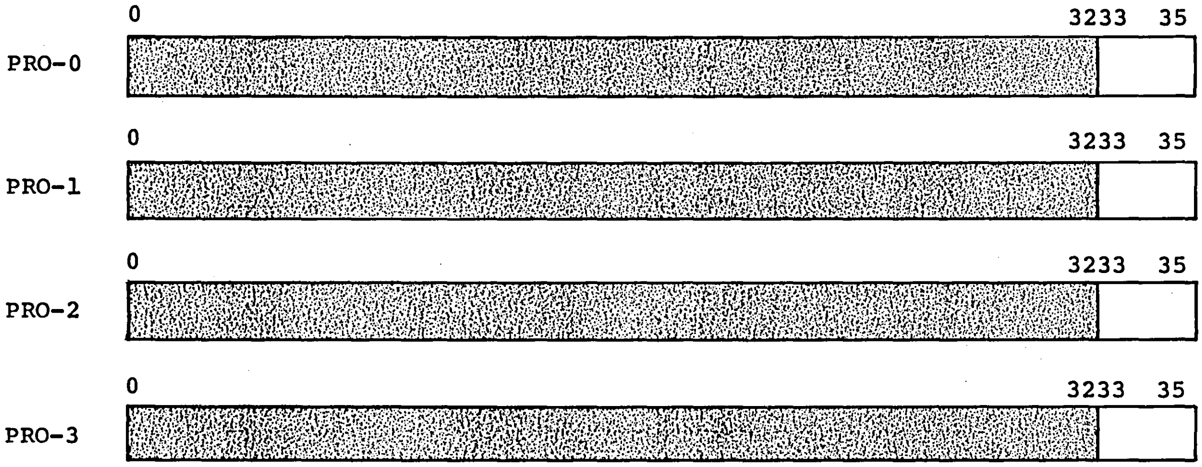
NOTE: During Startup, the address in the upper half of this cell points to the post-mortem dump location within Startup itself.

.CRCIC



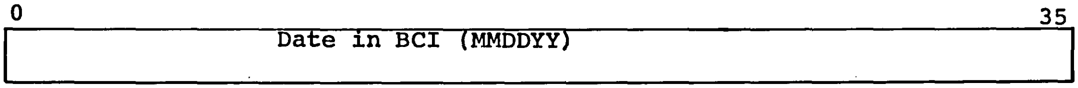
Bits 0-32 Not used
33-35 Port number on Memory Controller zero to which that IOC is connected.

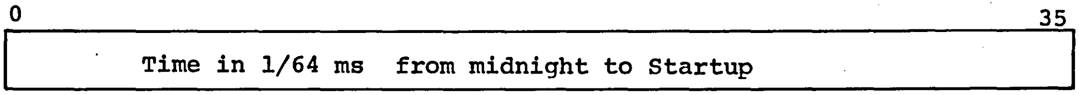
.CRCMC



Bits 0-32 Not used
33-35 Port number on Memory Controller zero to which that processor is connected.

.CRDAT

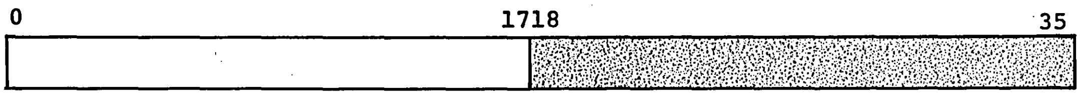




.CRDIT

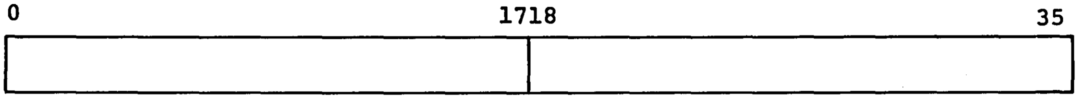
Table is used in conjunction with GECALL and is referenced by bits 3-5 of the .CRMDD entry, which gives the relative position of the system file within this table.

Word 0



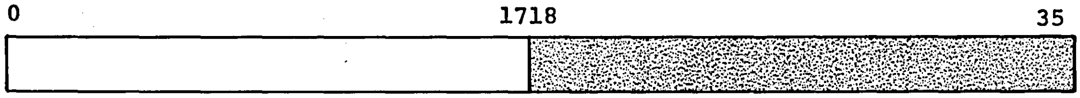
Bits 0-17 SCT address for device holding file 1
18-35 Not used

Word 1



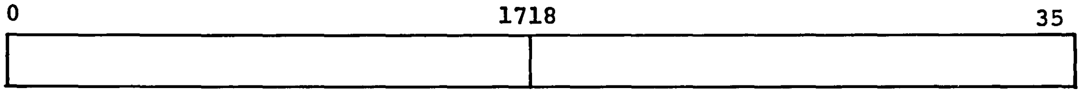
Bits 0-17 Link number
18-35 Number of links

Word 2



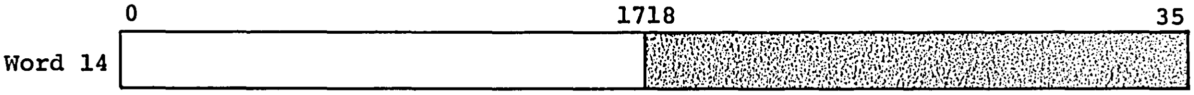
Bits 0-17 SCT address for device holding file 2
18-35 Not used

Word 3

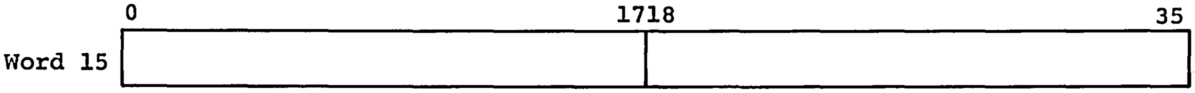


Bits 0-17 Link number
18-35 Number of links

•
•
•
•

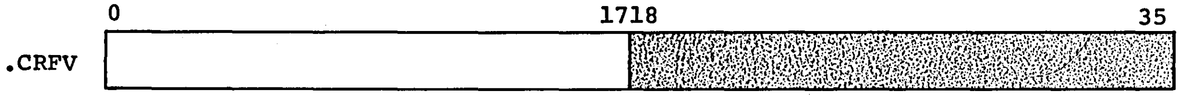


Bits 0-17 SCT address for device holding file n
18-35 Not used

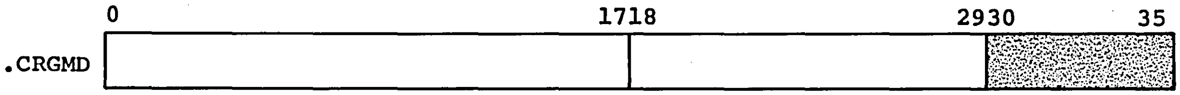


Bits 0-17 Link number
18-35 Number of links

This table holds the SCT address, starting link number and number of links for each file defined in the module directory (.CRMDD)



Bits 0-17 Absolute address of the first cell of the fault vectors
18-35 Not used



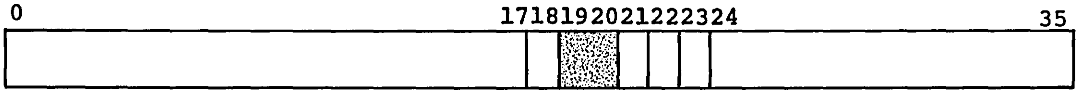
Bits 0-17 Absolute address of the start of the GECALL program table
18-29 Number of entries in the table
30-35 Not used

.CRI01-4

There is a four word entry for each channel (PUB) on each IOC.

The entries reside within the IOC mailbox area from Primary Mailbox Location plus 100$_8$to PMBX plus 177$_8$.

.CRI01

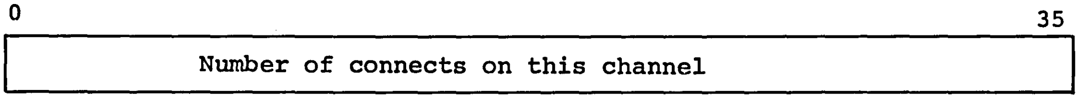


Bits 0-17 Absolute address of active I/O entry. Zero if non-busy.
18= 1, Special Interrupt has occurred
19-20 Not used
21= 1, This is the primary crossbar channel
22= 1, Do not use this channel
23= 1, Crossbar exists on this channel
24-35 True Ch IX of next Xbar (IOC*256 plus PUB*4)

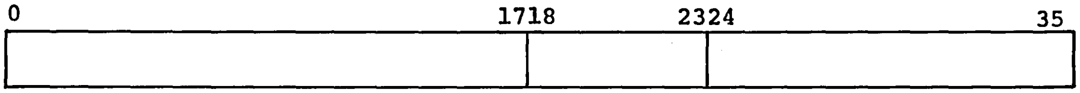
.CRI02

0
35
Time of day of connect

.CRI03

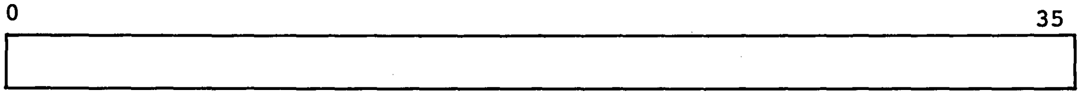


.CRI04



Bits 0-17 Number of errors on this channel
18-23 Number of errors in last n connects
24-35 Logical channel index (IOC*256 plus PUB*4)

.CRIOC



Bits 0-35 = 0, configured IOCs are IOCC
0-35 ≠ 0, configured IOCs are IOCB

.CRLST

0 17	18 35

<u>RAM</u>

Bits 0-17 SCT address of system library file
18-35 Starting Link number

.CRLST

0 17	18 35

<u>TAPE</u>

Bits 0-17 SCT address of system library file
18-35 Device name in BCI

.CRALB

0 17	18 35

Bits 0-17 SCT address of second library file
18-35 Device name in BCI

<u>.CRMCM</u>

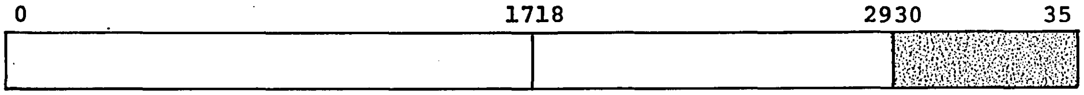
0 35
Memory controller mask for memory controller - 0

0 35
Memory controller mask for memory controller - 1

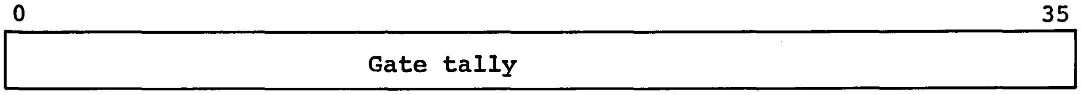
0 35
Memory controller mask for memory controller - 2

0 35
Memory controller mask for memory controller - 3

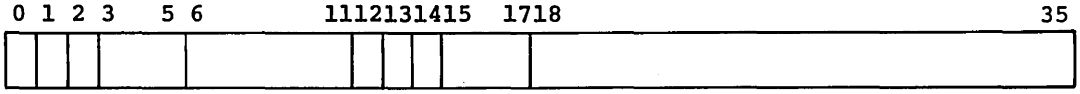
.CRMDD



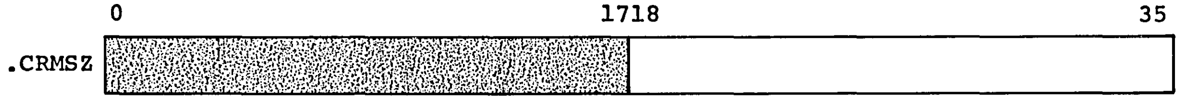
Bits 0-17 Absolute address of the first entry in the .CRMDD table
18-29 Number of entries in the table
30-35 Not used



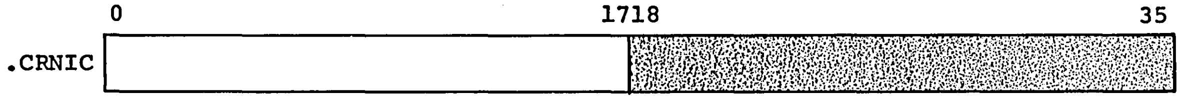
.CRMDD Table Entry



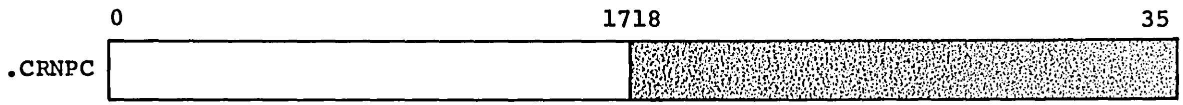
Bit 0= 1, Module is HCM and re-entrant
1= 1, Module is not re-entrant
2= 1, Module busy
3-5 Relative position of system file in .CRDIT
6-11 Program number
12= 1, Patch for this program exists in CRPCH
13= 1, This program required
14= 1, This program loaded
15-17 Type = 0 - GEPR
1 - SSA
2 - HCM
4 - Absolute program
5 - Relocatable program
18-35 Hard core monitor address or rel. block number



Bits 0-17 Not used
18-35 Number of 512-word blocks of core memory



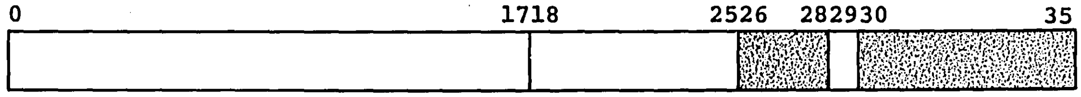
Bits 0-17 Number of IOCs configured
18-35 Not used



Bits 0-17 Number of processor configured
18-35 Not used

.CRPCH

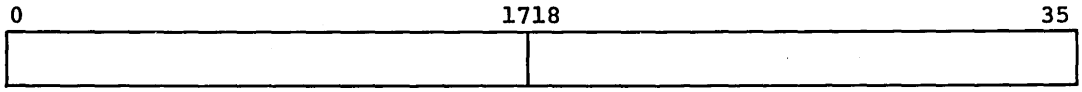
The lower half of cell .CRPCH is the modifier that will be used in an RPTX instruction to search the patch table at load time.



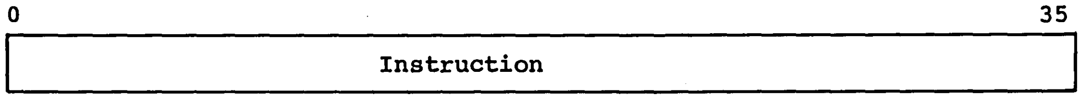
Bits 0-17 Absolute address of start of the system patch table
18-25 Number of table entries
26-28 Not used
29 TZE terminator bit for the RPTX
30-35 Not used

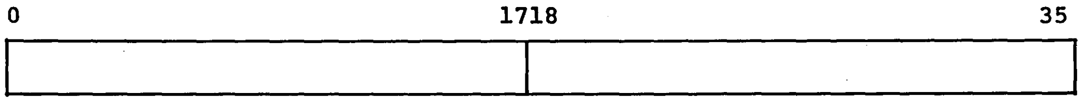
Format of System Patch Table

A maximum of 256 patches may be entered.

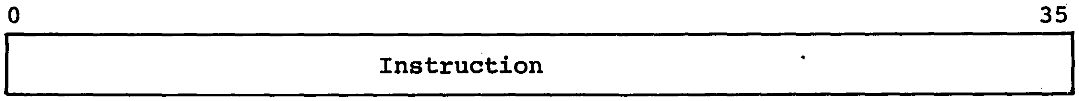


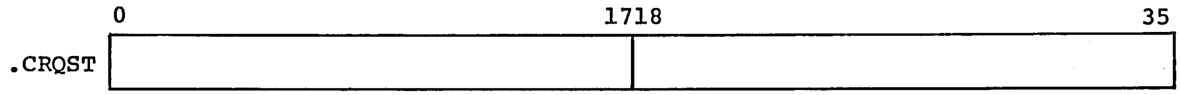
Bits 0-17 Module or GECALL number
18-35 Address of instruction



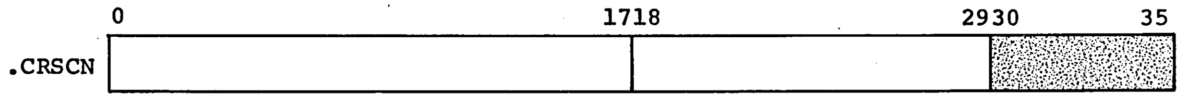


Bits 0-17 Module or GECALL number
18-35 Address of instruction



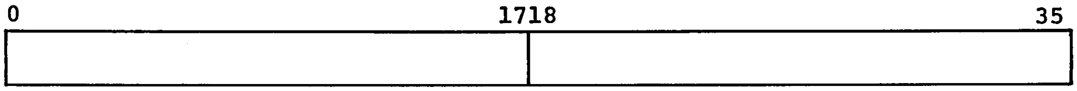


Bits 0-17 SCT address of Q* file
18-35 Starting link number



Bits 0-17 Absolute address of system name table
18-29 Number of entries
30-35 Not used

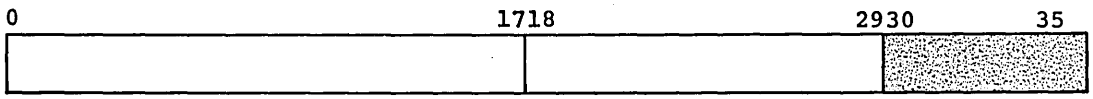
<u>Format of System Name Table Entry:</u>



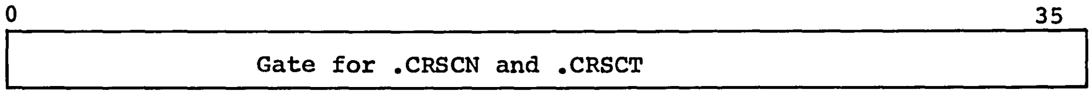
Bits 0-17 Address of device SCT table
18-35 System name (3 BCI characters)

.CRSCT

Each channel (PUB) that contains more than one device, or a shared file device, has a four-word secondary configuration table for each device.



Bits 0-17 Address of first secondary SCT table
18-29 Number of entries in system secondary tables
30-35 Not used

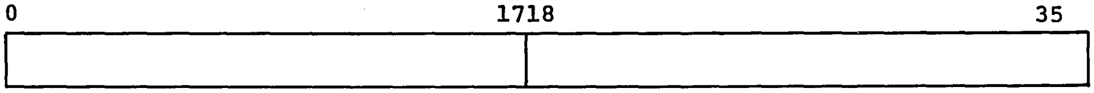


.CRSYT

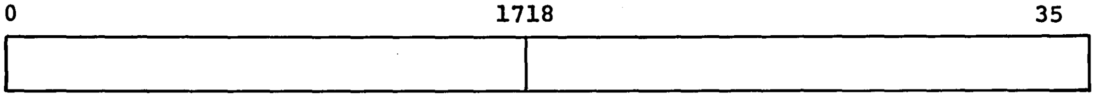
The devices to which SYSOUT links are assigned must all be of the same type.

The number of links assigned to each SYSOUT device will be equal to the total number of links assigned to SYSOUT divided by the number of devices assigned to SYSOUT.

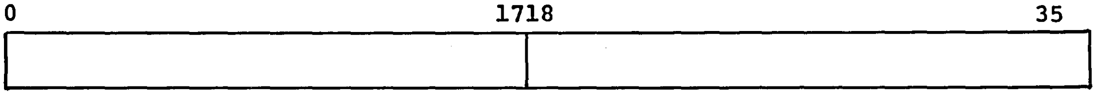
The link numbers refer to the start of a group of contiguous links on each device.



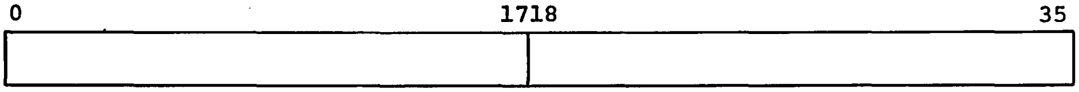
Bits 0-17 Number of devices on which SYSOUT links are assigned
18-35 Total number of links assigned



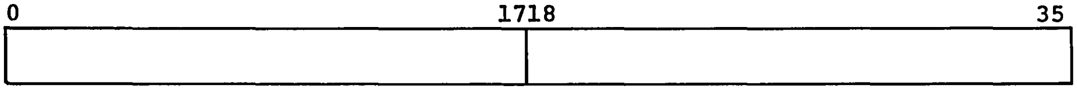
Bits 0-17 SCT pointer for device 1
18-35 Device 1 starting link number



Bits 0-17 SCT pointer for device 2
18-35 Device 2 starting link number



Bits 0-17 SCT pointer for device 3
18-35 Device 3 starting link number

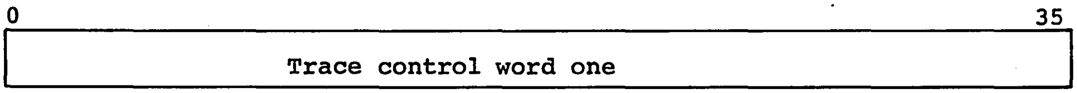


Bits 0-17 SCT pointer for device 4
18-35 Device 4 starting link number

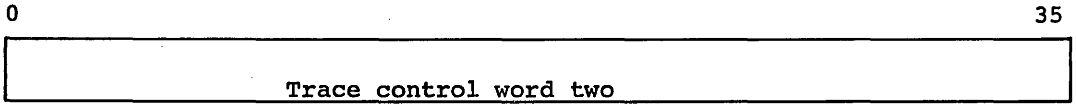
.CRTCT

In trace words one and two, a one bit in any position disables trace entries for that particular type of trace entry. A zero in any position enables tracing.

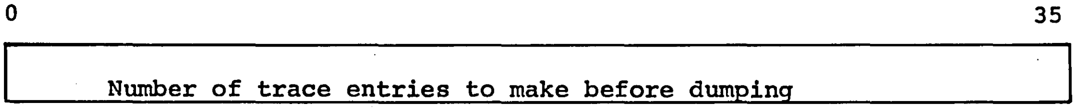
Word 1



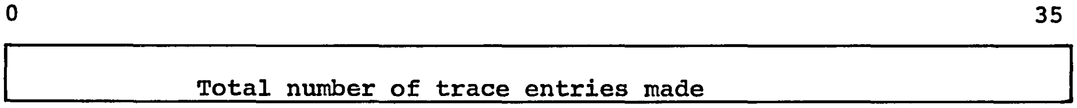
Word 2



Word 3

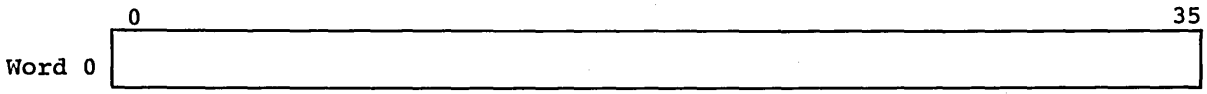


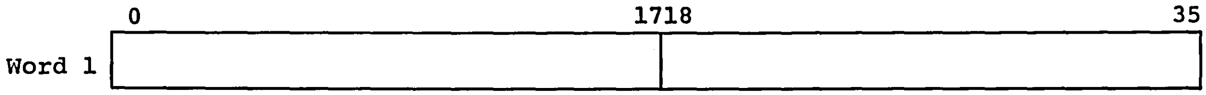
Word 4



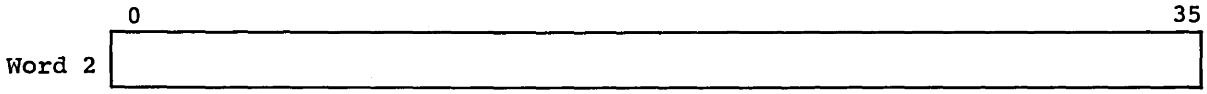
ACCOUNTING FILE

ACTIVITY ACCOUNTING RECORD

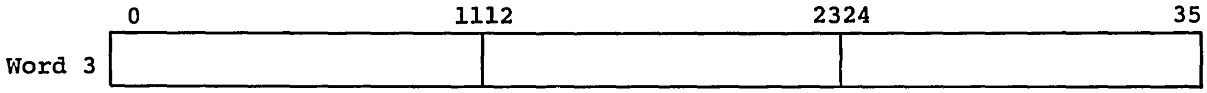




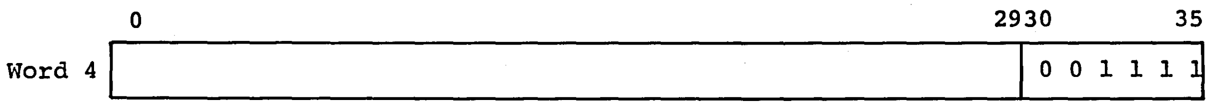
Bits	0-17	Record size
	18-35	Record type (1)



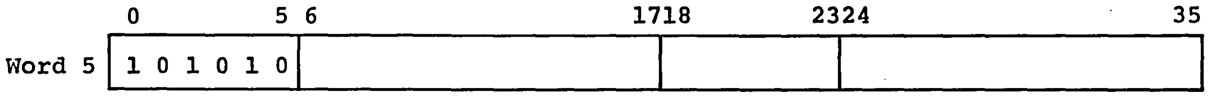
Bits	0-35	GE-600 System identification



Bits	0-35	Date
		0-11, Month
		12-23, Day
		24-35, Year

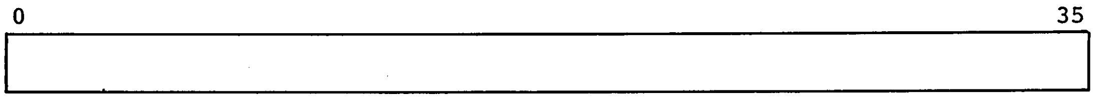


Bits	0-29	SNUMB
	30-35	Octal 17



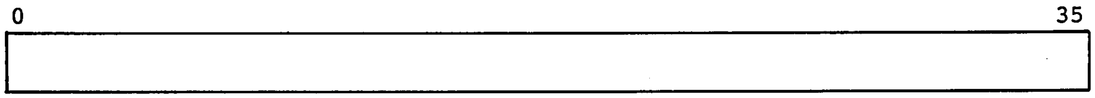
Bits	0-5	Octal 52
	6-17	Activity number
	18-23	EOF type
	24-35	Abort code

Word 6



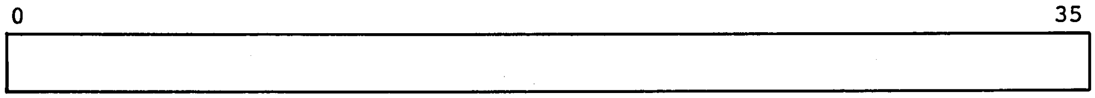
Bits 0-35 Activity mnemonic

Word 7



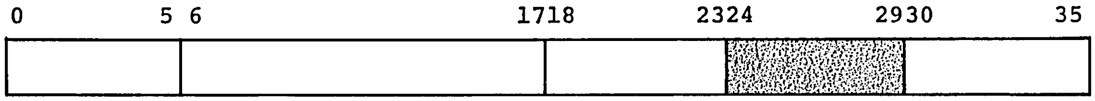
Bits 0-35 Start time (clock pulses)

Word 8



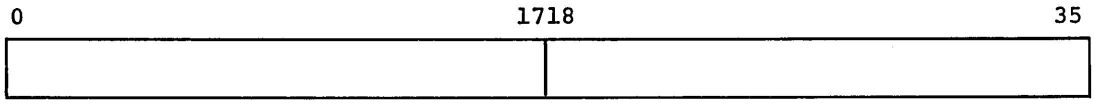
Bits 0-35 Stop time (clock pulses)

Word 9



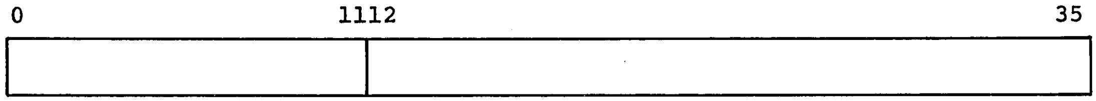
Bits 0-5 Core urgency
6-17 Number of activity compressions
18-23 Current urgency
24-29 Not used
30-35 Initial urgency

Word 10



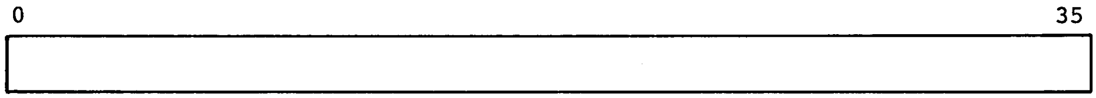
Bits 0-17 SYSOUT limit
18-35 SYSOUT count

Word 11

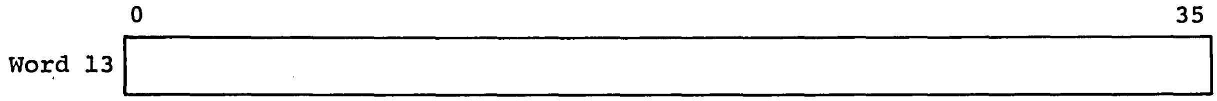


Bits 0-11 Initial number of memory blocks
12-35 Activity sequence number

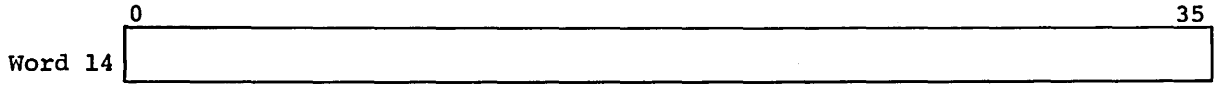
Word 12



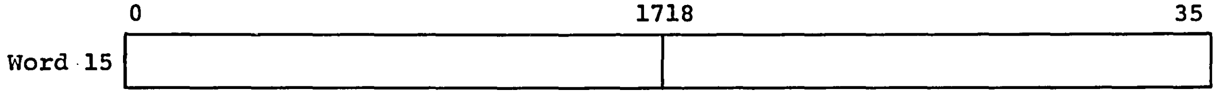
Bits 0-35 Memory-time product (K*MS)



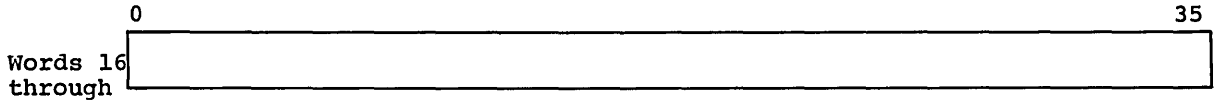
Bits 0-35 Processor time used (clock pulse)



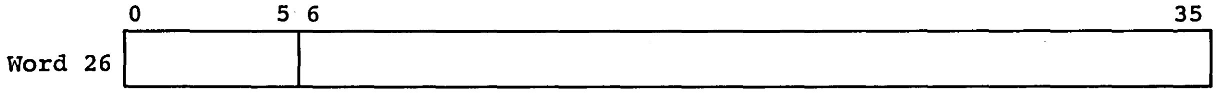
Bits 0-35 Processor time limit (clock pulse)



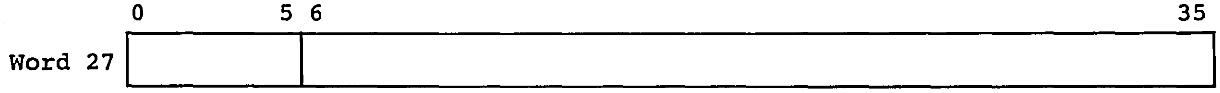
Bits 0-17 Number of peripheral entries
18-35 Number of report entries



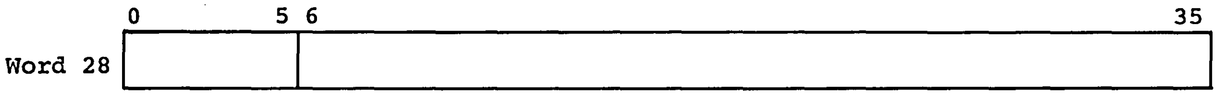
Bits 0-35 Columns 13-72 of $ IDENT card



Bits 0-5 Device
6-35 GECALL time (clock pulses)

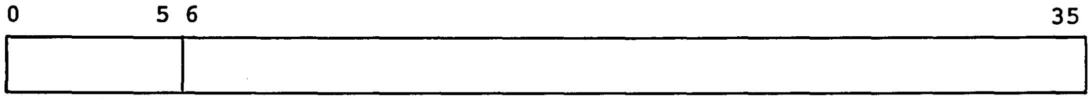


Bits 0-5 Device
6-35 Pushdown time (clock pulses)



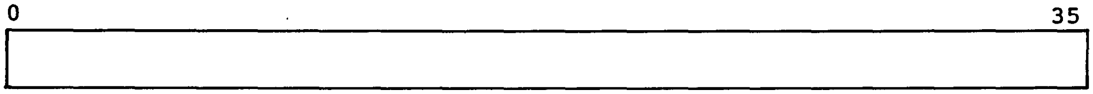
Bits 0-5 Device
6-35 Slave service area I/O time (clock pulses)

Word 29



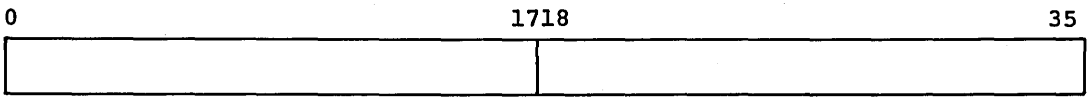
Bits	0-5	Device
	6-35	GESYOT time (clock pulses)

Word 30 through n-2



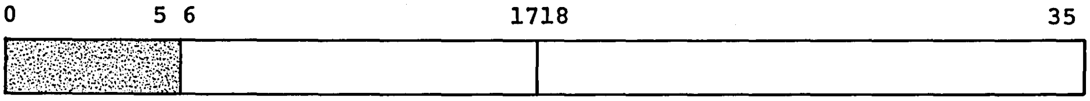
Bits	0-35	Peripheral file entry (5 words per entry) These 5-word entries are detailed on the following pages.

