

**SYSTEM CENTURY®
DIGITAL CENTRAL OFFICE****SOFTWARE ORGANIZATION AND FUNCTIONAL DESCRIPTION****CONTENTS**

Par.		Page	1.02	This section is reissued to describe and illustrate changes made to the software since release of issue 2. Significant changes and additions are indicated by bars in the margins.
1.	Introduction	1		
2.	Related Information	1		
2.01	General	1		
2.02	Related Publications	1	2.	RELATED INFORMATION
3.	General Software Organization	1	2.01	General.
3.01	Overview	1		A Sales and Instructional Literature Index of latest available publications can be obtained from your Stromberg-Carlson sales representative or from Publications Services, Stromberg-Carlson Corporation, P.O. Box 1698, Sanford, Florida 32771.
3.02	System Characteristics	2		
3.03	System Initialization	2		
4.	Call-Processing Software	3	2.02	Related Publications.
4.01	Overview	3		The following publications are recommended for use with this document:
4.02	CP Characteristics	4		
4.03	Cluster Arrangements	4		
5.	Maintenance and Administration Software	8		
5.01	Overview	8		
5.02	M&A Characteristics	8	01-025-05	Glossary of Digital Switching Terms and Abbreviations
5.03	Administrative Tasks	17		

ILLUSTRATIONS23-001-02 **SYSTEM CENTURY DCO,
General Description**

Fig.		Page	3.	GENERAL SOFTWARE ORGANIZATION
1.	Software Size	2	3.01	Overview.
2.	Call-Processing Control Loop	3		The SYSTEM CENTURY DCO software organization is subdivided into two groups: the CP software and the M&A software. The CP software controls the processing of a call and has the traditional translator/marker functions assigned to it, such as doing translations, selecting terminations, and marking or unmarking timeslot interchange paths. The M&A software controls the communications between hardware-redundant CPs; supervises and controls all input/output (I/O) to the mass-storage device (MSD) where copies of the latest system tables, diagnostics and administrative programs, and CP programs (for startup and recovery) are stored; controls the automatic generation of test calls (routine testing); and monitors system operation by checking a variety of parameters. Fault indications are relayed to the maintenance processor.
3.	Call-Processing System – Cluster Organization	5		
4.	Maintenance and Administration Software Organization	9		
5.	Maintenance and Administration Operating System	10		
6.	Directory Service Order Overlay	14		
1.	INTRODUCTION			
1.01	This section describes the software associated with a SYSTEM CENTURY Digital Central Office (DCO). It is organized in ascending order of detail (simple-to-complex) and subdivided into two major categories. Paragraph 4 describes the call-processing (CP) software and paragraph 5 describes the maintenance and administration (M&A) software.			

3.02 System Characteristics.

Although CP software and M&A software are distinctive in many ways, there is a degree of commonality. Both are structured on a cluster/module concept. The number of modules within a cluster group varies with the particular job. The cluster/module concept is arranged hierarchically (tree-structured) such that, in the calling scheme, a general ruling stipulates that a specific module can only be called by a module in a higher level cluster. Modules of the same cluster (internal calling) communicate, when required. Both the CP and M&A systems contain an executive cluster responsible for determining the next task to be processed. The CP executive cluster is considered an event scheduler, whereas the M&A executive cluster is a task scheduler. The M&A system depends functionally on an interrupt handler (IH) system. In this IH system, the executive cluster contains an interrupt vector table and an executive module that together enable the cluster to call or reactivate processing programs after the completion of interrupt service. An interrupt is caused by an event in an external I/O device and, when this interrupt is activated, causes a transfer of control to a specific location in memory, thereby breaking the normal flow of the routine in progress. The size of the CP and M&A software is shown in figure 1.

3.03 System Initialization.

a. Initialization, or startup, is the process of making the system operational (user ready to operate) from an inoperative state, such as after installing the DCO. Startup includes the loading of all programs

and permanent data bases (stored on an MSD) for the MP and the two CPs and placing the system into synchronized, redundant operation.

b. When power is applied, each of the CPs enters a wait condition. This condition prevents random executions while the memory is being loaded. The CPs remain in this condition until the MP causes them to start program execution at a set location.

c. A bootstrap loader program, which resides in a read-only memory (ROM), is executed by a switch on the MP central processing printed wiring board assembly. When the bootstrap executes, it causes an MP startup program to be read into the MP memory from the MSD. After the read operation, program control is automatically given to the startup program. This program causes the loading of all MP-resident software (from MSD) and the initialization of the MP and its peripherals. The MP startup process is then considered complete. To get either CP working, the call processor startup program (refer to Note at end of this paragraph) is manually requested and run. This program transfers the CP data base and a CP initialization program from the MSD to the CP memory. The initialization program initializes the port control store, timeslot interchange stores, and any other hardware memories associated with that CP. When the MP gets initialization-finished signals from the CP, it transfers the CP generic program from the MSD to the CP memory. In order to load and synchronize the other CP, a recovery program is manually requested and run.

	CP	M&A
MAIN MEMORY (K WORDS)	256	64
PER-PORT MEMORY (WORDS)	10	-
AUXILIARY MEMORY (K WORDS)	-	128 AVAILABLE
INSTRUCTIONS (K)	≈ 40	≈ 35 AUXILIARY MEMORY
MODULES	≈ 812	≈ 195

CP 5964A

Note. Auxiliary memory = floppy disk

Figure 1. Software Size

NOTE. These programs, like many other time-independent critical programs, are not permanently resident in the MP memory but are stored on the MSD. They are read into a common, reserved section of the MP memory, called the overlay area, when required. This greatly reduces the MP memory size requirement but also prevents more than one nonresident program from executing simultaneously.

4. CALL-PROCESSING SOFTWARE

4.01 Overview.

a. The call-processing software responds to requests from the telephony preprocessor. In turn, these requests are initiated by external events (for example, subscriber originations, digits received, releases, etc.). The response to the CP software is subjected to a priority determination that is implemented by three hardware event queues, corresponding to three priority levels. The requests by the telephony preprocessor for CP software responses are entered in the queues. The CP runs in an interrogation loop, constantly polling these queues, in order of priority, looking for a request response. This control loop is shown in figure 2.

b. The hardware event priority queues store (in first-in, first-out (FIFO) memory chips) the equipment numbers (ENs) of those ports to which the request for response relates (in approximate chronological order) or, in other words, those ports having active event codes. When the CP finds a queue entry, it interrogates the port store entry (PSE) of this EN. This PSE contains the status information for that port, which includes digits received, supervision bits, call state, and other related information concerning the call in progress on that port. There is one port store area (PSA) for each port in the system.

c. Each request for response in queue is handled as an independent call-state transition. Each request is processed to completion (that is, the call is advanced to the next state) before the next request is

