OPERATIONAL SOFTWARE CONTROL STRUCTURE

SOFTWARE SUBSYSTEM DESCRIPTION

2-WIRE NO. 1 AND NO. 1A ELECTRONIC SWITCHING SYSTEMS

CONTENTS PAGE

CONTENTS PAGE

Figures

CONTENTS PAGE

Tables

CONTENTS PAGE

1. GENERAL

INTRODUCTION

1.01 This section provides an introduction to the operational software control structure that is responsible for the orderly scheduling and execution of programs used in call processing, input-output management, and administrative control during the operation of the No. 1/1A Electronic Switching System (ESS). Information unique to No. 1A ESS applications is so noted; applications unique to No. 1 ESS are not described in this section.

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

1.03 Part 8 of this section provides a defined list of abbreviations and acronyms used in this section.

PURPOSE OF OPERATIONAL SOFTWARE CONTROL STRUCTURE

1.04 The No. 1/1A ESS software is composed of numerous programs, each of which is tailored to perform a specific function. Optimum system performance is dependent upon the proper scheduling of these programs. The operational software control structure provides an efficient and orderly transfer to the various programs to ensure the required performance.

SCOPE OF SECTION

1.05 The information provided in this section includes a description of the following:

(a) The operational control programs that administer the call processing work essential for the call processing functions of the No. 1/1A ESS.

(b) The administrative programs that perform the various functions necessary for the efficient operation of the No. 1/1A ESS.

1.06 This section is based on the 1AE4 version of the generic program.

2. BRIEF DESCRIPTION OF PIDENTS

2.01 Program identifications (PIDENTs) associated with operational software at the major control level are as follows:

(a) ECIO—No. 1A ESS Executive Control Input-Output Program—ECIO administers and schedules the execution of J- and H-level input-output programs and sets up the control structure for running base-level jobs.

(b) ECMP—No. 1A ESS Executive Control Main Program—ECMP administers and schedules the execution of all base-level (L-level) programs.

(c) AOVD—No. 1A ESS Automatic Overload Control—AOVD identifies overload conditions as they occur and initiates control strategies to reduce the demands on system resources.

(d) LLOD—No. 1A ESS Line Load Control and Toll Network Protection—LLOD provides a means of assuring acceptable grades of service to essential lines when a traffic overload exists.

3. NO. 1A ESS OPERATIONAL SOFTWARE

PROGRAM HIERARCHY

3.01 The No. 1A ESS programs are designed primarily to handle the call processing requirements of the office and, in addition, to provide for other operational, maintenance, and recovery functions.

3.02 All L-level programs collectively form the base-level program. Despite functional subdivisions that are made for various purposes, this is operationally a single program made up of a complex of loops without a beginning or an end. The core of this base-level program is the executive control main program (ECMP). In the absence of interrupts, the system operates on the base or L level.

A. Interrupt Structure

3.03 The interrupt structure is a hierarchy of ten interrupt levels, with A being the highest priority level and K, the lowest. An interrupt can seize control from the base level or from any interrupt level of lower priority. Interrupts are caused by various conditions in the central control (CC); for example, a time-out of a 5-ms interval by the system clock causes a J-level interrupt. When such conditions occur, the interrupt system causes the CC to stop its present program task, stores the program address at which the interrupt oocurred, and then transfers to the appropriate interrupt-level program. When interrupt processing is complete, control is returned to the program that was interrupted or to a safe point in the maintenance program.

3.04 The basic program control structure, including the hierarchy of the interrupt sources (A, B, C, D, E, F, G, H, J, and K), is illustrated in Fig. 1. The K-level interrupt is not used in the No. 1A ESS.

B. Base-Level Structure

3.05 Frequency classes A through E and interject represent the classes of base-level programs. Within each class, there is a fixed sequence of major program units called task dispensers. The majority of these are for call processing and administration. In general, they dispense program control to one or more task programs a consecutive number of times, depending on the number of tasks that the task dispenser program finds waiting. Occasionally, another task program is interjected into the flow between any two task executions. When a task program returns control to its task dispenser program, the latter checks to see if an interject request has been made. If so, the interject request will be executed before task dispensing is resumed.

FUNCTIONAL DESCRIPTION

3.06 Operational software can be functionally divided into the following parts:

- Call Processing
- Equipment Dependent Overhead
- Administrative Work.

3.07 Call processing work that has critical timing requirements (interfacing with the outside world) is performed on J-level. The bulk of call processing has less critical timing requirements and, thus, can be performed on base level.

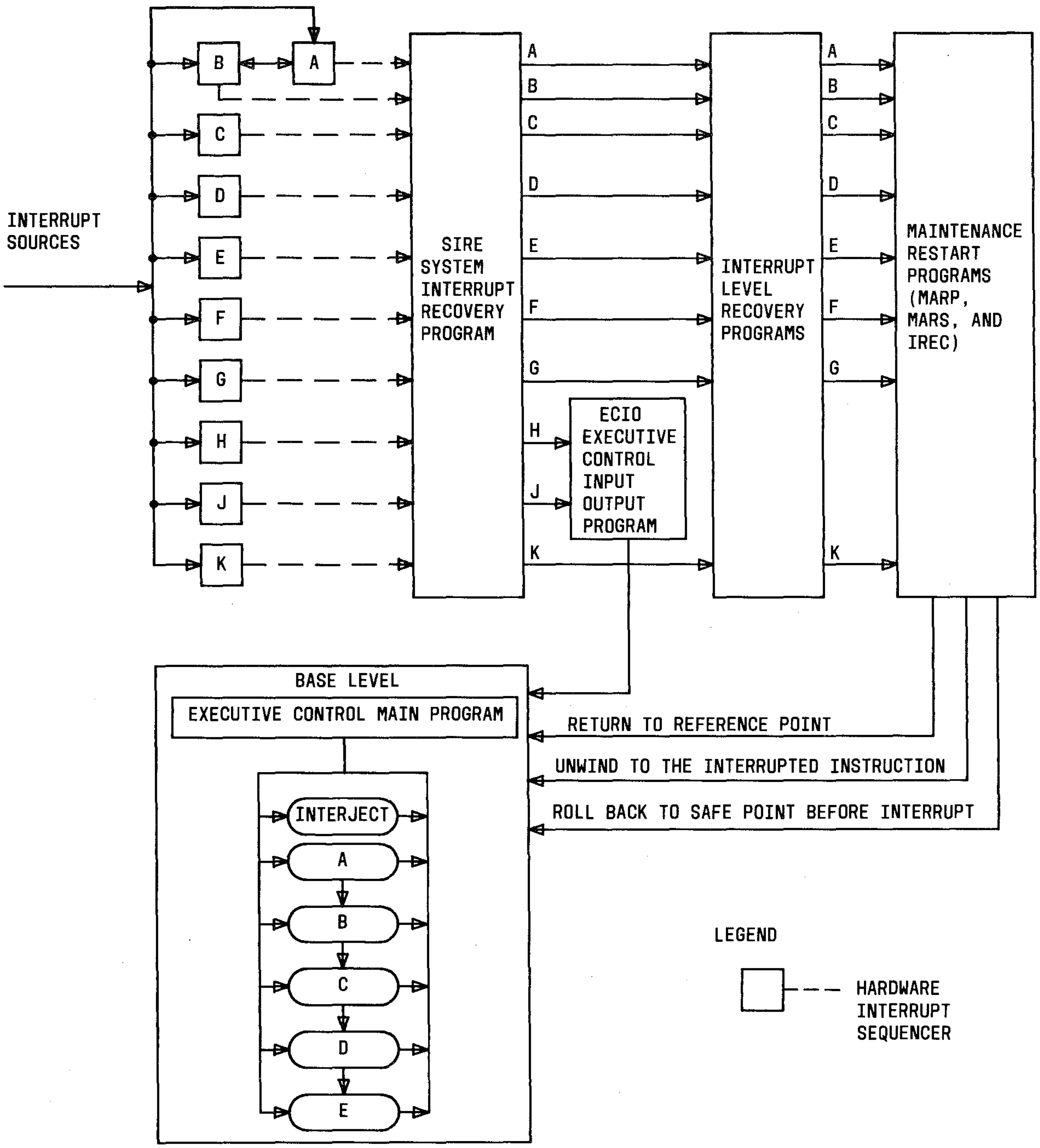
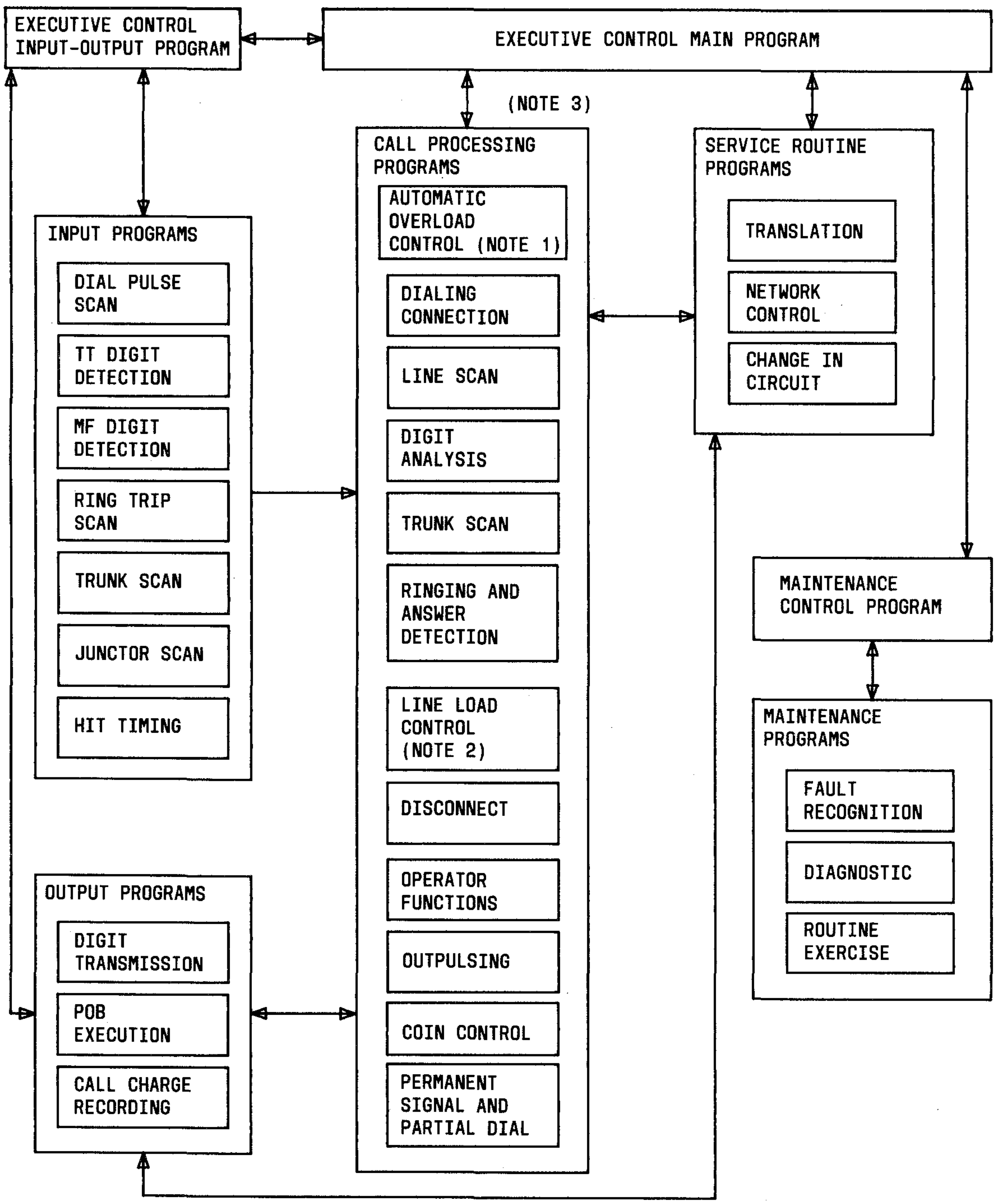