Word n-1



Bits	0-17	Media code
	18-35	Report code

Word n

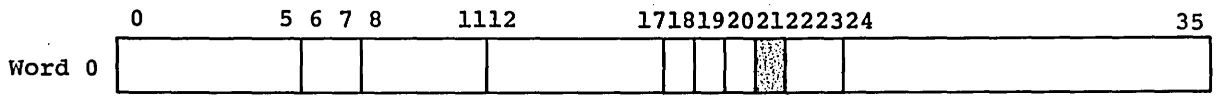


Bits	0-5	Not used
	6-17	Station identification
	18-35	Count

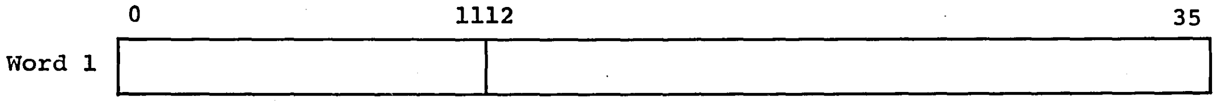
Words n-1 and n are repeated for each SYSOUT report.

PERIPHERAL FILE ENTRIES (on Activity Accounting Record)

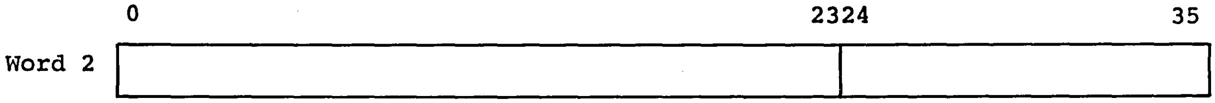
Magnetic Tape



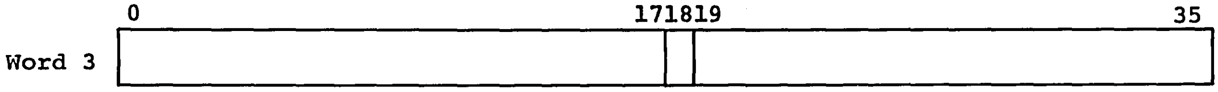
```
Bits     0-5   Device type
         6-17  Address
               6-7, IOC number
               8-11, Channel number
               12-17, Device number
           18= 1, Mount instruction issued (see Word 3)
           19= 1, Tape in Standby, Ready message issued
           20= 1, File name present
           21  Not used
        22-23  Disposition code
        24-35  File code
```



```
Bits     0-11  Alternate address
        12-35  Channel use time (milliseconds)
```

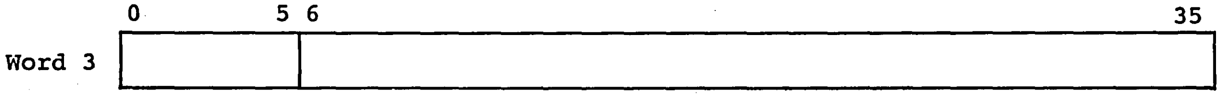


```
Bits     0-23  Number of connects
        24-35  Number of errors
```

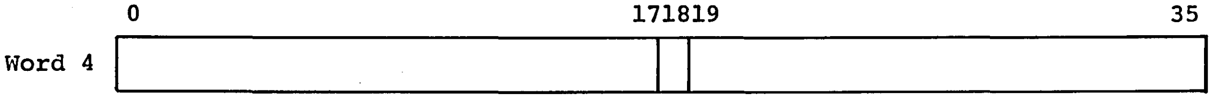


```
Bits     0-17  Starting record
           18  Density (=1, high; =0, low)
        19-35  Starting file
```

If bit 18 in Word 0 =1, Word 3 is as shown below:



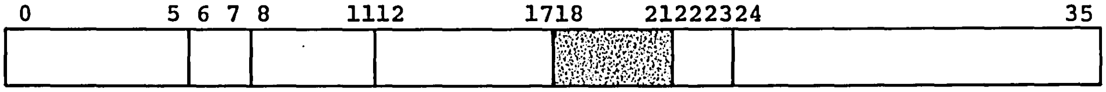
```
Bits     0-5   Octal 13
         6-35  File serial number in BCD, left-justified
```



Bits 0-17 Ending record
18 Density (=1, high; =0, low)
19-35 Ending file

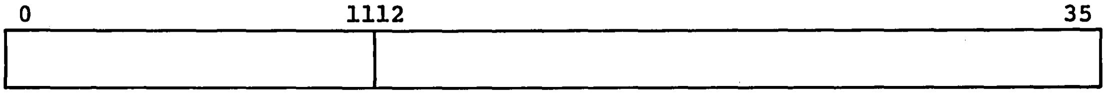
Unit Record Devices

Word 0



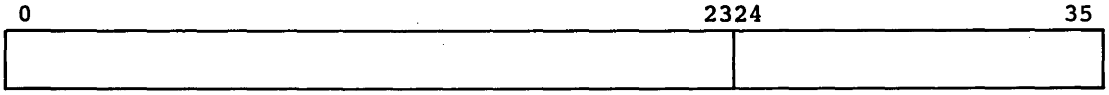
Bits	0-5	Device type
	6-17	Address 6-7, IOC number 8-11, Channel number 12-17, Device number
	18-21	Not used
	22-23	Disposition code
	24-35	File code

Word 1



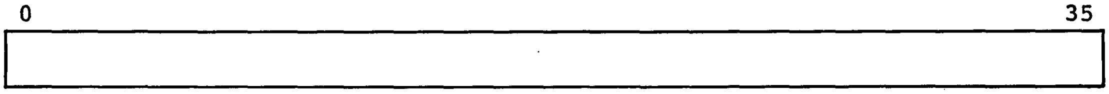
Bits	0-11	Zeros
	12-35	Channel use time (milliseconds)

Word 2



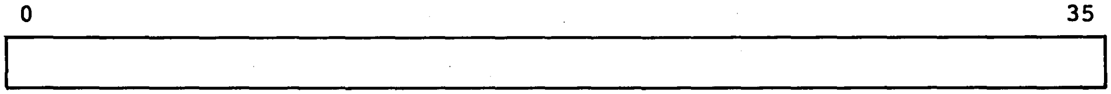
Bits	0-23	Number of connects
	24-35	Number of errors

Word 3



Bits 0-35 Time of allocation

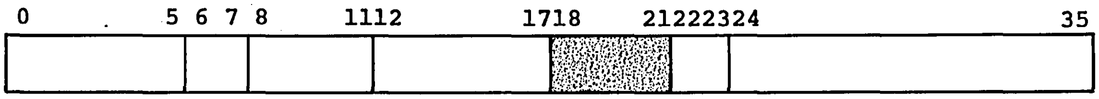
Word 4



Bits 0-35 Time of deallocation

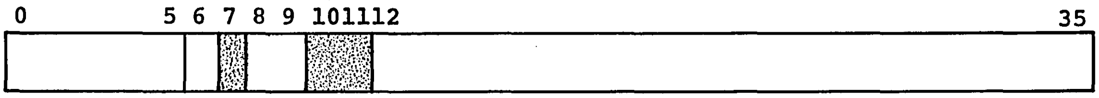
Mass Storage

Word 0



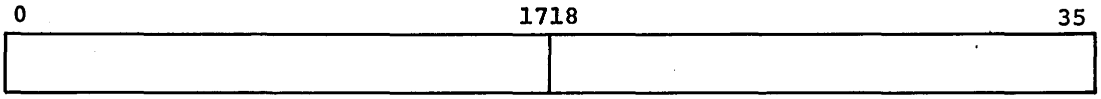
```
Bits    0-5    Device type
        6-17   Address
               6-7, IOC number
               8-11, Channel number
               12-17, Device number
        18-21  Not used
        22-23  Disposition code
        24-35  File code
```

Word 1



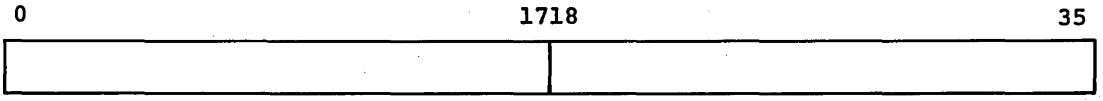
```
Bits    0-5    Catalog hash
          6 = 1, Catalog present
          7    Not used
          8 = 1, Random
            = 0, Linked
          9 = 1, Permanent
            = 0, Temporary
        10-11  Not used
        12-35  Channel use time (milliseconds)
```

Word 2



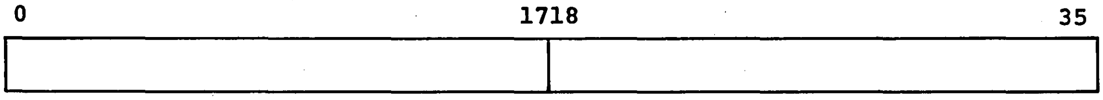
```
Bits     0-17  Initial number of links
        18-35  Final number of links
```

Word 3



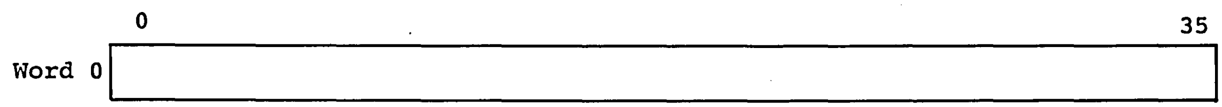
```
Bits     0-17  Initial first link
        18-35  Initial relative block
```

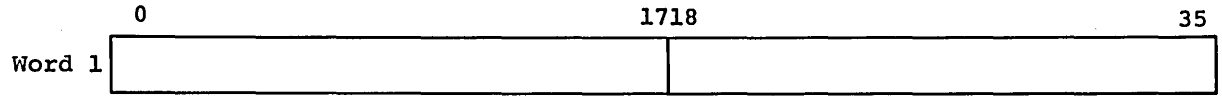
Word 4



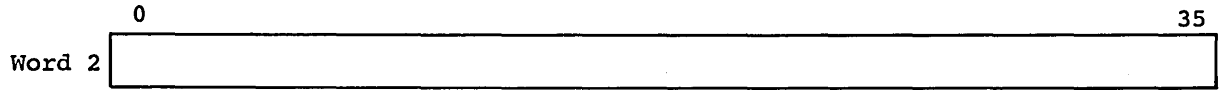
```
Bits     0-17  Final first link
        18-35  Final relative block
```

GEOT ACCOUNTING RECORD

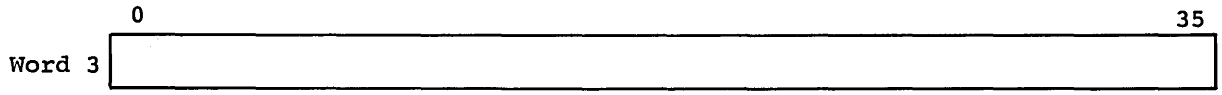




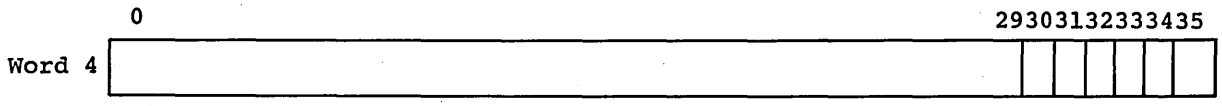
Bits	0-17	Record size (6)
	18-35	Record type (2)



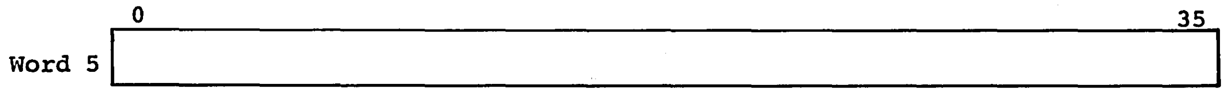
Bits	0-35	System identification from .CRSID



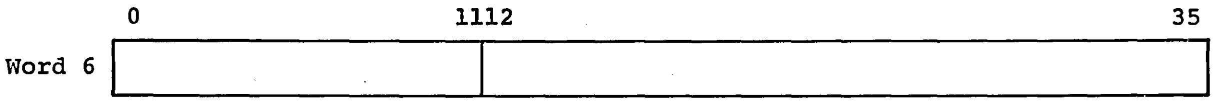
Bits	0-35	Date from .CRDAT



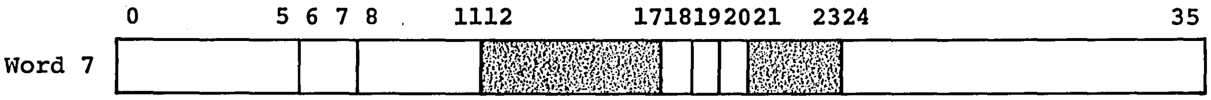
Bits	0-29	SNUMB
	30-35	Destination 30=1, Printer 31=1, Remote J* 32=1, Punch 33=1, Remote number 1 34=1, Remote number 2 35=1, Remote number 3



Bits	0-35	Time of day (clock pulses)

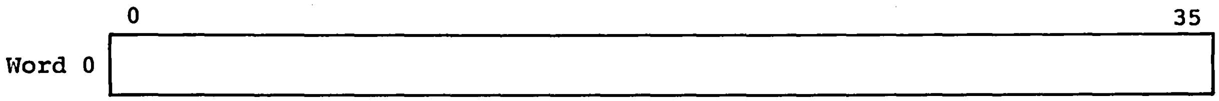


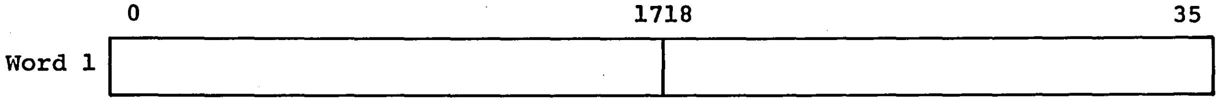
Bits	0-11	Number of reports
	12-35	Number of records



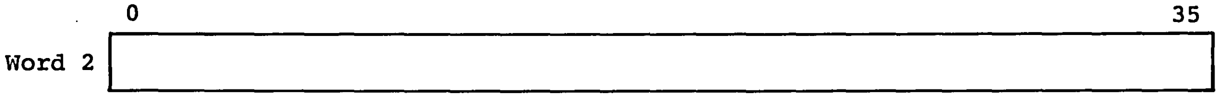
Bits	0-5	Device type
	6-7	IOC number
	8-11	Channel number
	12-17	Not used
	18=	1, Last destination for SYSOUT for this SNUMB
	19=	1, Destination is remote
	20	Zero
	21-23	Not used
	24-35	Remote station identification, if applicable

GEIN ACCOUNTING RECORD

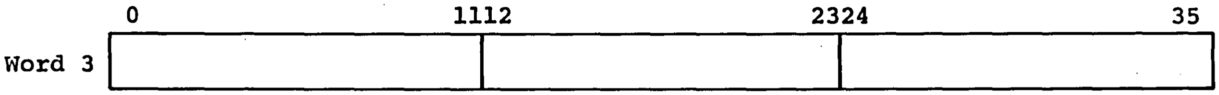




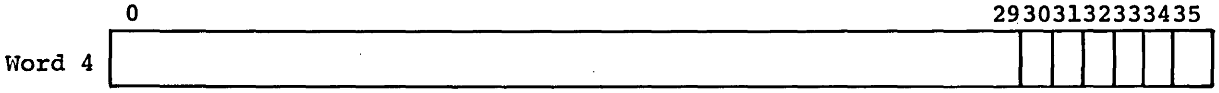
Bits 0-17 Record size (13)
18-35 Record type (5)



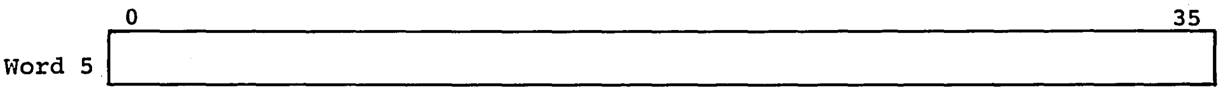
Bits 0-35 GE-600 System identification



Bits 0-35 Date
0-11, Month
12-23, Day
24-35, Year



Bits 0-29 SNUMB
30-35 Destination
30=1, Printer
31=1, Remote J*
32=1, Punch
33=1, Remote number 1
34=1, Remote number 2
35=1, Remote number 3



Bits 0-35 GEIN type (CR, IMC, Remote, T/S)

0 35

Word 6

Bits 0-35 Job start time (clock pulses)

0 35

Word 7

Bits 0-35 Job stop time (clock pulses)

0 35

Word 8

Bits 0-35 Input record count (card count)

0 35

Word 9

Bits 0-35 Core storage size

0 35

Word 10

Bits 0-35 Processor time used

0 35

Word 11

Bits 0-35 IOC time for input device

0 35

Word 12

Bits 0-35 IOC time for *J file

0 35

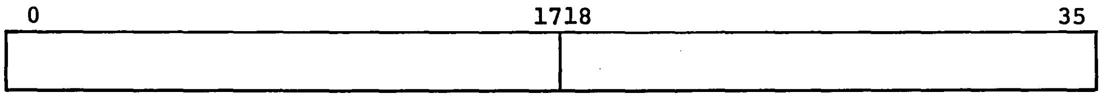
Word 13

Bits 0-35 IOC time for J* file

GESNAP ACCOUNTING RECORD

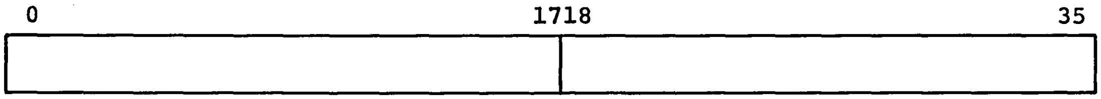
This record type is output to the accounting tape from GESNAP when Printer PR2 is not available and dedicated.

Word 0



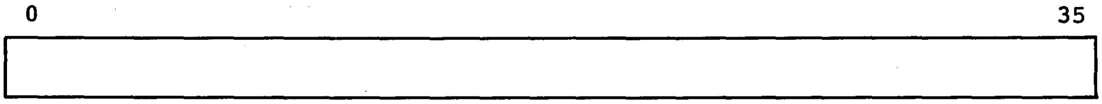
Bits 0-17 Block serial number inserted by ACTFL
18-35 Zero

Word 1



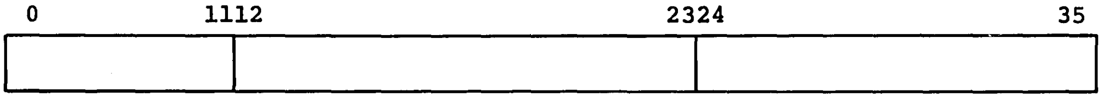
Bits 0-17 Record size (n-1)
18-35 Record type (6)

Word 2



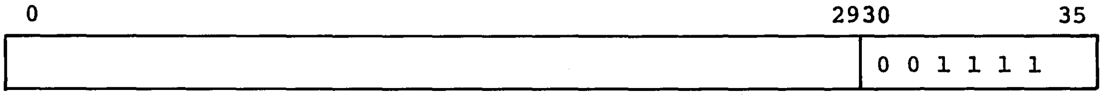
Bits 0-35 GE-600 System identification

Word 3



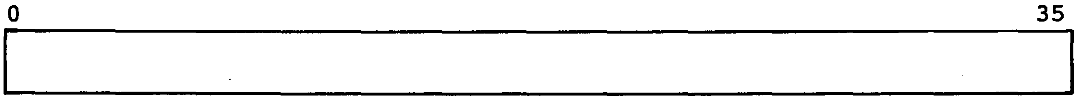
Bits 0-35 Date
0-11, Month
12-23, Day
24-35, Year

Word 4



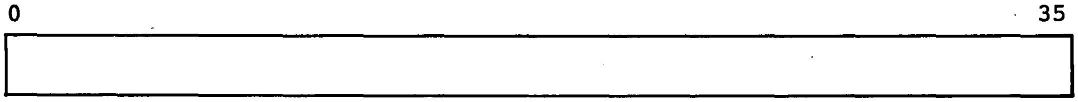
Bits 0-29 SNUMB
30-35 Octal 17

Word 5
thru
n-1



Bits 0-35 One line of formatted printer output.
(ready to be printed)

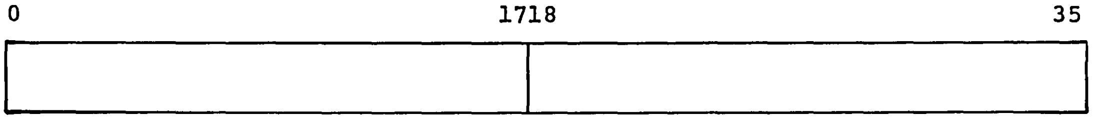
Word 41
≥ n



Bits 0-35 Slew Code

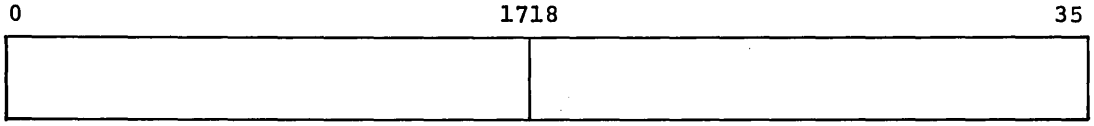
RESTART ACCOUNTING RECORD

Word 0



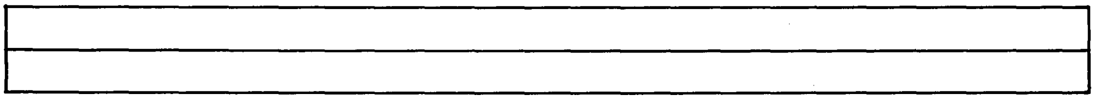
Bits	0-17	=4
	18-35	Zero

Word 1



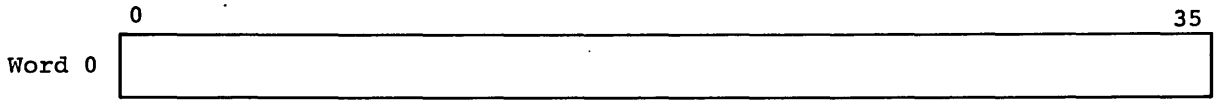
Bits	0-17	=3
	18-35	Record type =9, Normal restart occurred. =10, Next record is suspect. Data saved may be invalid.

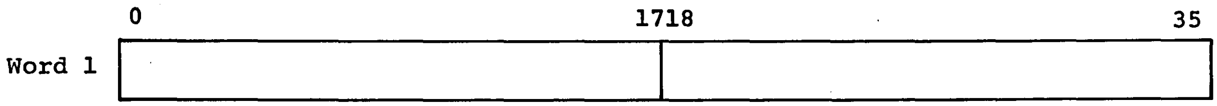
Word 2
thru
Word 4



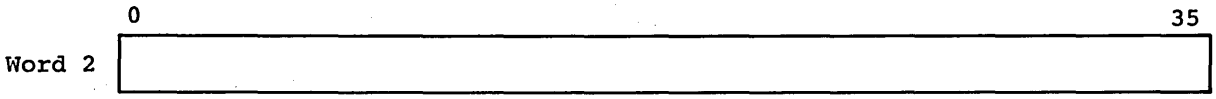
Bits	0-35	RESTART RECORD 00n =1 for normal restart. n is incremented by 1 for each type 10 record written (to a maximum of 3).

ERROR ACCOUNTING RECORD

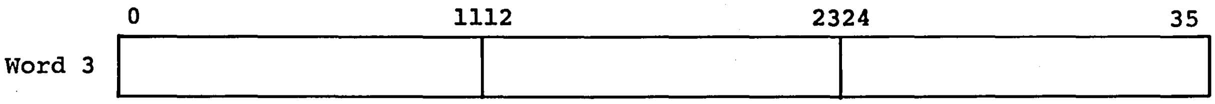




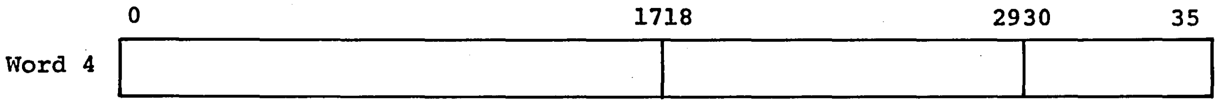
Bits	0-17	Record size (13)
	18-35	Record type (3)



Bits	0-35	GE-600 System identification



Bits	0-35	Date 0-11, Month 12-23, Day 24-35, Year

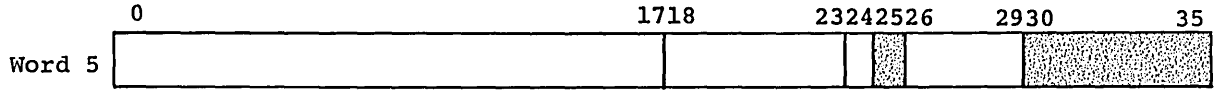


If Slave program,

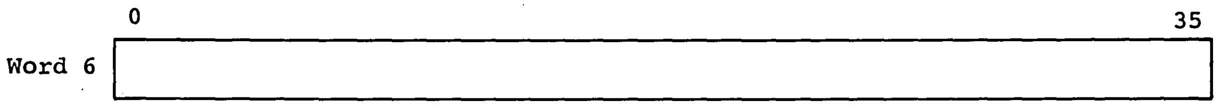
Bits	0-29	SNUMB
	30-35	Ignore character (octal 17)

If Master program,

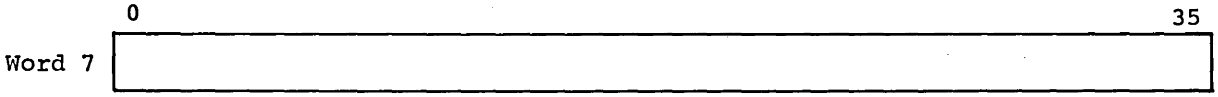
Bits	0-17	Program number
	18-35	Zeros



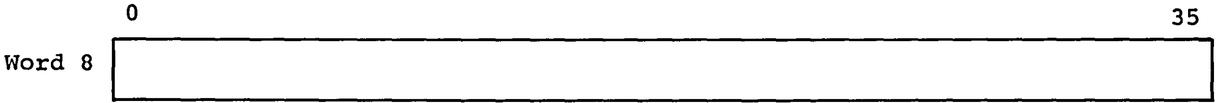
Bits 0-17 If slave, BCD account number; if master, zeros
18-23 Type code
24= 1, if GPROVD requested
25 Not used
26-29 GPROVD special options
=1, card reader or tape data alert
=2, tape "U" option
30-35 Not used



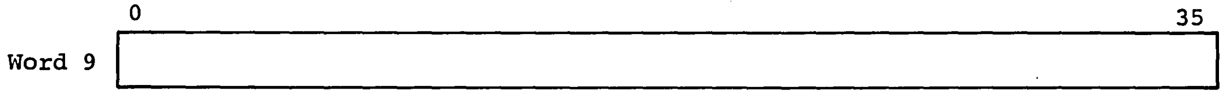
Bits 0-35 Time adjustment for internal clock (64th ms)



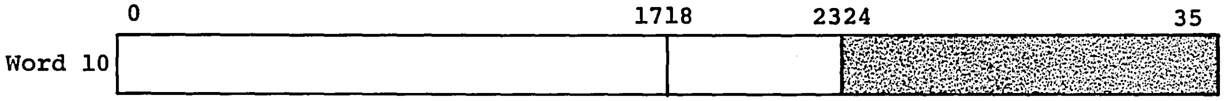
Bits 0-35 Time of interrupt-internal clock (64th ms)



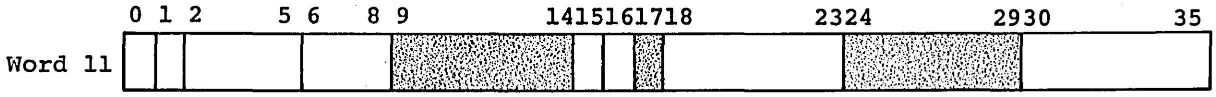
Bits 0-35 Time of connect-internal clock (64th ms)



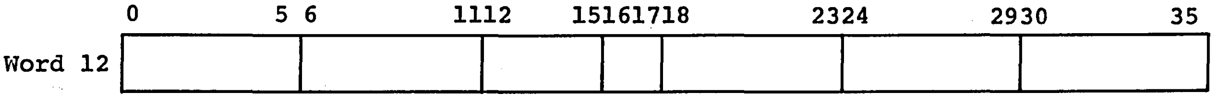
Bits 0-35 Number of connects on device (including this I/O)



Bits 0-17 Number of errors on device (excluding this I/O)
18-23 Error ratio (number of errors/64 connects)
24-35 Not used

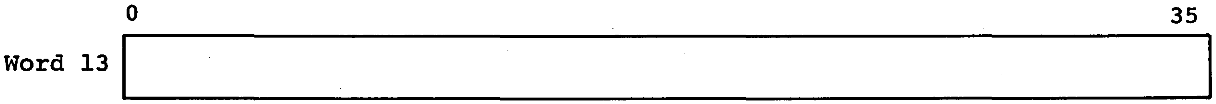


Bit	0	Sync bit
	1	Power bit
	2-5	Major status
	6-8	Substatus
	9-14	Not used
	15=	1, Lost interrupt
	16=	1, Initiation interrupt 0, Termination interrupt
	17	Not used
	18-23	IOC error
	24-29	Not used
	30-35	Record Residue



I/O Command (second if DS/DR)

Bits	0-5	Device command
	6-11	Device number
	12-15	Channel number
	16-17	IOC number
	18-23	IOC command
	24-29	Zeros
	30-35	Record count



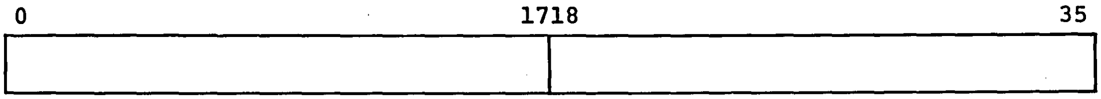
Bits	0-35	DS/DR, Seek address Tape, File serial number

DISPATCHER TABLES

DISPATCHER QUEUE

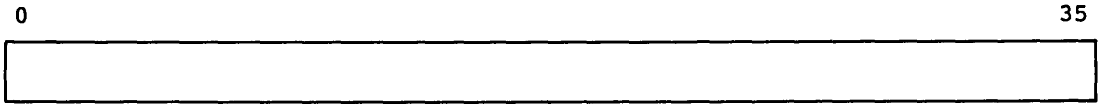
.CRDSP

Word 0



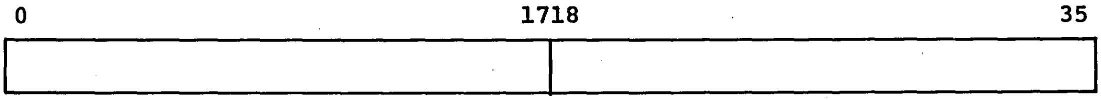
Bits	0-17	Number of processors waiting
	18-35	Number of processors running

Word 1



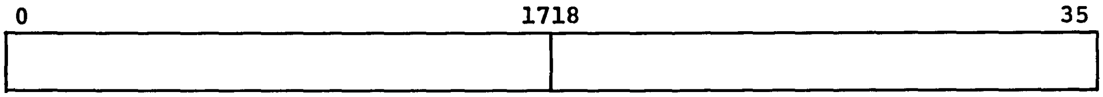
Bits	0-35	Gate

Word 2



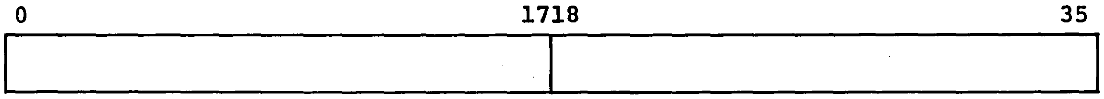
Bits	0-17	Processor number - Always 0
	18-35	Delay until Interrupt Signal (DIS) instruction

Word 3



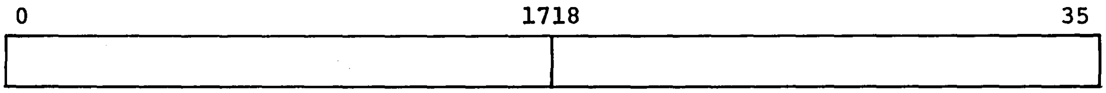
Bits	0-17	Processor number - Always 1
	18-35	DIS instruction

Word 4



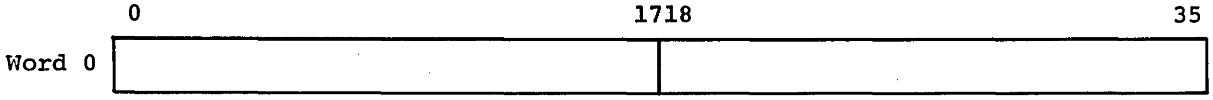
Bits	0-17	Processor number - Always 2
	18-35	DIS instruction

Word 5

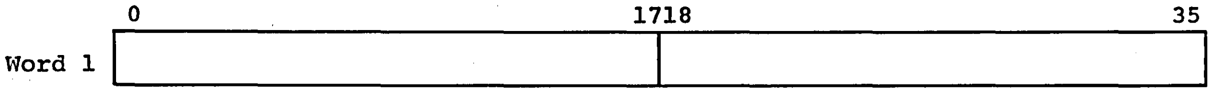


Bits	0-17	-1, Instruction Counter
	18-35	Transfer (TRA) instruction

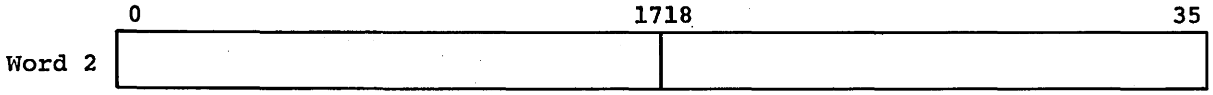
.CRPRQ



Bits	0-17	Program number of first program in the queue
	18-35	Program number of last program in the queue



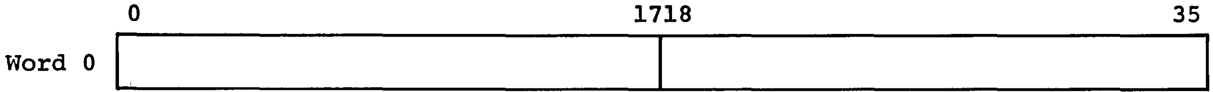
Bits	0-17	Program number of next program in the queue =0, If last entry
	18-35	Program number of previous program in the queue



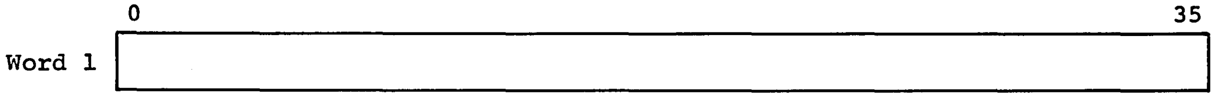
	Bits	0-17	Number of program in execution
		18-35	Number of processor executing
or		0-35	=0, Program not a candidate for execution

COURTESY CALL WAITING QUEUE

.CRCCS

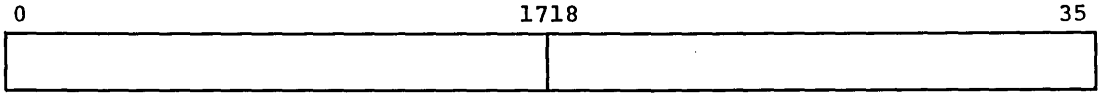


Bits	0-17	Address of first I/O queue in chain =0, None
	18-35	Address of last I/O queue in chain =0, None



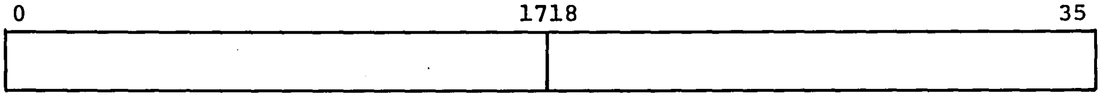
Bits 0-35 Gate

Word 2



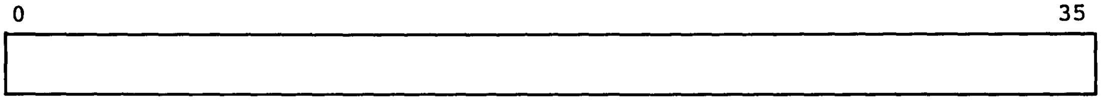
Bits 0-17 Number of entries in the queue
18-35 NOP instruction
(This is changed to a TRA when last courtesy call is paid.)