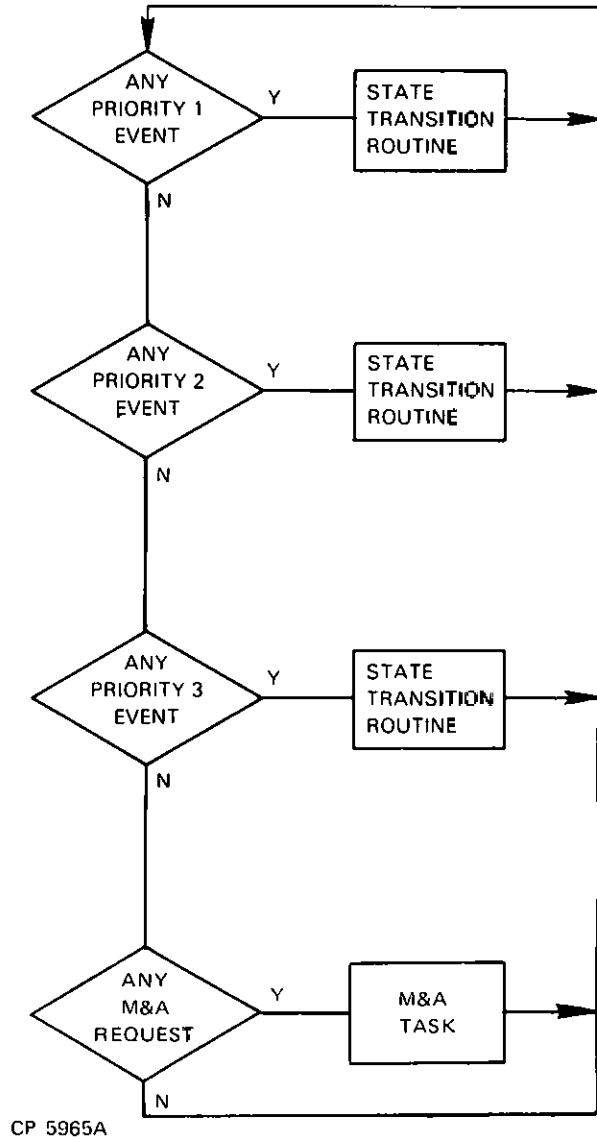


Figure 2. Call-Processing Control Loop

served. On completion of each state transition, the PSA assigned to each port is updated to show the new state of the call.

d. All transient call data associated with calls in progress is maintained in the PSAs of the associated ports. Calls in stable states present no load to the processor. Processing is invoked on a request basis by the telephony preprocessor when it detects a reportable external event.

4.02 CP Characteristics.

a. Program Redundancy.

During normal operation, an identical copy of the CP generic program resides in the memory of each call processor system (CPS). This program duality (mirror image), operating in a parallel synchronous environment, provides high system reliability and relieves software of excessive cross-writing. To control this program redundancy, the M&A system contains switchover logic that selects one CPS as the on-line system and gates its associated outputs to the external logic (common control, timeslot interchange, etc.). All installations contain identical programs so that, where economically feasible, conditional assembly can be used to make the program library efficiently match a specific installation.

b. Data Base.

Every installation has a unique data base, containing specific telephone company information. This data base contains specific information, such as directory number tables, routing information, COS table, etc. The data base size depends on the number of lines, the number of directory number blocks, the mix of single-party and multiparty lines and trunks, the position of the office within its local EAS environment (routing tables affected), etc. Data base operations are implemented by the data base utilities cluster.

c. Software Updating.

Field updating of the CPS is done by providing a new program/data base package, plus any subsequent requirement for additional random access memory. The program package and its corresponding data reside in this memory for normal operation of the CPS.

4.03 Cluster Arrangements.

a. Cluster Organization.

The call-processing clusters are grouped by levels (figure 3). The highest level contains the executive and the interrupt-handler clusters; they have upper management responsibilities in terms of the system operation. The second level is called the state transition level. This group of clusters causes the calls to advance from one state to the next. Examples of second-level clusters are: organization and dial tone, receiving digits, line-to-line, line-to-trunk, incoming

trunk, tandem, busy verification, and maintenance buffer handler. The third level supports the state transition level and is called the common sub-transition level. Clusters within this group are: purge, equipment connect, equipment release, translations, administrative change, and startup and sanity. The fourth-level group, called the utilities level, directly accesses external devices or system data bases and includes port store utilities, network utilities, and data base utilities clusters; the traffic measurement facility is optional.

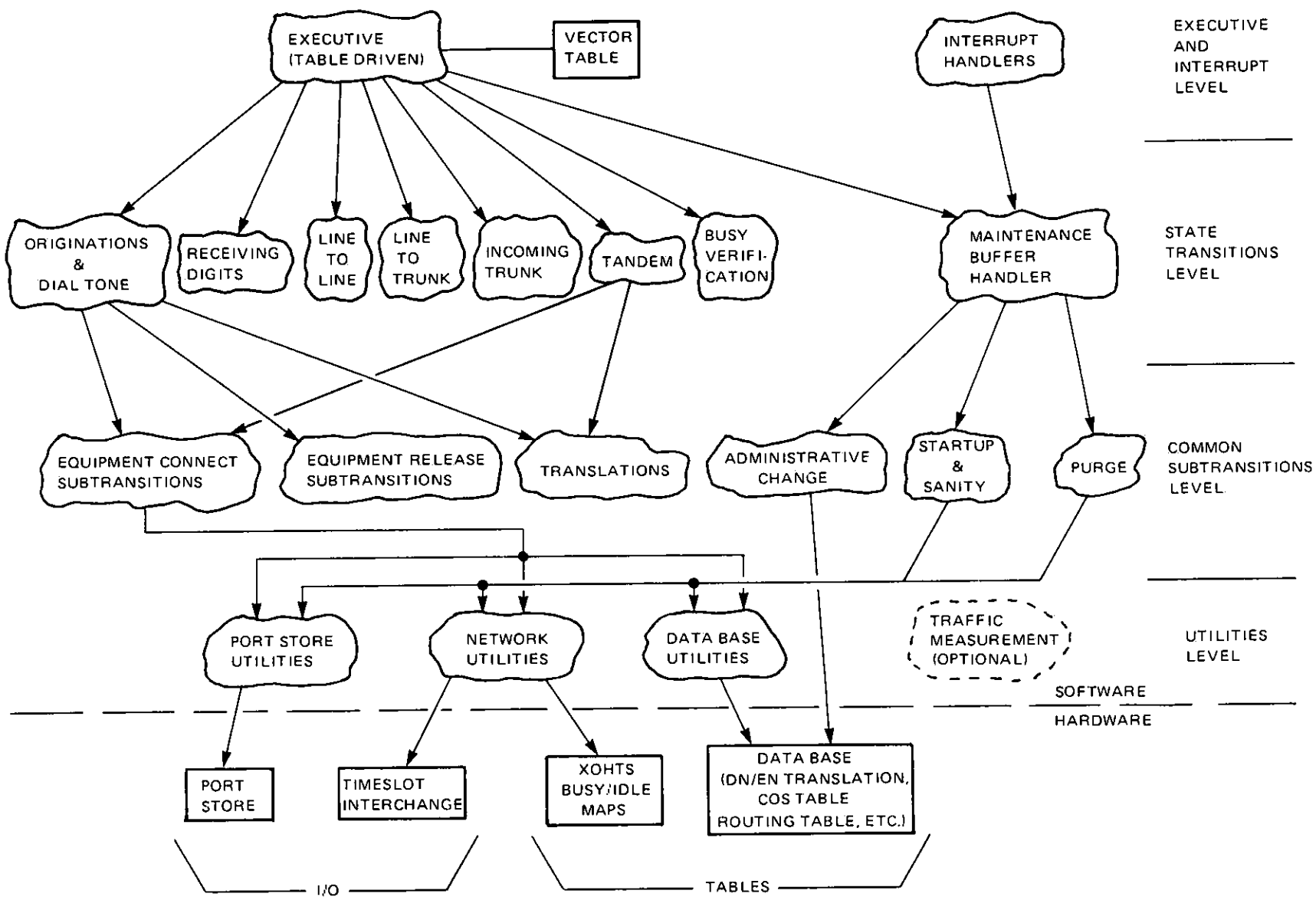
b. First-Level Clusters.

