# 3300 3500

# COMPUTER SYSTEMS MASTER

## GENERAL INFORMATION MANUAL

CONTROL DATA
CORPORATION

# 3300
# 3500

## COMPUTER SYSTEMS
## MASTER

GENERAL INFORMATION MANUAL

CONTROL DATA
CORPORATION

Additional copies of this manual may be obtained
from the nearest Control Data Corporation Sales
office listed on the back cover.

October, 1965
Pub. No. 60134100

CONTROL DATA CORPORATION
*Documentation Department*
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

# CONTENTS

# INTRODUCTION

The Control Data® Multiple Access Shared Time Executive Routine (MASTER) is a multiprogramming system that takes full advantage of the Executive mode and multiprogramming capabilities of the Control Data 3300 and 3500 Computer Systems. It is adaptable to applications involving multi-access on-line input/output with and without real time calculations as well as to conventional and batch processing applications.

MASTER increases the overall efficiency of the computing system by minimizing the idle time of the various processors: compute modules and data channels. It accomplishes this by simultaneous consideration of more than one job so that activities can be found for the processors whenever they become idle.

MASTER consists of a System Executive and an Operating System. The Operating System accepts jobs and divides them into executable entities called tasks, which are manipulated by the System Executive. The System Executive assigns tasks to available processors and administers communication among tasks. Interrupts provide a vehicle of communications between the System Executive operating in the monitor state and the Operating System (and user programs) operating in the program state. The Operating System operates in the program state because its functions appear as tasks to the System Executive. The monitor and program states are substates of the executive mode – a multiprogramming mode that is console selectable on the 3300/3500 computer systems.

The multiprogramming options in the 3300 and 3500 computer systems enable several jobs to be allocated to different sections of core and executed concurrently. Jobs are subdivided into program and input/output tasks, each of which is processed according to the priority of the associated job. Priority is determined by job class: background (highest), short duration, input/output bound, and compute bound. Background jobs are usually peripheral processing routines; heavily I/O bound and requiring little central processor time. Definition of short, I/O, and compute jobs is established by system parameters. In addition to the normal priority scheme, one emergency job may be submitted at any time and initiated immediately.

As a result of multiprogramming logic, MASTER tries to keep active one job from each class at all times. A job is active when it has tasks in core memory ready for execution. Because execution of tasks is interruptible, no one task may dominate the compute module except an emergency job. When an interrupt occurs during execution of a task in the program state, control reverts to the System Executive which processes the interrupt and then returns control to one of the active tasks so that it begins or resumes execution.

A mass storage input/output system loads, unloads, and organizes files on mass storage devices. It standardizes file formats and descriptions, and user interface regardless of hardware type.

Whenever possible, standard error procedures permit the system to recover. When recovery is not possible, the user is informed. The system and the operator exchange information on a comment medium.

# SYSTEM CONFIGURATION 1

## 1.1 HARDWARE

MASTER operates on a 3300 or 3500 computer system consisting of a variety of functional modules selected to meet specific or general-purpose requirements. The system may be expanded with ease as data processing requirements increase. MASTER utilizes the executive mode and relocation features of the 3300 and 3500 systems.

A typical 3300 or 3500 computer system for MASTER includes:

Central Processor Unit (3304 or 3504)

Four to eight Data Channels
(one 3307 or 3507 bidirectional 24-bit high speed parallel data channel; three to seven 3306 or 3506 bidirectional 12-bit parallel data channels)

32 to 65K Magnetic Core Storage
(3309 or 3509 storage module, 8,192 words plus one to three 3302 or 3502 storage modules, 16,384 words)

Multiprogramming Module (10028)

Disk Files and Controller

Drum Files and Controller

Card Reader (buffered) and Controller

Card Punch (buffered) and Controller

Printer and Controller

Two to four Magnetic Tape Units and Controller

Console with Typewriter (3301 or 3501)

### PERIPHERAL EQUIPMENT

The efficiency of any time-sharing program hinges on optimum use of peripheral equipment. In the typical system, MASTER uses the drum for system storage and temporary storage of user programs; and it uses disk storage for user files. Users do not have direct access to the card reader, card punch, or printer in systems having only one of each of these devices. MASTER reserves the card reader for preparation of input job files and reserves the card punch and printer for processing of output files.