Word 3



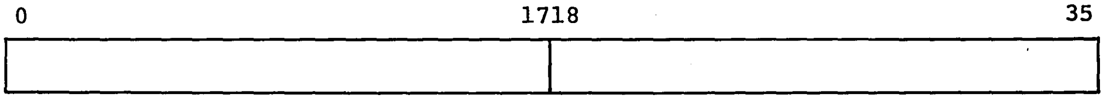
Bits 0-17 TRA instruction
18-35 Transfer Address

.CRLAL



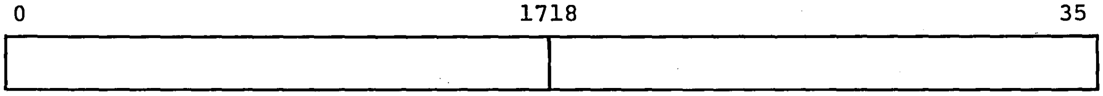
Bits 0-35 Gate

GEPOP



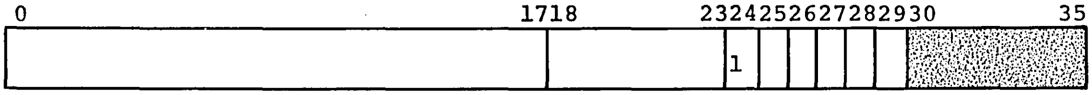
Bits 0-17 Lower Address Limit (LAL) if job is in core
18-35 Zero

SYSOUT



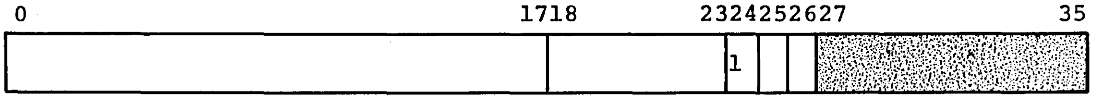
Bits 0-17 LAL if job is in core
18-35 Zero

SYSOUT



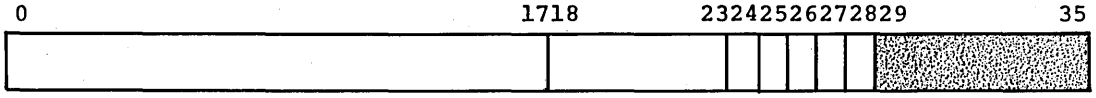
Bits 0-17 =0, Job swapped out
18-23 Urgency
24 Always 1
25 =1, A request has been made to start this program
26 =1, Program is in process of returning to core
27 =1, Operator requested KILL
28 =1, Operator requested TERM
29 =1, New job
30-35 Not used

PERIPH
ALLOC

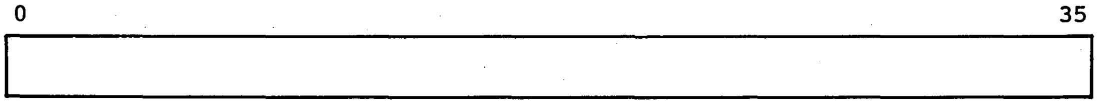


Bits 0-17 LAL if job is in core
=0, Job swapped out
18-23 Urgency
24 Always 1
25 =1, A request has been made to start this program
26 =1, Program is in process of returning to core
27-35 Not used

TIME
SHAR.



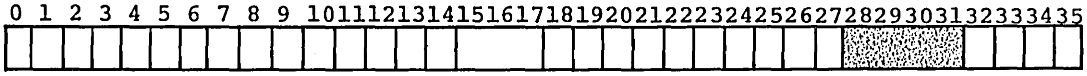
Bits 0-17 LAL if job is in core
=0, Job swapped out
18-23 Urgency
24 Always 1
25 =1, A request has been made to start this program
26 =1, Program is in process of returning to core
27 =1, Operator requested KILL
28 =1, Operator requested TERM
29-35 Not used



Bits 0-35 All zero, No Such Job

.STATE WORD

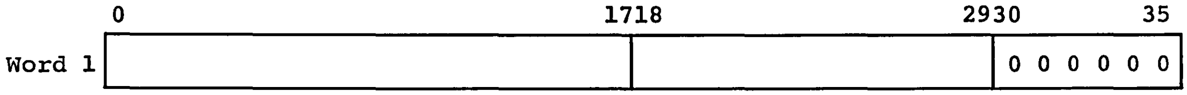
This .STATE WORD is used by all programs having a Slave Service Area.



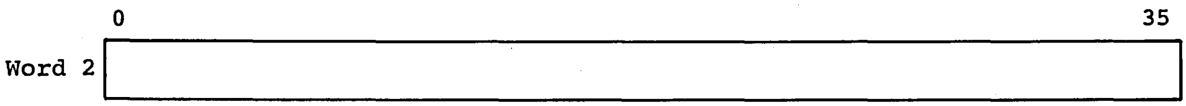
Bit 0 =1, In execution
1 =1, Program number in queue
2 =1, Program loading from SSA
3 =1, GEPR in control
4 =1, GEPR needed
5 =1, Abort in control
6 =1, Abort request received
7 =1, Swap in control
8 =1, Must swap this program
9 =1, Program in Courtesy Call
10 =1, At least one Courtesy Call waiting
11 =1, Waiting for core store size change
12 =1, SYSOUT writing
13 =1, Relinquished
14 =1, Roadblock
15-17 Abort Substate
=000, FINI
=001, FAULT
=010, BORT
=011, KILL/TERM
18 =1, MME .EMM allowed
19 =1, Reset bit 18 after fault
20 =1, A disaster fault has occurred
21 =1, Do not set abort bit on GEPR
22 =1, Wrapup done
23 =1, Enable request while swapping out
24 =1, MME's inhibited for time-sharing system
25 =1, Swap this
26 =1, First half of SSA in use for I/O
27 =1, An I/O has completed since last Link
28-31 Not used
32 =1, No.3 processor cannot execute this
33 =1, No.2 processor cannot execute this
34 =1, No.1 processor cannot execute this
35 =1, No.0 processor cannot execute this

TRACE OUTPUT

TYPE 0 - TERMINATE INTERRUPT

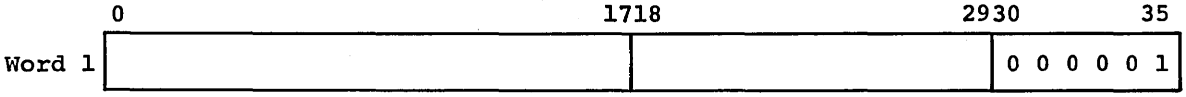


Bits 0-17 Instruction Counter
18-29 Indicator Register
30-35 Type Number

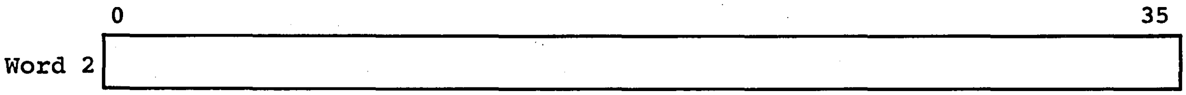


Bits 0-35 Time of day increment in 64th ms, right-justified

TYPE 1 - INITIATION INTERRUPT

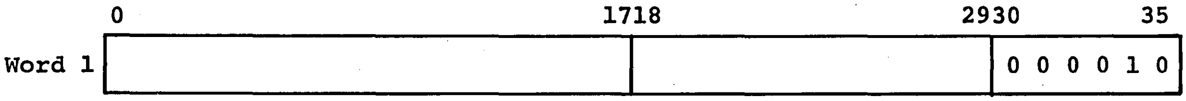


Bits 0-17 Instruction Counter
18-29 Indicator Register
30-35 Type Number

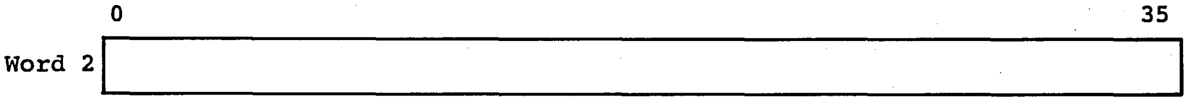


Bits 0-35 Time of day increment in 64th ms, right-justified

TYPE 2 - SPECIAL INTERRUPT



Bits 0-17 Instruction Counter
18-29 Indicator Register
30-35 Type Number



Bits 0-35 Time of day increment in 64th ms, right-justified

TYPE 3 - IOC INTERRUPT COUNTERS

	0 11	12 17	18 23	24 29	30 35
Word 1					0 0 0 0 1 1

Bits 0-11 Zero
12-17 Special Interrupt Counter
18-23 Termination Counter
24-29 Initiation Counter
30-35 Type Number

	0 35
Word 2	Zero

Bits 0-35 Time of day increment in 64th ms, right-justified

TYPE 4 - PROCESS TERMINATE INTERRUPT

	0 5	6 11	12 29	30 35
Word 1				0 0 0 1 0 0

Bits 0-5 Program Number
6-11 Queue Count
12-29 I/O Entry
30-35 Type Number

	0 35
Word 2	

Bits 0-35 Interrupt Queue Entry

TYPE 5 - PROCESS INITIATION INTERRUPT

	0 5	6 11	12 29	30 35
Word 1				0 0 0 1 0 1

Bits 0-5 Program Number
6-11 Queue Count
12-29 I/O Entry
30-35 Type Number

	0 35
Word 2	

Bits 0-35 Interrupt Queue Entry

TYPE 6 - PROCESS SPECIAL INTERRUPT

	0 5	6 11	12 29	30 35
Word 1	▨		▨	0 0 0 1 1 0

Bits	0-5	Not Used
	6-11	Queue Count
	12-29	Not Used
	30-35	Type Number

	0 35
Word 2	

Bits	0-35	Interrupt Queue Entry

TYPE 7 - CONNECT

	0 17	18 29	30 35
Word 1		▨	0 0 0 1 1 1

Bits	0-17	I/O Entry Address
	18-29	Not Used
	30-35	Type Number

	0 35
Wbrd 2	

Bits	0-35	Primary Mailbox including IOC number

TYPE 10 - SSA MODULE - .CALL

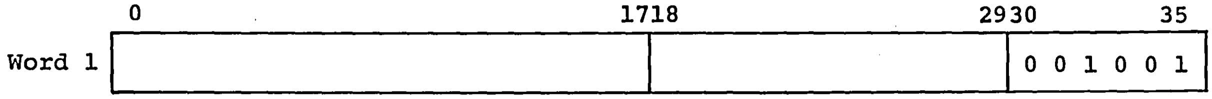
	0 17	18 29	30 35
Word 1			0 0 1 0 0 0

Bits	0-17	Instruction Counter
	18-29	Indicator Register
	30-35	Type Number

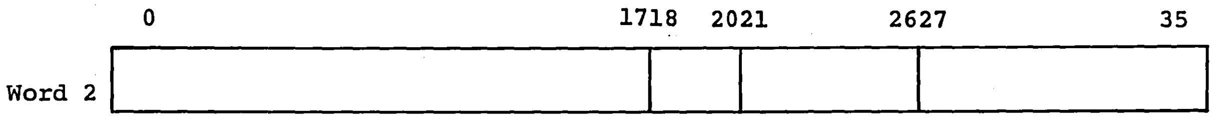
	0 17	18 20	21 26	27 35
Word 2				

Bits	0-17	Module Number
	18-20	Processor Number
	21-26	Program Number
	27-35	Entry Point

TYPE 11 - SSA MODULE - .GOTO

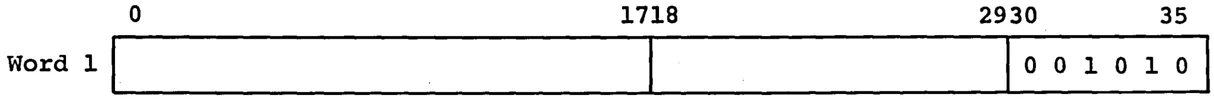


Bits 0-17 Instruction Counter (from)
18-29 Indicator Register (from)
30-35 Type Number

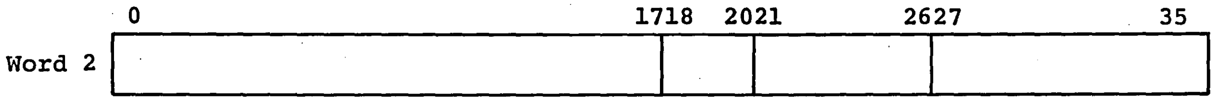


Bits 0-17 Module number
18-20 Processor number
21-26 Program number
27-35 Exit point

TYPE 12 - SSA MODULE - .EXIT

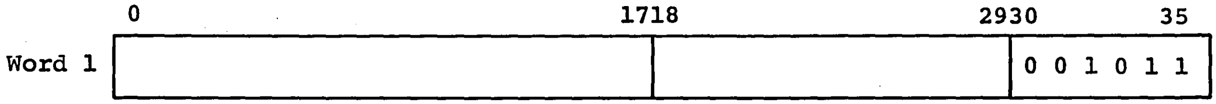


Bits 0-17 Instruction Counter
18-29 Indicator Register (from)
30-35 Type Number

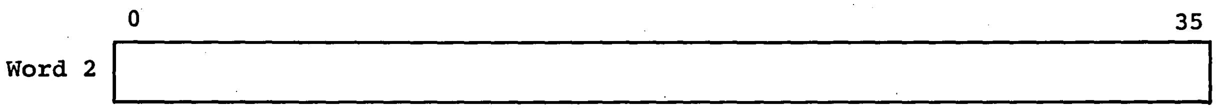


Bits 0-17 Instruction Counter (to)
18-20 Processor Number
21-26 Program Number
27-35 Exit Point

TYPE 13 - DISPATCH TO PROGRAM

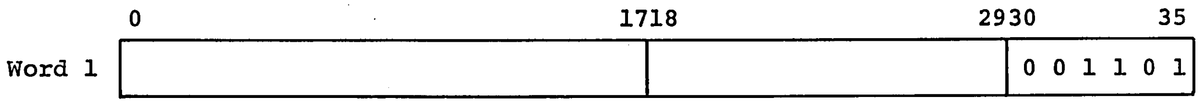


Bits 0-17 Instruction Counter
18-29 Indicator Register
30-35 Type Number

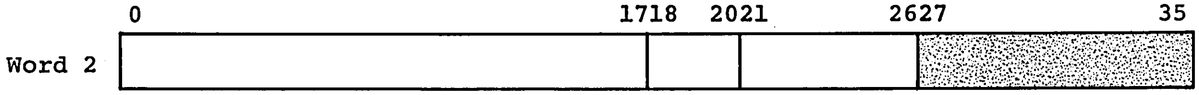


Bits 0-35 Time of day increment in 64th ms, right-justified

TYPE 15 - MME FAULT

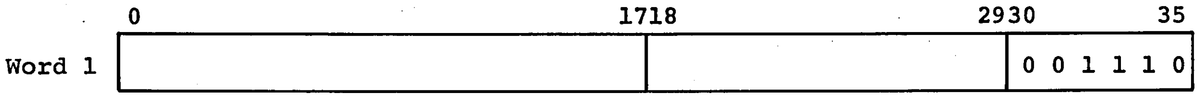


Bits 0-17 Instruction Counter
18-29 Indicator Register
30-35 Type Number

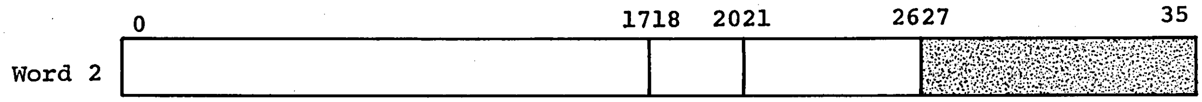


Bits 0-17 MME Type
18-20 Processor Number
21-26 Program Number
27-35 Not Used

TYPE 16 - OTHER FAULT

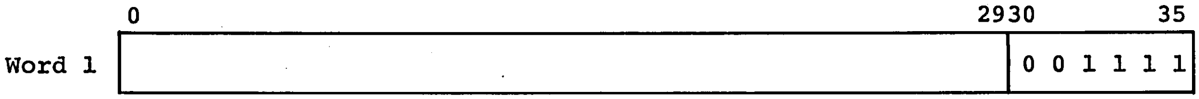


Bits 0-17 Instruction Counter
18-29 Indicator Register
30-35 Type Number

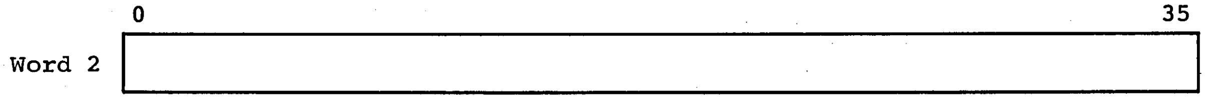


Bits 0-17 Fault Type
18-20 Processor Number
21-26 Program Number
27-35 Not Used

TYPE 17 - RETURN FROM INTERRUPT PROCESSOR



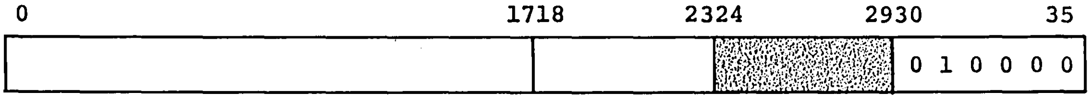
Bits 0-29 All zeros = no program, ≠ zero = program
30-35 Type Number



Bits 0-35 Time of day increment in 64th ms, right-justified

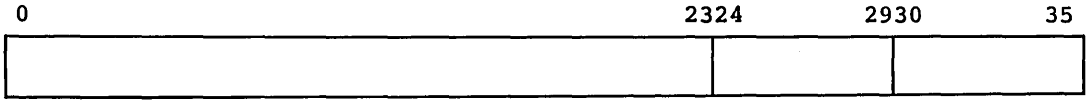
TYPE 20 - GEPR

Word 1



Bits	0-17	Absolute address of device SCT
	18-23	Program Number
	24-29	Not Used
	30-35	Type Number

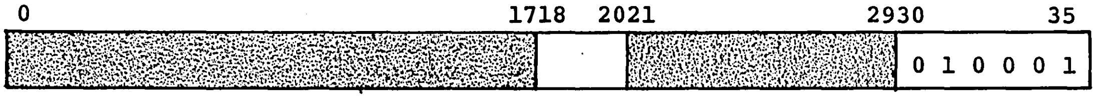
Word 2



Bits	0-23	Status Return Word 1
	24-29	4 * Channel Number
	30-35	Status Return Word 1

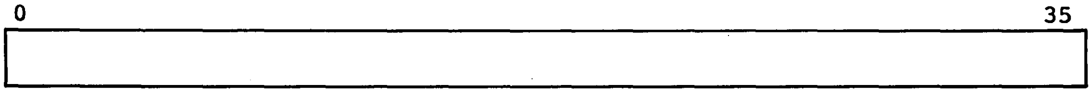
TYPE 21 - PROCESSOR IDLE

Word 1



Bits	0-17	Not Used
	18-20	Processor Number
	21-29	Not Used
	30-35	Type Number

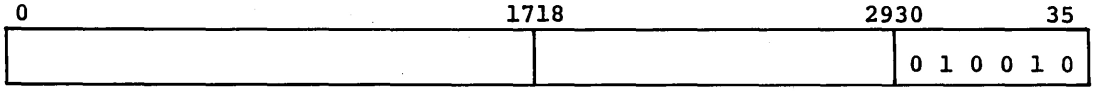
Word 2



Bits	0-35	Time of Day

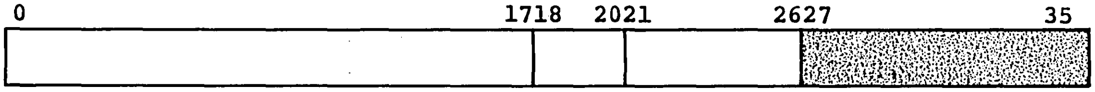
TYPE 22 - ENTER STIO

Word 1



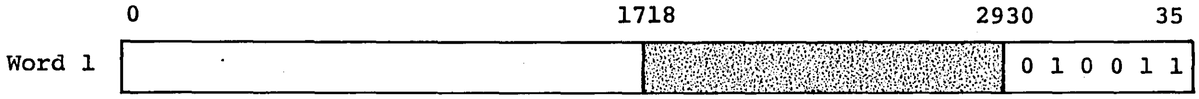
Bits	0-17	Instruction Counter of CALL
	18-29	Logical Primary Channel Index
	30-35	Type Number

Word 2

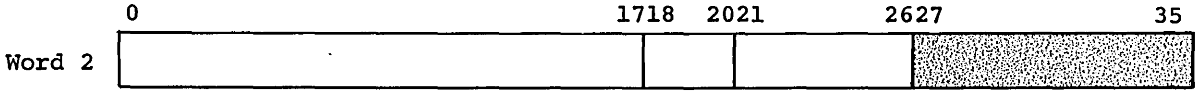


Bits	0-17	I/O Entry
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

TYPE 23 - LEAVE STIO

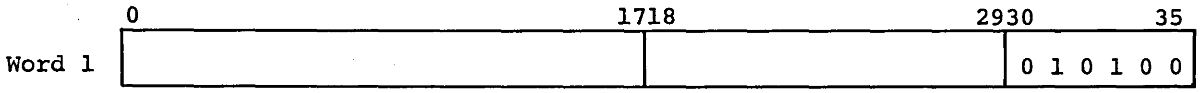


Bits	0-17	Instruction Counter of CALL
	18-29	Not Used
	30-35	Type Number

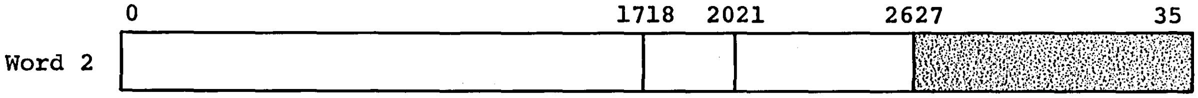


Bits	0-17	0 - not started, I/O Entry - started
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

TYPE 24 - ENTER STRET

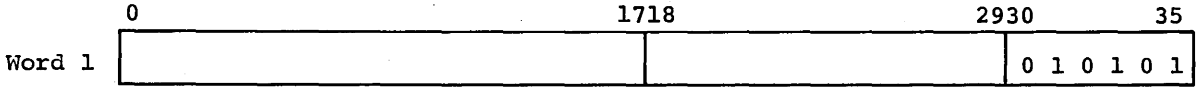


Bits	0-17	Instruction Counter of CALL
	18-29	Logical Primary Channel Index
	30-35	Type Number

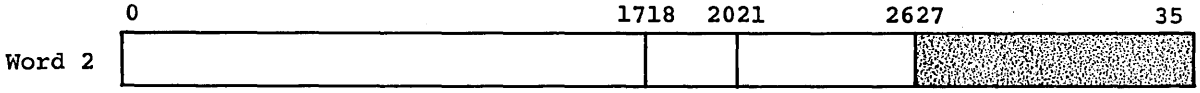


Bits	0-17	I/O Entry
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

TYPE 25 - LEAVE STRET

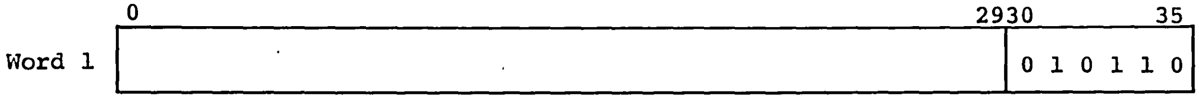


Bits	0-17	Instruction Counter of CALL
	18-29	Logical Primary Channel Index
	30-35	Type Number

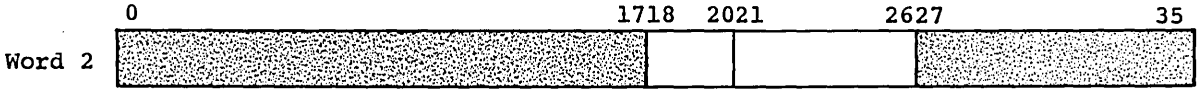


Bits	0-17	I/O Entry
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

TYPE 26 - ROADBLOCK BROKEN, PROGRAM NUMBER INTO DISPATCHER QUEUE

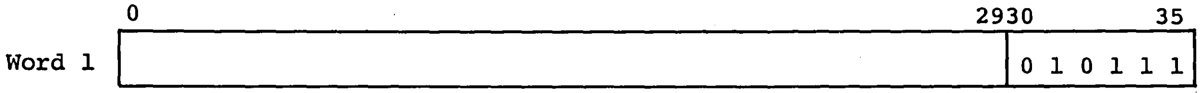


Bits	0-29	Zero
	30-35	Type Number

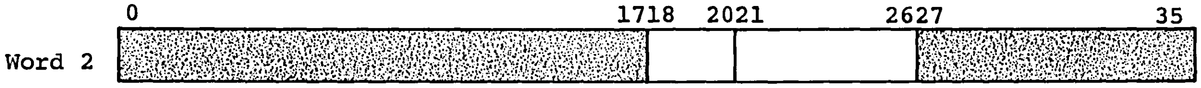


Bits	0-17	Not Used
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

TYPE 27 - RELINQUISH BROKEN, PROGRAM NUMBER INTO DISPATCHER QUEUE

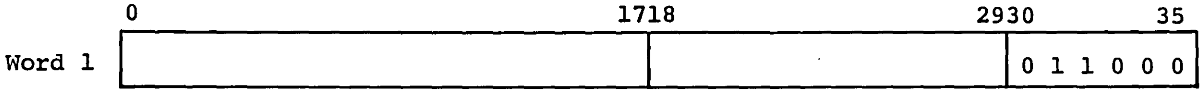


Bits	0-29	Zeros
	30-35	Type Number

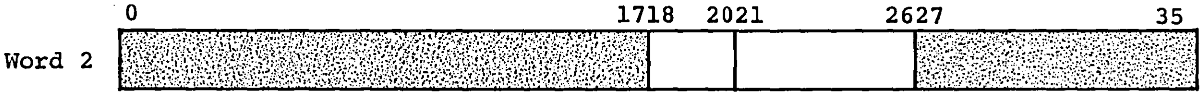


Bits	0-17	Not Used
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

TYPE 30 - INTERRUPTED PROGRAM IN COURTESY CALL, FRONT OF DISPATCHER QUEUE



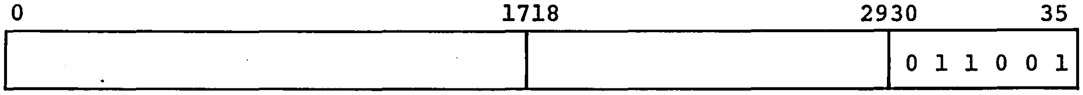
Bits	0-17	Program Number Instruction Counter
	18-29	Indicator Register
	30-35	Type Number



Bits	0-17	Not used
	18-20	Processor Number
	21-26	Program Number
	27-35	Not Used

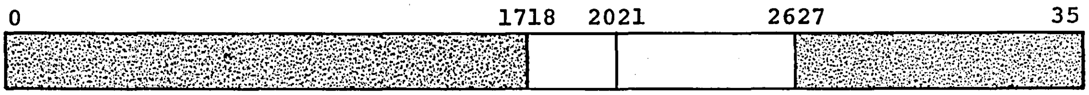
TYPE 31 - INTERRUPTED PROGRAM MAIN LEVEL, END OF DISPATCHER QUEUE

Word 1



Bits 0-17 Program Number Instruction Counter
18-29 Indicator Register
30-35 Type Number

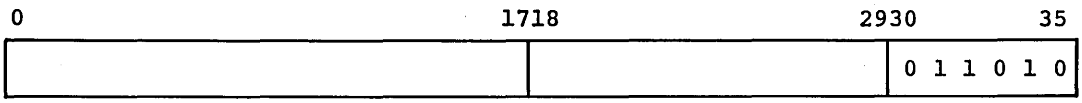
Word 2



Bits 0-17 Not Used
18-20 Processor Number
21-26 Program Number
27-35 Not Used

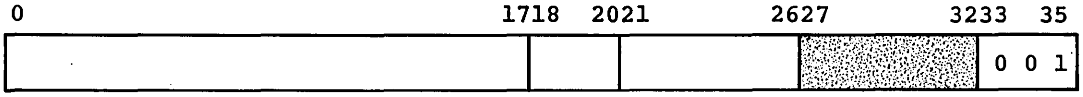
TYPE 32 - CALL CHANNEL MODULE, INTERRUPT EP1

Word 1



Bits 0-17 I/O Entry
18-29 Logical Channel Index
30-35 Type Number

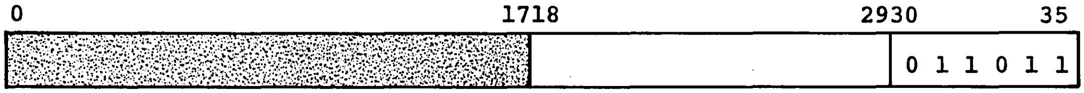
Word 2



Bits 0-17 Module Number
18-20 Processor Number
21-26 Program Number
27-32 Not Used
33-35 Type Number

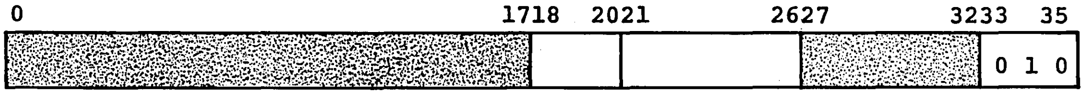
TYPE 33 - CALL CHANNEL MODULE, REQUEST EP2

Word 1



Bits 0-17 Not Used
18-29 Logical Channel Index
30-35 Type Number

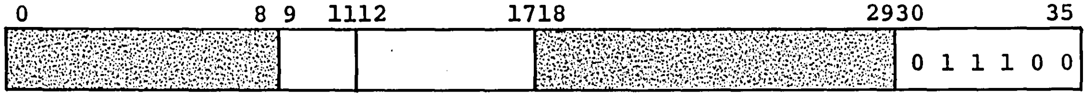
Word 2



Bits 0-17 Module Number
18-20 Processor Number
21-26 Program Number
27-32 Not Used
33-35 Type Number

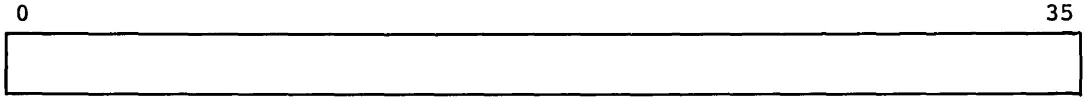
TYPE 34 - START GEPR

Word 1



Bits	0-8	Not Used
	9-11	Processor Number
	12-17	Program Number
	18-29	Not Used
	30-35	Type Number