1. The executive cluster determines the next event

to be processed, sets up the environment for the transition routine, and then gives control to this routine. This cluster consists of five modules: SCAN, TRAN, GOTRAN, RCHAND, and CKRET. The SCAN module scans the port store for an event to be processed. When an equipment number requiring service is found, the SCAN module transfers control to the TRAN module to handle the event. The TRAN module is the main module for the processing of transition-routine scheduling. This module handles interfacing of the busy-verification cluster. GOTRAN does the address calculation of the transition routine. RCHAND handles the processing of all return codes given back to modules SCAN, TRAN, and GOTRAN. CKRET is used by SCAN, TRAN, and GOTRAN to check return codes from subroutines called by them. If necessary, CKRET calls the RCHAND module. The only table used by the executive cluster is the transition-routine vector table. The executive cluster calls all transition-routine clusters but is not called by any other routine.

2. The interrupt-handler cluster processes inter-

rupts to the call-processing software. It handles interrupts generated by the CP itself and external interrupts from either the system hardware or the maintenance processor. This cluster also handles traps. The interrupts/traps handled are: call processor error traps (odd address, bus timeout, stack overflow, and memory parity); illegal instruction; input/output; power fail/restore; emulator; user trap; external event line; sanity check; and MP interrupts (message processing, synchronization, and startup).



CP 5362B

Figure 3. Call-Processing System - Cluster Organization

c. Second-Level Clusters (State Transition).

1. The origination and dial-tone transitions cluster reacts to seizure events, finds and connects the required tone ports and receiving devices, and proceeds to the call state necessary for digit collection. The modules of this cluster are designated by supervisory events (for example, idle to interdigit timeout, equipment busy tone to timeout/release, dial tone - DP to digits received, busy tone to timeout/release, etc.) that cause module processing, which updates the state of the call for the next event.

2. The receiving-digits (RD) cluster handles the digit-collection call responsibility from local subscriber and interoffice trunk calls. Of the modules that make up the RD cluster, one is for each of the three permissible events from each of the four input states (dial-pulse dialing, DTMF dialing, receiving dial pulse, and receiving multi-frequency), plus three for the bulk of the digits-received event that is common to all four states.

3. The line-to-line (L-L) cluster handles the progress of a call from its transition, starting in the ringing state to release, including the features of revertive calls. The modules in this cluster include a separate functional routine for each event occurring in the L-L section of the call progress charts. Examples of these events are: ringing, ring trip, talking L-L, ring halt, circle digit DP or TD, await release, tick tone, etc.

4. The line-to-trunk (L-T) cluster contains modules necessary to complete all types of L-T calls (sending DP or MF) after they have progressed through origination and dial-tone and dialing cluster states and have been determined as L-T type calls. Examples of the transitions implemented by a particular event are: sending DP, seize out MF, await answer, talking L-T, CAMA sending, sending MF, timed disconnect, guard, await CAMA answer, trunk lockout, etc.

5. The incoming trunk transitions cluster contains modules that handle the processing of incoming trunk events (release, timeout, etc.). Each module

represents events detected while in the incoming-trunk state (busy tone, equipment verifying, recorded announcement ringing, talking, hold, etc.). When an event is detected, the appropriate module is vector selected by the executive cluster. The modules update the state of the call for a new event.

6. The tandem cluster processes the events that occur during trunk-to-trunk signaling and conversation. Modules in this group include: tandem send (DP), tandem out (MF), tandem send (MF), await-answer tandem, talking tandem, etc.

7. The busy verification cluster inserts and removes the operator busy verification trunk and the conference circuit to/from the current network configuration of the line being verified. Modules in this group include: insert/remove verification, verification trace/hunt, etc.

8. The maintenance buffer handler controls the I/O buffers used to interface with the MP. It decodes messages from the MP, calling on the administration and startup/sanity clusters to do specific functions, and maintains a queue of messages for the MP. Modules in this group relate to message buffer operations, administrative queues, etc.

d. Third-Level Clusters (Common Subtransitions).

1. The purge cluster purges (clears) equipment numbers (ENs) generated by the modules when an error is detected. It also purges ENs generated by itself. This cluster contains one module. It is called by the executive cluster after return from a transition module, if a purge condition code is returned by the module. The purge cluster does the following:

- a) Erases all switch card store paths associated with the EN to be purged.
- b) Idles all network paths selected and associated with the EN to be purged.
- c) Changes the EN port store area to idle.

2. The equipment-connect cluster contains those modules required to connect one terminal to another. These modules include all-path hunting and path marking. Each module represents a unique connection to be made and, given the appropriate inputs, attempts to establish the designated connection. A task status (done/not done) indication is returned to the calling module after each module-level task is tried.

3. The equipment-release (ER) cluster contains those modules required to release one terminal from another. These modules handle timeslot interchange path unmarking and port store updates for the terminals involved. Each module represents a unique release to be made. Given the appropriate inputs, each module releases the designated connection. All error indications are relayed back to the transitions cluster for trouble reporting. Modules in this ER cluster are: MF outpulsing for release tone, ring, sender, receiver, connection, and verification; disconnect EN; halt ringing; guard trunk; receiver-sender idle for line, trunk, and EN; release verification; line and trunk lockout; and disable through supervision.

4. The translations routines enable a fixed generic program to operate in various environments by providing a buffer between the CP functions and installation-dependent parameters. The code point and number translators interpret incoming digits to determine proper disposition of service requests. This cluster has three modules called the code point translator (CXLTR), the directory number translator (DNTON), and the equipment number translator (ENTODN). From an I/O viewpoint, the CXLTR is given a string of digits and produces a route treatment index. The DNTON module (local calls only) is presented with a DN and outputs a corresponding EN and ring code. The ENTODN handles the reverse of this.

5. The administrative-change cluster indirectly interfaces with the M&A system by way of the maintenance buffer handler cluster. In the M&A system, directory service orders are inputted to the MP by an operator typing messages at the console. The MP interprets these messages, which are passed across the data link by way of the regular

maintenance buffer handler protocol. The administrative change routines process these messages, interrogating or updating the appropriate data base tables as directed by the message, and generate appropriate responses to the MP by way of the data link. This cluster is one of the largest in the CPS.

6. The startup and sanity test cluster contains the following modules: sanity check, startup (STRTUP), switch erase (SWERAS), initialize port store area (INPSA), and clear queues (CLRQUE). The stand-alone sanity check module provides a quick diagnostic look at the CP computer, testing the instruction set and device controllers. A report of the test results is sent to the MP. The other modules (STRTUP, SWERAS, INPSA, CLRQUE) idle all ports and paths and clear all tables related to dynamic assignments of paths. All port store areas are set up by deriving the necessary information from the CP data base.

e. Fourth-Level Clusters (Utilities).