Fig. 1—Basic Program Control Structure

3.08 Similarly, equipment dependent overhead with critical timing requirements (eg, digit scans) is performed on J level, while other work (supervisory trunk and line scans) in this functional category is performed on base level.

3.09 Both levels have administrative executive control programs which are responsible for the jobs performed on their respective levels (Fig. 2). Other administrative functions include automatic overload control and line load control.



NOTES:
1. THE AUTOMATIC OVERLOAD CONTROL PROGRAM CONTAINS BOTH CALL PROCESSING AND OVERLOAD CONTROL ROUTINES.
2. LINE LOAD CONTROL IS DEPICTED UNDER CALL PROCESSING SINCE CALL PROCESSING IS AFFECTED WHEN LINE LOAD CONTROL IS ACTIVE.
3. THE INTERCONNECTION LINES INDICATE THE GENERAL FLOW OF INFORMATION AMONG PROGRAMS. THESE LINES DO NOT DEPICT SPECIFIC PROGRAM RELATIONSHIPS OTHER THAN THE RELATIONSHIP TO THE EXECUTIVE CONTROL MAIN PROGRAM AND TO THE TYPES OF PROGRAMS. THE EXECUTIVE CONTROL MAIN PROGRAM SCHEDULES THE WORK OF ALL CALL PROCESSING PROGRAMS.

Fig. 2—Program Functional Grouping

4. NO. 1A ESS EXECUTIVE CONTROL INPUT-OUTPUT PROGRAM (ECIO)

GENERAL

4.01 Tasks associated with the system input-output functions are called "nondeferrable" tasks. These tasks must be performed punctually and usually repetitively; otherwise, critical data (eg, dial pulses) will be lost. "Deferrable" tasks (executed on the base or L level) are associated with data already in the system and are not, therefore, as critically synchronized to real time as the nondeferrable tasks.

4.02 To ensure punctual performance of the nondeferrable tasks without impairing the efficiency of the deferrable tasks, an interrupt facility is incorporated in the system. Every 5 milliseconds (ms) a system clock activates a J-level interrupt which gives control to the executive control input-output program (ECIO). Some input-output tasks are performed every 5 ms; others are performed at multiples of 5 ms and still others, at multiples of 20 ms. Thus, the input-output activities are classified into high-priority tasks and low-priority tasks according to the frequency and urgency with which these tasks must be performed.

4.03 Low-priority tasks can be delayed for a few milliseconds without adverse effect on the operation of the system. This will be the case when the coincidence of input work under a peak traffic load causes the system to take more than 5 ms to complete the high- and low-priority tasks. In this event, the H-level interrupt (actually high-priority J-level) will occur and the low-priority work will be interrupted. The accumulated high-priority work will again be performed before returning to the low-priority work that was interrupted. When the nondeferrable tasks or jobs have been completed, return is made to the interrupted deferrable program on base level. This cycle of activity is illustrated in Fig. 3.

ORGANIZATION

4.04 PIDENT ECIO uses three call store timetables to administer the scheduled execution of input-output routines, one for H-level or high-priority tasks and two for J-level or low-priority tasks. A fourth call store timetable, designated ECIOTABL, is used to set the flags that control certain base-level activities.

A. High-Priority Timetable

4.05 This timetable (Fig. 4) and its associated transfer table (Table A) are used to dispense the H-level tasks. The transfer table consists of 23 consecutive words, each of which is used to store the starting address of an input-output routine assigned to the high-priority timetable. The global name of each routine, as well as the associated PIDENT names, is listed in Table A. Blank words in the transfer table provide space for starting addresses of future input-output routines.

4.06 The timetable consists of 24 consecutive words, each of which is associated with a particular 5-ms interval within a 120-ms cycle. A count kept in word E4HCNT identifies the current 5-ms interval and its associated timetable word.

4.07 Within each timetable word, bits 0 through 22 are associated with the input-output routines whose starting addresses are stored in the corresponding words (0 through 22) of the transfer table. Thus each column of the timetable is associated with an input-output routine. Each column has an activity bit which is part of activity bit word E4HACT. If the activity bit for a particular column is equal to 1, the 1s marked within that column designate the 5-ms intervals during which the associated input-output job is scheduled for execution. The timetable shows, for example, that jobs 0, 1, 3, 4, 10, 19, and 22, when active, are scheduled every 5-ms interval, while others are scheduled less frequently.

B. Low-Priority Timetables

4.08 There are two low-priority timetables, henceforth referred to as J0 and J1. These timetables and their associated transfer tables are used to dispense the J-level tasks.

J0 Tables

4.09 The J0 transfer table (Table B) consists of 23 consecutive words, each of which is used to store the starting address of an input-output routine assigned to the J0 low-priority timetable. The global name of each routine, as well as the associated PIDENT name, is listed in Table B. Blank words in the transfer table provide space for starting addresses of future input-output routines.