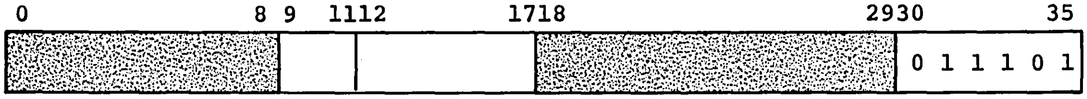
Word 2



Bits	0-35	Time of Day

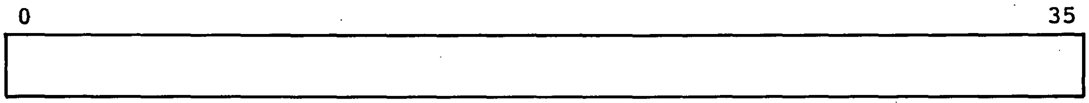
TYPE 35 - START ABORT

Word 1



Bits	0-8	Not Used
	9-11	Processor Number
	12-17	Program Number
	18-29	Not Used
	30-35	Type Number

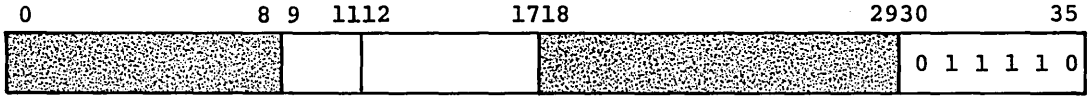
Word 2



Bits	0-35	Time of Day

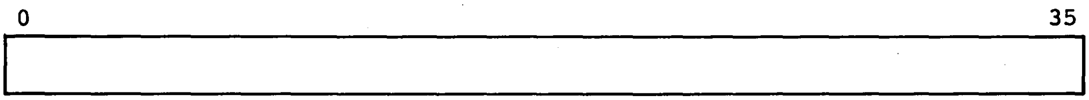
TYPE 36 - START SWAP

Word 1



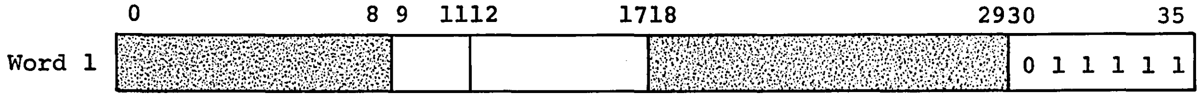
Bits	0-8	Not Used
	9-11	Processor Number
	12-17	Program Number
	18-29	Not Used
	30-35	Type Number

Word 2

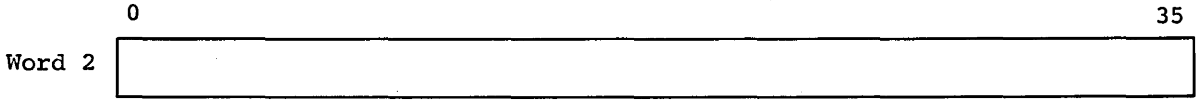


Bits	0-35	Time of Day

TYPE 37 - START COURTESY CALL

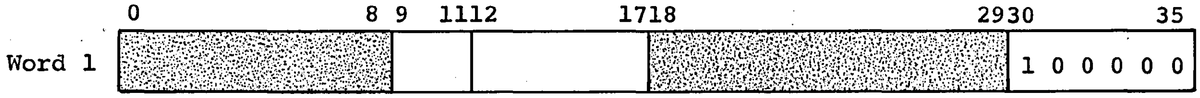


Bits 0-8 Not Used
9-11 Processor Number
12-17 Program Number
18-29 Not Used
30-35 Type Number



Bits 0-35 Time of Day

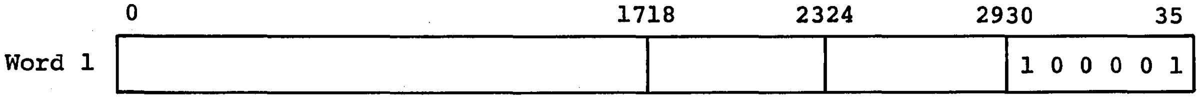
TYPE 40 - LEAVE GEPR, ABORTS, SWAPS, COURTESY CALL



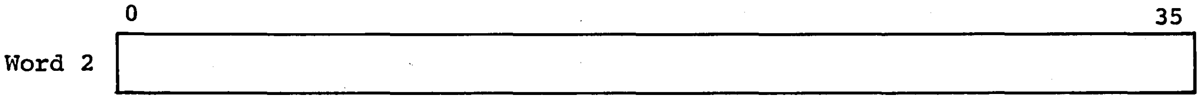
Bits 0-8 Not Used
9-11 Processor Number
12-17 Program Number
18-29 Not Used
30-35 Type Number

Bits 0-35 Time of Day

TYPE 41 - ENABLE PROGRAM



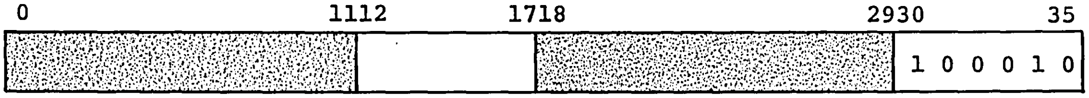
Bits 0-17 Enable Program Number
18-23 Program Number
24-29 Urgency
30-35 Type Number



Bits 0-35 Time of Day

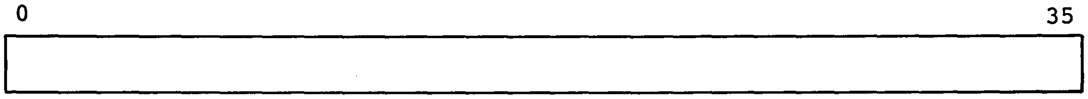
TYPE 42 - START JOB

Word 1



Bits	0-11	Not Used
	12-17	Program Number
	18-29	Not Used
	30-35	Type Number

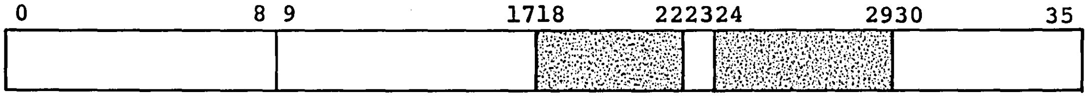
Word 2



Bits	0-35	Time of Day

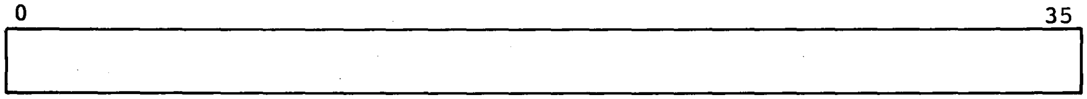
TYPE 43 - START COMPACTION/SWAP

Word 1



Bits	0-8	Program number
	9-17	Number jobs to be moved or swapped
	18-22	Not used
	23	0 = Move, 1 = Swap
	24-29	Not used
	30-35	Type number

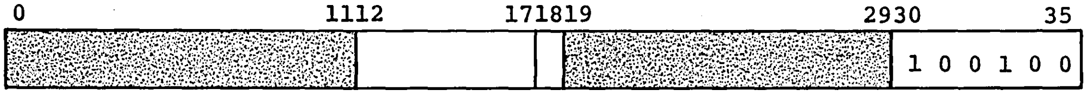
Word 2



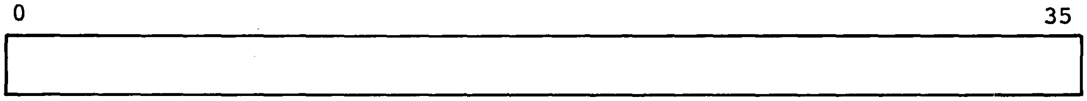
Bits	0-35	Time of Day

TYPE 44 - END OF COMPACTION/SWAP

Word 1



Bits	0-11	Not Used
	12-17	Program Number
	18=	0, Move 1, Swap
	19-29	Not Used
	30-35	Type Number

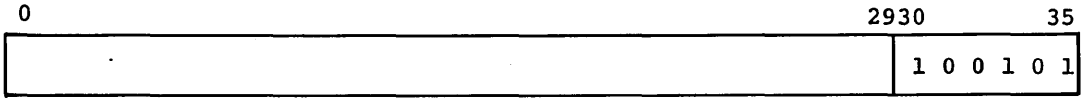


Word 2

Bits	0-35	Time of Day

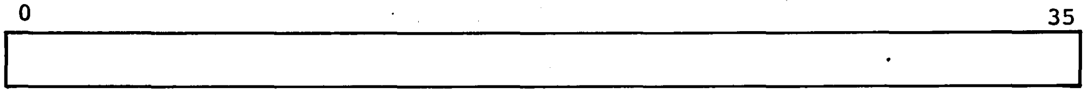
TYPE 45 - GEIN BEGUN

Word 1



Bits	0-29	SNUMB
	30-35	Type Number

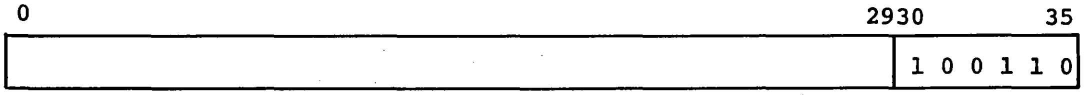
Word 2



Bits	0-35	Time of Day

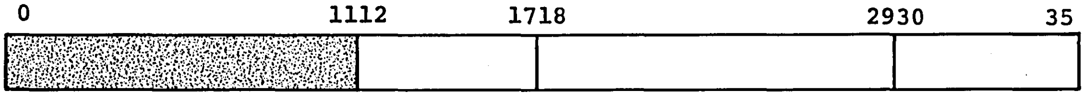
TYPE 46 - PROGRAM NUMBER ASSIGNED

Word 1



Bits	0-29	SNUMB
	30-35	Type Number

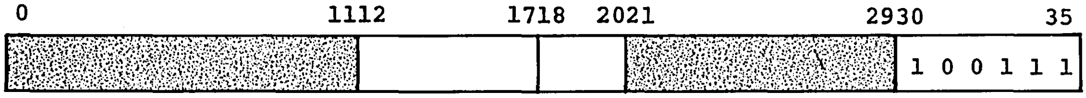
Word 2



Bits	0-11	Not Used
	12-17	Program Number
	18-29	Station Type
	30-35	Type

TYPE 47 - JOB TO PERIPHERAL ALLOCATION

Word 1

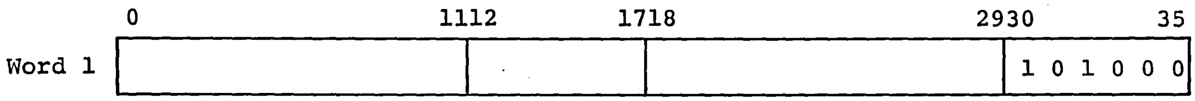


Bits	0-11	Not Used
	12-17	Program Number
	18-20	0 = CR, 1 = IMCV, 2 = REMOTE GEIN, 3 = FROM T/S
	21-29	Not Used
	30-35	Type Number

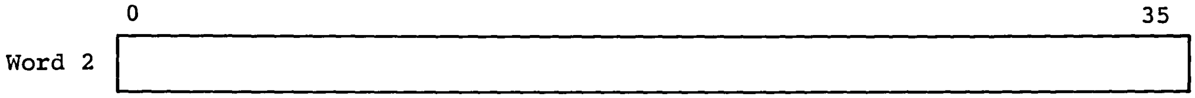
Word 2

Bits	0-35	Time of Day

TYPE 50 - ACTIVITY TO CORE ALLOCATOR

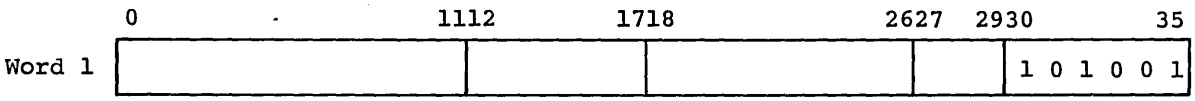


Bits	0-11	Activity Number
	12-17	Program Number
	18-29	Urgency
	30-35	Type Number

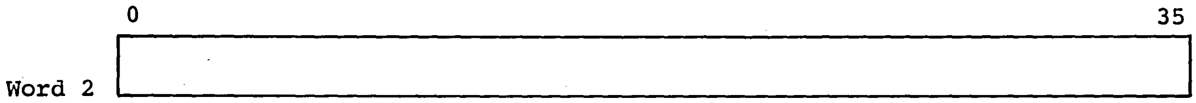


Bits	0-35	Time of Day

TYPE 51 - EOJ, EOX, ACTIVITY

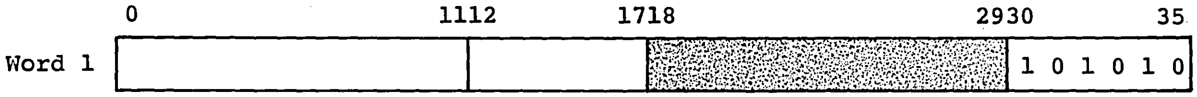


Bits	0-11	Abort Code
	12-17	Program Number
	18-26	Urgency
	27-29	0 = EOA, 1 = EOJ, 2 = ABORT
	30-35	Type Number

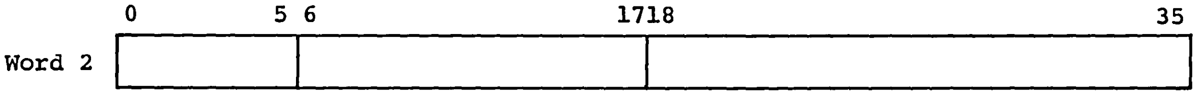


Bits	0-35	Time of Day

TYPE 52 - CANT ALLOCATE



Bits	0-11	Job Status
	12-17	Program Number
	18-29	Not Used
	30-35	Type Number



Bits	0-5	Device Type
	6-17	Number Needed
	18-35	Activity Sequence Number

TYPE 53 - LINK REFUSAL

```
        0                          1718      2324      2930        35
Word 1  |                            |         |         |1 0 1 0 1 1|
```

Bits	0-17	SCT Pointer
	18-23	Entry Number
	24-29	Program Number
	30-35	Type Number

```
        0                          1718                            35
Word 2  |                            |                               |
```

Bits	0-17	Number Requested
	18-35	Number Remaining

TYPE 54 - SNUMB READY FOR OUTPUT

```
        0                                              2930        35
Word 1  |                                                |1 0 1 1 0 0|
```

Bits	0-29	SNUMB
	30-35	Type Number

```
        0                                                          35
Word 2  |                                                            |
```

Bits	0-35	Time of Day

TYPE 55 - SNUMB STARTED PRINTING

```
        0                                              2930        35
Word 1  |                                                |1 0 1 1 0 1|
```

Bits	0-29	SNUMB
	30-35	Type Number

```
        0                                                          35
Word 2  |                                                            |
```

Bits	0-35	Time of Day

TYPE 56 - SNUMB STARTED PUNCHING

	0 29	30 35
Word 1		1 0 1 1 1 0

Bits 0-29 SNUMB
30-35 Type Number

	0 35
Word 2	

Bits 0-35 Time of Day

TYPE 57 - SNUMB FINISHED PRINTING

	0 29	30 35
Word 1		1 0 1 1 1 1

Bits 0-29 SNUMB
30-35 Type Number

	0 35
Word 2	

Bits 0-35 Time of Day

TYPE 60 - SNUMB FINISHED PUNCHING

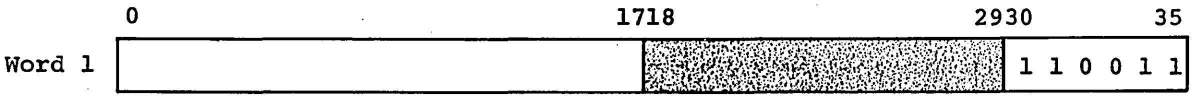
	0 29	30 35
Word 1		1 1 0 0 0 0

Bits 0-29 SNUMB
30-35 Type Number

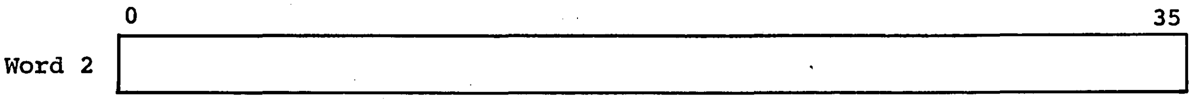
	0 35
Word 2	

Bits 0-35 Time of Day

TYPE 63 - IDLE CHANNEL AFTER PROCESSING II OR TI

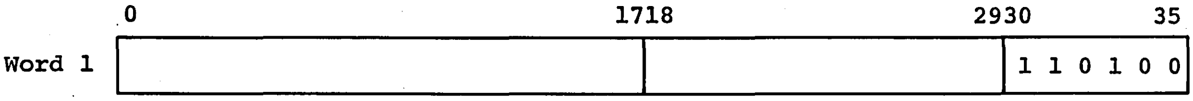


Bits	0-17	True Channel Index
	18-29	Not Used
	30-35	Type Number

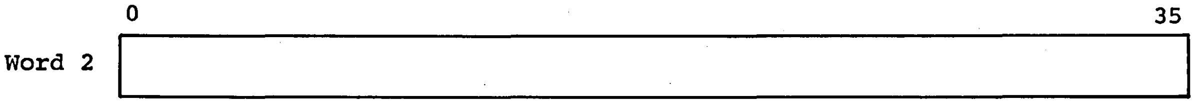


Bits	0-35	Time of Day

TYPE 64 - LENGTH OF CHANNEL QUEUE AT CONNECT



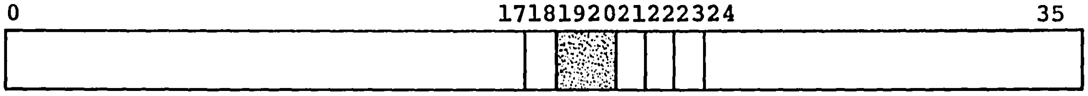
Bits	0-17	Length of Queue
	18-29	True Channel Index
	39-35	Type Number



Bits	0-35	Time of Connect

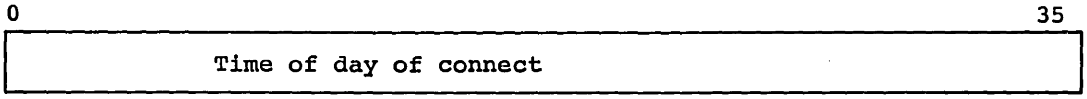
IOS TABLES
(Indexed by True Channel)
(.CRIO1, .CRIO2, .CRIO3, .CRIO4)

Word 1

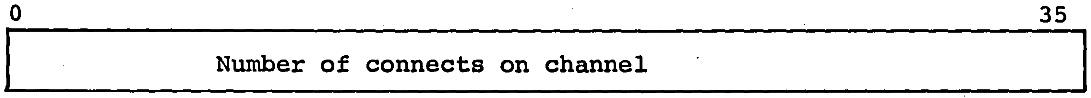


Bits 0-17 = Absolute address of active I/O entry
If zero, nonbusy
18 = 1, SI has occurred
19-20 = Not used
21 = 1, Primary channel on crossbar
22 = 1, Don't use this channel
23 = 1, Crossbar on this channel
24-35 = True channel index of next crossbar (IOC No. multiplied by 256) plus (Channel No. multiplied by 4)

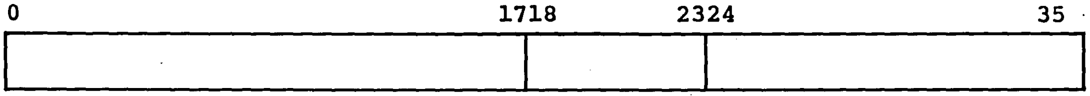
Word 2



Word 3



Word 4



Bits 0-17 = No. of errors on channel
18-23 = No. of errors in last n connects
24-35 = Logical Channel index (IOC No. multiplied by 256) plus (Channel No. multiplied by 4)

I/O STREAM TABLES

PRIMARY CHANNEL S.C.T. TABLE ENTRY (4-word)(.CRCT1,.CRCT2,.CRCT3,.CRCT4)
(Table indexed by logical channel)

	0 5	6	7 8	9	10	11	12 15	16 17	18	19	20	21	22	23	24 35
Word 1			0 0												

Bits 0-5 Device Type
6 = 1, Logical channel or device is temporarily stopped
7-8 Must be zero
9 = 1, GESPEC allowed on this channel
10 = 1, GESPECed entry linked for this channel
11 = 1, MME GESPEC requested on this channel
12-15 Primary channel number
16-17 IOC number
18 = 0, Assignable
= 1, Not Assignable
19 = 0, Available
1, Assigned
20 = 0, Not Dedicated
1, Dedicated
21 = 0, Primary
1, Secondary
22 = 0, Single File
1, Shared File
23 = 0, Single Device
1, Multiple Device
24-35 True channel index if primary channel; Logical primary channel index if secondary channel.
(IOC number multiplied by 256) plus (Channel number multiplied by 4)

	0 17	18 35
Word 2		

Bits 0-17 Absolute address of first entry in I/O queue; if zeros, empty
18-35 Absolute address of link table for file

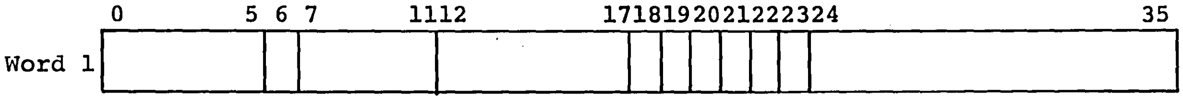
	0 17	18 35
Word 3		

Bits 0-17 Absolute address of last entry in I/O queue; if zeros, empty
18-35 Absolute address of secondary table, device 0, for multidevice channels

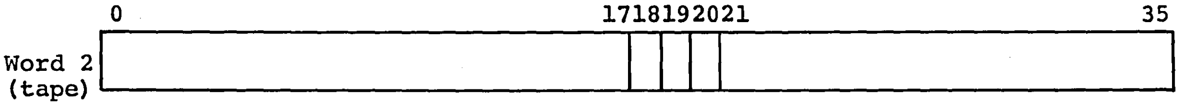
	0 17	18 23	24 29	30 35
Word 4				

Bits 0-17 Address of channel module identification word
18-23 Usage count (magnetic tape only)
24-29 Number of program last using device
30-35 A-register upper modification

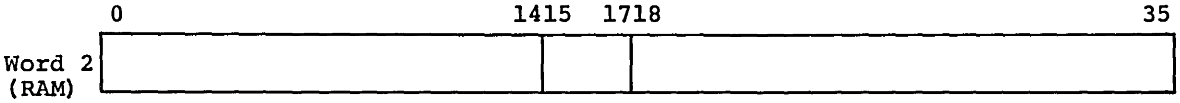
SECONDARY DEVICE S.C.T. TABLE ENTRY (4-word)
(One entry per device for multiple device channel)



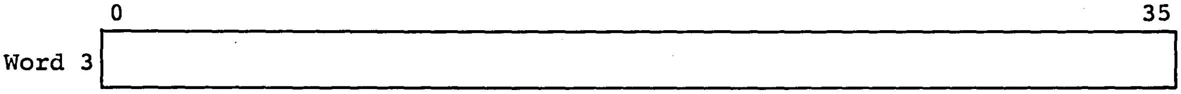
Bits 0-5 Device type
6 = 1, Logical channel or device is temporarily stopped
7-11 = Device number
12-17 If all ones, this is the last unit
18 = 0, Assignable
1, Not Assignable
19 = 0, Available
1, Assigned
20 = 0, Not Dedicated
1, Dedicated
21 = 0, Primary
1, Secondary
22 = 0, Single File
1, Shared File
23 = 0, Single Device
1, Multiple Device
24-35 Logical primary channel index (IOC number multiplied by 256) plus (Channel number multiplied by 4)



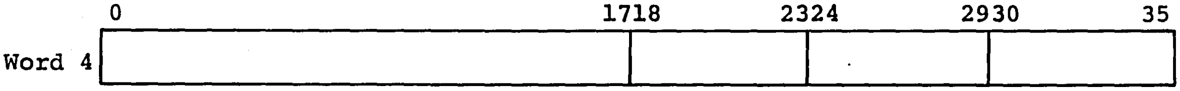
Bits 0-17 Number (+) of records since last EOF
18 = 0, Low density
1, High density
19 = 0, No EOT since rewind
1, EOT since rewind
20 = 0, Position good
1, Position lost
21-35 Number of EOF's passed



Bits 0-14 Usage count
15-17 Quantity of modules
18-35 Absolute address of link table



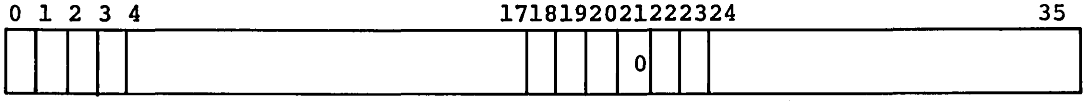
Bits 0-35 Number of connects on device



Bits 0-17 Number of errors on device
18-23 Number of errors in last n connects
24-29 Number of program last using device
30-35 Index of last unit allocated

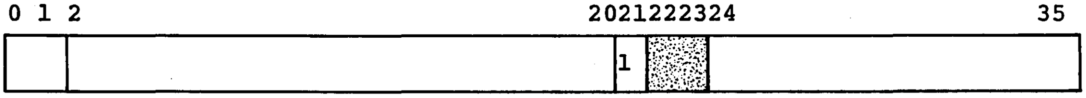
PAT POINTER ENTRY

Non-SYSOUT



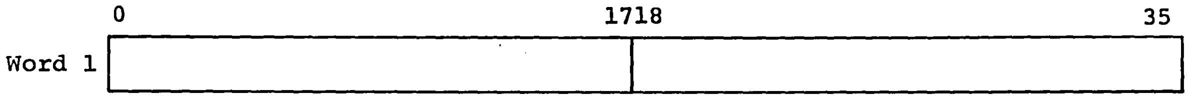
Bit 0 = 1, Read/Write permission
1 = 1, Write permission
2 = 1, Append permission
3 = 1, Execute permission
4-17 Offset to PAT entry from LAL
18 = 0, Allocated
1, Released
19 = 0, Primary
1, Secondary
20 = 0, No file request active
1, File request accepted
21 = 0, Not system device
22-23 Deallocation code 00 = Release
01 = Dismount
10 = Save
11 = Continue
24-35 File code

If bit 21 = 1, SYSOUT, the PAT pointer entry is as follows:

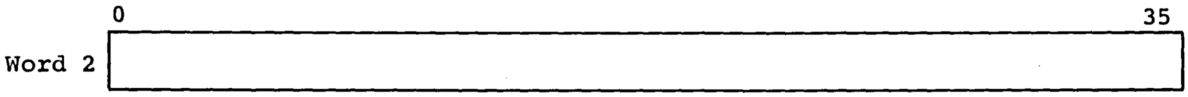


Bits 0-1 = 10, $ SYSOUT card
01, $ REMOTE card
00, Default, no file control card
2-20 Destination in remote
21 = 1, System device, SYSOUT
22-23 Not used
24-35 File code

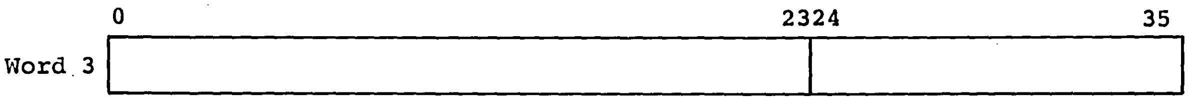
PAT ENTRY FOR PRINTER/PUNCH/READER



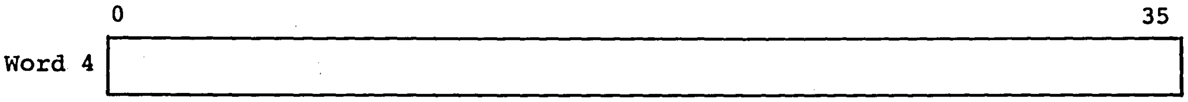
```
Bits    0-17  SCT pointer
       18-35  Logical unit designator (if present)
```



```
Bits    0-35  Channel Busy time
```



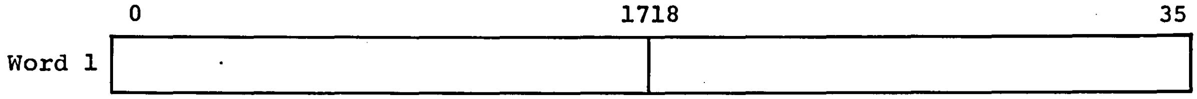
```
Bits    0-23  Number of connects on unit
       24-35  Number of errors
```



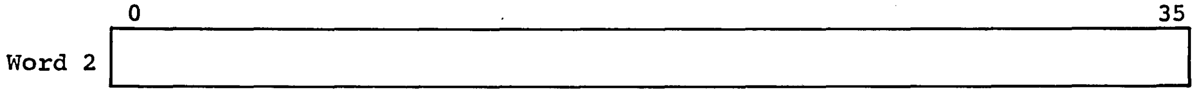
```
Bits    0-35  Time file released
```

Data is placed into words 3 and 4 only when file is released by MME GERELS prior to end of activity.

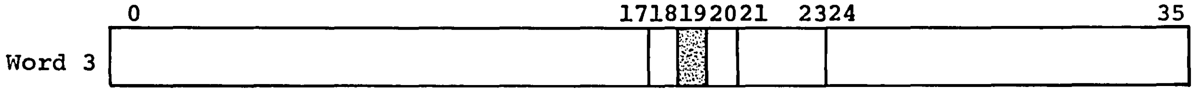
PAT ENTRY FOR MAGNETIC TAPE



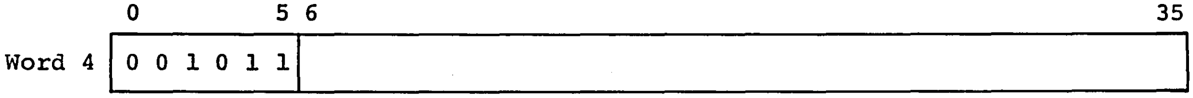
Bits 0-17 SCT pointer
18-35 Logical unit designator (if present)



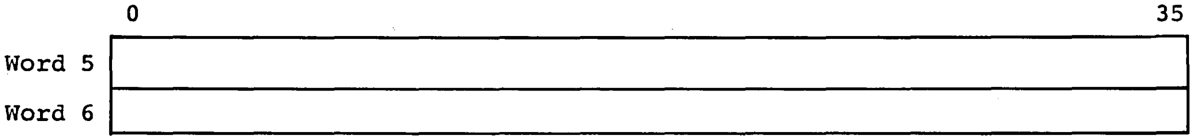
Bits 0-35 Channel Busy time



Bits 0-17 Secondary unit SCT pointer
18 = 1, File serial number present
0, File serial number not present
19 Not used
20 = 0, Name field not present
1, Name field present (see words 5 and 6)
21-23 Alternate Disposition code
24-35 Reel serial number



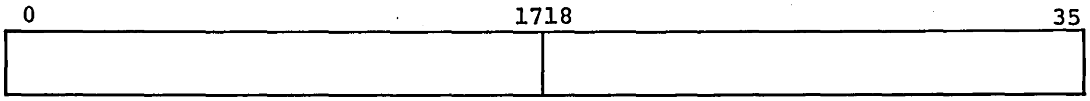
Bits 0-5 Octal 13
6-35 File serial number (BCD)



Bits 0-35 File Name (2 words). Present only when bit 20, word 3 is 1.