1. The port store utilities cluster allows general program accesses and updates of port store information. Each functional component (macro) of this cluster does a set of one or more retrievals or updates on a specified PSA. Each set of accesses/updates operates on a coherent group of fields within the PSA, corresponding to a given EN. The cluster data base consists of constants defining the PSA configuration. These constants are listed, by way of direct assignments, in macro DAPSU. This listing is part of the call-processing generic program library.

2. The network utilities cluster interfaces with the controller. This cluster is responsible for finding, marking, tracing, changing, and erasing network paths. The modules in this cluster do specific timeslot interchange operations. The basic operations required in setting up a conversation path consist of finding a path timeslot and marking the path. The basic operations required in releasing a conversation path consist of unmarking the path and idling the path timeslot. The data base of this cluster consists of software tables relating to COS, DN, and EN.

3. The data base utilities cluster provides access to the CPS software data base for the other clusters. This cluster interfaces with any module that must access the CP data base.

5. MAINTENANCE AND ADMINISTRATION SOFTWARE

5.01 Overview.

a. The SYSTEM CENTURY DCO system configuration uses three processors. Two (redundant) are dedicated to the call-processing tasks, and the third processor serves the maintenance and administrative functions. This third, or maintenance processor (MP), besides having other DCO system responsibilities, provides the interface for telephone company personnel (or other users) and automatically controls the operation of test access relays in port circuits for the purpose of placing test calls. The MP is the central processing unit (CPU) for the M&A operating system (O/S). Generally, an O/S provides access to the features of the CPU and also organizes the CPU and peripherals (TTY, floppy disk, etc.) into a useful tool for dedicated applications. The M&A software is organized by programs and is shown in figure 4.

b. The normal state of the DCO system is when the redundant components (CPs, timeslot interchange, port control system, etc.) are all operating in synchronism. In this state, call processing is carried out as usual, with the MP executing routine tests and administrative changes in a multiprogram mode. When a fault is detected, generally by means of an interrupt to the MP, the error handler causes a message to be printed, identifying the interrupting source together with any associated data. The programs currently in progress at the MP are suspended, and first-level diagnostics (memory-resident routines) are initiated. First-level diagnostics use the following to determine if switchover is required: detector identity, which is determined by parity checks; watchdog timer, or other timeouts (DMA, SYNCH), when a faulted side is immediately known; test calls, which compare linkages (CP not disturbed); and sanity tests, which consist of checksum tests and read/write checks. As a result of first-level diagnostics, one of the following actions (along with the printout) is taken:

1. If the cause of the fault cannot readily be determined, as in the case of a nonrecurring transient or insufficient information, the system returns to normal operation.

2. If a software fault occurs, the system returns to normal by way of a reset or restart.

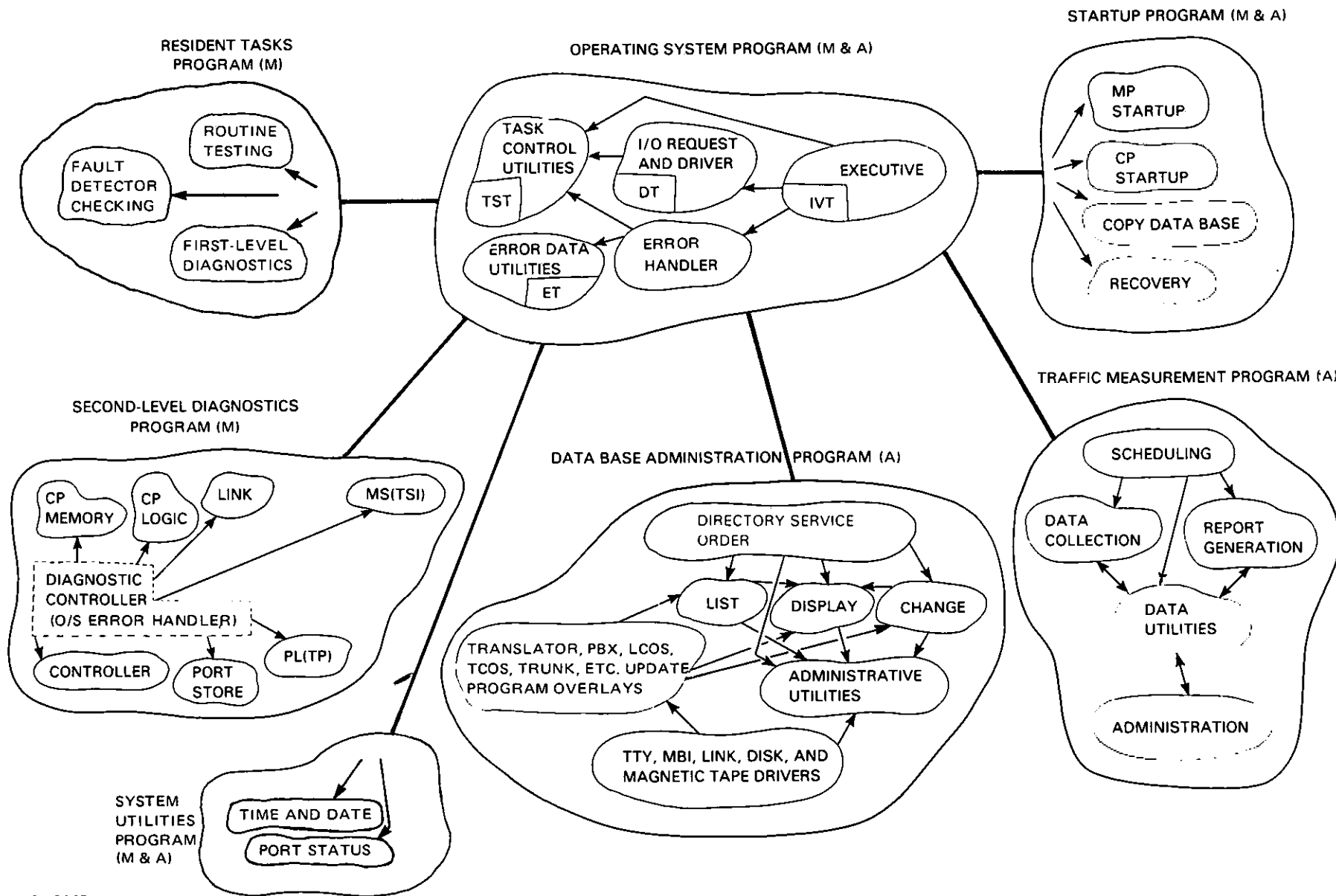
3. If a faulted side has been determined, the system goes into a single-side fault state, by way of the switchover logic, removing the faulted side from service and maintaining the fault-free side on-line. This causes appropriate second-level diagnostic scheduling.

4. If both sides fault, or if a nonredundant component faults, the system goes into a degraded state, since call processing is affected. Second-level diagnostics are now scheduled. These diagnostics run and isolate the fault to a specific PWBA card. If a specific card is already indicated, then the diagnostics verify the faulty board. A message is printed, stating the outcome of these tests. At the conclusion of second-level tests, the proper masks are set, and the dual programming mode can be reestablished. If the fault cannot be verified, then a recovery is initiated, and the system attempts to return to normal operation. This allows for transient errors that can cause fault detector operation. Second-level diagnostics can be used by the operator to further isolate repetitive faults existing in both sides.