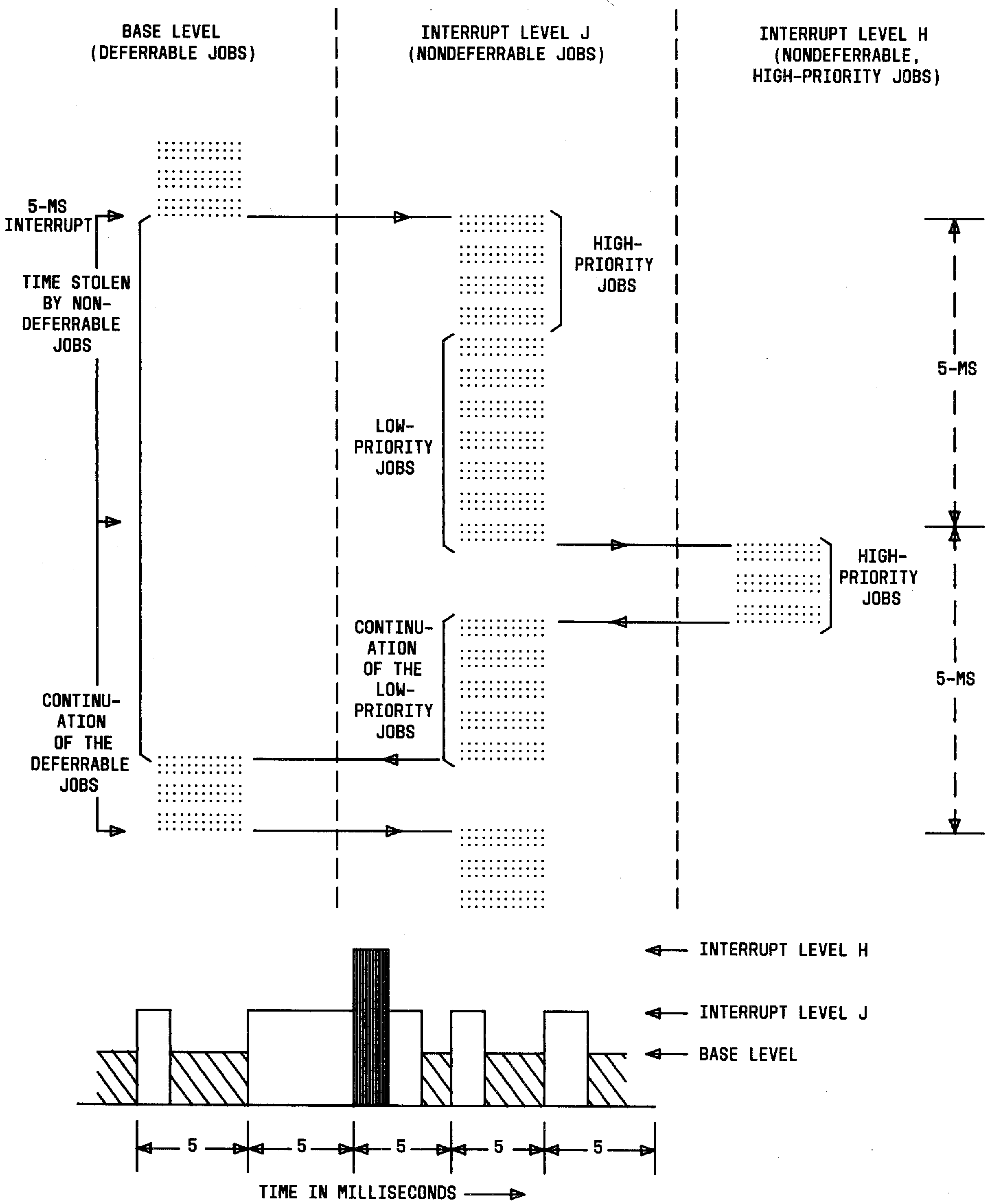


Fig. 3—Call Program Operation

E4HCNT: COUNT OF 5-MS INTERVALS (0-23)

HJOBS	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E4HACT	1	1	1	1		1	1					1	1		1	1	1	1	1	1		1	1
TIMETABLE WORDS 0	1			1		1	1					1	1			1		1	1	1		1	1
1	1			1									1				1		1	1		1	1
2	1			1									1		1			1	1	1		1	1
3	1			1									1				1		1	1		1	1
4	1			1									1			1		1	1	1		1	1
5	1			1									1				1		1	1		1	1
6	1			1								1	1		1			1	1	1		1	1
7	1			1									1				1		1	1		1	1
8	1			1		1							1			1		1	1	1		1	1
9	1			1									1				1		1	1		1	1
10	1			1									1		1			1	1	1		1	1
11	1			1									1				1		1	1		1	1
12	1			1			1					1	1			1		1	1	1		1	1
13	1			1									1				1		1	1		1	1
14	1			1									1		1			1	1	1		1	1
15	1			1									1				1		1	1		1	1
16	1			1		1							1			1		1	1	1		1	1
17	1			1									1				1		1	1		1	1
18	1			1								1	1		1			1	1	1		1	1
19	1			1									1				1		1	1		1	1
20	1			1									1			1		1	1	1		1	1
21	1			1									1				1		1	1		1	1
22	1			1									1		1			1	1	1		1	1
23	1	1	1	1									1				1		1	1		1	1

Fig. 4—High-Priority Timetable (5-ms)—Call Store Layout

4.10 The timetable (Fig. 5) consists of 20 consecutive words, each of which is associated with a particular 5-ms interval witin a 100-ms cycle. A count kept in word E4JCNT identifies the current 5-ms interval and its associated timetable word.

4.11 The relationship between the J0 timetable and the J0 transfer table is identical to that explained in paragraph 4.07, with one exception: the activity bit word is E4JACT. Figure 5 shows that jobs 1, 9, 19, and 22, when active, are scheduled every 5-ms interval, while others are scheduled less frequently.

J1 Tables

4.12 The J1 transfer table (Table C) consists of 23 consecutive words, each of which is used to store the starting address of an input-output routine assigned to the J1 low-priority timetable. The global name of each routine, as well as the associated PIDENT name, is listed in Table C.

TABLE A

TRANSFER TABLE FOR HIGH-PRIORITY JOBS (HJOBS)

JOB NAME	WORD	GLOBAL/LOCAL	PIDENT
Emergency Action Idle Mode Initialization	0	ESKIPO	ECIO1A00
Service Circuit Diagnostic Timing	1	TNDJLP	TNKC1A00
Unassigned	2		
General Maintenance Timing	3	MACS47	MACR1A00
Compatibility Check	4	PGI13	ECIO1A00
Digit Scan (2)	5	RSCAN2	RSCN1A00
Digit Scan (1)	6	RSCAN1	RSCN1A00
Outgoing Revertive Scan (1)	7	OGRSC1	OGGT1A00
Outgoing Revertive Scan (2)	8	OGRSC2	OGGT1A00
Unassigned	9		
Audit Segment Timing	10	SASGMT	SADT1A00
Long Audit Segment Network Counter	11	SAQETM	SADT1A00
Unassigned	12		
Unassigned	13		
Unassigned	14		
Unassigned	15		
Multifrequency Outpulsing	16	MFPULS	MFOP1A00
Centrex Maintenance Timing	17	CXMAIO	CXMA1A00
Unassigned	18		
In-Phase Peripheral Execution	19	QEINPH	QEPR1A00
Emergency Action Long-Timer Test	20	EACNTL	MIRVRECV
Recycle H-Flags Job Control	21	ECHP03	ECIO1A00
End of H-Flag Processing	22	ECHGBN	ECIO1A00

Note: Refer to PK-6A001 for data in other generics.

Blank words in the transfer table provide space for starting addresses of future input-output routines.

4.13 The timetable (Fig. 6) consists of 10 consecutive words, each of which is associated with a particular 20-ms interval within a 200-ms cycle. A count kept in word E4JCT1 identifies the current 20-ms interval and its associated timetable word.

4.14 The relationship between the J1 timetable and the J1 transfer table is identical to that explained in paragraph 4.07, with one exception: the activity bit word is E4JAC1. Figure 6 shows that job 3, when active, is scheduled every even 20-ms interval and jobs 4 and 5, every odd 20-ms interval, while others are scheduled less frequently.

C. Base-Level Timetable (ECIOTABL)

4.15 This timetable is used by ECIO to set flags for certain types of jobs to be performed on base level. Figure 7 illustrates the call store layout of ECIOTABL which consists of 40 consecutive words, each of which is associated with a particular 5-ms interval within a 200-ms cycle. A count kept in EX2INDX identifies the current 5-ms interval and its associated timetable word. The columns in ECIOTABL correspond to frequency classes, supervisory trunk or line scanning work, or certain miscellaneous jobs to be performed on base level. Each column has an activity bit which is part of activity bit word EX2ACT. If the activity bit for a particular column is equal to 1, the 1s marked within that column designate the scheduled rate of execution for that particular job. The timetable shows, for example, that jobs represented by columns 0 and 9, when active, are scheduled every 5-ms interval, while others are scheduled less frequently. Frequency classes A, B, C, D, and E in columns 4 through 8 are set at 10-ms, 20-ms, 40-ms, 50-ms, and 100-ms intervals, respectively. This establishes the minimum average intervisit time for each class. A detailed description of the manner in which the classes are actually visited is given in paragraphs 5.11 through 5.21.

OPERATION

4.16 A general flow diagram of J- and H-level interrupt processing via ECIO is shown in Fig. 8. The 700-nanosecond clock pulses in the central control (CC) are counted, and every 5 ms the counting circuit generates an output signal which interrupts the base-level program in progress at that time. The 5-ms clock requests both H- and J-level interrupts every 5 ms by setting H- and J-source flip-flops. The H level is normally inhibited, allowing the J level to assume control. Program control is transferred via hardwired circuitry to the system interrupt recovery program (SIRE) which stores the state of the control flip-flops and the contents of the CC registers in the J-level interrupt bin. SIRE transfers program control via the Processor Application Transfer Table (PATT) to global ECJLEV in ECIO.

A. Processing High-Priority J-Level Tasks

4.17 The primary operations performed in ECJLEV are as follows:

(a) The high-priority timetable counter (E4HCNT) is incremented by 1 and is used as an index to read the appropriate row in the timetable.

(b) The row read from the high-priority timetable is ANDed with the activity bit word (E4HACT). The ANDed results are inserted into call store word E4HJOBS. This word then contains a 1 in every position in which the timetable entry and the activty bit word contain a 1. Each position marked by a 1 indicates a task that is active and scheduled for excution.

(c) If all bits in E4HJOBS are 0, transfer is made to the low-priority timetable routine (ECLP01) where processing of low-priority tasks begins. If one or more E4HJOBS bits are equal to 1, the position of the rightmost 1 is saved for indexing into the high-priority transfer table (HJOBS—Table A). The rightmost 1 in E4HJOBS is itself set to 0 to allow the next 1, if any, to be recognized later as the new rightmost 1.

(d) The modified E4HJOBS word is stored in E4HSTR.

(e) The address of the HJOBS table is indexed with the value saved for the rightmost 1 to obtain the transfer table entry containing the address of the input-output routine corresponding to the rightmost 1.