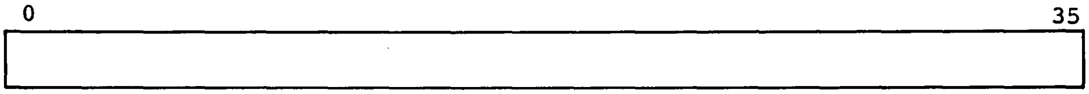
PAT ENTRY FOR MASS STORAGE FILE

Word 1



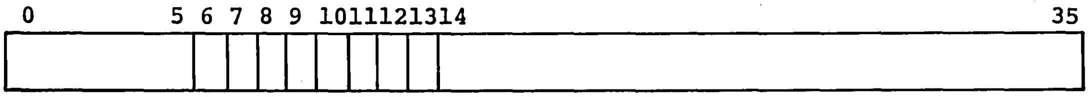
Bits 0-17 S.C.T. pointer for device
18-35 Logical unit designator

Word 2



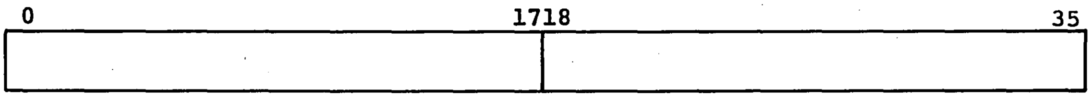
Bits 0-35 IOC time for this file

Word 3



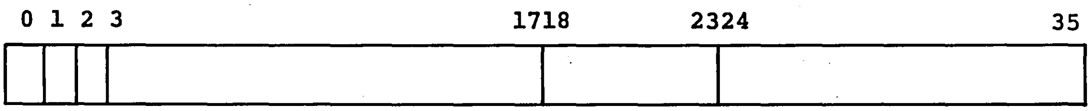
Bits 0-5 Hash of master catalog name
6 = 0, Catalog exists
1, Catalog does not exist
7 = 0, Last description is in core
1, Last description is not in core
8 = 0, Linked
1, Random
9 = 0, Temporary
1, Permanent
10 = 0, Link assignment not entirely in core, parallel access denied
1, Link assignment all in core, parallel access permitted
11 = 0, This user is the creator of this file
1, This user is not the creator of this file
12-13 Abort disposition code
14-35 Catalog Seek address

Word 4



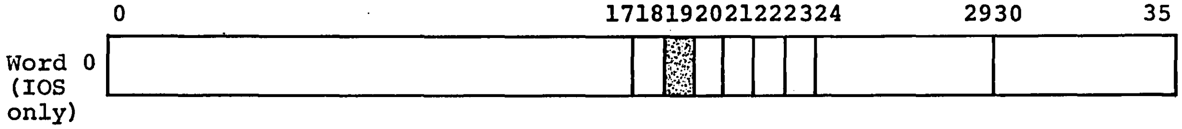
Bits 0-17 Total number of links assigned
18-35 Current relative block number within core description

Word 5

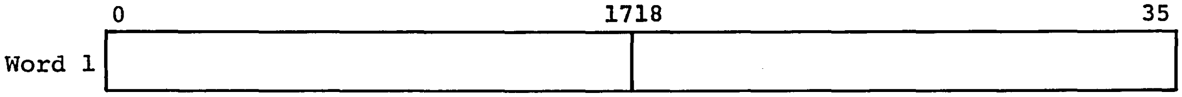


Bit 0 = 0, Last link string in core
1, Not last link string
1 = 0, Part of this file, space measured in links
1, Part of this file, space measured in blocks
2 = 0, Not first link string of file
1, First link string (rewound position)
3-17 Beginning link number of string
If bit 1 = 0
18-35 Number of links
If bit 1 = 1
18-23 Block within link
24-35 Number of blocks

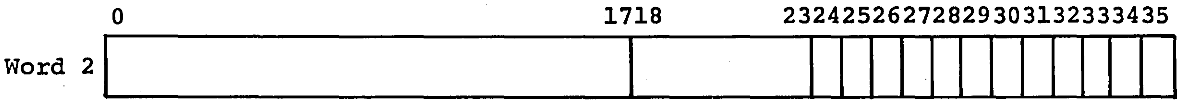
I/O ENTRY



Bits	0-17	Absolute address of next entry in queue (GEPR status word 2, upper)
	18-23	IOS Status
	18=	1, Stopped
	19	Not used
	20=	1, Seek completed (DSU200)
	21=	1, Seek started (DSU200)
	22=	1, GESPECed entry
	23=	1, In transmission
	24-29	Logical device index/angle
	30-35	This entry queue status
		Status = 0, Open
		1, Building
		2, Linked
		3, In transmission
		4, GEPR
		5, Courtesy Call waiting
		6, Linked/Removed
		9, Courtesy Call/Removed
	30=	1, Error Threshold



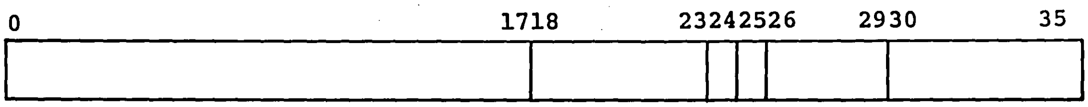
Bits	0-17	Absolute address of previous entry in queue
	18-35	GEPR status word 1



Bits	0-17	Absolute address of device SCT table
	18-23	Bypass count
	24=	1, Absolute CCA
	25=	1, GEPR Override
	26=	1, GPROVD and DCWPTR
	27=	1, Master TYIO
	28=	1, Do not abort
	29=	1, No RAM EOF recovery
	30=	1, DCW generated in SLV 60-67 (GEPR)
	31=	1, SYOT I/O
	32=	1, PMX contains seek address
	33=	1, CCA offset, pay CC in MM
	34=	1, Preselected channel (Index in SMX3)
	35=	1, Issue RESS for angle (DR20-IOS)

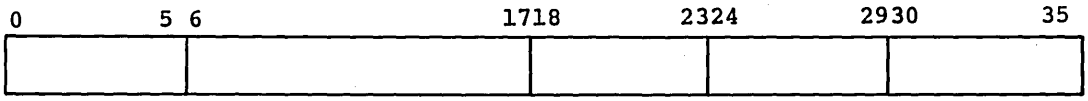
I/O ENTRY (Contd.)

Word 3



Bits	0-17	Offset to PAT PTR/File Code word in SSA (Off to LAL)
	18-23	GEPR error count
	24	Reissued I/O, Reset SCT stop (GEPR)
	25	Special GEPR command
	26-29	GEPR options 1= Data alert override 2= U-option on unrecoverable tape parities
	30-35	This entry program number

Word 4



If Seek/R-W request (Bit 32, Word 2 = 1)

Bits	0-35	Seek address

If I/O command (Bit 32, Word 2 = 0)

Bits	0-5	Device command
	6-17	Must be zero
	18-23	I/O command
	24-29	Must be zero
	30-35	Record count

Word 5

Bits	0-35	Secondary Mailbox No. 1 First DCW, data address relative to IOC LAL

I/O ENTRY (Contd.)

	0 17	18 35
Word 6		

Bits 0-35 Secondary Mailbox No. 2
0-17 DCWP+1, (offset to LAL)
18-35 Limits flag (0, +, or -)

	0 5	6 17	18 23	24 29	30 35
Word 7				0 0 0 0 0 0	

Bits 0-35 Secondary Mailbox No. 3
0-5 Device command
6-17≠ 0, Preselect Index
18-23 I/O command 2
24-29 Must be zero
30-35 Record count

	0 17	18 35
Word 8		

Bits 0-35 Secondary Mailbox No. 4
0-17 DCWP (offset to LAL)
18-35 GEPR status word 2, lower

	0 17	18 35
Word 9		

Bits 0-17 Status return pointer (offset to LAL)
18-35 Courtesy Call address (offset to LAL or ABS)

	0 35
Word 10	

Bits 0-35 IOS temporary storage
(Relative block number for EOF recovery, random file)

CONTROL RECORDS FOR TYPE 0 (Absolute) and TYPE 1 (Relocatable) ELEMENTS

BLOCK 1 ONLY

	0	35
Word 1	Checksum of data blocks which follow	
Word 2	Checksum of relocatable blocks which follow	
Word 3	Checksum of this block	
Word 4	Name of element	

	0	17	18	35
Word 5				

Bits 0-17 Entry address
18-35 Origin

	0	17	18	35
Word 6				

Bits 0-17 Number of blocks of relocation bits
18-35 Number of blocks of data

	0	35
Word 7	DCW to load the data block	

•
•
•
•
•

All DCW addresses are incremented by 1024 words.

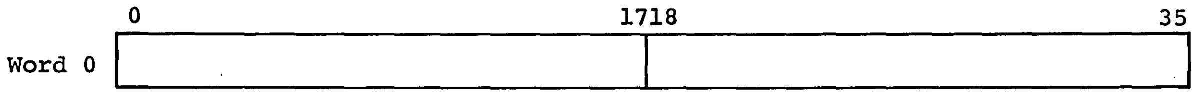
NEXT N BLOCKS

Data to be loaded as per DCW's.

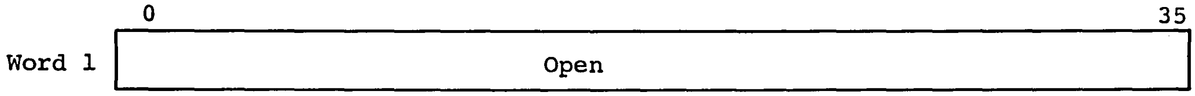
Next (M+17)/18 blocks--for
Type 1 elements only

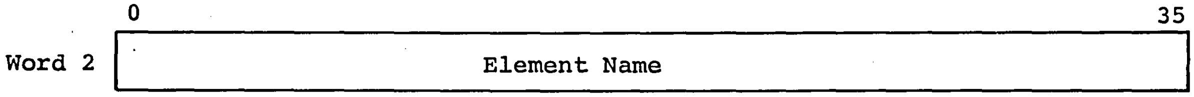
Relocation bits--one per half-word--
proceeding from left to right

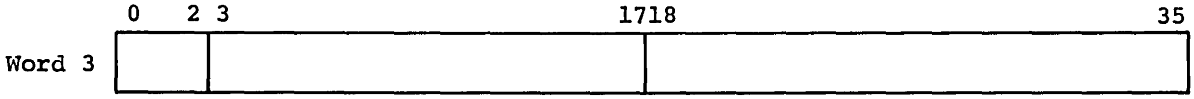
FORMAT OF CATALOG OF ELEMENTS



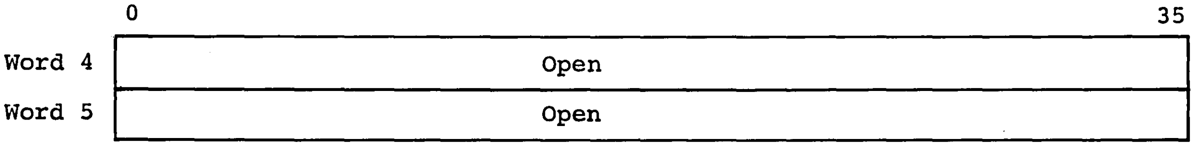
Bits	0-17	Block Number of available space
	18-35	Next catalog block number







Bits	0-2	= 0, Absolute program 1, Relocatable program 2, Select file 3, ASC II file from TTY
	3-17	Number of blocks containing element
	18-35	Block number of element

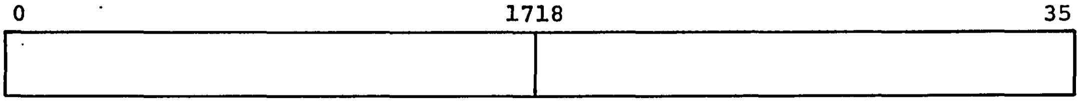


Words 4 and 5 - Data as defined by element type

Words 2 through 5 are repeated for each element

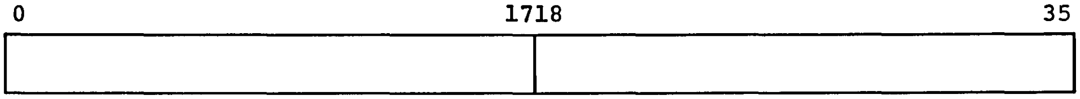
FORMAT OF AVAILABLE SPACE BLOCK WITHIN FILE OF ELEMENTS

Word 0



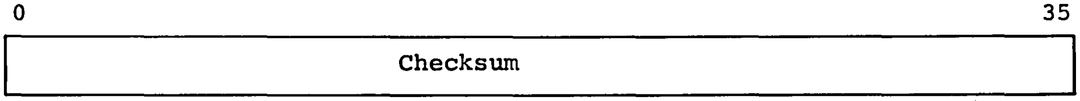
Bits 0-17 Zeros
18-35 Relative block number of the next available space. If zeros, this is the last available block.

Word 1 thru Word N*-2



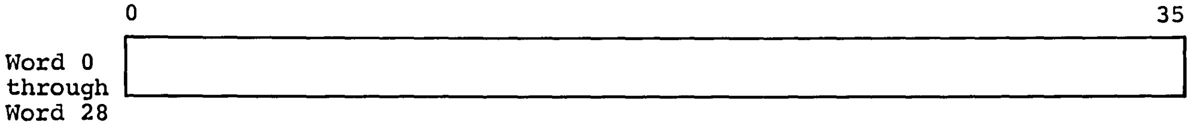
Bits 0-17 Relative block number of an available block in this file.
18-35 Number of blocks

Word N*-1

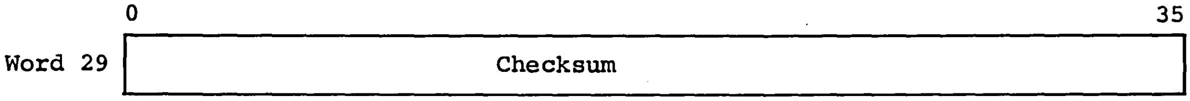


*N is the number of words per addressable block on this device.

AVAILABLE LINK TABLE
On a Random Device

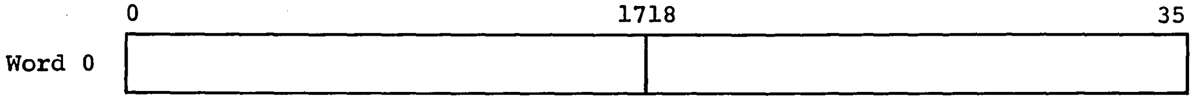


Any bit = 1, LINK is used
= 0, LINK is available

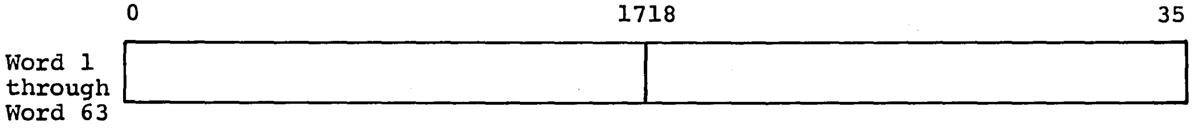


The available LINK table is saved in Block 0 of LLINK 2 on every random access device and describes the availability of LINKS 0 through 1043 (29 words x 36 bits per word = 1044). Since this is greater than the number of LINKS on any device, nonexistent LINKS appear as unavailable LINKS.

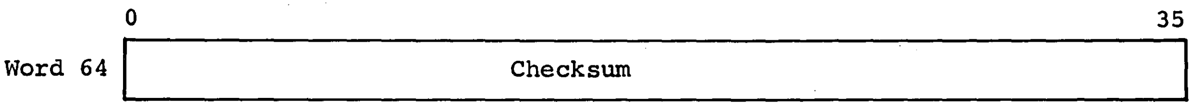
In Memory

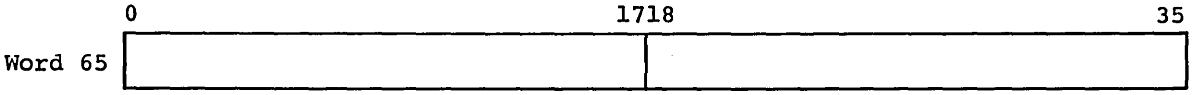


Bits 0-17 Number of LINKS reserved for file extension and not used for file creation (Threshold)
18-35 Total number of LINKS available on this device



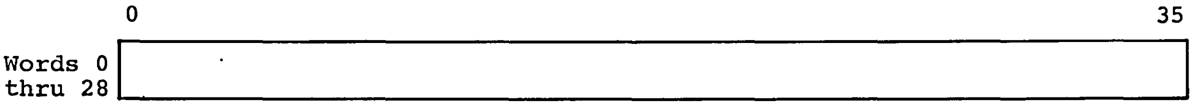
Bits 0-17 LINK number
18-35 Number of LINKS



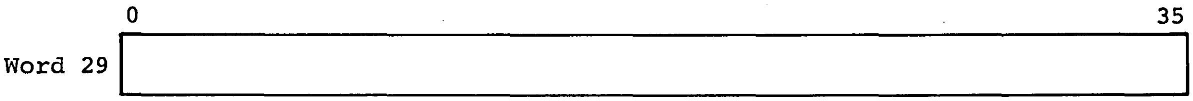


Bits 0-17 Initial number of LINKS available on the device
18-35 Block number of next available LLINK

AVAILABLE LLINK TABLE



Bits 0-35 Binary bits representing sequential LLINKs
any bit =1, LLINK is used
any bit =0, LLINK is available

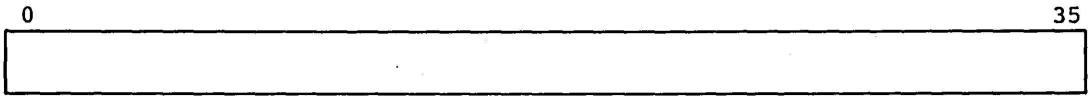


Bits 0-35 Checksum

The available LLINK table is saved in Block 0 of LLINK1 on every random access device.

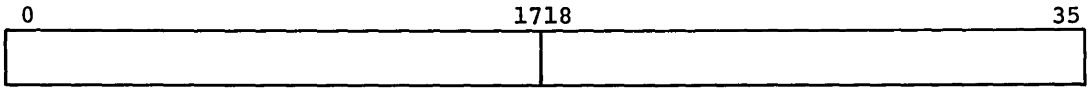
TAPE CATALOG RECORD

Word 0



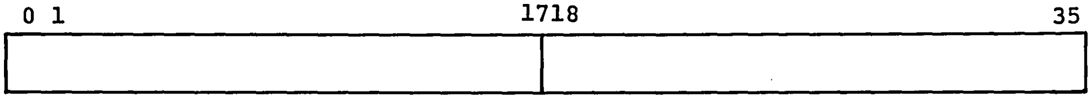
Bits 0-35 Checksum of this record

Word 1



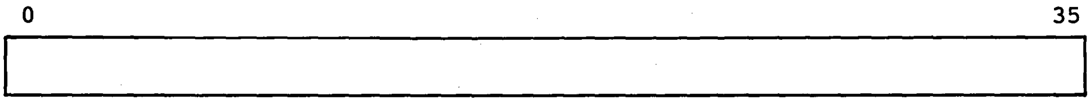
Bits 0-17 This record number
18-35 Previous catalog record number

Word 2



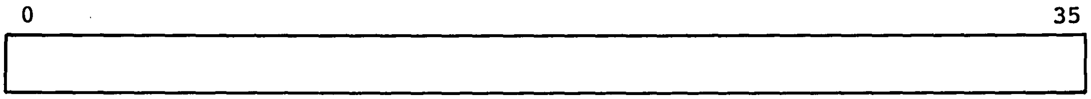
Bit 0 = 0, Non-terminal catalog record
= 1, Terminal catalog record
Bits 1 - 17 Not used
18 - 35 Must be non-zero

Word 3



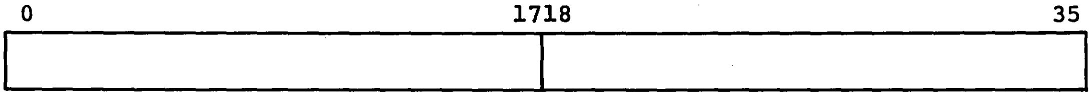
Bits 0-35 Number of entries in this record multiplied by 2

Word 4



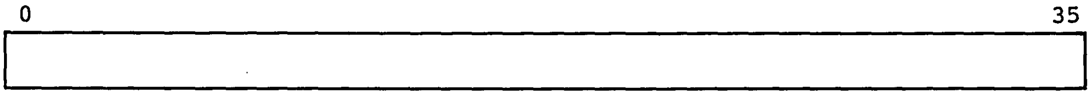
Bits 0-35 Name 1

Word 5



Bits 0-17 Starting record number
18-35 Number of records

Word 6



Bits 0-35 Name 2

	0 — 17	18 — 35
Word 7		

Bits 0-17 Starting record number
18-35 Number of records

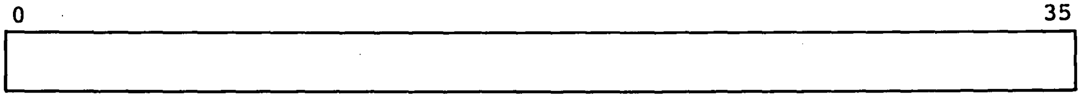
•
•
•

Through word 63

EXTEND EDIT

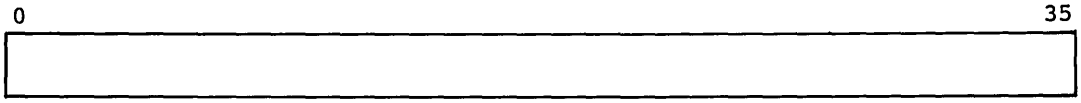
TAPE (CATALOG RECORD)

Location



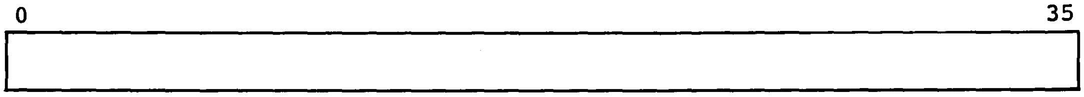
Bits 0-35 Checksum of this record

Loc.+1



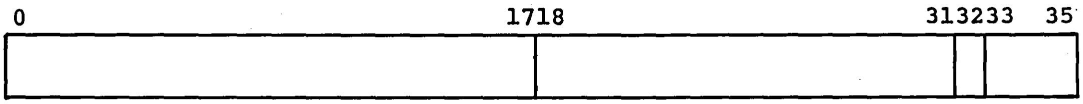
Bits 0-35 Minus 1

Loc.+2



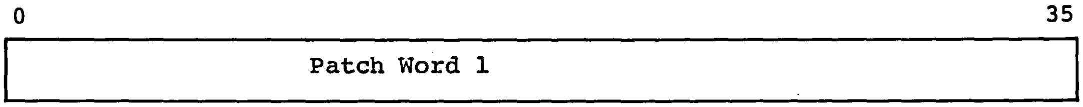
Bits 0-35 BCI name

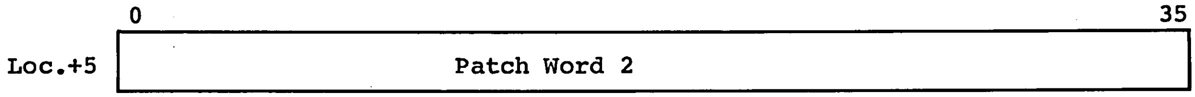
Loc.+3

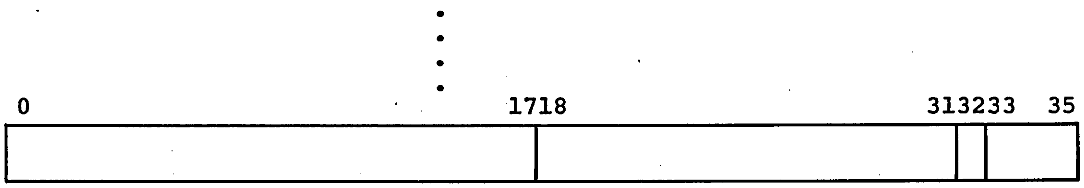


Bits 0-17 Location plus relative address
18-31 Relocation bits
32= 1, Relocation specified on SYSNAME card
33-35 Number of consecutive patch words; must be less than or equal to 7, for which 14 relocation bits are available indicating upper and lower relocation.

Loc.+4

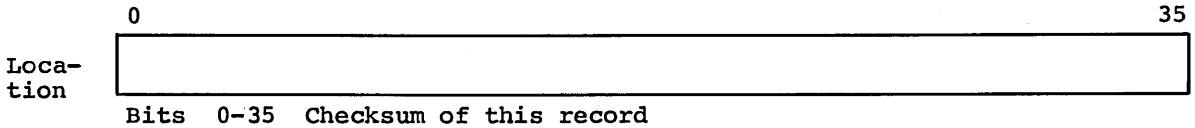




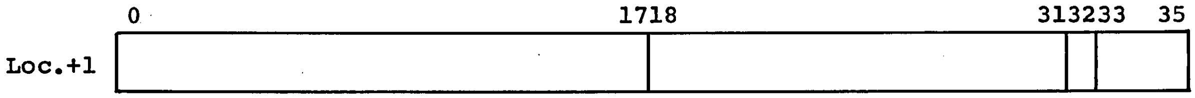


Bits 0-17 Location plus relative address
18-31 Relocation bits
32= 1, Relocation specified on SYSNAME card
33-35 Number of consecutive patch words

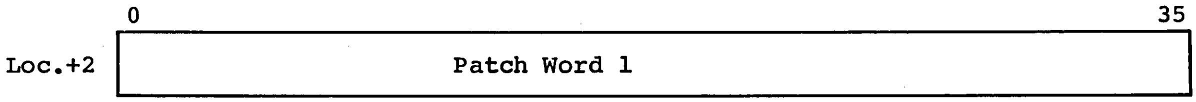
TAPE (CONTINUATION RECORD)

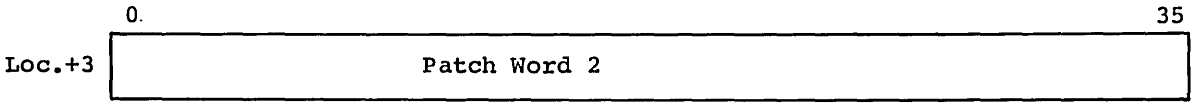


Bits 0-35 Checksum of this record

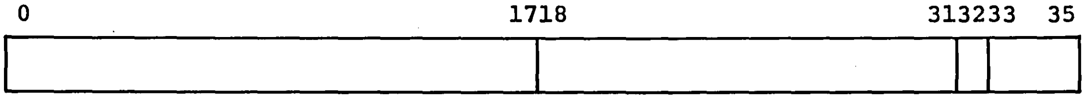


Bits 0-17 Location plus relative address
18-31 Relocation bits
32= 1, Relocation specified on SYSNAME card
33-35 Number of consecutive patch words, must be less than or equal to 7, for which 14 relocation bits are available indicating upper and lower relocation.

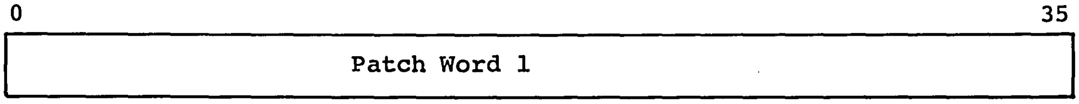




•
•
•



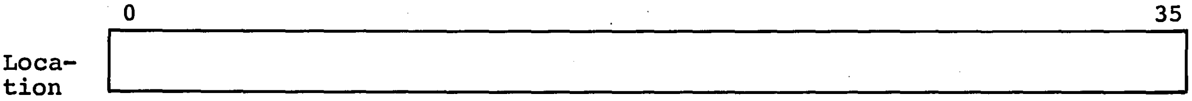
Bits 0-17 Location plus relative address
18-31 Relocation bits
32= 1, Relocation specified on SYSNAME card
33-35 Number of consecutive patch words



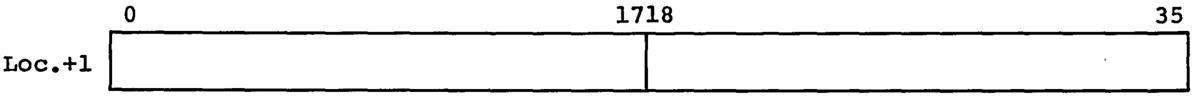
•
•
•
•

thru
Loc.+63

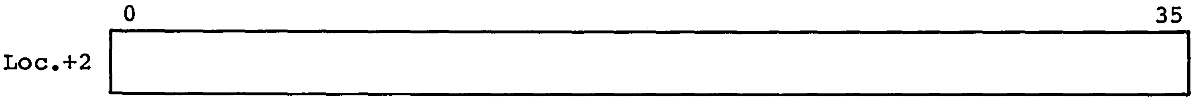
SHARED DEVICE (CATALOG RECORD)



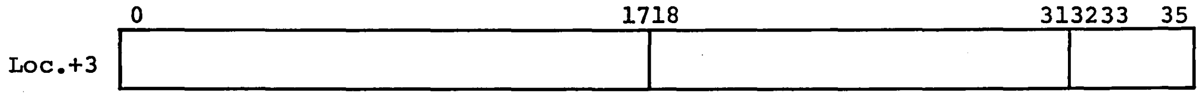
Bits 0-35 Checksum of this record



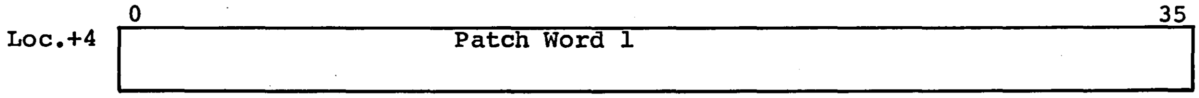
Bits 0-17 = Zero
18-35 Next catalog block number

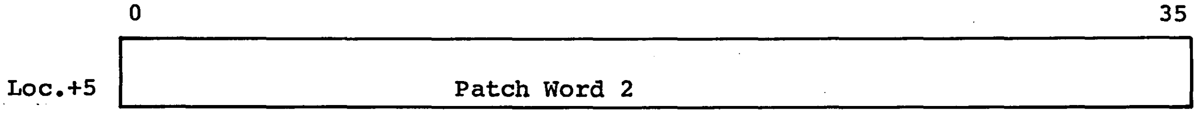


Bits 0-35 BCI name



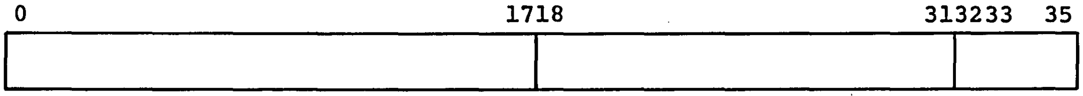
Bits 0-17 Location plus relative address
18-31 Relocation bits
32= 1, Relocation specified on SYSNAME card
33-35 Number of consecutive patch words; must be less than or equal to 7, for which 14 relocation bits are available indicating upper and lower relocation.



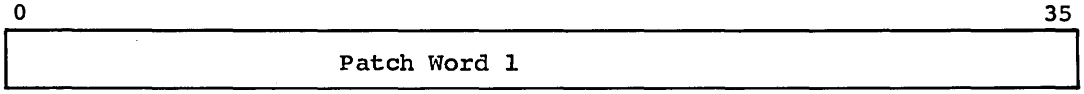


•
•
•

through



Bits 0-17 Location plus relative address
18-31 Relocation bits
32= 1, Relocation specified on SYSNAME card
33-35 Number of consecutive patch words



•
•
•

Loc.+39/63

CPB-1488

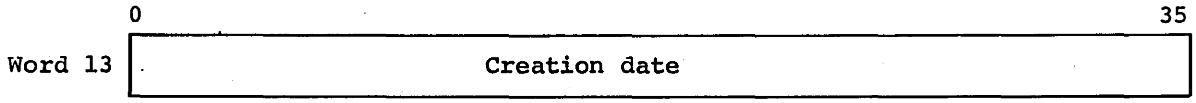
CATALOG BLOCKS

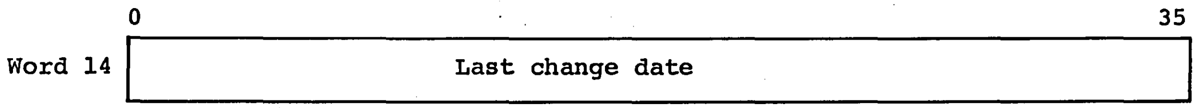
TYPE 0 - CATALOG DESCRIPTION

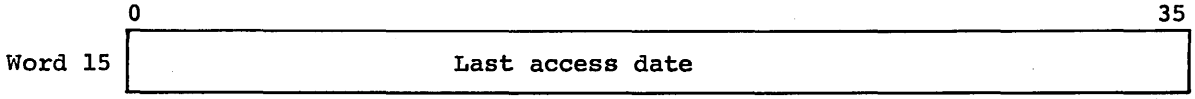
	0 5	6 35
Word 1	0 0 0 0 0 0	▨

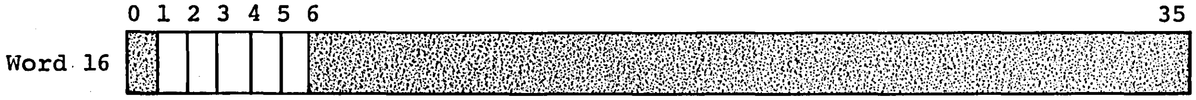
Bits 0-5 Type identification
6-35 Not used

	0 35
Word 2	Address of next LLINK in user's chain
Word 3	Address of previous LLINK in user's chain
Word 4	Address of previous Type 0
Word 5	Address of continuation - Type 1
Word 6	Address of permission block - Type 4
Word 7 and 8	BCI of catalog name
Word 9 and 10	BCI of catalog originator
Word 11 and 12	BCI of password

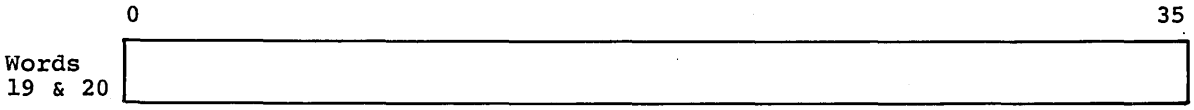




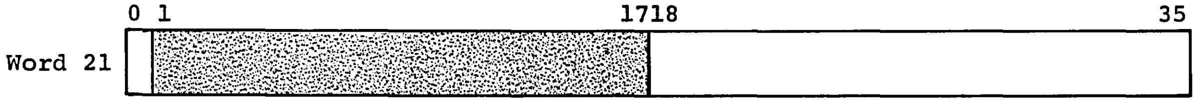




Bits	0-1	Not used
	2=	1, Gross Read
	3=	1, Gross Write
	4=	1, Gross Append
	5=	1, Gross Execute
	6-35	Not used



Bits 0-35 BCI of subordinate file or Catalog Name



If subordinate file is on the same device,

Bits	0-17	Word position within LLINK
	18-35	Absolute block number of LLINK

If subordinate file or catalog is on another device,

Bit	0 = 1	
	1-17	Not used
	18-35	BCI of device name

•
•
•

	0	35
Words n-3 & n-2		

Bits 0-35 BCI of subordinate file or Catalog name

	0	35
Word n-1	Address of subordinate catalog	

(Same as word 21)

	0	35
Word n	Checksum	

TYPE 1 - CATALOG DESCRIPTION

	0 5	6 35
Word 1	0 0 0 0 0 1	

Bits 0-5 Type Identification
6-35 Not used

	0 35
Word 2	Address of next LLINK in user's chain

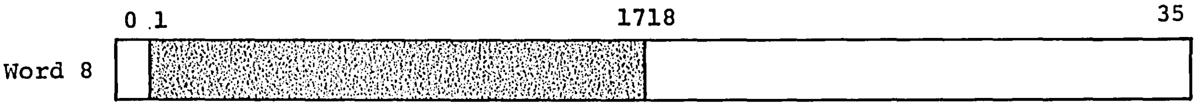
	0 35
Word 3	Address of previous LLINK in user's chain

	0 35
Word 4	Address of previous Type 0 or 1

	0 35
Word 5	Address of continuation Type 1

	0 35
Words 6 & 7	

Bits 0-35 BCI of subordinate file or Catalog Name



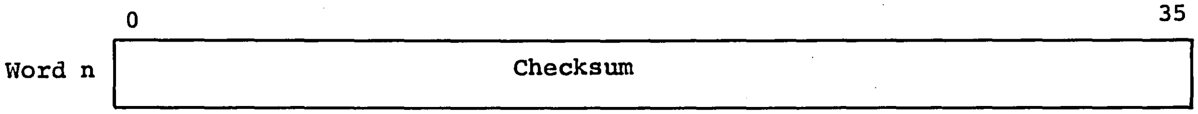
If subordinate file is on the same device,