Concurrently as jobs are being read from the card reader, they can also be read from other units such as magnetic tape designated by the operator as input units.

All 12-bit I/O data channels are pooled; each channel is assigned to a task with the highest priority as it becomes idle. In a system with one 24-bit data channel, this high-speed processor is reserved for the Operating System which uses it for communication with the drum.

## EXPANDABILITY

The system can be easily expanded to include new input/output equipment, Satellite® computers, additional compute modules, and core memory up to 262K.

## RELIABILITY

Continuous operation is essential when doing concurrent processing for a number of users. To attain this end, MASTER provides reasonable recovery procedures for hardware errors. When recovery is not complete, the system continues operation with a reduced configuration.

## EXECUTIVE MODE

The executive mode may be selected at the console when MASTER is initiated; it includes two states – monitor state and program state.

### Monitor State

With executive mode selected, a master clear results in setting the central processor to the monitor state. The processor also reverts to monitor state whenever interrupt occurs while operating in the program state. All 3300 and 3500 instructions are executable; only the System Executive in MASTER may operate in the monitor state.

### Program State

All user programs and the Operating System in MASTER are executed in the program state.

While in the program state, an attempt to execute certain instructions causes an executive interrupt to occur. These instructions include:

- halt and pause
- I/O operation initiation
- I/O status interrogation
- interrupt
- inter-register transfers that might alter the contents of register file locations 00 through 37

To handle the above operations, MASTER recognizes a complete set of I/O and special-purpose macro requests. These requests result in transfer of control to the System Executive through use of coded halt instructions. When the halt instruction is attempted, interrupt occurs and the monitor state of operation is established. The System Executive processes the interrupt and either performs the requested function or calls a task to perform the function, and selects a task to execute.

MASTER disables the interrupt system part of the time while in the monitor state. Before returning to the program state, the System Executive enables all or part of the interrupt system depending upon the task to be executed.

## RELOCATION

Relocation in the 3300 and 3500 Computer Systems contributes two vital features to MASTER: It makes it possible for the user to expand core memory storage to a maximum of 262,144 words, and provides protection of the many user and Operating System programs currently residing in core memory.

Core memory is divided into program address groups each of which incorporates 00000 to $77777_8$ addresses. Each group is divided into 16 pages of $4000_8$ words. Thus, for maximum possible storage there are 128 pages.

With the standard 3000 series 15-bit address only one group may be addressed; addressing all eight groups requires additional logic using a 3-bit program group number. Pages in a group can be addressed by using the upper 4 bits of the standard 15-bit address. Combining the 3 bits and the 4 bits produces a 7-bit page addressing scheme through which any of the 128 pages can be referenced.

For each program group, 16 page indexes maintained by the System Executive in a page file implement relocation addressing during execution of a task and permit or inhibit reading and writing in the indexed pages.

When an address is referenced, during execution of a task, the page number is read from the page file and appended to the remaining 11 bits of the program address to obtain the actual 18-bit memory address.

The smallest portion of a page allocatable to a task is 1/4 page (512 words).

## 1.2 SOFTWARE

Compatibility

MASTER input/output is file oriented. However, MASTER accepts with minor changes tape-oriented software utilizing CIO, the input/output routine of 3200 SCOPE. A conversion routine is loaded whenever programs reference CIO. This routine converts CIO calls to Mass Storage Input/Output (MSIO) calls and interprets file reference so that proper files are maintained.

A version of the 3000 Series Comprehensive Assembler (COMPASS), available with MASTER, includes all the instructions provided on the 3300 and 3500 computers.

MASTER accepts with minor changes object programs produced by FORTRAN, COBOL, and other special languages for the 3200 computer. The compilers, however, are modified from the 3200 version so that they will generate the correct calling sequences. All other standard 3200 software packages — including those with overlays — will run using the compatibility features of MASTER.

Library Preparation and Maintenance

A PRELIB facility in MASTER permits installations to correct, update, or replace library and system routines with a minimum of effort. Installation parameters permit arbitrary decisions that affect the nature of the time-sharing system so that a user may adapt a system to his needs. For example, parameters permit a user to alter the emphasis to be placed on multiple access jobs, the time discrimination between short and long jobs, and the estimates of core requirements.