(f) A transfer is made to the address of the input-output routine. When the input-output task has been completed, transfer is made back to the next instruction in ECJLEV.

(g) Call store word E4HSTR [step (d) above] is read. If all bits of the word are 0, transfer is made to the low-priority timetable routine (ECLP01) where processing of low-priority tasks begins. If one or more E4HSTR bits are equal to 1, the position of the new rightmost 1 is saved for indexing into the high-priority transfer table (HJOBS). The rightmost 1 is itself set to 0 to allow the next 1, if any, to be recognized later as the new rightmost 1.

TABLE B

TRANSFER TABLE FOR 5-MS LOW-PRIORITY JOBS (J0JOBS)

JOB NAME	WORD	GLOBAL/LOCAL	PIDENT
Supervisory Scans	0	SUSC_TS	CCLT0000
Emergency Action Idle-Mode Job Control	1	ELIDLE	ECIO1A00
Incoming SXS Digit Scan	2	SXSIST	SXSI1A00
Output EADAS Characters	3	EXAMIT	EDAS1A00
Power-Cross Scan	4	QEPXBE	QEXC1A00
Fast Answer Junior Register Processing	5	FAJRSN	FJRP1A00
Centrex Maintenance I/O Timing	6	CXMAI0	CXMA1A00
Centrex Key and Lamp Order I/O	7	CXCCIO	CXIO1A00
Timed Scan Junior Register (TSJR) Scan	8	DITSJR	DITS1A00
POB Execution During Emergency Action	9	QEINPH	QEPR1A00
Incoming SXS AIT Scan	10	SXSAIT	SXSI1A00
Supervisiory Fast Scan	11	SUSC_IO	CCLT0000
SXS Trunk Side Scan	12	SUISTS	CCLT1A00
Long Audit Segment Network Counter	13	SAEQTM	SADT1A00
Abandon and Interdigtal Scan	14	RSAITS	RSCN1A00
Increased Centrex Data Link I/O	15	CXCCIO	CXIO1A00
Ring-Trip Scan	16	SUTRIP	SURG1A00
Unassigned	17		
POB Execution	18	QEBEGN	QEXC1A00
Generic Utility	19	LULPTREM	LULPUTIL
Unassigned	20		
Process J1 Flags	21	ECJP20	ECIO1A00
End of J-Level Processing	22	ECJEND	ECIO1A00

Note: Refer to PK-6A001 for data in other generics.

(h) The modified word is again stored in E4HSTR, and steps (e) and (f) are performed.

(i) Steps (g) and (h) are executed repetitively until, one by one, the high-priority flag bits are removed. At this point, transfer is made to the low-priority timetable routine (ECLP01).

B. Processing Low-Priority J-Level Tasks

4.18 Before ECLP01 initiates the processing of low-priority tasks, the H-source flip-flop is reset and the H-level interrupt is uninhibited to permit H-level interrupts to occur. In addition,

E4JCNT	COUNT OF 5-MS INTERVALS (0-19)																						
E4JACT	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TIMETABLE WORDS	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1			1	1					1	1	1		1							1	1	1
1	1			1				1						1	1			1				1	
2	1			1				1			1	1		1		1		1			1	1	1
3	1	1		1									1	1	1			1	1	1		1	
4	1			1				1			1	1		1				1	1		1	1	1
5	1			1	1									1	1							1	
6	1			1				1			1	1		1			1	1		1	1	1	1
7	1	1		1					1	1				1	1	1		1				1	
8	1			1				1			1	1		1				1	1		1	1	1
9	1			1				1						1	1			1	1	1		1	
10	1			1	1						1	1		1							1	1	1
11	1	1		1				1						1	1			1				1	
12	1			1				1			1	1		1		1		1			1	1	1
13	1			1									1	1	1		1	1	1	1		1	
14	1			1				1		1	1	1		1				1	1		1	1	1
15	1	1		1	1									1	1							1	
16	1			1				1			1	1		1				1		1	1	1	1
17	1			1					1					1	1	1			1			1	
18	1			1			1	1			1	1		1				1	1		1	1	1
19	1	1		1				1						1	1		1	1	1	1		1	

JOJOBS

Fig. 5—J0 Low-Priority Timetable (5-ms)—Call Store Layout

certain base-level activities are scheduled for execution as follows:

(a) The base-level timetable counter (EX2INDX) is incremented by 1 and is used as an index to read the appropriate row in the timetable (ECIOTABL—Fig. 7). The counter is recycled when it reaches the size of the table.

(b) The row read from ECIOTABL is ANDed with the activity bit word (EX2ACT). This allows jobs in the table to be turned on and off. The ANDed results are inserted (to preserve outstanding flags) into call store word EX2CONTROL. This word is interrogated routinely in ECMP at global entry ECMPNXJB (paragraphs 5.08 through 5.10).

4.19 The low-priority tasks are processed in the same manner as that previously explained for high-priority tasks. After the proper timetable entry has been selected from the low-priority 5-ms timetable (J0_5_MS—Fig. 5) and ANDed with the activity bit word (E4JACT), the ANDed results are inserted in call store word E4J0JOBS. If all bits in the word are 0, transfer is made to global ECJGBN where CC registers G, X, Y, Z, K, J, F, and L are restored and the order GBNHJ is given. This instruction inhibits the H-level interrupt, resets the J-source flip-flop, restores the B register

TABLE C

TRANSFER TABLE FOR 20-MS LOW-PRIORITY JOBS (J1JOBS)

JOB NAME	WORD	GLOBAL/LOCAL	PIDENT
Unassigned	0		
Unassigned	1		
Unassigned	2		
Revertive Pulse Reception	3	RVINIT	RVRC1A00
Revertive Pulse Reception	4	RVCPDS	RVRC1A00
Panel Call Indicator Outpulsing	5	OGPCIP	OGPC1A00
Outgoing Trunk Start-Pulse Scan	6	OGSPSC	OGWN1A00
Outgoing Revertive Release Scan	7	OGRLSC	OGGT1A00
Unassigned	8		
Trunk DP Outpulsing (Off-Hook)	9	DPOFHK	DPGE1A00
Trunk DP Outpulsing (On-Hook)	10	DPONHK	DPGE1A00
Unassigned	11		
Unassigned	12		
Multiple Bit Scan	13	MUMBSR	MUMB1A00
Unassigned	14		
Unassigned	15		
Centrex Lamp Order Queue	16	CNLPQE	CNLP1A00
Unassigned	17		
Unassigned	18		
Centrex Data Link Restore	19	CXMARD	CXMS1A00
Unassigned	20		
Recycle J-Flags Job Control	21	ECJP03	ECIO1A00
End of J-Flag Processing	22	ECJP2R	ECIO1A00

Note: Refer to PK-6A001 for data in other generics.

and the control flip-flops, and returns control to the interrupted base-level program.

4.20 If one or more bits in E4J0JOBS are equal to 1, the rightmost 1 technique is used for processing the low-priority tasks scheduled for execution. If bit 21 is on (scheduled every fourth entry in the low-priority 5-ms timetable), transfer is made to routine ECJP20 where the tasks scheduled by the low-priority 20-ms timetable (J1_20_MS—Fig. 6) are processed in the same manner as that previously explained for the tasks scheduled by the low-priority 5-ms timetable. After the proper timetable entry has been selected from the low-priority 20-ms timetable (J1_20_MS) and ANDed with the activity bit word (E4JAC1), the ANDed results are

	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E4JCT1	COUNT OF 20-MS INTERVALS (0-9)																						
E4JAC1	1	1		1			1			1			1	1		1	1	1	1	1			
TIMETABLE WORDS 0				1										1						1			
1																		1	1				
2							1						1							1			
3				1												1	1	1	1				
4		1					1			1										1			
5				1										1				1	1				
6																				1			
7							1						1					1	1				
8				1												1	1			1			
9	1						1			1								1	1				

J1JOBS

Fig. 6—J1 Low-Priority Timetable (20-ms)—Call Store Layout

inserted in call store word E4J1JOBS. If all bits in the word are 0, transfer is made to global ECJGBN which returns control to base level. If one or more bits in E4J1JOBS are equal to 1, the rightmost 1 technique is again used for processing those tasks that are scheduled for execution.

4.21 Every 100 ms a bit is set which results in the counter for the low-priority 5-ms timetable being recycled to 0 and the interject source bit being set. This 100-ms interject flag (IN1SINJS set in ECIO local ECJP03) causes the execution of an ECMP routine (ECTK00) which updates the system clock and administers execution of a list of tasks from a 100-ms timetable. If ECJP03 finds that the previous interject request has not been answered, it increments a 100-ms correction register (E4TM+5) to cause the system clock to be corrected by 100 ms when ECMP enters the clock correction routine (ECADDE).

4.22 Every 200 ms the counter for the 20-ms timetable is recycled to 0, and return is made to low-priority 5-ms timetable processing. At this point, J-level work is completed with a transfer back to base-level processing via global ECJGBN.