Bits 0-17 Word position within LLINK
18-35 Absolute block number of LLINK

If subordinate file or catalog is on another device,

Bit 0 = 1
1-17 Not used
18-35 BCI of device name

•
•
•
•



TYPE 2 - FILE DESCRIPTION

	0 5	6 17	18 35
Word 1	0 0 0 0 1 0	▨	

Bits 0-5 Type identification
6-17 Not used
18-35 Device name

	0 35
Word 2	Address of next LLINK in user's chain

	0 35
Word 3	Address of previous LLINK in user's chain

	0 35
Word 4	Address of previous Type 0 or 1

	0 35
Word 5	Address of continuation - Type 3

	0 35
Word 6	Address of permission block - Type 4

	0 35
Words 7 & 8	

Bits 0-35 BCI of file name

	0 35
Words 9 & 10	

Bits 0-35 BCI of file originator

```
             0                                            35
            ┌────────────────────────────────────────────┐
Words       │                                            │
11 & 12     └────────────────────────────────────────────┘
```

Bits 0-35 BCI of password

```
             0                                            35
            ┌────────────────────────────────────────────┐
Word 13     │               Creation date                │
            └────────────────────────────────────────────┘
```

```
             0                                            35
            ┌────────────────────────────────────────────┐
Word 14     │             Last change date               │
            └────────────────────────────────────────────┘
```

```
             0                                            35
            ┌────────────────────────────────────────────┐
Word 15     │             Last access date               │
            └────────────────────────────────────────────┘
```

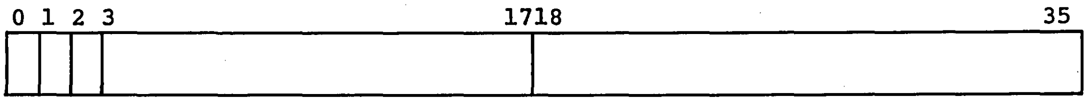
```
             0 1 2 3 4 5 6     1112    1718               35
            ┌─┬─┬─┬─┬─┬─┬───────┬───────┬──────────────────┐
Word 16     │ │ │ │ │ │ │       │       │                  │
            └─┴─┴─┴─┴─┴─┴───────┴───────┴──────────────────┘
```

```
Bit      0 = 1, File being written
         1 = 1, File being appended
         2 = 1, Gross read
         3 = 1, Gross write
         4 = 1, Gross append
         5 = 1, Gross execute
      6-11   Type of file
     12-17   Number of readers
     18-35   Maximum size - in LLINKs
```

```
             0                                            35
            ┌────────────────────────────────────────────┐
Words       │                                            │
17 & 18     └────────────────────────────────────────────┘
```

Bits 0-35 BCI of system Master Catalog Name

Words 19 thru n-1   The required number of words to describe the file, formatted as follows:



Bit 0 = 0, No file request active
1, File request accepted
1 = 0, Not system device
1, System device, SYSOUT
2 = 1, First link string
3-17 Beginning link number of string
18-35 Number of links

Word n   Checksum

TYPE 3 - FILE DESCRIPTION

Word 1

0 5	6 35
0 0 0 0 1 1	▨

Bits 0-5 Type identification
6-35 Not used

Word 2

0 35
Address of next LLINK in user's chain

Word 3

0 35
Address of previous LLINK in user's chain

Word 4

0 35
Address of previous Type 2 or 3

Word 5

0 35
Address of continuation - Type 3

Words 6 thru n-1 The required number of words to describe the file, formatted as follows:

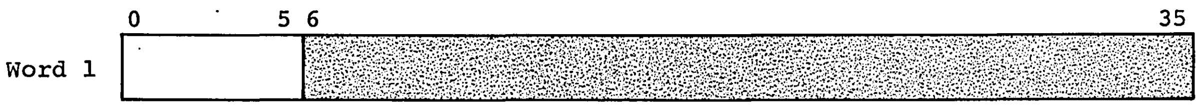
0	1	2	3 17	18 35

Bit 0 = 0, No file request active
1, File request accepted
1 = 0, Not system device
1, System device, SYSOUT
2 = 1, First link string
3-17 Beginning link number of string
18-35 Number of links

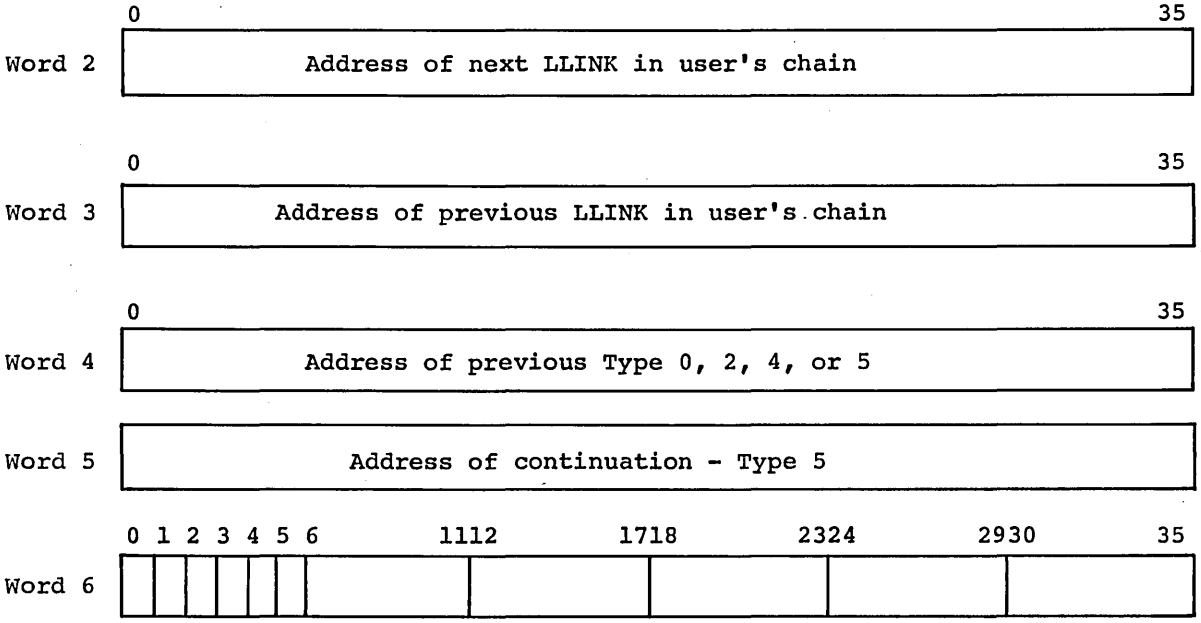
Word n

0 35
Checksum

TYPE 4 or 5 - PERMISSIONS



Bits	0-5	Type identification, 4 or 5
	6-35	Not used


0
35
Word 2
Address of next LLINK in user's chain
0
35
Word 3
Address of previous LLINK in user's chain
0
35
Word 4
Address of previous Type 0, 2, 4, or 5
Word 5
Address of continuation - Type 5
0 1 2 3 4 5 6
1112
1718
2324
2930
35
Word 6

Bits	0-5	For user 1
	0=	1, Read permission
	1=	1, Write permission
	2=	1, Append permission
	3=	1, Execute permission
	4=	1, Purge permission
	5=	1, Modify permission
	6-11	Same as above, for user 2
	12-17	Same as above, for user 3
	18-23	Same as above, for user 4
	24-29	Same as above, for user 5
	30-35	Same as above for user 6

Words 7, 8 & 9	Same format as word 6, for users 7 through 24

	0	35
Word 10 & 11		

Bits 0-35 BCI of user 1

	0	35
Word 12 & 13		

Bits 0-35 BCI of user 2

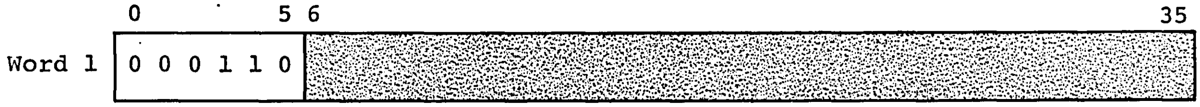
•
•
•
•

	0	35
Word n-2 & n-1		

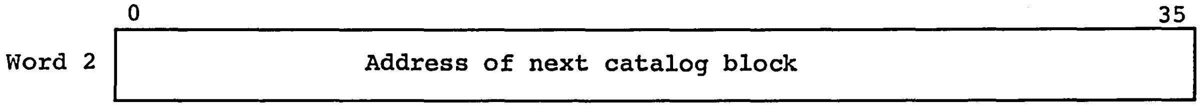
Bits 0-35 BCI of user 24

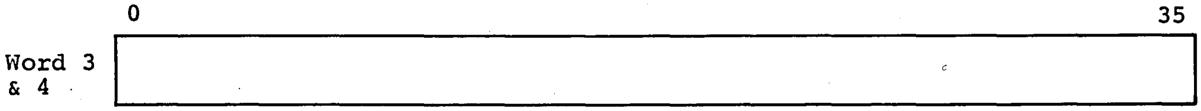
	0	35
Word n	Checksum	

TYPE 6 - SYSTEM MASTER CATALOG

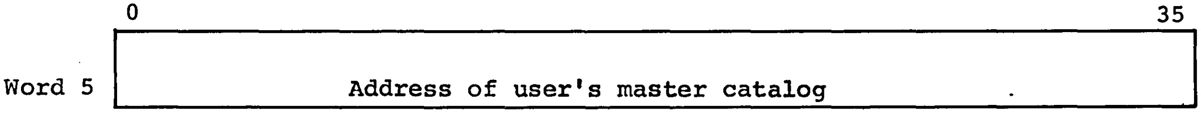


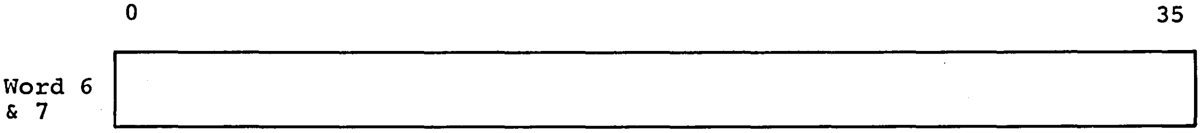
```
Bits   0-5   Type identification
       6-35  Not used
```



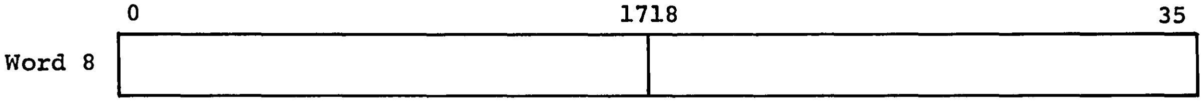


```
Bits   0-35  BCI of Master Catalog Name
```

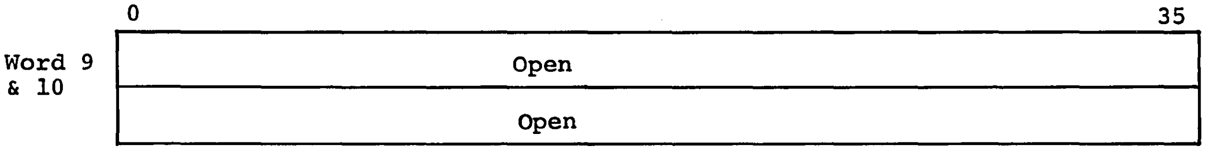




```
Bits   0-35  BCI of catalog originator
```



```
Bits   0-17  Number of blocks used
      18-35  Maximum blocks allowed
```



	0	35
Word 11 & 12		

Bits 0-35 BCI of password

	0	35
Word 13	Total resources allowed	

	0	35
Word 14 thru Word n	Resources used to date	

•
•
•

	0	35
Word n+1 & n+2		

Bits 0-35 BCI of Master Catalog Name

	0	35
Word n+3	Address of user's master catalog	

	0	35
Word n+4 & n+5		

Bits 0-35 BCI of catalog originator

	0	1718	35
Word n+6			

Bits 0-17 Number of blocks used
18-35 Maximum blocks allowed

	0	35
Word n+7 & n+8		

	0	35
Word n+9 & n+10		

Bits 0-35 BCI of password

	0	35
Word n+11	Total resources allowed	

	0	35
Word n+12 thru m	Resources used to date	

•
•
•

	0	35
Word m+1	Address of previous catalog block	

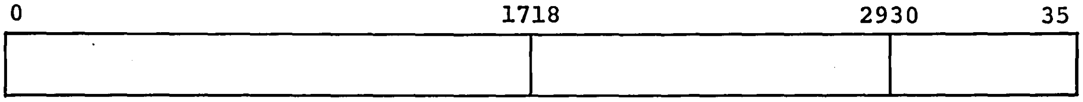
	0	35
Word m+2	Checksum	

CORE ALLOCATOR TABLES (USED ONLY BY GEPOP)

URGENCY QUEUE

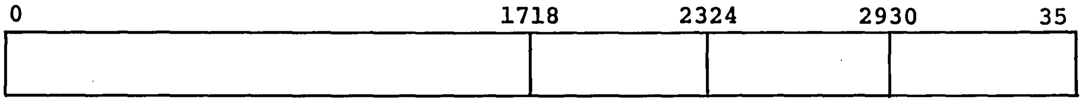
The urgency queue is a threaded list of all jobs in core in order of decreasing urgency. By definition, PNPOP is always the most urgent job in core. An RPL may be used to scan this queue. The entry is associated with its program number by its relative position within POUQ. Length of queue is .NRPRG+1.

POUQ



Bits 0-17 Absolute address of highest urgency job in queue
18-29 Zeros
30-35 Program number of lowest urgency job

P1



Bits 0-17 Pointer to next lower urgency job
18-23 Urgency
24-29 Status (octal) = 00 - Job in core
01 - Job not in core
02 - Job swapping out
05 - Core compaction/swapping being done to make room for job
07 - Core being compacted upward to satisfy GEMORE made by program
17 - Core being compacted downward to satisfy GEMORE made by program

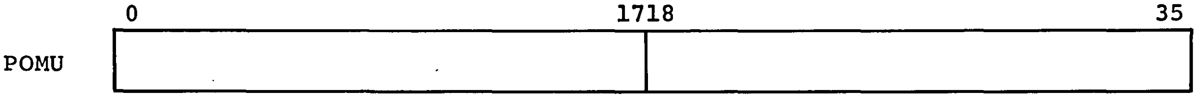
30-35 Program number of next higher urgency

P2
thru
P12

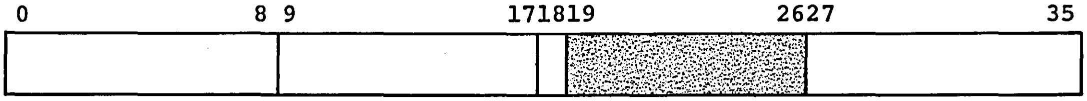
.
.
.
.

MEMORY UTILIZATION TABLE

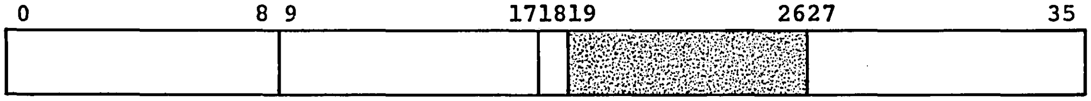
For 256k system, last absolute address entry in table is 777 (octal), the address of the last block of memory, rather than 000, the first illegal memory address. Length of table is .NRPRG+2.



Bits	0-17	Zero
	18-35	Number of 512-word blocks

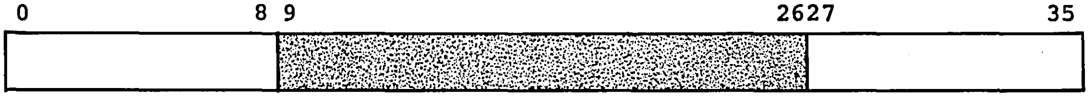


Bits	0-8	Absolute address of first allocator block
	9-17	Program number in allocator block
	18=	1, =7 Job to be moved for memory compaction
	19-26	Not used
	27-35	Number of 512-word blocks available between this and next job



Bits	0-8	Absolute address of next allocator block
	9-17	Program number in allocator block
	18=	1, =7 Job to be moved for memory compaction
	19-26	Not used
	27-35	Number of 512-word blocks available between this and next job

⋮



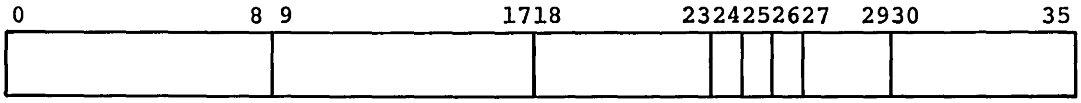
Bits	0-8	Absolute address of end of memory
	9-26	Not used
	27-35	Zero

CORE DEMAND TAPE TABLE

Length is .NRPRG+4

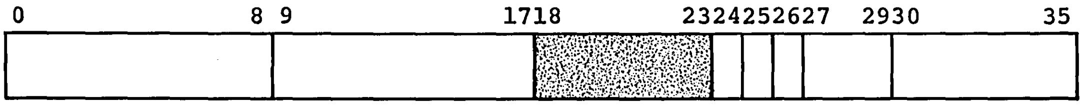
POPQ2

Ordinary Slave Core Request



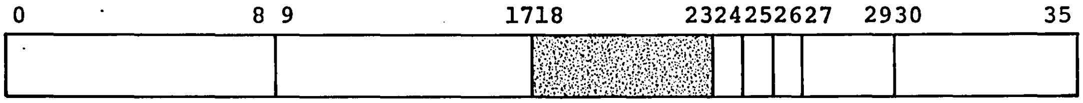
```
Bits    0-8    Zero
        9-17   Number of 512-word blocks of core needed
       18-23   Zero
          24=  1, Job must be allocated at the extreme high end of
               core and must have BAR set to access up to 2 to the 18th
          25=  1, Job should be allocated toward lower end of core.  It
               will be in system for a long time and should be kept out
               of memory compaction.
          26=1, A privileged slave.  When executing, it may enter
               master mode with a MME EMM.
       27-29   Number of 1024-word Slave Service Area blocks
       30-35   Zero
```

System Function



```
Bits    0-8    System program to GECALL
        9-17   Number of 512-word blocks of core needed
       18-23   Not used
          25=  1, Job should be allocated toward lower end of core.  It
               will be in system for a long time and should be kept out
               of memory compaction.
          26=  1, A priviledged slave.  When executing, it may enter
               master mode with a MME EMM.
       27-29   Number of 1024-word Slave Service Area blocks
       30-35   Zero
```

GEMORE Request



Bits 0-8 Additional number of 512-word blocks requested

9-17 Number of 512-word blocks already assigned

18-23 Not used

24= 1, Job must be allocated at the extreme high end of core and must have BAR set to access up to 2 to the 18th

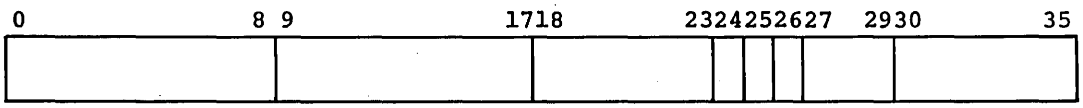
25= 1, Job should be allocated toward lower end of core. It will be in system for a long time and should be kept out of memory compaction.

26= 1, A privileged slave. When executing, it may enter master mode with a MME EMM.

27-29 Number of 1024-word Slave Service Area blocks

30-35 Zero

Job Swapped Out of Core



Bits 0-8 Zero

9-17 Number of 512-word blocks needed

18-23 Zero

24= 1, Job must be allocated at the extreme high end of core and must have BAR set to access up to 2 to the 18th.

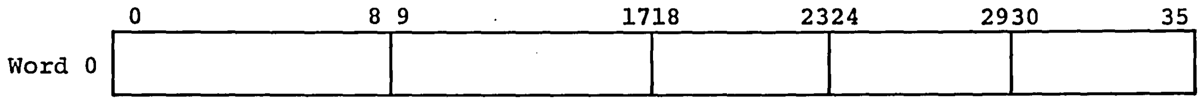
25= 1, Job should be allocated toward lower end of core. It will be in system for a long time and should be kept out of memory compaction.

26= 1, A privileged slave. When executing, it may enter master mode with a MME EMM.

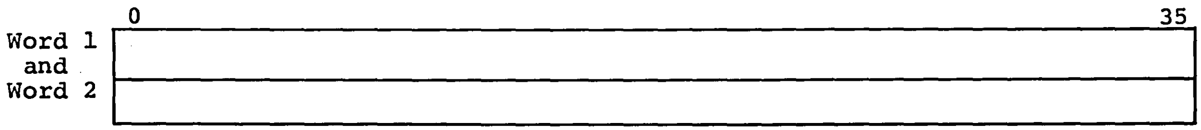
27-29 Number of 1024-word Slave Service Area blocks

30-35 Number of links of pushdown file.

Format of GEPOP Entries



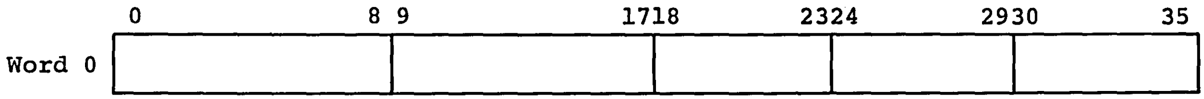
Bits		
	0-17	Message type
	0-8	Module number
	9-17	Entry point
	18-23	Output station or console
	24-29	Output station or console
	30-35	Message source = 00, Console
		01, Allocator
		02, GEMORE
		03, REMOTE
		04, Interrupt Processor
		05, Timer
		06, GEMREL



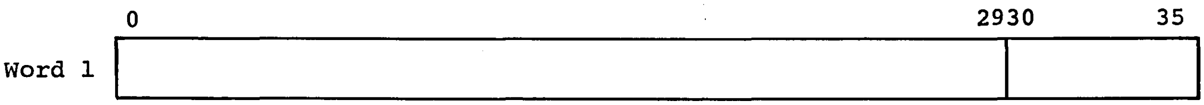
Bits		
	0-35	Message text. Each message is two data words. For console messages, this is the argument of the message.

GEPOP Entries

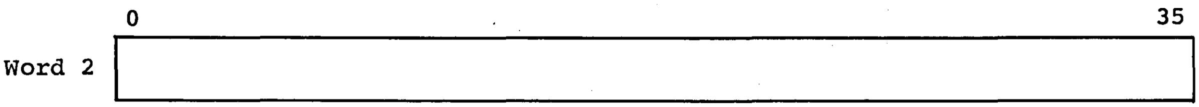
Job Ready to Move



Bits		
	0-8	.MPOP3
	9-17	Entry point 7
	18-23	Output station or console
	24-29	Output station or console
	30-35	Message source

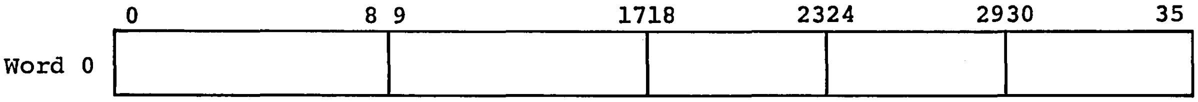


Bits	0-29	Zero
	30-35	Program number

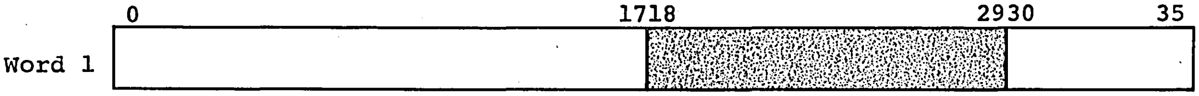


Bits	0-35	Zero

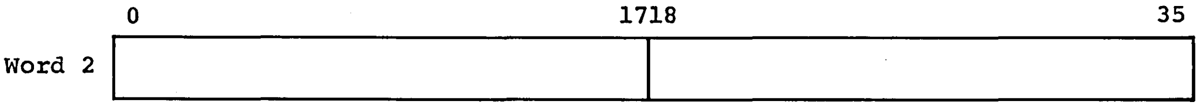
Job Ready to Swap



Bits	0-8	.MPOP3
	9-17	Entry point 8
	18-23	Output station or console
	24-29	Output station or console
	30-35	Message source



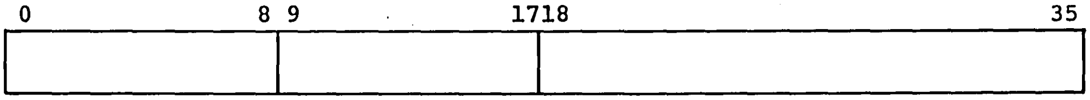
Bits	0-17	Number of links on pushdown file
	18-29	Not used
	30-35	Program number



Bits	0-17	SCT pointer for pushdown file
	18-35	Link number on pushdown file

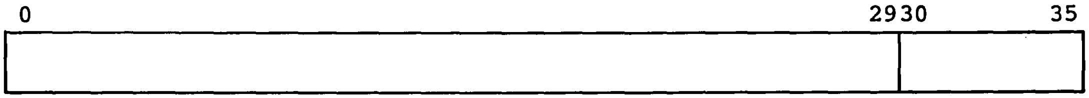
Card Reader Released and Re-GESPECED After GEIN

Word 0



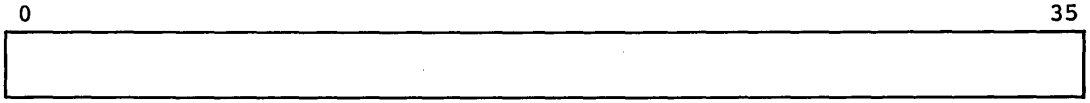
Bits	0-8	.MPOPM
	9-17	Entry point 1
	18-35	Zero

Word 1



bits	0-19	Zero
	30-35	Program number

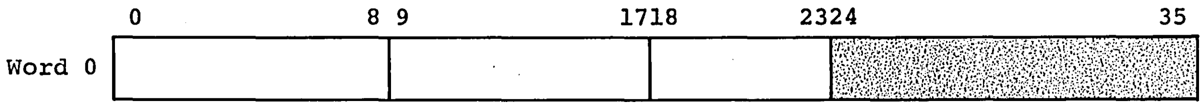
Word 2



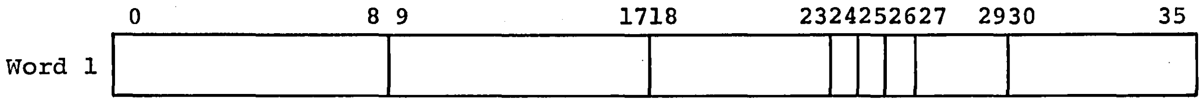
Bits	0-35	Zero

CORE ALLOCATION MESSAGES

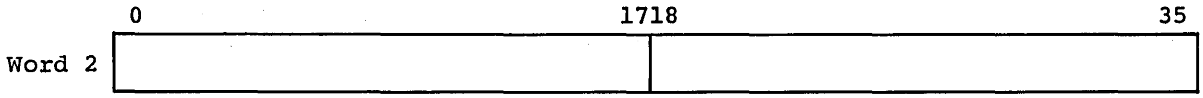
Start New Job



Bits	0-8	.MPOP3
	9-17	Entry Point 1
	18-23	Program number
	24-35	Not used

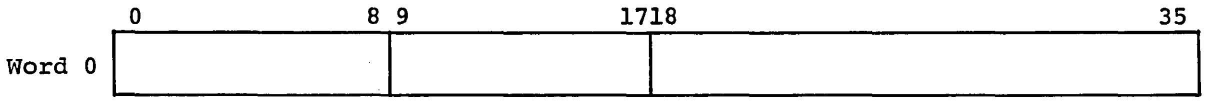


Bits	0-8	Module number or zero. If zero, ordinary slave program load procedure followed. If not zero, module will be GECALLed.
	9-17	Number of blocks, core required
	18-23	Urgency
	24=	1, =7 Job must be allocated at extreme high end of core and have BAR set to access up to 2 to the 18th.
	25=	1, =7 Job should be allocated toward lower end of core
	26=	1, =7 Privileged slave
	27-29	Number of 1024-word Slave Service Area blocks
	30-35	Input station identification



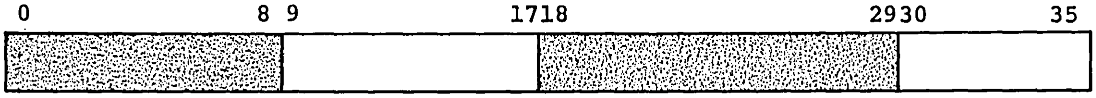
Bits	0-17	SCT table address
	18-35	Link number of file containing SSA for boot

End of Job - Core Release



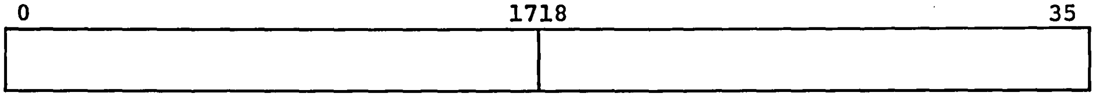
Bits	0-8	.MPOP3
	9-17	Entry point 2
	18-35	Zero

Word 1



Bits	0-8	Not used
	9-17	Allocator action code
	18-29	Not used
	30-35	Program number

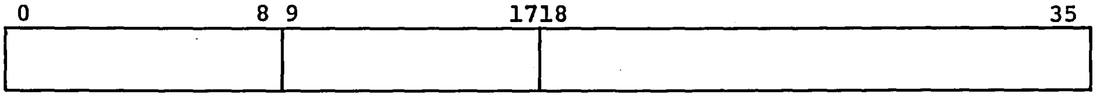
Word 2



Bits	0-17	Address of End message
	18-35	Number of words in End message

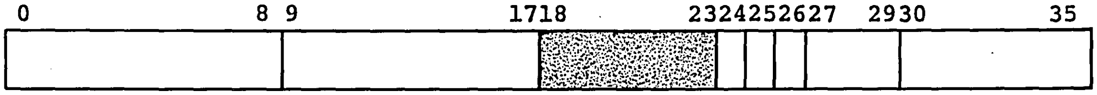
Core Release from GEMREL

Word 0



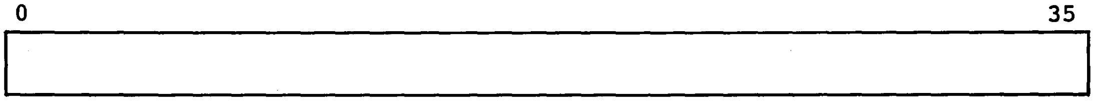
Bits	0-8	.MPOP6
	9-17	Entry point 2
	18-35	Zero

Word 1



Bits	0-8	Not used
	9-17	Number of 512-word blocks of Core
	18-23	Not used
	24=	1, =7. Job must be allocated at extreme high end of core and have BAR set to access up to 2 to the 18th.
	25=	1, =7 Job should be allocated toward lower end of core.
	26=	1, =7 Privileged slave
	27-29	Number of 1024-word Slave Service Area blocks
	30-35	Program number

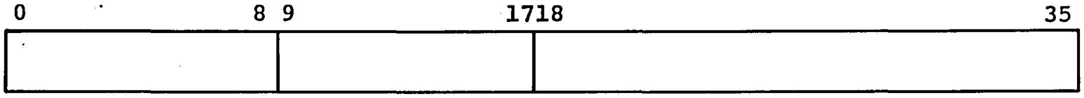
Word 3



Bits	0-35	Zero

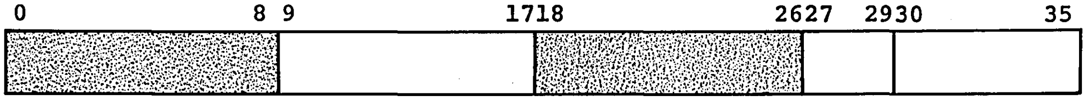
Core Demand From GEMORE

Word 0



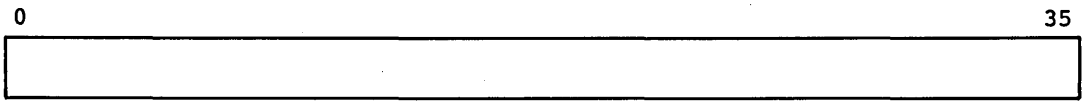
Bits	0-8	.MPOP3
	9-17	Entry point 4
	18-35	Zero

Word 1



Bits	0-8	Not used
	9-17	Number of 512-word blocks of core
	18-26	Not used
	27-29	Number of 1024-word Slave Service Area blocks
	30-35	Program number

Word 2

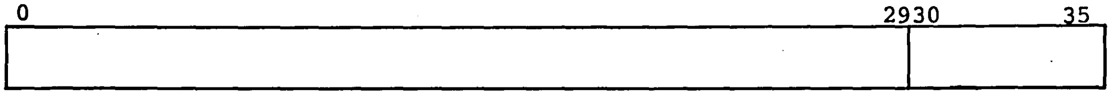


Bits	0-35	Zero

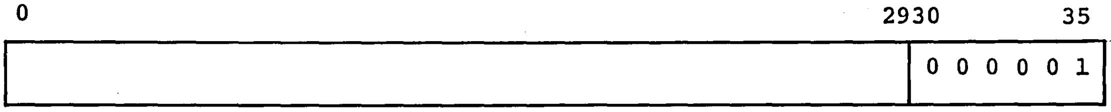
SYSOUT

OPERATOR INTERFACE FOR SYSOUT RECOVERY (QUEUE ENTRIES)