5.02 M&A Characteristics.

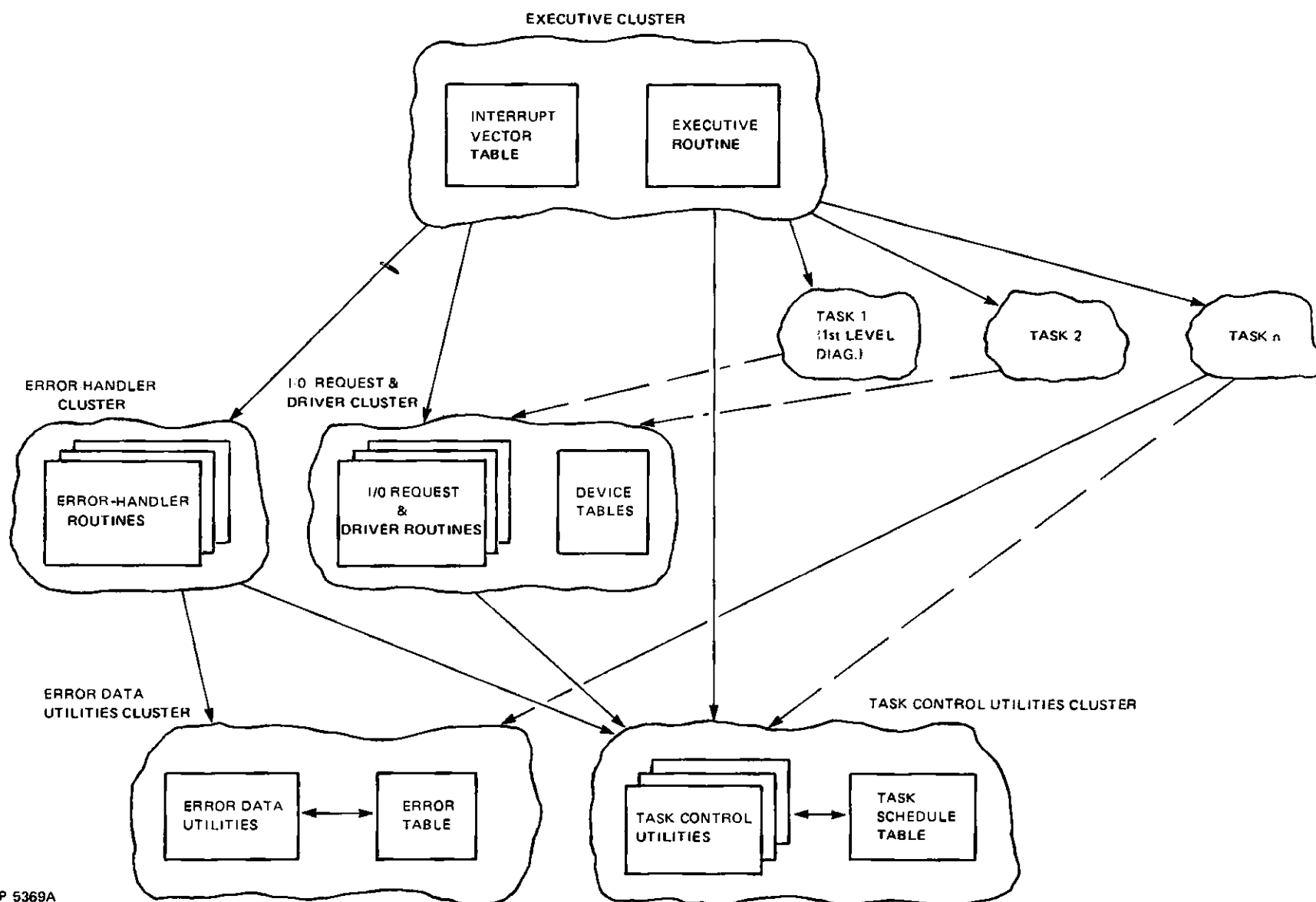
a. Operating System Program (Fig. 5).

1. The operating system (O/S) of the M&A software is a collection of related clusters that controls the scheduling and execution of all processing tasks (diagnostics, routine testing, recovery, startup, and administrative functions) of the maintenance, diagnostic, and administrative program functions of the SYSTEM CENTURY DCO. It also supervises and coordinates all input/output operations and responds to error conditions. The O/S cluster group includes: the executive cluster, the task control utilities cluster, the error-handler cluster, the input/output cluster, and the error data utilities cluster.



CP 5966B

Figure 4. Maintenance and Administration Software Organization



CP 5369A

Figure 5. Maintenance and Administration Operating System

2. The executive cluster schedules and dispatches processing tasks. This control over tasks is directed by the task schedule table (TST). This cluster consists of an interrupt vector table (IVT) and two modules. The IVT contains a procedure pointer and a processor status word for each interrupt class. The occurrence and acknowledgement of an interrupt results in automatic access to the procedure pointer/status word pair for the class of interrupt, and in the automatic (hardware-controlled) dispatching to the procedure indicated by the pointer. The two modules in this cluster are the executive and the keyboard monitor. The executive module controls the dispatching of processing tasks (diagnostics, routine testing, etc.), following completion of interrupt processing. The algorithm of this module is controlled by the contents of the TST. The keyboard monitor module accepts commands entered at the console and requests the scheduling of the required task(s).

3. The task control utilities cluster contains routines for accessing and updating the task schedule table (TST). The contents of the TST determine what programs are run and how (dual-program mode, single-program mode, time delays, privileged mode, etc.) routines of this cluster can be called by processing tasks and the O/S. This allows tasks to cause the scheduling or activation of other tasks and the loading into the overlay area. This cluster contains several tables (TST, control word, initial-entry point, time word, stack pointer, and priority pointer) and various modules. Examples of associated modules are: trap handler, save (for example, CPU) registers, scheduler, descheduler, time-delayed scheduler, enter privileged state, terminate privileged state, lockout task, unlock task, etc.

4. Modules within the error-handler cluster are invoked in response to message interrupts generated by the maintenance bus interface or the CP/MP link. They are also invoked in response to interrupts generated by system errors. This cluster

is a program control system that responds to all error detection signals of the diagnostic hardware and software of the SYSTEM CENTURY DCO. This cluster records a sensed error in the error-category vector (ECV) and attempts to activate appropriate fault isolation software after sequentially comparing the ECV contents with an error dictionary.

5. The I/O request and driver cluster transfers I/O requests by processing tasks to the O/S by way of the I/O trap instruction. This instruction results in a program-generated interrupt that is vectored, or steered, to the appropriate I/O routine of this cluster. The processing task that requested service is then suspended from executing until service completion. This suspension can lead to the activation of a separate task. All I/O operations requested by a task are done under interrupt control. The routines of this cluster always return control to the executive cluster.

6. The error-data utilities consist of routines to manipulate (set and reset bits in) the error-category vector (ECV) and to compare the ECV with the contents of the corresponding error dictionary (ED). The ECV is a collection of contiguous 16-bit words, where each bit corresponds to a specific class/category of potential errors/faults of the DCO system. The setting of this bit is the responsibility of the error-handler routine associated with that class of faults. The ECV manipulation routines of the error-data utilities provide the handlers with convenient means to set/reset arbitrary bits of the ECV. The ED consists of a fault table and a corresponding diagnostic action (DA) table. The automatic processing required to correlate fault symptoms includes the searching of the fault table for an entry that matches the ECV. When a match is found, indicated by the corresponding DA table entry, processing is initiated. This cluster contains a utility routine to facilitate the searching of the ED.

b. Resident Tasks Program.