C. Processing H-Level (High-Priority J-Level) Tasks

4.23 If J-level tasks are not completed before the next 5-ms clock interrupt, an H-level interrupt is triggered by the clock interrupt. Program control is then hardwired to SIRE. SIRE stores the contents of the CC registers and the state of the control flip-flops in the H-level interrupt bin and transfers program control via PATT to global ECHLEV in ECIO. The H-level interrupt counter is incremented for later use by ECMP in making a correction in the system clock. (This is necessitated by the fact that for each H-level interrupt, a J-level interval is lost. When this happens, there is a 5-ms delay in setting the 100-ms interject flag which results in the system clock being updated by 100 ms.) Transfer is made to global ECHP01 where the H-level execution of high-priority tasks is performed in the same manner as that explained in paragraph 4.17 with one exception. In paragraph 4.17 (i), the last bit in E4HSTR transfers control to global ECHGBN where the J-level source is cleared if it is found to be inhibited; CC registers G, X, Y, Z, J, K, F, and L are restored; and the order GBN is given to return control to the interrupted program.

4.24 If a 5-ms interrupt request is made during an H-level execution of high-priority tasks,

						RCFILL			IN_TL	TL_CT	RP_TL	MAC_FILL	IN_LS	RP_LS	TIMCALC	E (EINC)	D (DINC)	C (CINC)	B (BINC)	A (AINC)	HIGH_TK	INBAND	IOCPSPEC	HI_OCC
BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECIOTABL+ 0															1					1				1
1															1				1		1			1
2															1					1				1
3															1			1			1			1
4															1					1			1	1
5															1				1		1			1
6															1		1			1				1
7															1						1			1
8													1		1					1				1
9															1				1			1		1
10															1					1				1
11															1			1			1			1
12															1	1				1				1
13															1				1		1			1
14															1					1			1	1
15															1						1			1
16															1		1			1				1
17															1				1		1			1
18									1						1					1				1
19															1			1				1		1
20															1					1				1
21															1				1		1			1
22															1					1				1
23															1						1			1
24															1					1			1	1
25															1				1		1			1
26															1		1			1				1
27															1			1			1			1
28															1					1				1
29															1				1			1		1
30															1					1				1
31															1						1			1
32															1	1				1				1
33															1				1		1			1
34															1					1			1	1
35															1			1			1			1
36															1		1			1				1
37															1				1		1			1
38									1						1					1				1
39															1							1		1

Fig. 7—Base-Level Timetable (ECIOTABL)—Call Store Layout

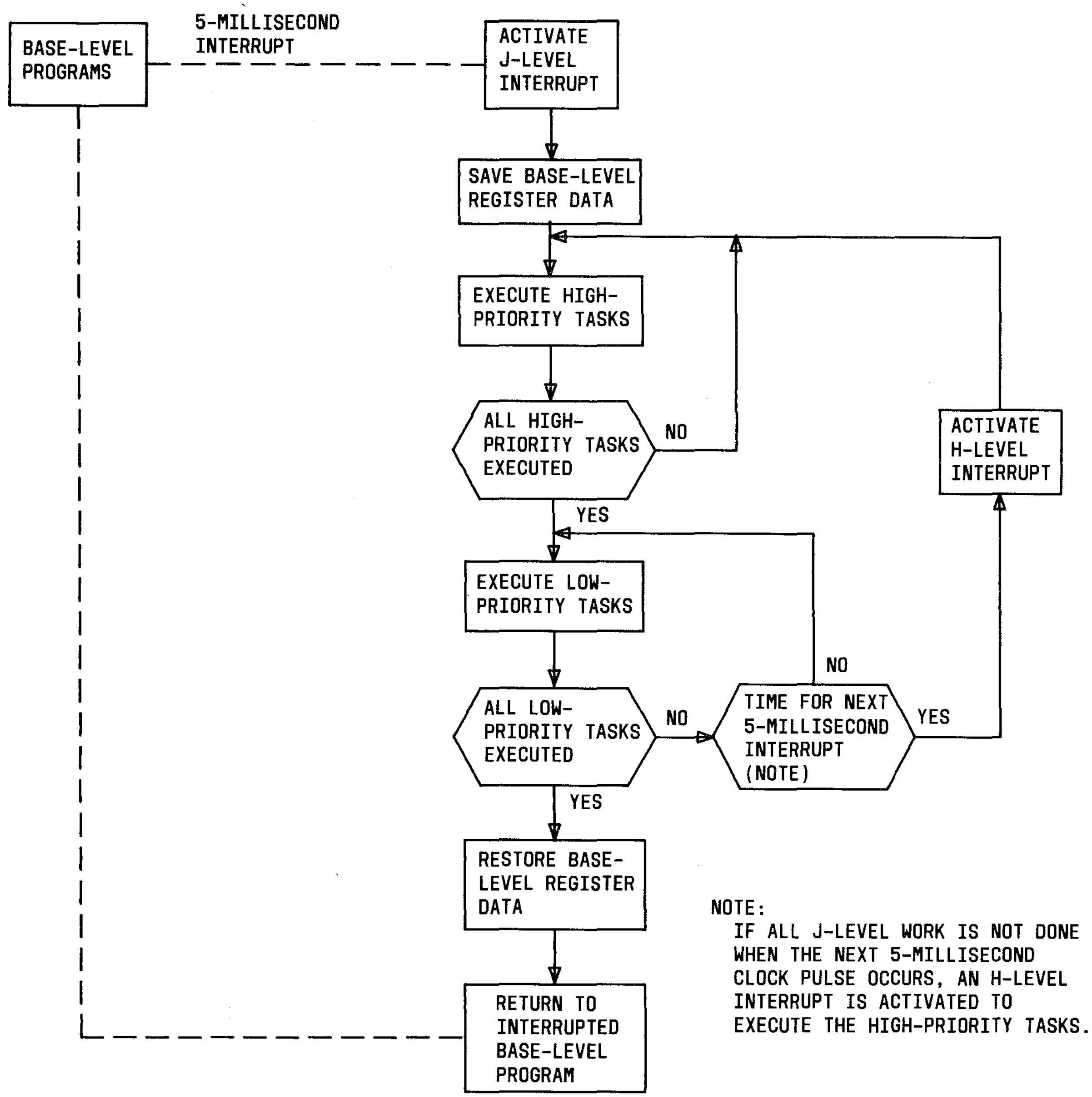


Fig. 8—J- and H-Level Interrupt Processing—Flow Diagram

the H-source flip-flop is set. No interrupt occurs because H level cannot interrupt itself; however, another H-level interrupt will occur immediately at the conclusion of the present H level.

4.25 If a higher-level interrupt occurs while an H- or J-level interrupt is in progress, the interrupt level audit interface program (IREC) is responsible for running any input-output audits necessary for a return to normal processing. A check made to determine the status of the input-output (I/O) tasks at the time the interrupt occurred is accomplished by a transfer to global ECJOBI in ECIO with a return to IREC. If it is necessary to run any input-output audits (eg, trunk scanner audit, junctor scanner audit, POB audit, etc), IREC transfers control to ECIO global ECJ0BA. After the necessary audits have been processed, return is made to IREC. Since the higher-level interrupt may leave indices into the J- and H-level timetables out of range or otherwise invalid, IREC transfers to ECIO global ECINIT for J- and H-level pointer initialization and auditing via transfer to SADAINIT (Section 231-045-215).

5. NO. 1A ESS EXECUTIVE CONTROL MAIN PROGRAM (ECMP)

GENERAL

5.01 The executive control main program (ECMP) controls the execution of all base-level programs. Those receiving control directly from ECMP are primarily the task dispenser programs that unload hoppers, queues, or other buffers; some task programs (eg, traffic measurements, dial tone speed tests); the maintenance control program; and some input-output programs.

5.02 ECMP contains several task dispensers that give control to other programs on appropriate schedules or on request. These are the programs that administer timetables and ordered bits buffers.

5.03 Other task dispensers included in ECMP are responsible for administering timing queues, the miscellaneous interject hopper, the ring trip hopper, and the long job queue.

5.04 In addition, ECMP maintains and uses the system clock and calendar in performing its control functions.

DESCRIPTION OF BASE-LEVEL SCHEDULING

5.05 In the No. 1 ESS, the main program performs its control function on the base level with the use of six priority classes of programs. The highest priority class is called interject. The other five classes are A, B, C, D, and E in descending order by frequency of examination. The main program delivers control to the five frequency classes according to the following pattern which is repeated endlessly:

...ABACABADABACABAEABACABADABACAB...

5.06 The No. 1A ESS has refined and enlarged the design of the base-level programs. The new algorithm incorporates two major concepts. First, the base-level scheduling universe is expanded to include a domain external to the existing five frequency classes. Second, the idea of performing or looking for work only when it is "time to do so" is incorporated.

5.07 The base-level frequency classes are retained and remain internally intact from a functional viewpoint. However, the method of visiting the classes is changed. In general, a given class or job external to the frequency classes is not entered unless its flag has been set via global ECLP01 in ECIO, indicating that it is time to execute that particular class or job.

5.08 EX2CONTROL, a call store word generated in ECIO, contains the outstanding flags of jobs to be performed for a particular entry in ECIOTABL, which is the base-level timetable (see paragraph 4.15 for a detailed description). This word is interrogated routinely on base level at global entry ECMPNXJB (Fig. 9).

5.09 At the end of a MAC_FILL routine maintenance segment, routine ECMPNXJB is entered and EX2CONTROL is checked for flags. If there are none, then, by definition of the algorithm, it is not time to transfer to any base-level classes to look for work or to execute trunk or line scanning or any other ECIOTABL jobs. The maintenance control program continues to be reentered as long as no flags are found. Thus, all available system real time is spent performing maintenance work.