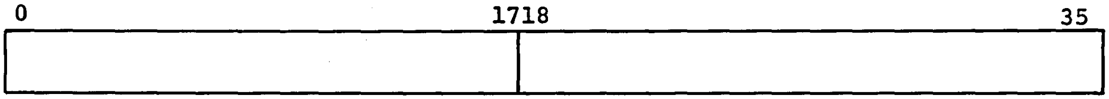
KILLS



Bits	0-29	SNUMB
	30-35	Zero

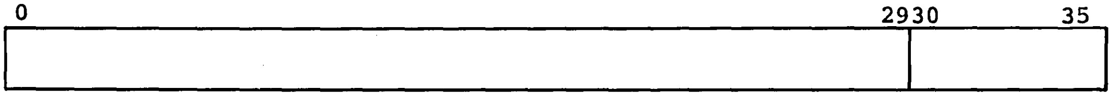


Bits	0-29	Zero
	30-35	Code number (1)

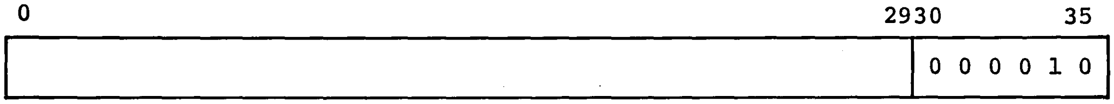


Bits	0-17	Activity number
	18-35	Report code

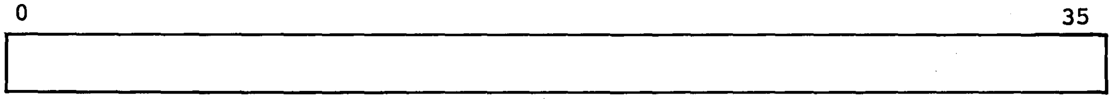
PRIOR



Bits	0-29	SNUMB
	30-35	Zero

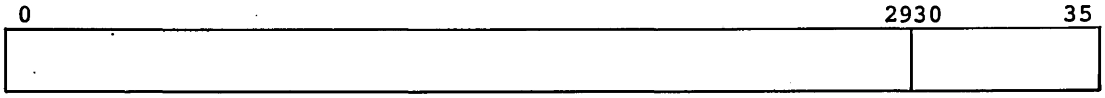


Bits	0-29	Zero
	30-35	Code number (2)

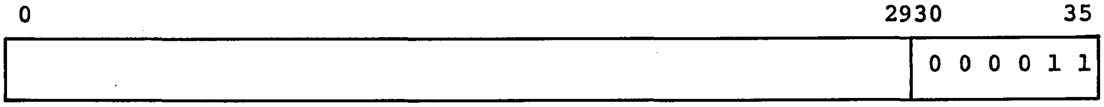


Bits	0-35	Zero

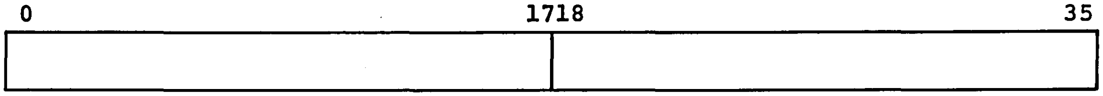
REPRT



Bits 0-29 SNUMB
30-35 Zero

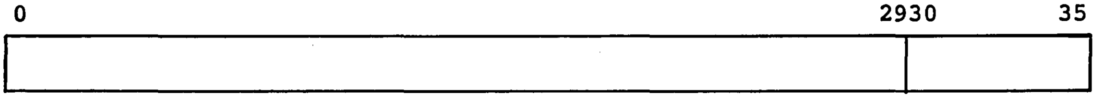


Bits 0-29 Zero
30-35 Code number (3)

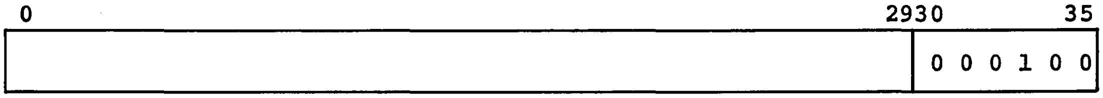


Bits 0-17 Activity number
18-35 Report code

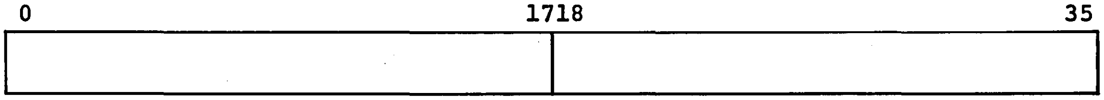
REPNC



Bits 0-29 SNUMB
30-35 Zero



Bits 0-29 Zero
30-35 Code number (4)



Bits 0-17 Activity number
18-35 Report code

RLSE

0 17	18 35

Bits 0-17 SCT address
18-35 Zero

0 29	30 35
	0 0 0 1 0 1

Bits 0-29 Zero
30-35 Code number (5)

0 5	6 17	18 29	30 35
			▨

Bits 0-5 IOC number
6-17 Channel number
18-29 Device number
30-35 Not used

OPERATOR INTERFACE FOR SYSOUT REDIRECT

PURGE ALL

0 29	30 35

Bits 0-29 SNUMB or zero
30-35 Must be zero

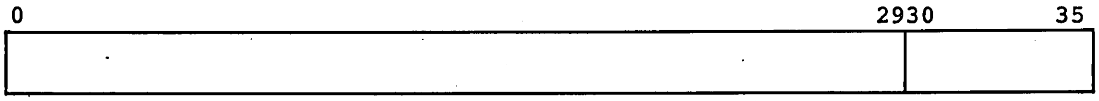
0 29	30 35
	0 0 0 1 1 0

Bits 0-29 Zero
30-35 Code number (6)

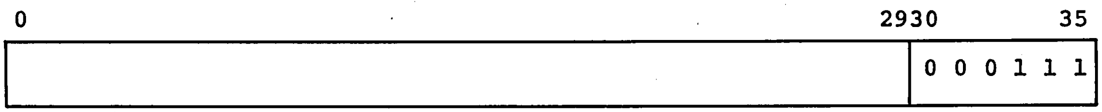
0 35

Bits 0-35 Zero

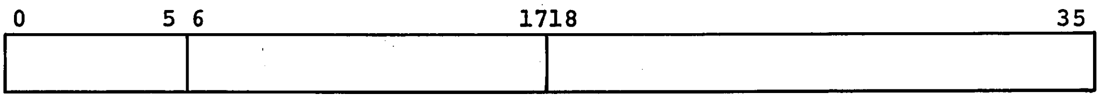
PURGE ID



Bits 0-29 SNUMB or zero
30-35 Must be zero

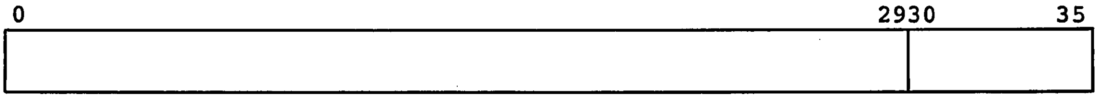


Bits 0-29 Zero
30-35 Code number (7)

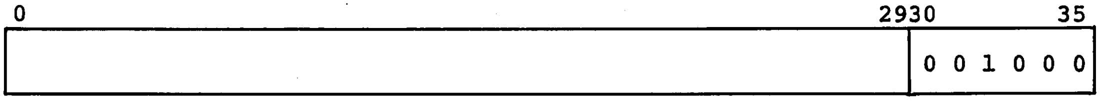


Bits 0-5 Zero
6-17 Remote station identification
18-35 Zero

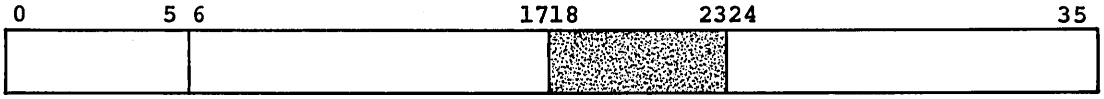
REDRC ID ID



Bits 0-29 SNUMB or zero
30-35 Must be zero

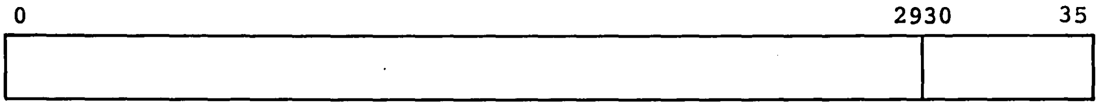


Bits 0-29 Zero
30-35 Code number (8)

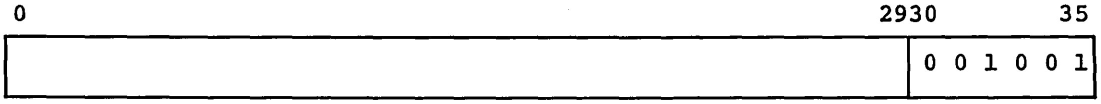


Bits 0-5 Zero
6-17 Remote station identification (from)
18-23 Not used
24-35 Remote station identification (to)

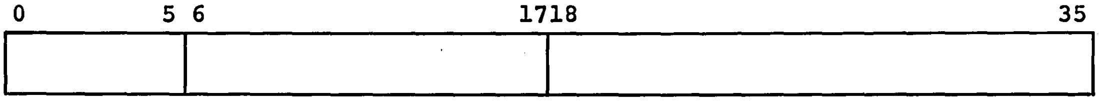
REDRC ID ONL



Bits	0-29	SNUMB or zero
	30-35	Must be zero

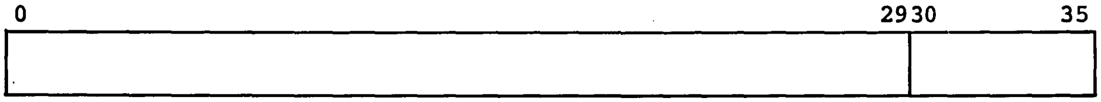


Bits	0-29	Zero
	30-35	Code number (9)

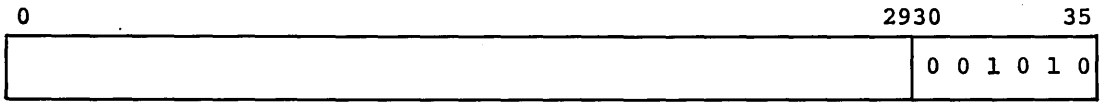


Bits	0-5	Zero
	6-17	Remote station identification
	18-35	Zero

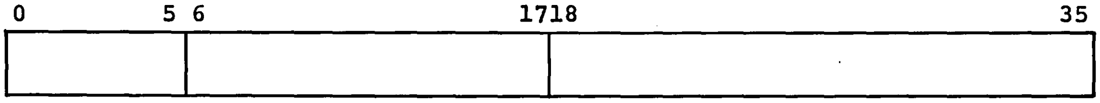
REDRC PRT ID



Bits	0-29	SNUMB or zero
	30-35	Must be zero

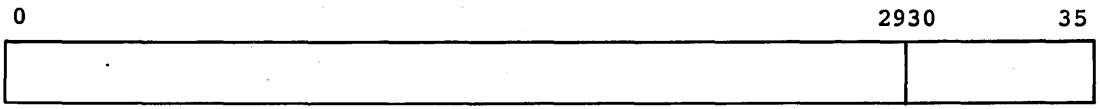


Bits	0-29	Zero
	30-35	Code number (10)

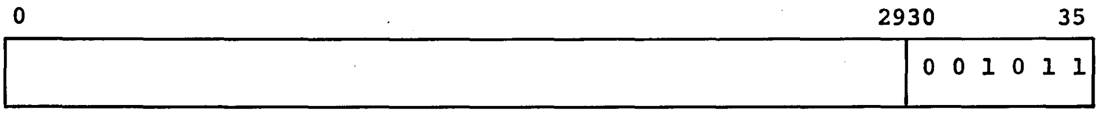


Bits	0-5	Zero
	6-17	Remote station identification
	18-35	Zero

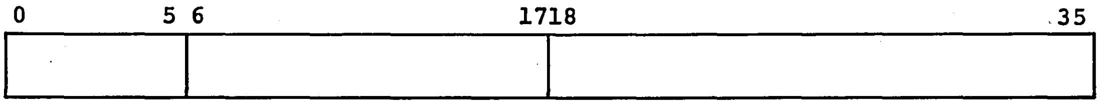
REDRC PNC ID



Bits 0-29 SNUMB or zero
30-35 Must be zero

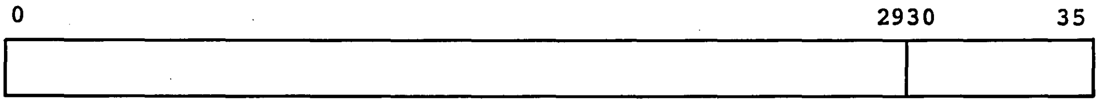


Bits 0-29 Zero
30-35 Code number (11)

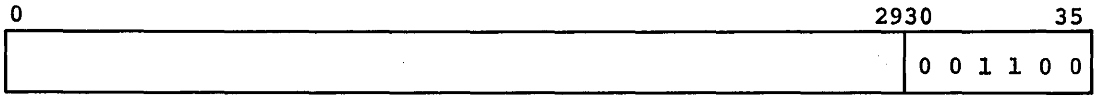


Bits 0-5 Zero
6-17 Remote station identification
18-35 Zero

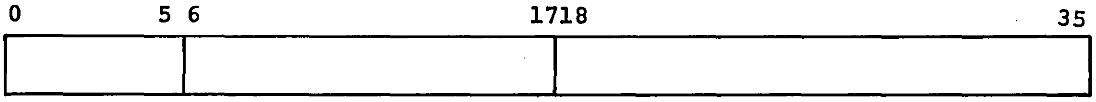
REDRC ALL ID



Bits 0-29 SNUMB or zero
30-35 Must be zero

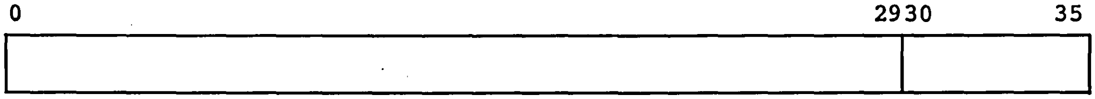


Bits 0-29 Zero
30-35 Code number (12)

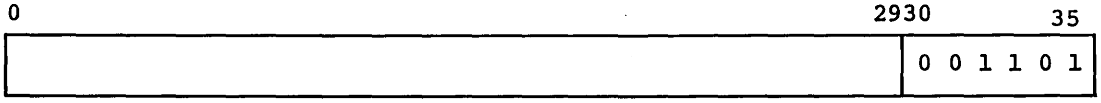


Bits 0-5 Zero
6-17 Remote station identification
18 35 Zero

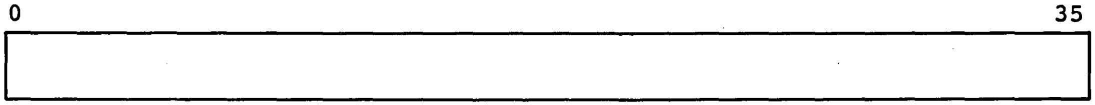
PURGE ONL



Bits	0-29	SNUMB or zero
	30-35	Must be zero

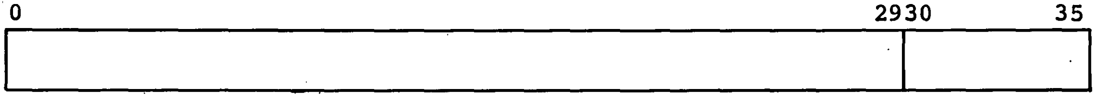


Bits	0-29	Zero
	30-35	Code number (13)

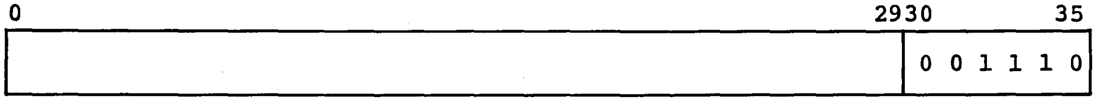


Bits	0-35	Zero

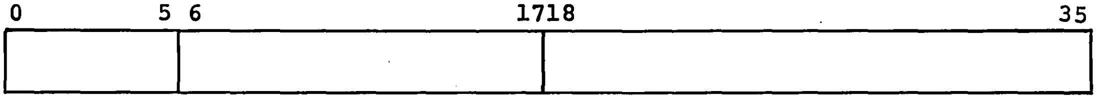
REDRC ONL ID



Bits	0-29	SNUMB or zero
	30-35	Must be zero

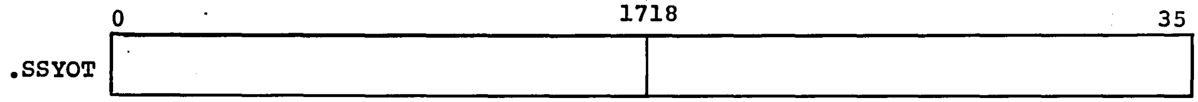


Bits	0-29	Zero
	30-35	Code number (14)



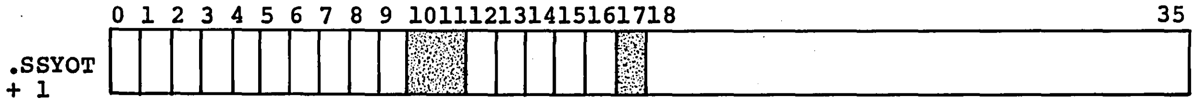
Bits	0-5	Zero
	6-17	Remote station identification
	18-35	Zero

Slave Service Area Reserved for SYSOUT



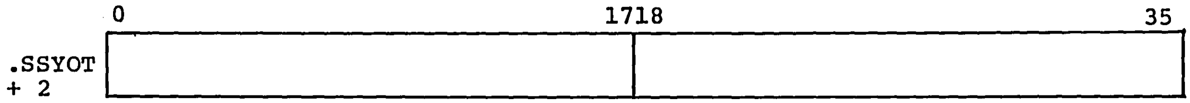
Internal Symbol - LIMIT

Bits	0-17	Maximum record count
	18-35	Current record count



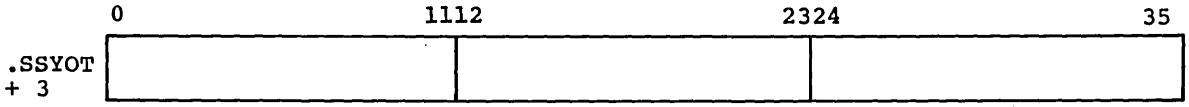
Internal Symbols - STATE and LURSW

Bits	0-15	Current state
	0=0,	Initialized and Not in Courtesy Call
	=1,	Not Initialized or in Courtesy Call
	1=0,	I/O is not in progress
	=1,	I/O is in progress
	2=0,	Not waiting at main level
	=1,	Waiting at main level
	3=0,	No buffers
	=1,	Have extra SSA buffer
	4=0,	No more I/O needed
	=1,	More I/O needed
	5=0,	No DCW string
	=1,	Building a DCW string
	6=0,	Buffer not full
	=1,	Buffer full
	7=0,	Buffer empty
	=1,	Buffer being filled
	8=0,	Current report not in buffer
	=1,	Current report in buffer
	9=0,	No EOF on this report
	=1,	EOF on this report
	10-11,	Not used
	12=0,	Number of reports within limits
	=1,	Too many reports in this activity
	13=0,	No $ REPORT cards given
	=1,	$ REPORT cards given
	14=0,	Number of Remote destinations (less than or equal to 3)
	=1,	Too many Remote destinations (greater than 3)
	15=0,	Current request (if MME) has file code
	=1,	Current request does not have file code
	16=0,	Current MME from slave
	=1,	Current MME from SSA
	17	Not used
	18-35	Relative location of user's status word



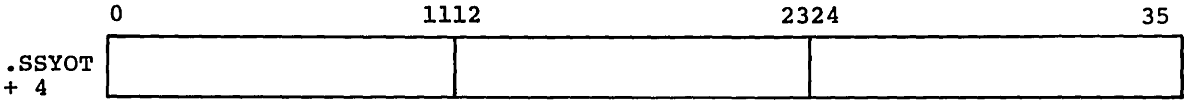
Internal Symbols - NORMC and DESTN

Bits 0-17 Number of reports
18-35 Current mask for destination



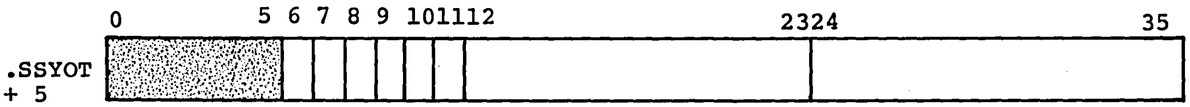
Internal Symbol - RMTLK

Bits 0-11 Next printer link
12-23 Next link for destination 1
24-35 Destination code 1



Internal Symbol - RMTLK + 1

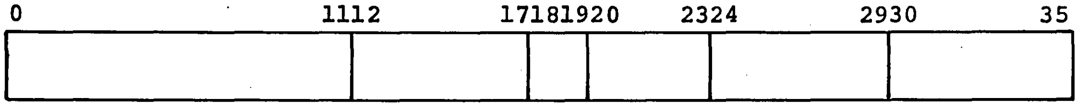
Bits 0-11 Next punch link
12-23 Next link for destination 2
24-35 Destination code 2



Internal Symbol - RMTLK + 2

Bits 0-5 Not used
6=1, Print
7=1, Remote J*
8=1, Punch
9=1, Remote no. 1
10=1, Remote no. 2
11=1, Remote no. 3
12-23 Next link for destination 3
24-35 Destination code 3

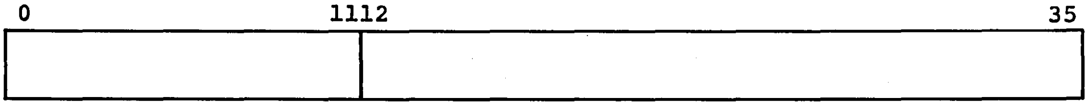
.SSYOT
+ 6



Internal Symbol - RMLST (first word of entry)

Bits	0-11	Current link
	12-17	Number of blocks left in 320
	18-19	Fences left in link
	20-23	Number of 320's to next fence
	24-29	Media code
	30-35	Report code

.SSYOT
+ 7



Internal Symbol - RMLST (second word of entry)

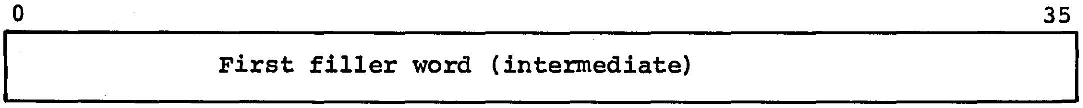
Bits	0-11	First link
	12-35	Current count

.SSYOT
+ 8
through
.SSYOT
+ 23

Internal Symbol - RMLST

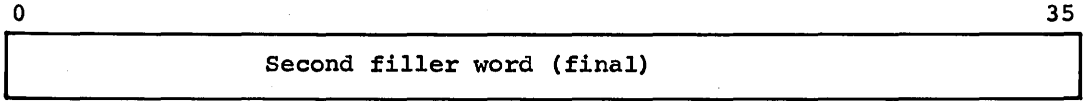
2-word entries (same as above) for each of 9 reports and/or 3-word entries for contents of up to 6 $ REPORT cards set by allocator

.SSYOT
+ 24



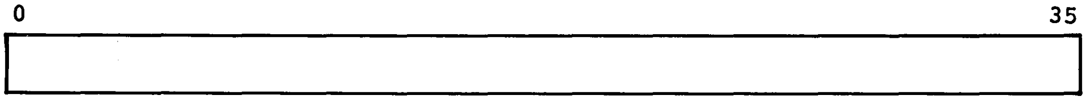
Internal Symbol - DCWFA

.SSYOT
+ 25



Internal Symbol - DCWFB

+ 26



Internal Symbol - LNKNO
Transient Use Symbol - RMMSK

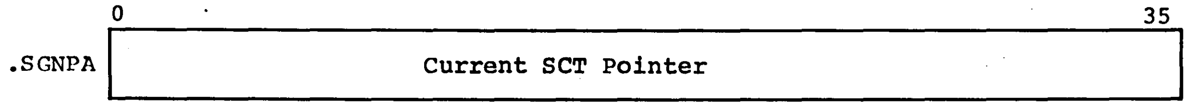
	0	35
.SSYOT + 27 through + 33	DCW list	

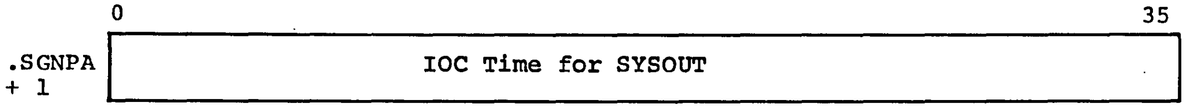
Internal Symbol - DCWST

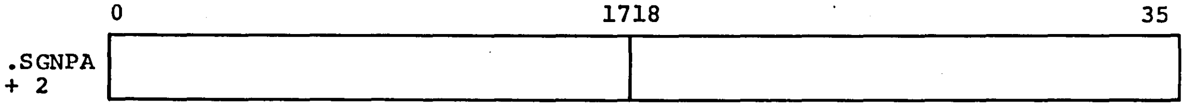
	0	35
.SSYOT + 34 & + 35	Status Words	

Internal Symbol - STATW
Transient Use Symbols - BKDTA
ABLKN

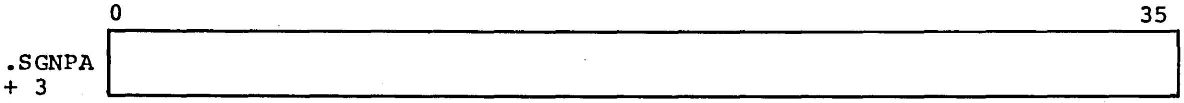
J*PAT For All Slave Programs Including GEIN and ALOC



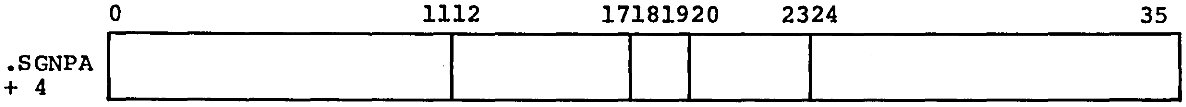




Bits 0-17 SCT pointer for beginning
18-35 First link number for beginning



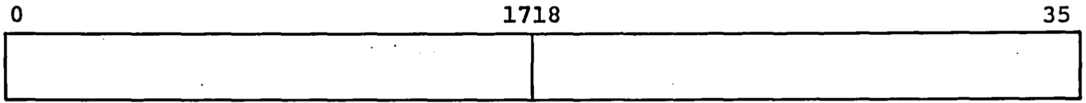
Word reserved to record positioning data for last I/O for error detection and recovery.



Bits 0-11 Link number of location of next execution report line
12-17 Number of addressable blocks left in current 320-word block
18-19 Number of fences left in this link
20-23 Number of 320-word blocks left to this fence
24-35 BLINK number for first link of J* for this job.

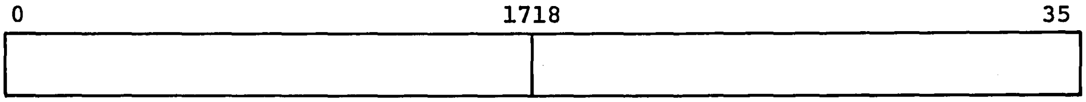
Communication Region SYSOUT Table

.CRSYT



```
Bits    0-17  Number of devices (of same type) on which SYSOUT
              links are assigned(1 ≤ devices ≤ 13)
       18-35  Total number of links assigned
```

.CRSYT
+ 1



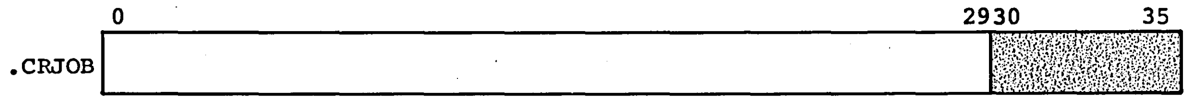
```
Bits    0-17  SCT pointer for device 1
       18-35  Link number of the first of n consecutive links
              assigned on device 1, where n is the number of links
              divided by the number of devices
```

.CRSYT
+ 2
through
+ 13

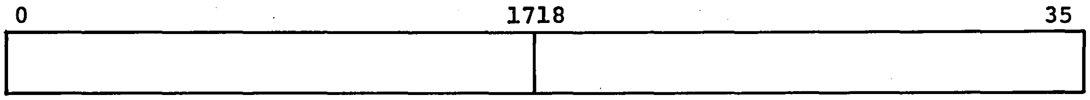
Same as .CRSYT + 1, for devices 2 - 13

LIST FORMATS FOR MOVING JOBS

To Allocator From GEIN

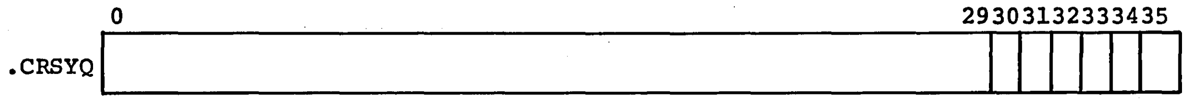


Bits	0-29	Job sequence number
	30-35	Not used

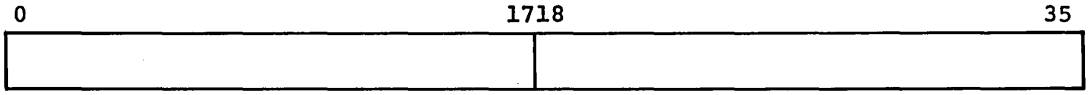


Bits	0-17	L(SCT) Table entry for J* device
	18-35	First link number assigned to J*

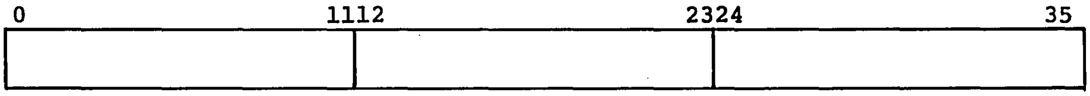
To GOUT When Job is Finished



Bits	0-29	Job sequence number
	30=1,	Print
	31=1,	Remote J*
	32=1,	Punch
	33=1,	Remote number 1
	34=1,	Remote number 2
	35=1,	Remote number 3



Bits	0-17	L(SCT) Table entry for J* device
	18-35	First link number assigned to J*

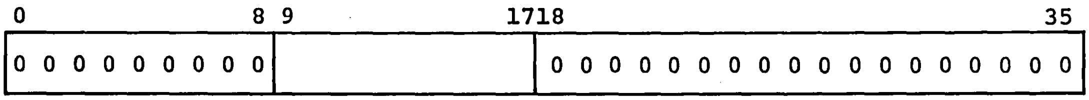


Bits	0-11	Identification number of remote device 1
	12-23	Identification number of remote device 2
	24-35	Identification number of remote device 3

SPECIAL RECORD CONTROL WORDS

(Contained in data produced by SYSOUT for GOUT to process)

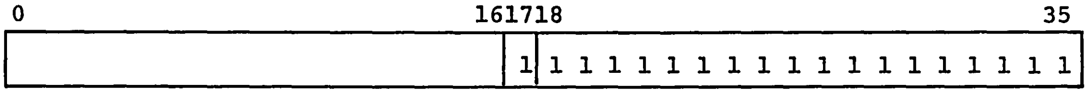
Filler Record



Bits	0-17	Record size in number of words
	0-8	Zero
	9-17	Number of words in record
	18-35	Record type - must be zero

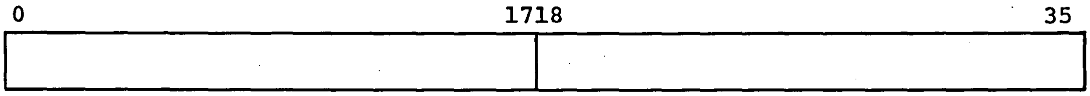
Link Pointer on J*

Word 1



Bits	0-17	Record size in number of words
	0-16	Zero
	17=	One word record
	18-35	Record type

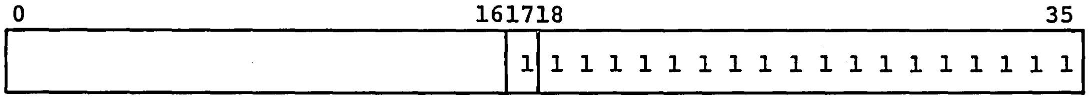
Word 2



Bits	0-17	SCT pointer
	18-35	Link number

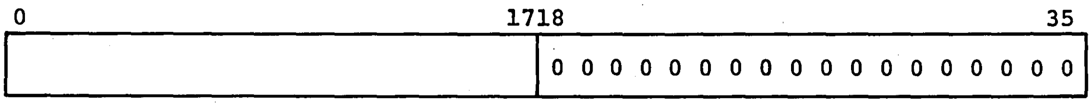
Link Pointer in User Reports

Word 1



Bits	0-17	Record size in number of words
	0-16	Zero
	17=	One word record
	18-35	Record Type

Word 2



Bits	0-17	SYSOUT link number
	18-35	Must be zero

End of Activity Data Record

0 11	12 17	18 35
	0 1 1 0 0 0	1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0

Bits 0-17 Record size in number of words
0-11 Zero
12-17 30 words
18-35 Record type

End of Job Flag

0 17	18 35
	1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0

Bits 0-17 Zero
18-35 Record type

$ REPORT Card Data Record

0 11	12 17	18 35
	0 1 0 0 1 0	0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0

Bits 0-17 Record size in number of words
0-11 Zero
12-17 22 words
18-35 Record type

Header Word for Multi-Print-Line Logical Records

0 8	9 17	18 35
		0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits 0-17 Record size in number of words
0-8 Zero
9-17 Number of words
18-35 Record type

Dump Record in Binary to be Expanded

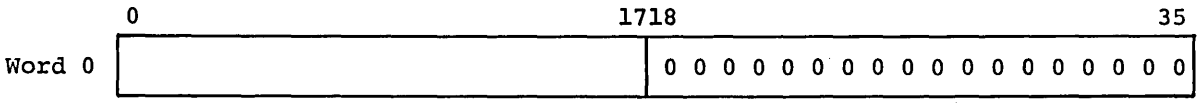
0 8	9 17	18 35
		0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0