# JOBS AND JOB FLOW 2

## 2.1 JOB SUBMISSION

Jobs are submitted to MASTER in three ways:

First, the user (or the operator acting for the user) submits jobs in a standard form to some input device such as a card reader or tape unit.

Second, the operator submits jobs with commands from the console typewriter. An operator-requested job is typically a request for execution of a background routine such as card-to-drum or drum-to-print. MASTER converts the operator request to a standard job form.

Third, MASTER may request a background job. For example, MASTER might initiate a routine that drives a standard device. Jobs may be accepted at any time from some device other than a standard input device. Both devices could be simultaneously providing input to the Operating System.

Regardless of how a job is submitted, it is eventually presented to MASTER as a job file. A job submitted from the operator or a running program usually already exists as a job file in the system library. In the case of standard user-submitted jobs, the input routine creates a job file and assigns a unique job name for each job. The input routine for the job passes to the Operating System all information available on control cards with the job.

The input routine makes calls to MSIO to create a job file. MSIO allocates, opens, writes, and closes the file.

### EMERGENCY JOBS

An emergency job may be submitted to MASTER at any time and will receive immediate attention. Other jobs will be suspended if necessary to permit immediate execution. MASTER accepts only one emergency job at a time. It may be submitted by an operator command from the typewriter, or may be triggered through a real-time interrupt.

## 2.2 JOB ORGANIZATION

Standard user jobs are presented in deck form. A job deck consists of control cards and any program decks and data cards required by the job. Every job begins with a job identification card (JOB) and ends with an end-of-job card. MASTER control cards in addition to these are the schedule control card (SCHED) and the file control card (FILE). These and any other cards to be included between the JOB and end-of-file cards depend on the requirements of the job.

### JOB CARD

This card contains the word JOB followed by the parameters for:

job charge number

job identification (name)

time limit in minutes after which the job is terminated (including input/output time)

line limit for the printer output

### SCHED CARD

This card contains the name SCHED followed by parameters giving:

time estimate

class

core estimate

peripheral equipment requirements

The time estimate for the job is the most likely running time; it identifies a long or short job to MASTER. In contrast, the time limit on the JOB card limits job execution time.

In the class field, the user estimates whether the job is input/output or compute bound. From the class and time fields MASTER determines the priority classification.

The user also estimates the number of 512-word blocks of core memory required by the job. When conscientiously supplied, core estimates aid efficiency by preventing the system from attempting to initiate a job that requires more storage than is currently available.

In the peripheral equipment fields, the user may specify non-standard peripheral requirements of a job; the information may occupy several fields. When no specifications are given, MASTER assumes that the job has no special requirements. Units such as the card reader, card punch, and printer are indicated only when the job is to control one or more of these units itself. Because most jobs will use files maintained by the system on mass storage devices, SCHED cards should contain little or no peripheral equipment information.

When a SCHED card is not used with a job, parameters internal to MASTER but determined at the installation are used for time, class, and core.

### FILE CARDS

In general, each installation has a standard directory maintained on a mass-storage device. Unless otherwise specified, MASTER assumes that all user input/output requests address files in the standard directory. If a file is to be read that is not in the standard directory, or written on a device such as a magnetic tape unit, it must be declared prior to execution. The declaration is in the form of a FILE card on which a file identifier is equated to a specified type of hardware. Subsequent FILE cards permit two identifiers to be equated. Non-standard input files must also be declared as input so that the Operating System will direct the operator to service the input device when MASTER initiates the job. Output files require only a device type declaration.

## 2.3 JOB FLOW

User jobs are read in from a card reader or magnetic tape by a background job that transfers the jobs to a job stack on some mass storage device, usually a disk. Jobs may be simultaneously presented from many sources. Additional input devices such as remote stations merely involve adding background routines to the Operating System.

Once a job has been transferred to the disk or drum, the input routine notifies MASTER of the existence of the job. MASTER determines whether or not the job requires emergency action, and if so, MASTER immediately begins monitoring the job.