5.10 If one or more flags are found set in EX2CONTROL, in general, the highest priority (rightmost 1) of these is honored and control is passed to the appropriate routine. In the case of the frequency classes A through E (corresponding to jobs AINC, BINC, CINC, DINC, and EINC, respectively), a somewhat different procedure is followed.

A. Frequency Class Flags (ECIOTABL)

5.11 Flags for frequency classes A, B, C, D, and E are set at 10-ms, 20-ms, 40-ms, 50-ms, and 100-ms intervals, respectively (Fig. 7). This establishes the minimum average intervisit time for each class. The method of scheduling classes A through E gradually transforms from a priority into a frequency scheme as traffic increases and available real time decreases. This is accomplished by an indexed transfer which determines the class of job to be performed. The 4-bit index, contained in call store location EX2NEXT, may be thought of as a progression indicator in the following sequence:

...ABACABADABACABAE...

The five descriptions that follow correspond to finding each of the frequency class flags (A through E) as the highest priority flag set (Fig. 10).

Class A Flag Set

5.12 If the A flag is the highest priority flag set (AINC entry), EX2NEXT is read and its value is incremented by one and stored (through a 4-bit mask to provide automatic recycling to 0 when the value of the index is 15). The old contents of EX2NEXT are then used to index into one of 16 slots in table AADR. Five possibilities exist for the value of the index in EX2NEXT. It may point to either an A, B, C, D, or E class execution. The actions taken in these cases are as follows:

(a) If EX2NEXT points to class A execution, a transfer is made to that class (ATBL entry), and all of the class A jobs are executed via transfers to task dispenser programs using the stack return address option. After execution of the last task dispenser program, the A flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. Class A jobs and the associated globals and PIDENTS are listed in Table D. Each frequency class has an ordered bits buffer (OBB) dispenser program which is administered by ECMP; the OBB dispenser program for class A is the fourth entry in Table D. This type of buffer consists of a call store word (E4AOBB for class A), each bit of which serves as a flag for the associated program. The address of each program is in an accompanying transfer table (AJOBS) in a position corresponding to the flag's bit position in the OBB word. Class A ordered bits buffer jobs and their associated globals and PIDENTs are listed in Table E. The ordered bits buffer dispenser program (ECAO00) loads the contents of the ordered bits buffer word (E4AOBB) into register K and zeroes E4AOBB. Then K is ORed with the OBB activity bit word (E4AACT) to determine which programs are active. The result is stored in K and the active programs are executed, using the rightmost 1 technique (paragraph 4.17) to index into the AJOBS table. When all class A ordered bits buffer jobs have been completed, return is made to the next (fifth) entry in the class A jobs table (Table D) and eventually to ECMPNXJB as explained in the first paragraph of this step.

(b) If EX2NEXT points to a class B execution, a check is made to see if it is time to do class B (B flag set). If so, a transfer is made to that class, and all of the class B jobs are executed. At the conclusion of class B, the B flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. If the B flag is not found to be set, class A is executed and, at its conclusion, the A flag in EX2CONTROL is zeroed

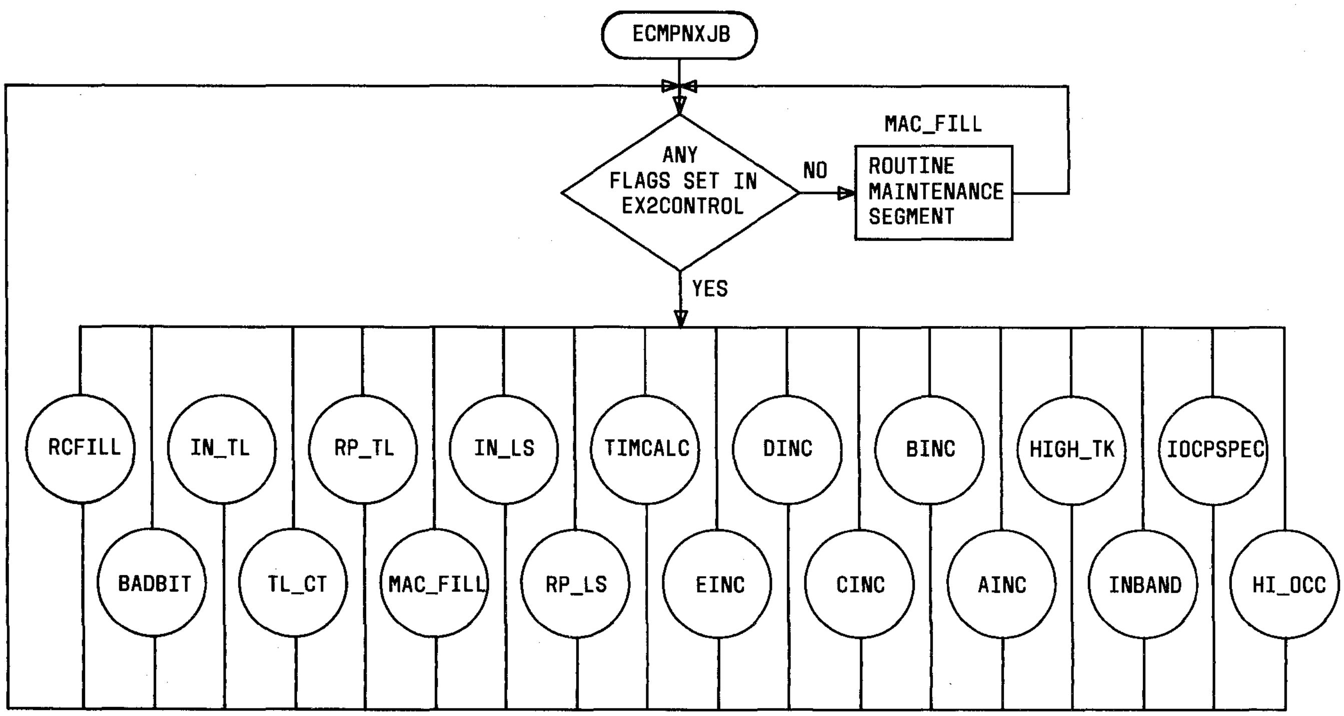


Fig. 9—ECMPNXJB

and control is returned to ECMPNXJB. This latter situation means that although it was class B's turn to be scheduled, it was not time to do the class B jobs.

(c) The actions taken when EX2NEXT points to a class C, D, or E execution are equivalent to those just described for class B.

Class B Flag Set

5.13 If the B flag is the highest priority flag set (BINC entry), then by definition of the scheduling algorithm it is not time to do class A work. Thus the question "Whose turn is it?" is applicable only to classes B, C, D, and E. As such, the upper 3 bits of the 4-bit index in EX2NEXT are used to index into 1 of 8 slots in the BADR table containing the class execution sequence BCBDBCBE (same as the AADR table with all of the class A executions removed). EX2NEXT is now updated by first zeroing the least significant bit, then adding 2, and storing through a 4-bit mask (for automatic recycling). This procedure advances EX2NEXT to the next slot in the BADR table and also points EX2NEXT to a class A execution in the AADR table (guaranteeing that A will be the next class executed if its flag was set prior to a return to ECMPNXJB).