Bits 0-17 Record size in number of words
0-8 Zero
9-17 Number of words
18-35 Record type

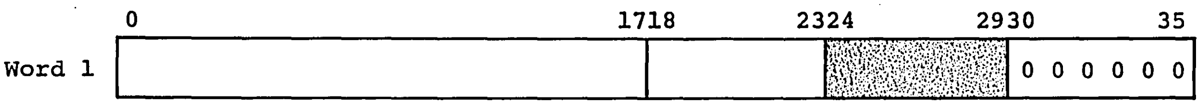
PERIPHERAL ALLOCATOR

Q* FILE

FIFO List

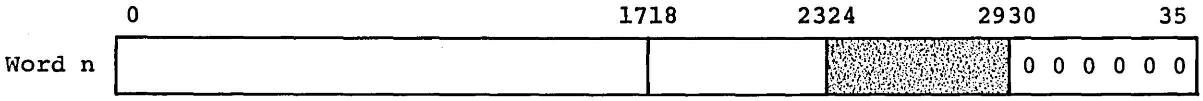


Bits	0-17	Number of entries
	18-35	Must be zero



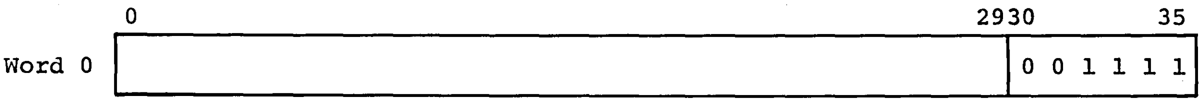
Bits	0-17	Control block pointer 1
	18-23	Urgency 1
	24-29	Not used
	30-35	Must be zero

•
•
•
•



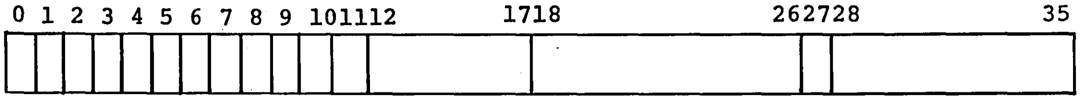
Bits	0-17	Control block pointer n
	18-23	Urgency n
	24-29	Not used
	30-35	Must be zero

Job Control Table (One per job)



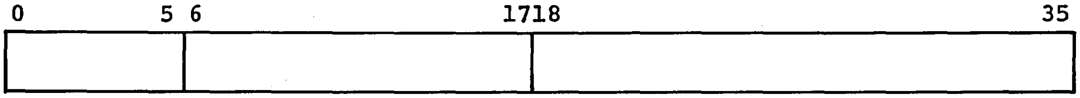
Bits	0-29	Job sequence number
	30-35	GERTS/TS code

Word 1



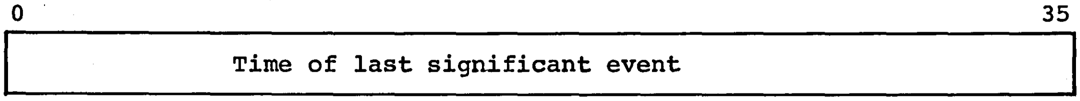
Bits 0-11 Job status
0=1, Activity abort
1=1, Kill or terminate
2=1, Hold
3=1, Limbo
4=1, Allocation delayed
5=1, Allocation complete
6=1, Tape Ready wait
7=1, New job
8=1, New activity
9=1, Sieve
10=1, Critical tape job
11=1, Tapes are saved
12-17 Program number
18-26 Memory blocks needed
27= 1, Job from time-sharing system
28-35 Current activity number

Word 2

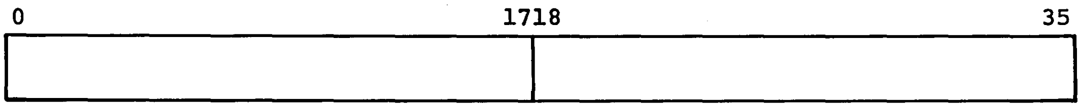


Bits 0-5 Device type needed
6-17 Number of devices needed
18-35 Allocation activity count

Word 3

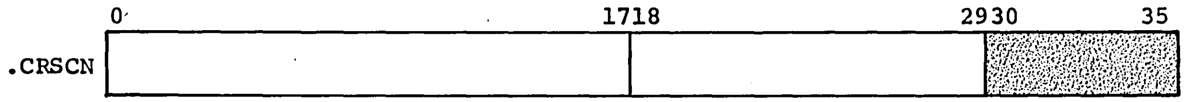


Word 4

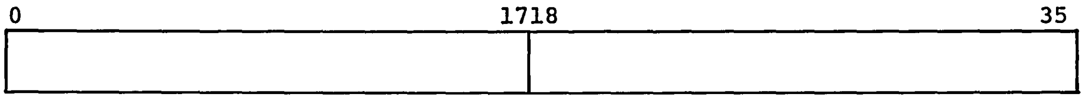


Bits 0-17 Control data (J*/*J) SCT pointer
18-35 Control data (J*/*J) Link number

SYSTEM NAME TABLE

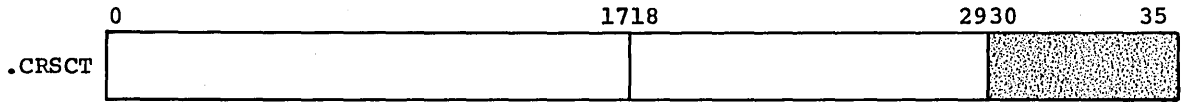


Bits	0-17	Address of name table
	18-29	Number of entries in table
	30-35	Not used

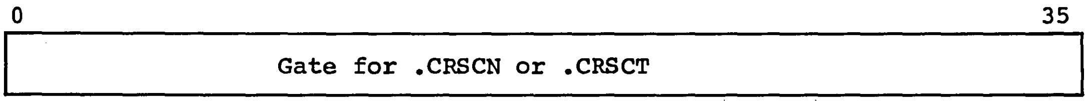


Bits	0-17	Address of device SCT table
	18-35	Device name

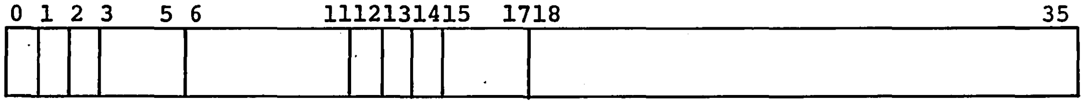
SECONDARY SCT POINTER



Bits	0-17	Address of first SCT
	18-29	Number of entries in secondary tables
	30-35	Not used

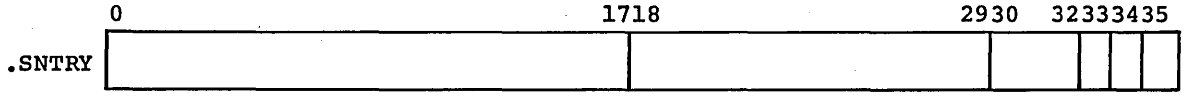


.CRMDD (one for each module)



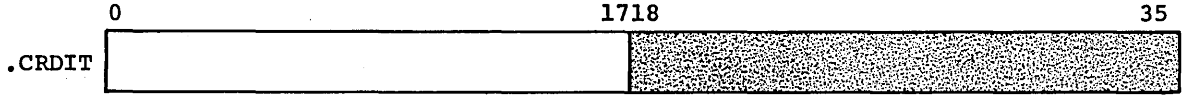
Bit 0 = 0=greater than HCM and Re-entrant
1 Not Re-entrant
2 Busy
3-5 Device table index
6-11 Program number
12= 1, Patch for this program exists in .CRPCH
13= 1, This program required
14= 1, This program loaded
15-17 Type= 0, GEPR
1, SSA
2, HCM
4, Absolute program
5, Relocatable program
18-35 HCM address or relative block number

ENTRY WORD IN MODULE

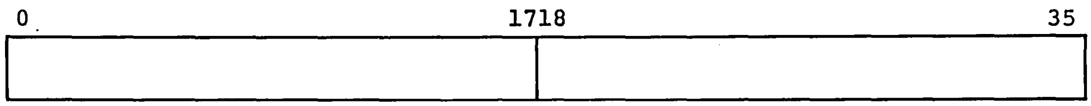


Bits 0-17 Module number
18-29 Module length
30-32 Type= 0, GEPR
1, SSA
2, HCM
4, Absolute program
5, Relocatable program
33 Always 1
34= 1, Module busy (in SSA only)
35= 1, Not Re-entrant

DEVICE INDEX TABLE

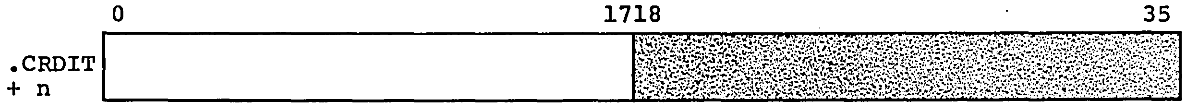


Bits 0-17 SCT address for device holding File 1
18-35 Not used

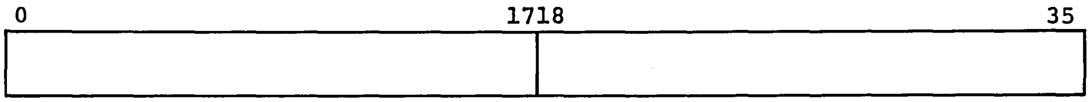


Bits	0-17	Link number
	18-35	Number of links

⋮



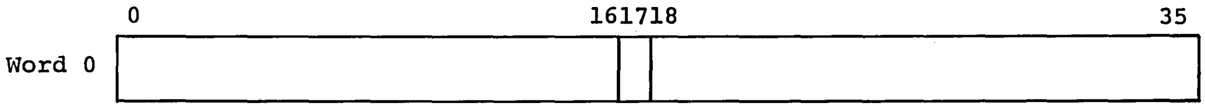
Bits	0-17	SCT address for device holding File n
	18-35	Not used



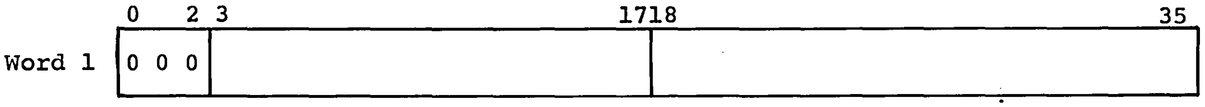
Bits	0-17	Link number
	18-35	Number of links

J* FILE POINTERS

Data File Pointer (J*)

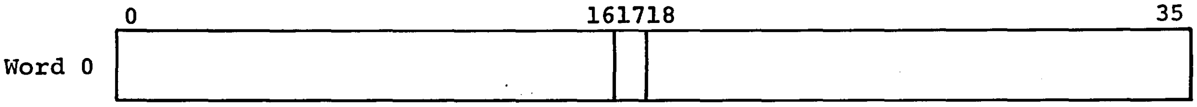


Bits	0-16	Zero
	17=	1
	18-35	Zero

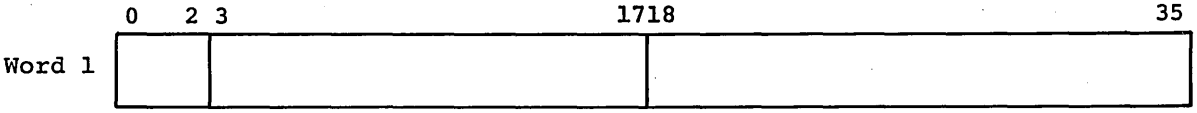


Bits	0-2	Zero
	3-17	Number of STD records
	18-35	Relative *J record number

R* File Pointer (*J)

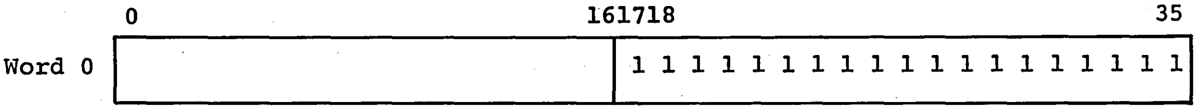


Bits	0-16	Zero
	17=	1
	18-35	Zero

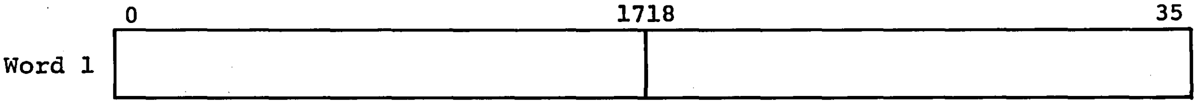


Bits	0-2	010 or 011
	3-17	Number of STD records
	18-35	Relative *J record number

Control File Pointer (*J)

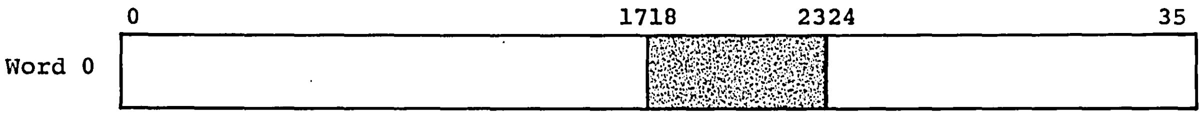


Bits	0-16	Zero
	17-35	Ones

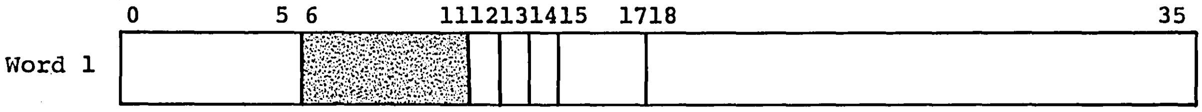


Bits	0-17	SCT pointer of continuation of *J
	18-35	Link number of continuation of *J

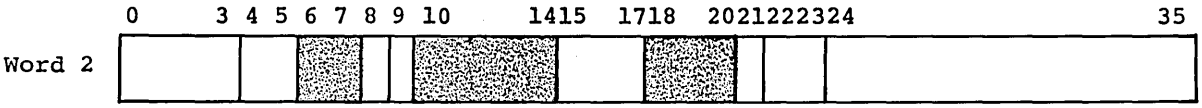
ALLOCATION DETAIL ENTRY



```
Bits    0-17  Number of words in entry
       18-23  Not used
       24-35  Reel sequence number (for magnetic tape only)
```



```
Bits    0-5   Device type
        6-17= Random access device--Number of links required
            = Magnetic tape--as follows:
              6-11  Not used
                12= 1, if file serial number present
                13= 1, if tape not ready
                14= 1, if file name present
              15-17 Number of tapes required
       18-35  Logical unit or device name
```



```
Bits    0-3   File access
        4-5   Alternate disposition code
        6-7   Not used
          8= 0, Linked
           = 1, Random
          9= 0, Temporary
           = 1, Permanent
       10-14  Not used
       15-17  Number of options
       18-20  Not used
          21= 1, SYSOUT device
       22-23  Disposition code
              00=R
              01=S
              10=D
              11=C
       24-35  File code
```

	0 5	6 35
Word 3	0 0 1 0 1 1	

Optional - Present when device is magnetic tape

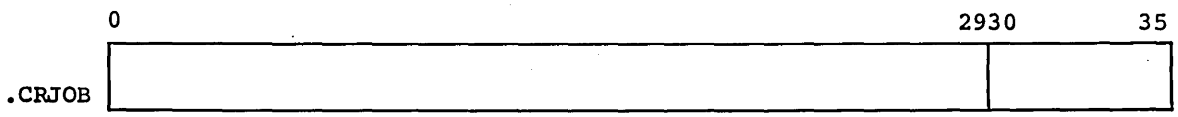
Bits 0-5 = 13
6-35 File serial number or 99999 in BCD

	0 35
Words 4 & 5	

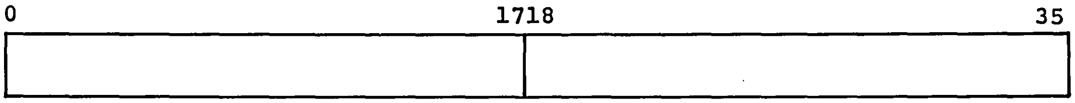
Optional - Present when bit 14, word 1 = 1

Bits 0-35 Filename in BCD

ALLOCATOR INPUT QUEUE ENTRY



Bits	0-29	Job sequence number
	30-35	Program number (new job)



For a new job entry from GEIN:

Bits	0-17	J* SCT pointer
	18-35	Link number of beginning of J*

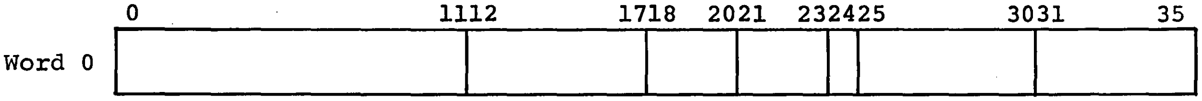
For an old job:

Bits	0-17	Action code = 0, Delete .CRJOB entry 1 , New activity 2, End job 3, Kill or terminate request 4, Hold request 5, Run request 6, New urgency 7, Unused
Bits	18-35	If action code = 0, 1, or 2, program number If action code = 6, urgency value If action code = 3, 4, 5, or 7, not used

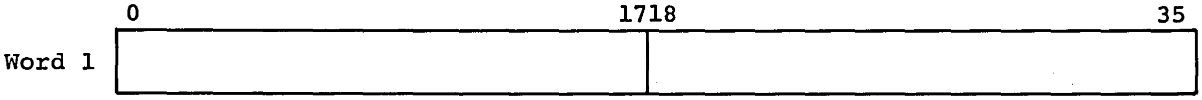
.MRGIN

TABLE T1

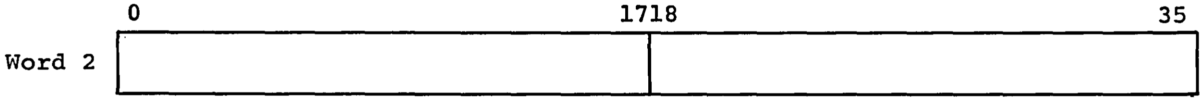
Table contains one 5-word entry for each job waiting for more data blocks. Last operation code is (octal) 25 or 26.



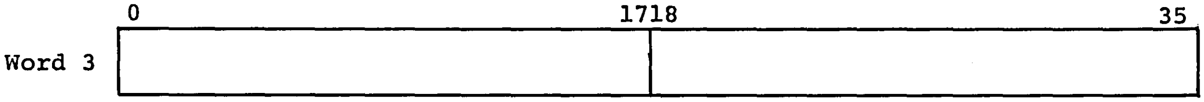
Bits 0-35 Zero if entry not occupied
0-11 Terminal identification
12-17 Program number assigned to that job
18-20 DN-30 processor index, 0, 1, 2, or 3
21-24 Status
23= 1, Error detected in Courtesy Call 2 routine but "Input Accepted" already sent to the GE-600. Error processing in Courtesy Call 1 is delayed until an "Accept Input" is received for same terminal.
25-30 Terminal type
31-35 Last operation code, 25 or 26 (octal)



Bits 0-17 Number of data blocks received for the job
18-35 Number of blocks allowed on the file for this job



Bits 0-17 Offset to PAT pointer
18-35 File code



Bits 0-17 Offset to PAT entry
18-35 Number of link strings in PAT

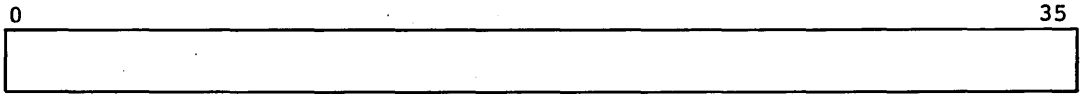
	0	35
Word 4		

Bits 0-35 Word pointed to by DCW associated with a Seek instruction DCW in Table TB (word 3 of Table TB)

TABLE TB

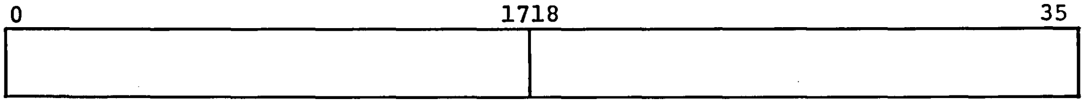
Table contains one 6-word entry for each buffer able to receive data blocks from terminals.

Word 0



Bits 0-35 Transfer to Courtesy Call 2 (TSX2 CC2)

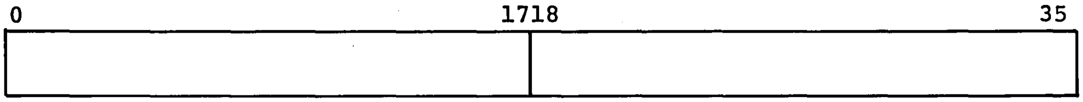
Word 1



Bits 0-35 All ones
or
0-17 DN-30 processor Index 2 if the corresponding buffer is linked to a DATANET-30 and ready to receive a data block from this DATANET-30, or if the buffer is occupied by a data-block during the write operation of this data block onto the disc.
18-35 Pointer to an entry in Table T1 if the buffer is occupied by a data block for a job corresponding to this entry in T1.

Words 2 through 5 are used for the MME GEINOS calling sequence to write a data block onto disc/drum.

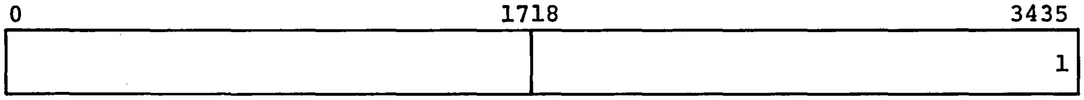
Word 2



DCW for Data Transfer

Bits 0-17 Buffer address
18-35 Must be zero

Word 3



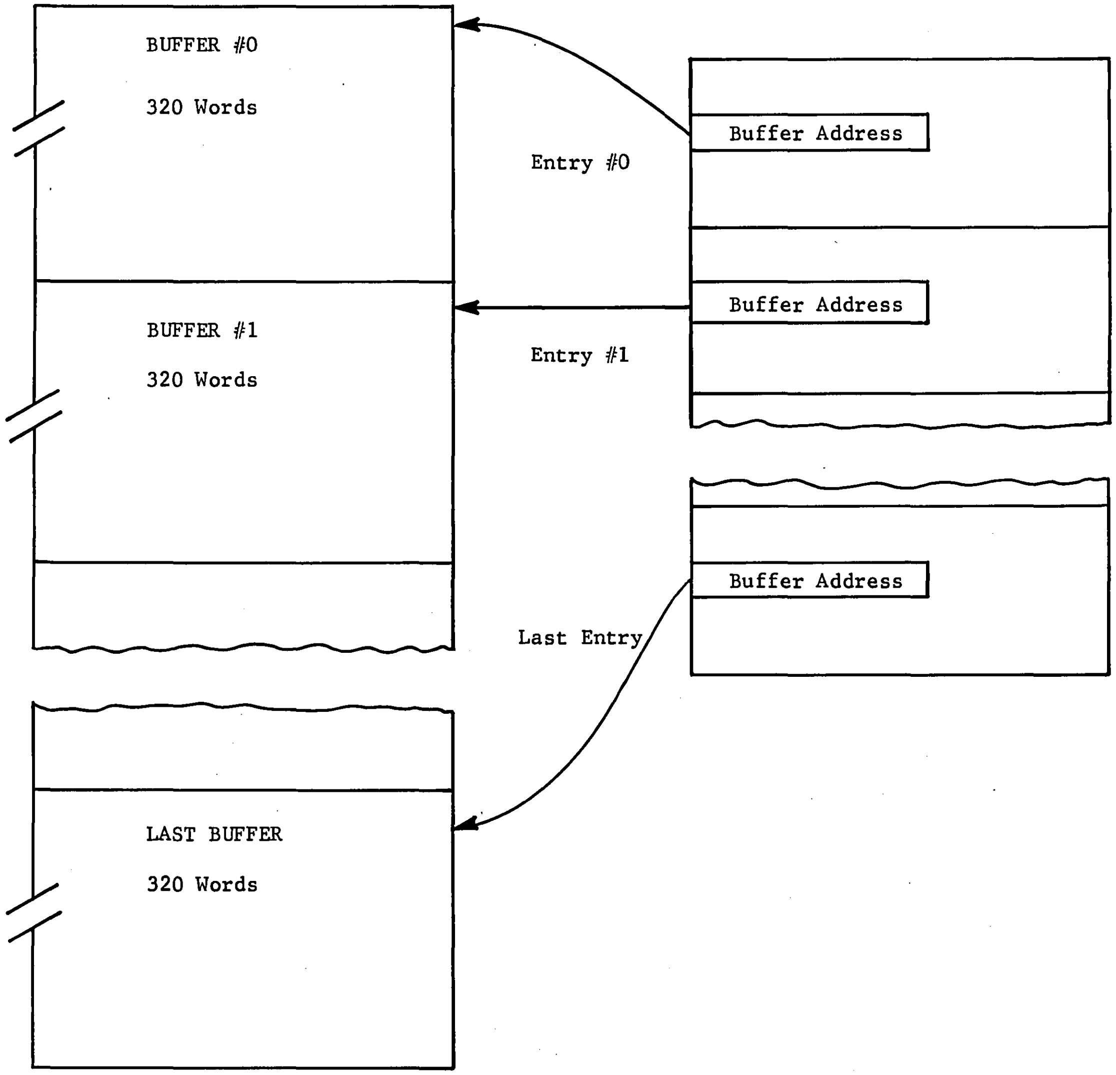
DCW for Seek command

Bits 0-17 Address of word 4 of corresponding entry in table T1
18-34 Must be zero
35 One

	0	35
Word 4	Status Word 1	

	0	35
Word 5	Status Word 2	

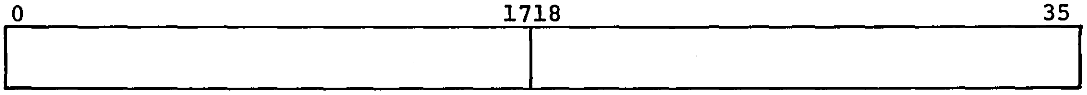
TB TABLE-I/O BUFFER RELATIONSHIP



DATANET-30 TABLE

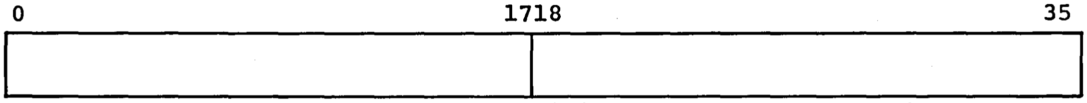
(One 2-word entry for each DATANET-30 processor.)

STDN



Bits 0-35 All ones
or
0-17 Pointer to entry in T1 when an ACCEPT INPUT has not been answered because of a shortage of buffers.
18-35 Pointer to entry in TB when a buffer is linked to this DATANET-30.

NJDN



Bits 0-17 Number of jobs in T1 waiting for more data blocks through this DATANET-30.
18-35 Must be zero

.MDNET

LINE INTERFACE TABLES

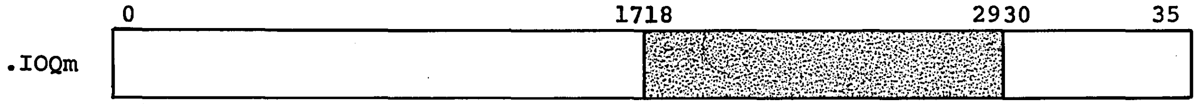
The HCM tables, .CRREQ, .CRPAR, and .CRMSC, point to tables in .MDNET as shown below:

.CRREQ--.IOQA, .IOQB,.IOQC

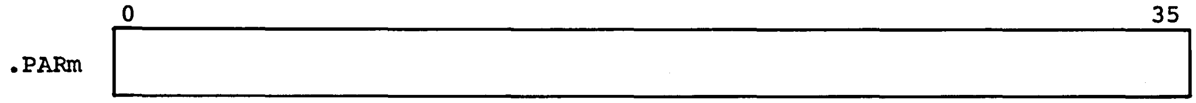
.CRPAR--.PARA, .PARB, .PARC

.CRMSC--.MSCA, .MSCB, .MSCC

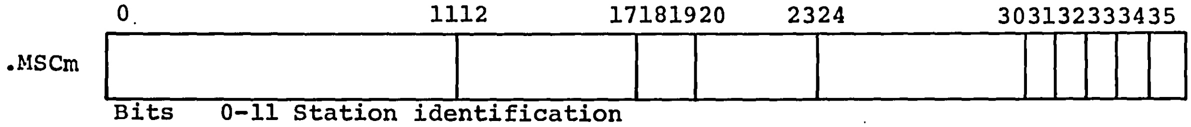
Each of these .MDNET tables contains one entry for each line in a DATANET-30. Up to three DATANET's may be used. The "A" tables are for the first, the "B" tables for the second, and the "C" tables for the third. In the descriptions given below, m may be A, B, or C.



Bits 0-17 Offset to associated I/O queue
18-29 Not used
30-35 Program number



Bits 0-35 BCD name when line is in connect to slave mode



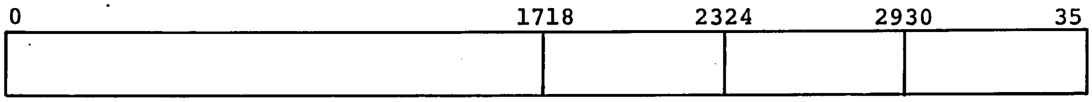
Bits 0-11 Station identification
12-17 Terminal type
18-19 DATANET-30 Index (0, 1, 2)
20-23 Current line interface mode

0 - Disconnect
1 - Idle
2 - Wait for activity termination
3 - Direct access
4 - Remote SYSOUT
5 - Remote GEIN
6 - Timesharing
7 - Connect to slave
8 - Connect to slave, wait for I/O queue
9 - GEPOP processing abort request

24-30 Line number
31 Zero
32= 1, SYSOUT processing status request (SYSOUT mode)
33= 1, GE-600 response to "Accept Input, Last of Current SNUMB" (remote GEIN mode), or 600 is waiting for response following Write or Write Then Read (Direct Access or Timesharing mode)
34= 1, Break received on this line
35= 1, Line is busy

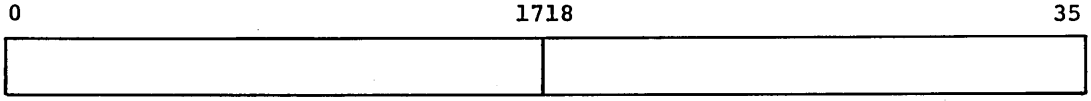
Each of the four following tables contains one entry per DATANET-30 and can be referenced by modules external to .MDNET through the entry point indicated.

.RLOG
(EP11)



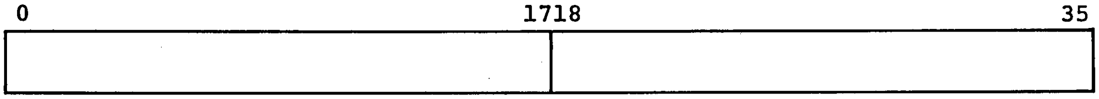
Bits 0-17 Logical channel index
18-23 Must be zero
24-29 DATANET-30 index (0,1,2)
30-35 DATANET-30 index multiplied by 3 (0,3,6)

.RRD1Q
(EP12)



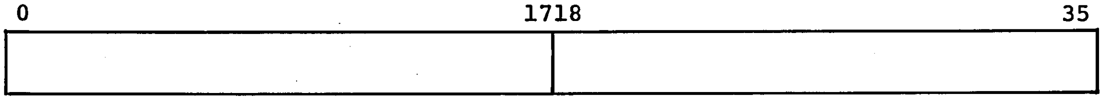
Bits 0-17 Absolute location of Read Response 1 I/O queue that remains permanently linked in GEPOP
18-35 Must be zero

R.GINQ
(EP13)



Bits 0-17 Relative address of Read Response 2 I/O queue for remote GEIN
18-35 Must be zero

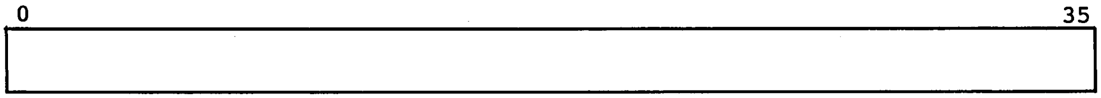
R.SOQ
(EP14)



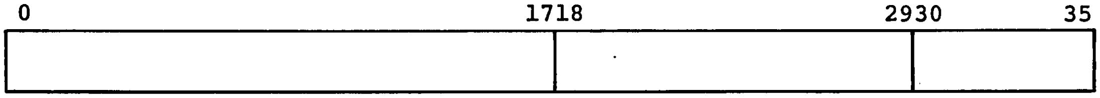
Bits 0-17 Relative address of last output queue used by GEOT (remote SYSOUT). Used to return status and pay courtesy call upon next request for output.
18-35 Must be zero.

.RINQ contains two words per entry and one entry for each allowable Direct Access or Timesharing program. The number of entries is currently set at 16.

.RINQ
(EP15)



Bits 0-35 BCD connect to slave name



Bits 0-17 Relative location of I/O queue containing slave program's remote inquiry information
18-29 Must be zero
30-35 Program number

INDEX

DOCUMENT REVIEW SHEET

TITLE: GE-625/635 GECOS-III Introduction and System Tables

CPB #: 1488

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: ____________________

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

Please cut along this line

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
PROCESSOR EQUIPMENT DEPARTMENT
13430 NORTH BLACK CANYON HIGHWAY
PHOENIX, ARIZONA 85029

ATTENTION: DOCUMENTATION B-90

FOLD

INFORMATION SYSTEMS

GENERAL ELECTRIC

LITHO U.S.A.