## JOB INITIATION

In initiating a new job, MASTER considers such variables as job class, equipment requirements, wait time, and system activity. All required I/O devices, such as tapes, card readers, printers, must be available before a job is initiated (except for emergency jobs).

The required number of units is reserved for the job until it is completed. If the system does not have the type of unit specified, the job is terminated The card reader and card punch, and the printer will never be directly available to the user if there is only one of each in a system.

MASTER gives priority first to emergency jobs and then to all background jobs which are predominately I/O jobs requiring little compute module attention. Because they generally drive slow speed peripheral equipment such as card readers and printers, background jobs require high priority to keep the devices going full speed.

With the exception of background jobs, of which several may be concurrently active, MASTER attempts to keep active at least one job from each of the three remaining job classes — short, input/output, and compute. A second job from a class is initiated (when another of the same class is active) only if one of the other classes has no job active or capable of being initiated.

## SHORT JOBS

The short class provides fast turnaround time to short jobs. With this scheme, several short jobs are likely to pass through the system for every long one from the I/O or compute classes. Short jobs can be eliminated by setting the installation parameter for job time criteria to zero which causes all short jobs to be classed as either compute or I/O.

## JOB PRIORITY

Within a job class, priorities are assigned on a first-in-first-out basis. However, the first job in the list might not be initiated because its storage requirements (memory and I/O) are high and cannot be satisfied by the available storage. Thus, a job submitted later may be initiated first. However, no job is permitted to wait longer than a limiting time set as an installation parameter. In addition, the operator may request that any job already submitted be given top priority.

### JOB FILES

MASTER, through its associated background routines, maintains for each job submitted, an input file, an output file, and a punch file (if requested). As a job is submitted, the input routine opens the input file on the standard directory and copies the job from the input device (card or tape unit). The input file contains the entire job (or control cards to collect the files making up the job) following the JOB and SCHED cards.

After a job has been initiated, MASTER opens the input file for a job to be monitored and interprets control statements.

MASTER assures that logical file names are properly equated to physical files. For example, within a program a user may refer to his input file as INPUT. However, many jobs may concurrently refer to the logical name INPUT and the system must insure uniqueness of names. MASTER therefore checks for FILE control cards equating logical names to physical files. When a FILE card equates a standard unit logical name (INPUT, PUNCH, PRINT, etc.) to a physical file, MASTER makes the appropriate MSIO calls to accomplish this equating. When there is no control card for INPUT, however, MASTER calls MSIO to equate INPUT to the name of the physical job file.

When the job terminates, an output routine copies the information from the output file onto the system output device, typically a printer. If the job requests punching, MASTER opens a punch file on the standard device. At job termination, the output is punched by a background routine. The user may, however, suppress printing or punching through use of control cards.

## 2.4 TASKS

MASTER internally divides a job into one or more tasks of which there are two types – program tasks executed on compute modules, and I/O tasks executed on data channels.

As an example, consider a user job involving only compilation of an object program. Typically, the job to be compiled consists of one program task and several I/O tasks. The program task consists of the compilation. In general, each operation involving a data channel requires a separate I/O task; thus, each separate input or output operation involves an I/O task. A job to be compiled and executed involves two program tasks — one for compilation of the object program, and one for execution of the compiled job — in addition to several I/O tasks for compilation and execution.

Individual subroutines required by a job are loaded with a particular task within a job and require no intervention by the System Executive to be called and executed. Thus, they are not tasks regardless of whether they are included with the job or called from the library by control cards.

The process by MASTER of administering a job also involves several tasks such as scheduling, initiating, monitoring, and loading of jobs; and execution of I/O driver routines such as card-to-disk and disk-to-print. MASTER recognizes that certain tasks are ready for execution, assigns available processors to them on the basis of priorities, and maintains tables in which priorities and task status, and the relationship between tasks are recorded. The only distinction MASTER makes for its own tasks is the relaxation of certain restrictions governing the manner in which tasks request the execution of subordinate tasks, and in which subordinate tasks announce their completion.