5.14 The actions taken after indexing into the BADR table are similar to those previously described for the AADR table.

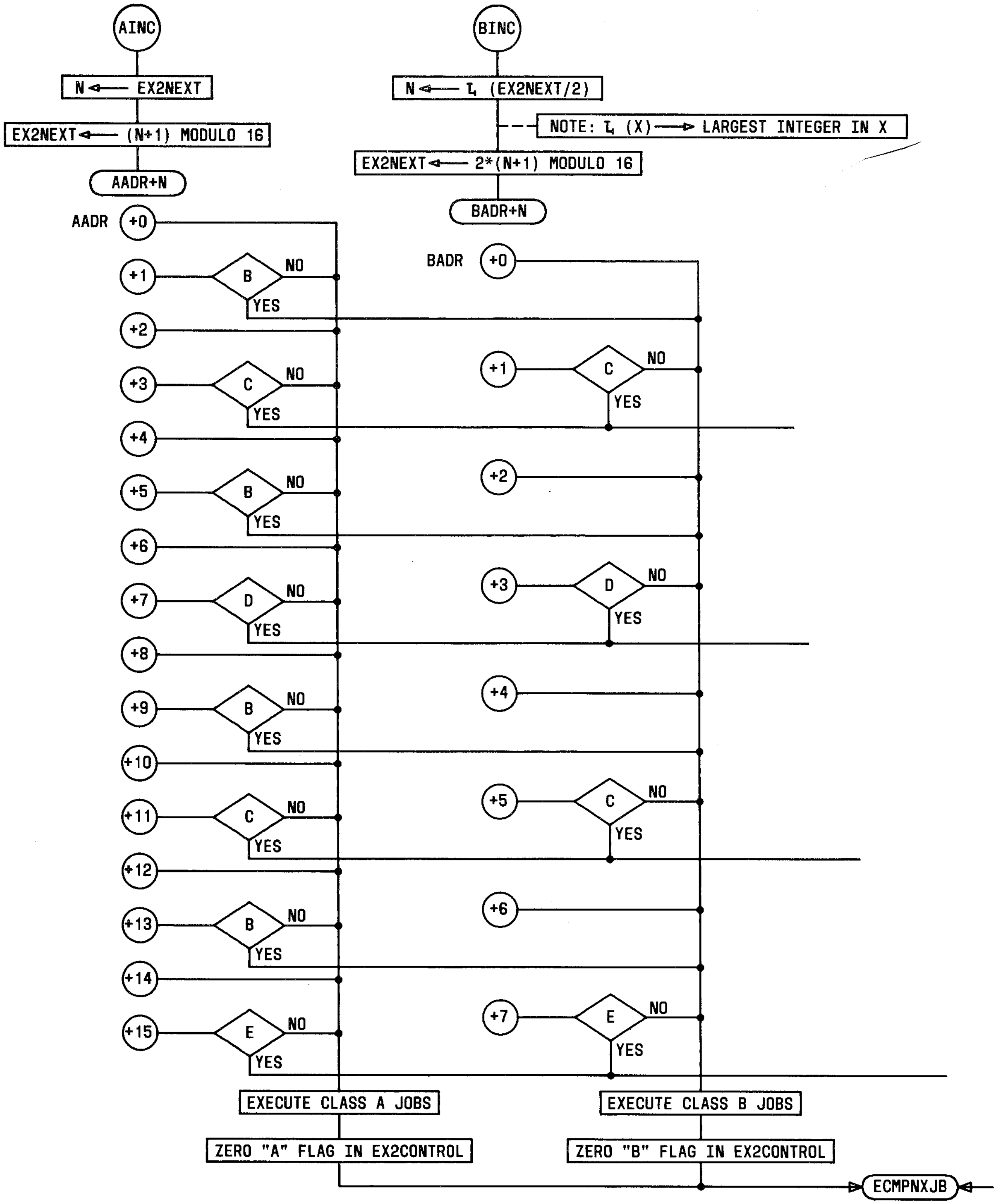
(a) If EX2NEXT points to class B execution, a transfer is made to that class (BTBL entry), and all of the class B jobs are executed via transfers to task dispenser programs using the stack return address option. After execution of the last task dispenser program, the B flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. Class B jobs and the associated globals and PIDENTs are listed in Table F. Note that the first entry in Table F is processed by a task dispenser (ECURTA) within ECMP. The ring trip scan program scans all ringing trunks for answers on local and incoming calls. When a trip is detected, the scan program loads the master scanner number (MSN) corresponding to the answered ringing trunk into the ring trip hopper. ECURTA receives control every class B entry and unloads the ring trip hopper. This routine performs the required MSN translation to find the call register associated with the call. The RI-PT return entry YRIPTS is used to notify the task program of the ring trip. Upon completion of hopper unloading, return is made to continue class B processing. The OBB dispenser program for class B (seventh entry in Table F) is administered by ECMP in the same manner as previously explained for class A (paragraph 5.12). The entry point for execution is ECBO00; the OBB word and the activity bit word are E4BOBB and E4BACT, respectively; and the transfer table is BJOBS. Class B ordered bits buffer jobs and the associated globals and PIDENTs are listed in Table G. An additional OBB dispenser program, the common programs ordered bits buffer (tenth entry in Table F), is administered by ECMP when class B is in control. The entry point for execution is ECCOM00, the OBB word is EX1DOBW1, and the transfer table is COMJOBS. The common programs OBB jobs and the associated globals and PIDENTs are listed in Table H. These jobs perform 1A processor work.

(b) If EX2NEXT points to a class C execution, a check is made to see if it is time to do class C (C flag set). If so, a transfer is made to that class, and all of the class C jobs are executed. At the conclusion of class C, the C flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. If the C flag is not found to be set, class B is executed and at its conclusion the B flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. This latter situation means that although it was class C's turn to be scheduled, it was not time to do the class C jobs.

(c) The actions taken when EX2NEXT points to a class D or E execution are equivalent to those just described for class C.

Class C Flag Set

5.15 If the C flag is the highest priority flag set (CINC entry), then by definition of the scheduling algorithm it is not time to do either class A or class B work. Thus the question "Whose turn is it?" is applicable only to classes C, D, and E. As such, the upper 2 bits of the 4-bit index in EX2NEXT are used to index into one of four slots in the CADR table containing the class execution sequence CDCE (same as the BADR table with all of the class B executions removed). EX2NEXT is now updated by first zeroing the 2 least significant



AINC
N ← EX2NEXT
EX2NEXT ← (N+1) MODULO 16
AADR+N
AADR
+0
+1
+2
+3
+4
+5
+6
+7
+8
+9
+10
+11
+12
+13
+14
+15
B
C
D
E
NO
YES
EXECUTE CLASS A JOBS
ZERO "A" FLAG IN EX2CONTROL
BINC
N ← Ⅰ (EX2NEXT/2)
NOTE: Ⅰ (X) → LARGEST INTEGER IN X
EX2NEXT ← 2*(N+1) MODULO 16
BADR+N
BADR
EXECUTE CLASS B JOBS
ZERO "B" FLAG IN EX2CONTROL
ECMPNXJB

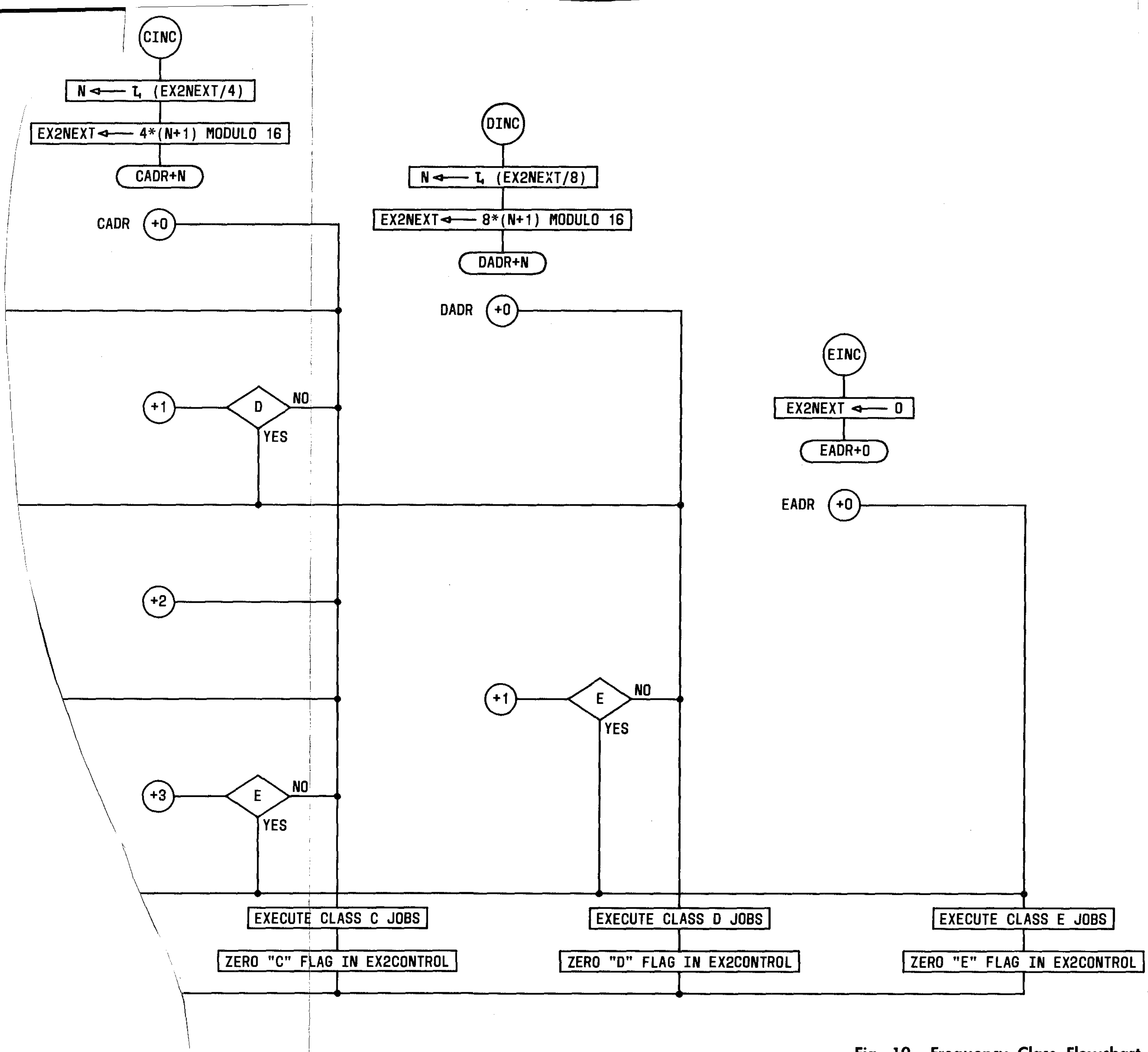


Fig. 10—Frequency Class Flowchart

TABLE D

CLASS A JOBS (ATBL)

JOB NAME	GLOBAL/LOCAL	PIDENT
High-Priority POB Result Hopper	PQPOBH	QEPR1A00
Maintenance POB Result Hopper	PQPOBM	NMFL1A00
Unexpected POB Result Hopper	PQPOBU	QEPR1A00
Class A Ordered Bits Buffer	ECAO00	ECMP1A00
TOUCH-TONE® Digit Hopper	ORUNTT	ORDL1A00
Multifrequency Digit Hopper	ICUNMF	ICAL1A00
Incoming Revertive Digit Hopper	ICUREV	ICRV1A00
Incoming SXS Reorder AIT Hopper	ISXSUH	ISXS1A00

Note: Refer to PK-6A001 for data in other generics.