1. In figure 4, routine testing exists as a task controlled by the executive cluster. Routine testing, in turn, controls the test call generator and CP sanity test subtasks. The purpose of routine testing is to continuously exercise and test the system. To test the system, the test access relays and the test call generator are normally used. These allow the MP to automatically place test calls between any two ports in the system, providing they are in different port groups. This testing function also periodically invokes (calls) sanity tests of the CPs and periodically initiates switchover. The routine testing is only done when the system is operating in a normal, nondegraded environment (both sides are operational and synchronized). Under these conditions, the testing function operates continuously in the MP background program mode. Sanity tests involve a program sequence, located in the CP memory, consisting of a processor instruction set. The result obtained by executing this sequence is compared with a hardware-stored, "correct" result. If this result doesn't occur (disagreement, nonresponsive processor) within a prescribed time, appropriate error processing is initiated. Other functions routinely checked are: detector operation, programmable logic operation, and CP checksums.

2. The execution of first-level diagnostics is controlled by a supervisory program called "first-level supervisor." This supervisor analyzes the different mismatches and initiates the proper diagnostic routine to check a suspected unit of the common control. After one or more diagnostic programs are run, the faulty side is identified. First-level diagnostics are initiated only when alarms (mismatches) exist between the two sides of the common control, and there is a lack of any side-dependent alarms (which normally cause automatic switchover). The first-level supervisor, in turn, is controlled by a program called "diagnostic controller." This controller transfers control to either the first-level diagnostics (FLD) or second-level diagnostics (SLD), checks for the alarm source, logs the different alarms in appropriate tables, and determines if any alarms are in the common control. If any alarms exist, then diagnostic control goes to FLD, otherwise

control goes to SLD. First-level diagnostics terminate after one of the following conditions: a decision is met concerning the faulty side, and the good side is switched-in (system goes to single-side fault); automatic switchover occurs (system goes to single-side fault); an alarm condition disappears (transient fault), and the system is restored to the synchronous mode (system "normal" state reaffirmed); or the faulty side cannot be identified (system "normal" state reaffirmed).

3. The fault detectors checking cluster is used to test the operation of various fault detection networks located throughout the DCO system (for example, common control, input/output, peripherals, etc.). This cluster can be executed on a routine basis (every 24 hours), upon manual request, and by automatic, diagnostic programming.

c. Startup Program.

This program has four clusters called: MP startup, CP startup, CP copy data base, and recovery. These clusters are described as follows:

1. MP Startup (MPSU).

The MPSU cluster bootstraps (reads and loads itself) the MP, loads all of its memory, initializes software areas and hardware, and starts the M&A executive cluster. The bootstrap loader enters the MP and begins execution of a loader. The loader then loads the MP system memory load (includes the MP startup (MPSU) and the MP startup teletypewriter handler (MPSUTH) modules) and passes control to the MPSU module. The MPSU module causes printing of the system identification (ID), which identifies the system stored on the MSD. Then it prints "Is This Correct? (Y/N)" and waits until either a character is inputted from the console or until 10 seconds have elapsed. If a Y(es) character was entered, then the MPSU continues; otherwise, the message "Wrong System--Halt" is printed and the MPSU terminates. If no character is entered, the MPSU continues after 10 seconds. The MPSU initializes the hardware and terminates with an exit to the M&A executive cluster.

2. CP Startup.

The CP startup cluster runs and resides, as an overlay, in the MP. Its function is to load,

initialize, and start one or both of the CPs. This can be done either by command or automatically, as required. This cluster is entered through the executive cluster and returns to the executive cluster on completion. The CP startup cluster transfers the CP data base from the MSD, through the CP/MP data link, to the CPs. It then transfers a CP initialization program from the MSD to the program area of the CP, starts program execution, and waits for a finished signal from the CPs. The initialization program extracts information from the data base and uses it to initialize the timeslot interchange and port stores. When the CP startup cluster gets the finished signals, it transfers the CP programs from the MSD to the CPs, starts executing CP programs and, if both CPs are being started, initiates synchronization.

3. CP Copy Data Base.

The CP copy data base cluster transfers the CP data base from either the on-line or the off-line CP to replace the old version of the data base stored on the MSD. This cluster runs as an overlay in the MP. Normal operation involves replacement of the old version by entering \$COPY ONL DB. Status information concerning the contents of the data base is stored in an MSD-resident data file, called the VDFILE. This file was created to prevent the loss of status information during an MP reload. To preserve integrity under any sequence of circumstances, the VDFILE is updated before the copy function, and after, to reflect the status of the storage files. The actual copy function is done a block at a time, where a block is the size of the largest space available in the overlay area. This area is a multiple of the number of words in a disk (MSD) sector. The data is first read from the data link into MP memory, then written to the MSD. When the copy function is done, the message "COPY END EXECUTION" is printed and the overlay area is released.

4. Recovery.

The recovery process consists of loading the inactive CP and all its associated memories and stores with a copy of all programs and data, including time-perishable data, as found on the active CP. The function is implemented as an overlay program for the MP; the program is able to directly access (without CP aid) the timeslot

links. The recovery process involves: copying of CP programs and its static data base; copying of the dynamic CP data base; copying of information in the timeslot interchange stores; copying of the port control store information; and clearing and resetting of the port control request queues.

d. Second-Level Diagnostics Program.

This program includes the following diagnostic clusters: CP memory, CP logic, link, controller, matrix switch (timeslot interchange) memory, port store, and programable logic (telephony preprocessor).

1. Controller Diagnostics Cluster.

This cluster of modules is resident on the floppy disk and is brought into the M&A memory where it executes as an overlay task. Whenever the operation of the controller is questionable, the controller diagnostics are called in by the diagnostic controller (O/S error handler). The controller diagnostics cluster assumes that the side being tested is off-line, unsynchronized, and the link card is fault free. Each of the cluster's modules test a portion of the controller with a specific test pattern that provides the diagnostic controller (O/S error handler) with valuable information to operate on.

2. CP Memory Diagnostics Cluster.

This cluster does a series of tests on CP memory banks to determine if the memory cards are operating properly. This test series writes and reads memory-bit patterns; error messages are printed on the console.

3. CP Logic Diagnostics Cluster.

This cluster tests for the following types of errors: branch instructions with each condition-code bit set, jump instructions for a specific mode, a certain word instruction for destination mode 0, a certain byte instruction for non-zero-type destination mode, etc. The functions of this cluster are to initiate pointers/variables, transfer test routines into CP memory, initiate CP logic tests (with the interrupt) by way of the link, and read the response by way of the link. Error messages are printed on the console.

4. Link Diagnostics Cluster.

The modules of this cluster test the following functional areas of the link: MP ability to access link registers, MP ability to transfer data to/from the CP memory by way of the link, MP ability to interrupt the CP, and CP ability to interrupt the MP. The diagnostic controller (O/S error handler) compiles link errors with other errors before a faulty-link message is reported.