## TASK LISTS

MASTER maintains lists of tasks to be executed on compute modules and tasks to be executed on I/O processors (data channels) and assigns tasks to processors on the basis of priority. An entry for a task contains all the information required by the Operating System for administration of the task. For program tasks, information includes task name, priority, physical storage allocation, kind of task (MASTER or user), contents of processor registers in event the task is interrupted, task status (one of several conditions registering its progress), linkages with other tasks, and accounting information (the accumulated execution time on each type of processor).

## TASK OBJECTS

For each type of task, program and I/O, the entity requiring allocation is the task object. For a program task, the task object is the program itself the instructions and the data areas. The required allocation is assignment of core storage. For an I/O task, the task object is a file, and the required allocation is assignment of the physical storage facility within some I/O device.

## DEFINED TASKS

In processing calls for tasks, the Operating System examines a list of defined tasks to determine if the requested task is permanent, temporary, copiable, or re-entrant.

When the task is not defined (MASTER does not recognize its name) the caller is notified that no connection can be made.

When the task is neither re-entrant nor copiable, memory is allocated for the task and only one call for the task may be connected at any instant. A task list entry is made for the first caller; additional callers are queued. Callers are recognized in the order in which calls are made unless a latter caller has a higher priority. When a task is defined as copiable, memory is allocated for the copy and a task list entry is made for each copy.

When the task is defined as re-entrant, a task list entry is made for each call. For a non-permanent, re-entrant task the first call requires memory allocation and establishment. An established task merely requires connection (and a new task list entry) for each call. Task execution proceeds independently for each caller. A re-entrant task is not self-modifiable; instructions in the task cannot be altered during execution of the task.

## TASK PROCESSING

After a task has been called, allocated, and its task list entry made, the System Executive transfers control to the task (in the program state) and begins execution. Execution of a program task continues until completion or until an interrupt occurs. Either returns control to the monitor state.

## INTERRUPTS

Interrupts may be voluntary or involuntary. A voluntary interrupt of a processor can occur only if the task in execution on that processor issues an executive request. This is accomplished by executing a halt instruction with the appropriate code bits. All other interrupts of the processor are involuntary. Two common involuntary interrupt conditions are: an external interrupt has occurred, signaling the completion of an I/O task; and an internal interrupt has occurred, indicating an arithmetic fault in some step of a program task.

Any interrupt occurring in the program state causes control to revert to the monitor state. For executive requests, the interrupted instruction is never executed; instead, the System Executive performs or initiates tasks to perform the request.

For involuntary interrupts, the interrupted instruction is not executed until MASTER returns control to the interrupted task. In the multiprogram environment, the point at which a particular task is resumed depends upon its priority with relation to all other active tasks. When control returns to the voluntarily interrupted task, it returns to the instruction following the executive request. For some executive requests, control will not return to the task until the function has been performed; for others, execution does not depend upon completion of the function and may continue immediately at the discretion of the System Executive.

## SYSTEM ACTIVITY

MASTER initiates jobs only when there are no tasks of higher priority to be done. The Operating System first seeks to eliminate any condition inhibiting current task processing. One such condition is the indefinite wait, the wait caused by a circularity of calls on non-re-entrant tasks. After detection of this condition, the Operating System:

generates a diagnostic message

generates a listable dump

terminates with an error indication of the task involved in the circular condition

Another condition which may inhibit job processing is the accumulation of a stack of requests for memory allocation. In this case, no job may proceed until more storage is allocated. A sufficient number of jobs are then suspended to provide enough storage for one of the jobs to proceed.

If job processing seems not to be inhibited, an attempt is made by the Operating System to initiate a new job or recall a suspended job. Suspended jobs receive priority over uninitiated jobs of the same class. However, before a job is initiated or recalled, the Operating System removes all completed jobs from the job lists.

# EXECUTIVE REQUESTS 3

When COMPASS, the Comprehensive Assembler for 3300/3500 MASTER, is used several executive requests in the form of macro instructions compensate for regular mnemonic instructions that generate machine instructions illegal in the program state. The assembler upon detecting a macro instruction generates a calling sequence for the executive in the form of a coded halt. During execution of the assembled program, the coded halt causes an executive interrupt; the System Executive processes the interrupt and performs — or calls a task to perform — work meaningful to the caller (the program using the macro instruction).

Most of the macros implicitly call tasks. That is, the System Executive automatically calls a task or set of tasks to perform the requested function. The caller knows only that the function will be performed without knowing the nature or variety of tasks involved in its performance.

MASTER also permits users to explicitly call tasks by using the CALL macro. Only those tasks that are known to the Operating System can be called. A user calls the task by name. Following the call are parameters that the called task requires. The caller passes parameters in the format and sequence in which the task expects them.

After a task is assembled, it is identified to the Operating System as a defined task, callable by name — usually by other COMPASS programs using the CALL statement.

In addition to MSIO macros for input/output, system macros include:

SELECT
REMOVE
TIME
DATE
CALL
BLOCK R
BLOCK W
STATUS
WAIT

## SELECT

SELECT sets a program interrupt so that upon detection of a specified fault during task execution, control transfers to a specified address. Parameters include the name of the fault (arithmetic overflow, divide, exponent overflow, BCD, search-move interrupt, illegal instruction interrupt), and the interrupt address.

REMOVE REMOVE deletes selection of a fault previously set by a SELECT. The only parameter is the name of the fault.

TIME TIME returns the time of day to the task making the request. TIME has no parameters.

DATE The DATE macro returns the date to the task making the request. DATE has no parameters.

CALL The CALL macro is a request for the System Executive to connect a call from the task making the CALL to the task named in the CALL. Parameters include the name of the callee, the status the caller is to assume after call connection (ready, wait, or terminate) whether or not the caller is going to abandon the call (not wait for call end), the number of words to be passed from the caller to the callee, whether or not the caller and callee are to share common blocks. The CALL may be followed by a BLOCK R or BLOCK W.

BLOCK R The BLOCK R macro requests the System Executive to pass to the callee named in the CALL macro the common blocks named with the BLOCK R macro as read only. The callee may not write in the named blocks. Parameters of BLOCK R include the number of common blocks named in the block statement, a flag for the System Executive telling whether or not the block statement is the last one associated with the CALL, and the names of the common blocks passed.

BLOCK W The BLOCK W macro requests the System Executive to pass the named blocks as locations that the callee can use for reading and writing. In all other respects it is the same as the BLOCK R statement.

RETURN The RETURN macro requests the System Executive to return control to the caller task; it is used only in callee tasks. The Operating System may remove task linkages in the task list and change the status of both caller and callee, according to parameters of the RETURN statement. RETURN parameters indicate whether or not transfer of call end is requested, and whether or not task termination is requested.

STATUS STATUS is a request for the System Executive to indicate to the requesting task the status of a particular task which the requesting task has previously called. The accompanying parameter is the name of the called task.

WAIT A task using a WAIT requests to be set to wait status by the System Executive and will resume execution upon fulfillment of conditions defined by the WAIT parameters.

# INPUT/OUTPUT CONTROL 4

## 4.1 I/O SATURATION LIMITS

With the advent of multiple access, time-shared software systems, the ability of a computer to perform many concurrent tasks is a measure of the hardware adaptability to a time-shared, multi-access environment. The actual number of permissible concurrent I/O tasks on a given machine is a function of many variables associated with logic design and hardware timing. For MASTER, an attempt has been made to analyze the I/O saturation limits of the 3300 as a function of:

The work load to be imposed upon the central memory from within and without the main processor.

The work load imposed upon the I/O control section (block control) by the main processor.

Maximum access rates to the I/O control section (block control) from peripheral devices.

Effective data rate of the peripheral unit under consideration; where the effective data rate is based upon the maximum delay in computer response allowable by the device before a lost data condition results.

To aid concurrent I/O processing, each unique device type is weighted against its maximum effective data rate. MASTER maintains a tally of all weighted devices currently active in the system. When I/O control receives a request from the Operating System to perform I/O on a device, I/O control, using the weight of the requested device, estimates if its inclusion in the system will exceed system saturation limits. If it exceeds the limits, the request is queued in the I/O task list with a ready status until activity reduces to a level allowing execution. Otherwise, transmission on the device is initiated.