5. Matrix Switch (Timeslot Interchange) Memory Diagnostics Cluster.

This cluster has a card test scheduler, a test module scheduler, and various test modules. When the diagnostic controller (O/S error handler) calls this cluster, the controller sets up an 8-bit vector field that selects one of eight memory cards for testing. After the test routines are completed, an 8-bit and a 16-bit vector field are filled with 1s and 0s. The 8-bit field indicates a pass/fail mode (bit set = tests failed), and the 16-bit field indicates the failure mode (bit 0 set = write operation; bit 1 set = read operation; bit 2 set = timeout; bit 3 set = enable; bit 4 set = data mismatch; and bits 5 through 15 = spare).

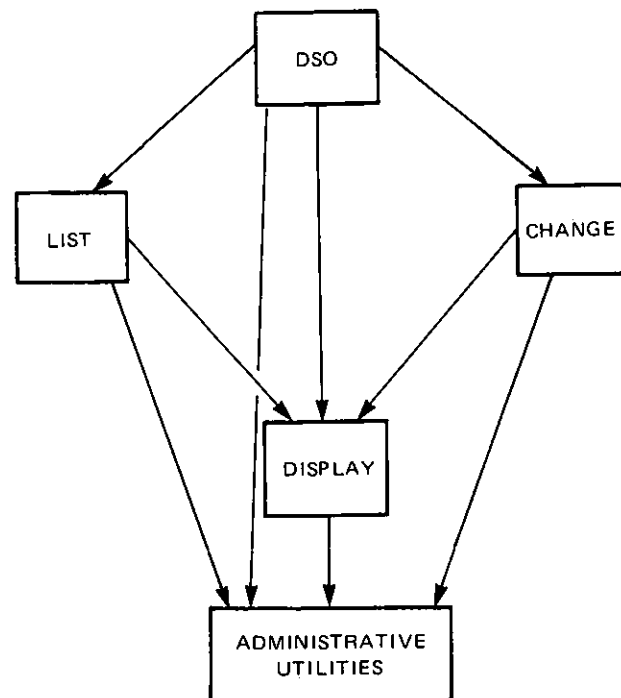
e. Data Base Administrative Program.

1. Directory Service Order (DSO) Program Overlay.

The DSO cluster is the main controlling cluster for the DSO overlay. Its functions are: to print the overlay heading, read requests, edit and interpret requests, and determine which other cluster to call to execute the request. The DSO interfaces with the CPs and the console by way of the executive cluster. The function of the DSO overlay is to display and list all directory number, location number, PBX, or trunk information in the CP data base on request. It also controls administrative changes to single-party and multiparty assignments in the data base. The DSO is built as an overlay task residing on the MSD. It is executed by making a program request (SDSO, carriage return) to the MP executive cluster. The DSO can do the following tasks: list and/or display directory number (DN), location number (LN), PBX group, and trunk group assignments; change the routing of an unassigned DN; assign a DN to an LN or an unassigned party of an LN; detach a DN from an

LN or a party of an LN; detach a whole multiparty LN; and change the COS, ringing code, or circle digit of an assigned DN/LN. (For this discussion, the term EN for equipment number, used in call-processor related text, and the term LN for location number, used in M&A related text, can be considered to be synonymous.)

a) Figure 6 shows the DSO overlay. The change cluster is called and controlled only by the DSO cluster and uses DISP for display. The change cluster interfaces with the CPs and the console by way of the executive, display, and DSO clusters.



CP 5361

Figure 6. Directory Service Order Overlay

b) The function of the list cluster is to print out a series of directory numbers. A DN (or LN) is specified when the list cluster is called (one or the other, not both). The list cluster prints the first DN/LN, using the display cluster. Then, the list cluster calculates the next DN/LN and loops back to print it out. This

continues until all numbers in the system are printed out or until the operator interrupts the operation with the escape (ESC) key.

c) The display cluster is loaded and controlled by either the DSO, PBX, or trunk overlays (applies to the list cluster also). It is called by various clusters within these overlays to do all display functions involving DN/LN assignments. It can handle unassigned DN/LNs, single-party/multiparty assignments, PBX assignments, and trunk group assignments.

d) The administrative utilities cluster contains a group of unrelated modules used by several other administrative clusters. Examples of the functions done by these modules are: ASCII to binary; binary-coded-decimal to ASCII; compare address of ASCII buffer and compare table; edit console input buffer by fields; fill buffer, 1 byte at a time; print heading every tenth pass; limit-check a string of bytes against an upper and lower limit; convert LN to port number; move string of bytes from a source to a destination address; print an administrative message code on console; print console output buffer and check for abnormal termination; and find current LN in PNLN table, equivalent to current port number given.

2. PBX Group Update Program Overlay.

The PBX group update administrative program creates and changes PBX and centrex group entries. The functions for PBX are list, change, and display. The list and display functions are identical to the list and display functions of the DSO. The change function adds or changes PBX groups one line at a time. The pilot number line is typed in first. If the pilot DN is unassigned, the program sets up a new PBX group. Each succeeding line adds or changes an EN assignment. A carriage return leaves a line as it was.

3. Trunk Group Update Program Overlay.

The trunk group update program contains list and change functions also. The list function can be requested either by trunk group number or by any one of the ENs within the trunk group. The change function adds or changes trunk groups, one line at a time. The trunk group number line is typed in first.

4. Translation Table Update Program Overlay.

The translation table update program administers changes to code translation assignments and screen-class/route-treatment assignments in the data base. It also provides the system operator with the capabilities to list and display all code translation and screen table information in the CP data base.

5. PABX Group Update Program Overlay.

The PABX group update program administers changes to the software trunk table (TKTBL). This program provides the capability to display a selected group, list all groups starting at a specified entry, change one or all parameters of a specified group, add a new member to an existing group, and delete a member from a group.

6. Line Class-of-Service Update Program Overlay.

The line class-of-service (LCOS) update program administers changes to the LCOS table information contained in the CP data base. This program provides the capability to add, delete, derive, or purge (clear) table entries, display or list all LCOS table entries, and list all equipment numbers that reference a specific LCOS table entry.

7. Trunk Class-of-Service Update Program Overlay.

The trunk class-of-service (TCOS) update program provides the capability to display a selected TCOS entry, list all TCOS entries starting at a specific entry, and change one or all valid parameters of a specified TCOS entry.

8. Route Treatment Update Program Overlay.

The route treatment update program provides the capability to list all entries, display an entry, add or delete entries, derive a new entry from an existing table entry, purge all unreferenced entries, and list all active references in the CP data base to a built-table entry. The program provides information relating to various routing methods (for example, custom-service, local-office, recorded announcement, trunk-group, partial-dial, etc.).

9. Subscriber Line Transfer Program Overlay.

The subscriber line transfer program provides the capability to list or display business-residence directory numbers and change or add to the CP data base table titled business/residence number subscriber line transfer.

10. Fixed-Called-Address Update Program Overlay.

The fixed-called-address (FCA) program provides the capability to list, display, change, add, or delete information relating to the FCA CP data base tables. These tables contain the ENs, associated DNs, and state timers needed to delay automatic transfer of DNs.

f. Traffic Measurement Program.