## 4.2 I/O FILES

In general, two types of files exist in the MASTER system: user files, which must be declared and processed by the user; and system files, which are processed by input/output control in MASTER. Structurally, the files are identical for they are both maintained by MASTER Mass Storage Input/Output (MSIO).

All user input/output is prepared for input to or output from MASTER on mass storage devices of the following types:

- magnetic tapes
- disk files
- drums
- data cells

The device used for the standard directory is a feature of the installation.

# OPERATOR COMMUNICATION 5

## 5.1 OPERATOR-TO-SYSTEM

The operator initiates communication with MASTER by pressing the Manual Interrupt Button on the console. Further communication is through a peripheral device assigned for the purpose — usually a console typewriter. Operator-initiated messages may:

Inform the system of a change in equipment status

Request a background job

Modify a job parameter (priority, time limit, etc.)

Request the state of a job

Request termination, skipping, or suspension of job known to the Operating System

Inform MASTER of completion of a requested action (for example, a tape has been mounted as requested)

## 5.2 SYSTEM-TO-OPERATOR

System messages to the operator are initiated by calls from the Operating System. Communication is through a peripheral device, usually the same as that used for operator-initiated communications. System-initiated messages may:

Signify start and completion of jobs and issue reports on all active jobs

Command servicing of peripheral equipment (for example, instruct operator to mount tapes or place cards in reader)

Command remedial action to alleviate an unusual or abnormal condition such as a hardware or program failure

CONTROL DATA

CORPORATION

## COMMENT AND EVALUATION SHEET

3300/3500 COMPUTER SYSTEMS
MASTER General Information Manual

Pub. No. 60134100 October, 1965

YOUR EVALUATION OF THIS MANUAL WILL BE WELCOMED BY CONTROL DATA CORPORATION. ANY ERRORS, SUGGESTED ADDITIONS OR DELETIONS, OR GENERAL COMMENTS MAY BE MADE BELOW. PLEASE INCLUDE PAGE NUMBER REFERENCE.

FROM NAME: ____________________

BUSINESS
ADDRESS: ____________________

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

**CONTROL DATA CORPORATION**
*Documentation Department*
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

FOLD FOLD

STAPLE STAPLE

3300/3500 MASTER GENERAL INFORMATION MANUAL

**CONTROL DATA SALES OFFICES**

ALAMAGORDO • ALBUQUERQUE • ATLANTA • AUSTIN, TEXAS • BILLINGS BIRMINGHAM • BOSTON • BOULDER, COLORADO • CAPE CANAVERAL CHICAGO • CINCINNATI • CLEVELAND • COLORADO SPRINGS • DALLAS DAYTON • DENVER • DETROIT • DOWNEY, CALIFORNIA • GREENSBORO, NORTH CAROLINA • HARTLAND, WISCONSIN • HONOLULU • HOUSTON HUNTSVILLE • MIAMI • MONTEREY, CALIFORNIA • INDIANAPOLIS • KANSAS CITY, KANSAS • LOS ANGELES • LAS VEGAS • MADISON, WISCONSIN MINNEAPOLIS • NEWARK • NEW ORLEANS • NEW YORK CITY • OAKLAND OMAHA • PALO ALTO • PHILADELPHIA • PHOENIX • PITTSBURGH ROCHESTER, NEW YORK • SACRAMENTO • SALT LAKE CITY • SAN BERNARDINO • SAN DIEGO • CONDADO, PUERTO RICO • SANTA BARBARA SAN FRANCISCO • SEATTLE • ST. LOUIS • TULSA • WASHINGTON, D. C.

ADELAIDE • AMSTERDAM • ATHENS • BOMBAY • CANBERRA • DUSSELDORF • FRANKFURT • HAMBURG • JOHANNESBURG • LONDON • LUCERNE • MELBOURNE • MEXICO CITY • MILAN • MONTREAL • MUNICH OSLO • OTTAWA • PARIS • ROME • STOCKHOLM • STUTTGART • SYDNEY TEHERAN • TEL AVIV • TOKYO (C. ITOH ELECTRONIC COMPUTING SERVICE CO., LTD.) • TORONTO • ZURICH

8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440

CONTROL DATA
CORPORATION

Pub. No. 60134100

Litho in U.S.A.