The traffic measurement and reporting system (TMRS) provides the operating company with various statistical traffic reports. These reports describe the switching system operation with respect to traffic load for various time intervals. The base entry system records peg-count and overflow-count measurements. A peg count is a count of system element usage and is incremented with each use. An overflow is a count of the unsatisfied requests for a system element (due to insufficient capacity); this count is incremented once for each unsatisfied request on a per-element/per-call basis. TMR systems, for the base entry DCO switches, are generally called traffic metering systems, a subset of traffic measurement. The measurements are derived from traffic events detected by the CP system. The computational workload of this TMRS is distributed between the CP and the MP. The traffic-event detection software resides in the CP, and the data collection, analysis, and reporting software resides in the MP. An interface, in the form of a traffic-event queue, is used to pass event information from the CP to the MP to compute various traffic measurements. The CP is responsible to the queue for detecting and loading traffic events. The MP is responsible to the queue for maintaining the man-machine interface to the TMRS and for unloading events from the queues before overflow.

g. System Utilities Program.

This program provides the system operator with the capabilities to enter, change, or print time-and-date information and to get status information relative to the port I/O (lines/trunks) section of the DCO system.

h. Field Expansion.

Changes to the CP program or data base involve procedures different from those of hardware, since two sides cannot operate in synchronism with dissimilar software loads. If the changes are in the CP program, then they are checked out at the factory, using computer-lab facilities (software simulators), and also on equipment-lab facilities, using an actual telephone system. Once the program has been thoroughly checked, a new disk is prepared that contains the proper versions of all programs the customer presently uses. The data base is not included on this new disk since Stromberg-Carlson Corporation does not normally have the customer's updated information. Several copies of this new disk are normally supplied to the customer. Using the "copy data base" command, the existing data base is copied onto the new disks. Once this is completed, the new CP program is loaded into the off-line CP. A recovery, with an immediate switchover, is then used to put the DCO in synchronous operation.

i. Fault Detection and Reporting.

Throughout the DCO system, detectors monitor various functions and alert the MP when irregularities exist. Some of the major detectors are: parity, timeouts, and mismatch.

1. Parity generation and checking is incorporated in the CP memory and port control store. Any parity errors are reported to the MP and also to the switchover logic.

2. Many timeout functions (DMA request, interrupt request, watchdog, sanity, process request, reply, timeslot interchange, etc.) are tested, and the MP is alerted if a timeout occurs. The switchover logic contains a watchdog timer circuit to monitor the processors (MP and CPs) by requiring a specific signal periodically from each processor. A timeout occurs if the watchdog timer does not receive this signal in time. The bus synchronization (SYNC) circuit forces an instruction-by-instruction SYNC and gives a reply timeout alarm if either CP fails to complete a bus transaction in time. A sanity test program is resident in both CPs and can be called by a specific hardware fault or by the MP; a timeout circuit on the switchover logic alerts the MP if the expected output word is not received from each CP in time.

3. For mismatch detection, the bus synchronization circuit compares data between CPs on each bus transaction. It provides an MP mismatch indication and also informs the MP whether the operation was a read or a write. Mismatch detectors also compare A and B system data between the timeslot interchange (TSI) and the controller, between the TSI and the supervisory bit buffer, and between the TSI and the port group circuit.

4. All circuits provided on a traffic basis, including all port types, are checked primarily by the generation of test calls. The pattern of the test calls is such that every major area of the system is checked frequently, at least once a day.

5. Other detection methods involve the telephony preprocessor (TPP) checking for proper correlation of control and sense bits as an indication of proper supervisory channel functioning. The TPP also detects the status of the port check bit generated by the port circuit. Another timeout is where checksums are strategically located within the CP software to allow the MP to validate the program and the data base load whenever it demands. The reporting of faults is represented by messages printed out on the console, caused by incoming detector signals. Data concerning detected faults is also printed out to allow manual monitoring of the diagnostic process. All diagnostic actions are reported on the console, plus messages concerning specific PWBA replacement. Remote reporting of faults is available, either as calls to signal a remote location by means of tones or as a remote console reporting over a dedicated line.

j. **Fault Correction and Recovery.**

For correcting faulty nonredundant equipment by replacement, each card is correctly identified to avoid any possible interruption of the customer's service. Card removal or insertion can cause minor transients leading to the printout of error messages. All messages require careful interpretation so that an accurate assessment of the replacement card can be made. When the faulted card is in the redundant equipment, the entire side will be out of service, and no error messages should result from card removal/insertion operations. The side is recovered

and put into synchronous operation after the replacement card is properly checked. All replacement actions warrant the placing of sufficient (quantity and quality) test calls to ensure proper reinstated operation. The recovery process involves restoring the system to normal operation when the on-line side is functioning correctly and the off-line side has been taken out of service (removed from synchronous operation) for any reason. Recovery can be initiated manually or automatically. Automatic recovery is usually attempted after transient failures. Manual recovery is usually attempted after faulty cards have been replaced. The recovery process consists of loading the inactive CP and all its associated memories and stores with a copy of all programs and data, including time-perishable data, as found on the active CP side. The recovery function is done through software in the call and maintenance processors.

5.03 Administrative Tasks.

a. **Overview.**

The configuration features and classes of service of a telephone switching office depend on how its data base is set up. The data base consists of a series of translation and directory tables. These tables are initially configured by an off-line computer lab program that sets up the tables, leaving room for future expansion. The administrative programs can make changes to the existing office configuration but cannot add new tables. They only make changes to existing tables. Specifically, they can change an entry in a table or can use up a spare location to create a new entry in an existing table. The administrative programs also provide a backup copy of the data base and monitor traffic/service related information. The responsibility for the execution of administrative functions is divided between MP and CP software. The MP is responsible for all decision making and all administrative I/O, while the CP does all data retrieval and changes. Administrative tasks are grouped into related function programs. Each program is a mass-storage resident overlay that must be read into the MP memory for execution. The operator calls the overlay by typing in a "\$" and the name, followed by a carriage return. The MP executive cluster recognizes the name, reads the overlay into memory, and starts executing the overlay. Before an operator

can request a function, the heading format requirement must be satisfied. Each program prints out a heading before it accepts any function request. Entry parameters can be lined up under the appropriate heading by using the tab feature. Individual functions are requested by typing in the function request command name and any parameters. Examples of request commands are: LIST, CHANGE, and DISPLAY. A function is terminated from the middle of an entry by operating the escape key. This restarts the overlay at a point just after the heading. Operating this key a second time causes an exit from the overlay and return to the executive level.

b. User-Related Information.

The operator communicates with the MP by way of the system console. The operator cannot directly access the CPs. All communications with the CPs are under the control of MP software. The system I/O

console is normally a TTY and has hard-copy display with an alphanumeric entry keyboard to do the administrative tasks. Turning the console on signals the MP executive cluster of pending operator entries. The system executive cluster verifies this signal by printing out a prompting (request for response) character. Entries are not processed until the carriage-return key is operated. An entry can be changed any time up until the carriage-return entry. Operating the backspace key eliminates characters, one at a time, starting with the last one entered and working back. A reverse slash is typed for each character eliminated. Typing a normal alphanumeric character causes entry to resume, starting with the position of the last character eliminated.

c. Data Base Administration.

This task is described in paragraph 5.02e.

d. Traffic Measurement and Reporting.

This task is described in paragraph 5.02f.