TABLE E

CLASS A ORDERED BITS BUFFER JOBS (AJOBS)

JOB NAME	GLOBAL/LOCAL	PIDENT
POB Queue	WQUPOB	WQUE1A00
Class-of-Service Tone-1 Queue	WQUCL1	WQUE1A00
Class-of-Service Tone-2 Queue	WQUCL2	WQUE1A00
200-ms Timing for Special Ringing Register	ECSRTI	ECMP1A00

Note: Refer to PK-6A001 for data in other generics.

bits, then adding 4, and storing through a 4-bit mask (for automatic recycling). This procedure advances EX2NEXT to the next slot in the CADR table, points EX2NEXT to a class B execution in the BADR table, and also points EX2NEXT to a class A execution in the AADR table. This gives classes A and/or B the first chance at being executed should either or both of their flags get set prior to returning to ECMPNXJB.

5.16 The actions taken after indexing into the CADR table are similar to those described for the AADR and BADR tables.

(a) If EX2NEXT points to class C execution, a transfer is made to that class (CTBL entry), and all of the class C jobs are executed via transfers to task dispenser programs using the stack return address option. After execution of the last task dispenser program, the C flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. Class C jobs and the associated globals and PIDENTs are listed in Table I. Note that the third entry in Table I is processed by a task dispenser (ECLONG) within ECMP. A program doing extensive processing with a call register can take a real-time break by placing

TABLE F

CLASS B JOBS (BTBL)

JOB NAME	GLOBAL/LOCAL	PIDENT
Ring Trip Hopper	ECURTA	ECMP1A00
Remove Dial Tone Hopper	ORURDT	ORDL1A00
Customer Dial Pulse Receiver Digit Hopper	ORUCDP	ORDL1A00
Outgoing Revertive Digit Hopper	OGRHOP	OGRV1A00
Outgoing Trunk Dial Pulse Digit Hopper	DPHPUL	DPOP1A00
Incoming Trunk Dial Pulse Digit Hopper	INCTDP	ICAL1A00
Class B Ordered Bites Buffer	ECBO00	ECMP1A00
Hit Scan Result Hopper	DIHSRH	CHGD1A00
Miscellaneous Scan Result Hopper	DITSJH	YMRG1A00
Common Programs Ordered Bits Buffer	ECCOM00	ECMP1A00

Note: Refer to PK-6A001 for data in other generics.

the register in the buffer called the long job queue via ECMP globals ECLJLD and ECLJLL. ECLONG receives control every class C entry and removes the registers from the queue, using the RI-PT return entry to give control to the programs that requested the buffering. When all entries have been processed, return is made to continue class C processing. The OBB dispenser program for class C (second entry in Table I) is administered by ECMP in the same manner as previously explained for class A (paragraph 5.12). The entry point for execution is ECCO00; the OBB word and the activity bit word are E4COBB and E4CACT, respectively; and the transfer table is CJOBS. Class C ordered bits buffer jobs and the associated globals and PIDENTs are listed in Table J.

(b) If EX2NEXT points to a class D execution, a check is made to see if it is time to do class D (D flag set). If so, a transfer is made to that class, and all of the class D jobs are executed. At the conclusion of class D, the D flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. If the D flag is not found to be set, class C is executed and at its conclusion the C flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. This latter situation means that although it was class D's turn to be scheduled, it was not time to do the class D jobs.

(c) The actions taken when EX2NEXT points to a class E execution are equivalent to those just described for class D.

Class D Flag Set

5.17 If the D flag is the highest priority flag set (DINC entry), then by definition of the scheduling algorithm it is not time to do either classes A, B, or C and thus the question "Whose turn is it?" is applicable only to classes D and E. As such, the most significant bit of the 4-bit index in EX2NEXT is used to indicate class D's turn (bit equals 0) or class E's turn (bit equals 1). EX2NEXT is updated by first zeroing the 3 least significant bits, then adding 8, and storing through a 4-bit mask (for automatic recycling). This procedure toggles the most significant bit of the index and also points EX2NEXT to class A in the AADR table, class B in the BADR table, and class C in the CADR table. Thus, these three classes will all have a chance at execution before another

TABLE G

CLASS B ORDERED BITS BUFFER JOBS (BJOBS)

JOB NAME	GLOBAL/LOCAL	PIDENT
Regular Ringing Register Queue	WQURRR	WQUE1A00
Special Ringing Register Queue	WQURRS	WQUE1A00
Coin-Control Register Queue	WQUCNR	WQUE1A00
Coin-Zone Queue	WQUCN8	WQUE1A00
Local Coin-Operator Queue	WQUCN9	WQUE1A00
Recent Change Message Update (General Entry)	RCMUGENT	RCMU0000
Carrier Group Alarm	CGSEGM	CGTB1A00
CAMA Register Port 0 Queue	WQULCR	WQUE1A00
PROCON RAM Pumpup Request	CXMSPU	CXMS1A00
Timer Check and Counter Initialization	AO5SEC	AOVD1A00
100-ms 1-Way Call Register Timing	YPOHMS	ECMP1A00
200-ms 2-Way Call Register Timing	YPOTM3	ECMP1A00

Note: Refer to PK-6A001 for data in other generics.

TABLE H

COMMON PROGRAMS ORDERED BITS BUFFER JOBS (COMJOBS)

JOB NAME	GLOBAL	PIDENT
Plant Measurements (Initial Entry)	PLNTCAMO	PLNTCMMN
Plant Measurements (Repeated Entry)	PLNTSAMO	PLNTCMMN
Base Level Return to Library Client	LIBRLDMP	LIBRTRP1
Partitioned Message Returns Mechanism	IOCPPCLI	IOCPTIM1
Requests for ADS Units	DUADMPDC	DUAD02
Output Messages for Data Unit Administration	DUADTTYO	DUAD04
Error Analysis Buffer Unloading	ERIFBFUD	ERIF
Output Message Starter	IOCPOMS1	IOCPOMS1

Note: Refer to PK-6A001 for data in other generics.

D or E, upon subsequent setting of any of their flags in EX2CONTROL.

5.18 The actions taken after determining class D's or class E's turn from EX2NEXT are similar to those described for the AADR, BADR, and CADR tables.

(a) If EX2NEXT points to class D execution, a transfer is made to that class (DTBL entry), and all of the class D jobs are executed via transfers to task dispenser programs using the stack return address option. After execution of the last task dispenser program, the D flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. Class D jobs and the associated globals and PIDENTs are listed in Table K. The OBB dispenser program for class D (first entry in Table K) is administered by ECMP in the same manner as previously explained for class A (paragraph 5.12). The entry point for execution is ECDO00; the OBB word and the activity bit word are E4DOBB and E4DACT, respectively; and the transfer table is DJOBS. Class D ordered bits buffer jobs and the associated globals and PIDENTs are listed in Table L.

(b) If EX2NEXT indicates a class E execution, a check is made to see if it is time to do class E (E flag set). If so, a transfer is made to that class, and all of the class E jobs are executed. At the conclusion of class E, the E flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB.

If the E flag is not found to be set, class D is executed and at its conclusion the D flag in EX2CONTROL is zeroed and control is returned to ECMPNXJB. This latter situation means that although it was class E's turn to be scheduled, it was not time to do the class E jobs.

5.19 If the E flag is the highest priority flag set (EINC entry), then by definition of the scheduling algorithm it is not time to do any of the other classes. As such, the index in EX2NEXT is simply recycled to 0, and class E jobs (ETBL entry) are executed via transfers to task dispenser programs using the stack return address option. After execution of the last task dispenser program, the E flag in EX2CONTROL is zeroed. Before returning control to ECMPNXJB, the interject activity bit is set and transfer is made to the ECMP interject program via global ECMPSOFT to ensure that interject work is executed at least once per base-level cycle, even if the interject request flag is erased by some error. Class E jobs and the associated globals and PIDENTs are listed in Table M.

5.20 The OBB dispenser program for class E (second entry in Table M) is administered by ECMP in the same manner as previously explained for class A (paragraph 5.12). The entry point for

TABLE I

CLASS C JOBS (CTBL)

JOB NAME	GLOBAL/LOCAL	PIDENT
Low-Priority POB Result Hopper	PQPOBL	QEPR1A00
Class C Orderd Bits Buffer	ECCO00	ECMP1A00
Long Job Queue	ECLONG	ECMP1A00
Outgoing MF Tone-List Report Hopper	MFTLRH	MFTL1A00
Outgoing Start Pulse-Detection Hopper	OGOUSP	OGTC1A00
Centrex Key Report Hopper	CXKEYH	CXKY1A00
Trunk Seizure and Answer Hopper	AOUTKS	AOVD1A00

Note: Refer to PK-6A001 for data in other generics.

TABLE J

CLASS C ORDERED BITS BUFFER JOBS (CJOBS)

JOB NAME	GLOBAL/LOCAL	PIDENT
Master Control Center Lamp Display	MCSLMD	MCCP1A00
EADAS Character Analysis	EDCHAR	EDAS1A00
PBX Relays Release Timing	PSTM01	PSPD1A00
Compatibility Check	PGM145	ECMP1A00
Network Camp-on Feature	NMFCON	NMFA1A00
TTY Work Register	TWRCON	TTWK1A00
TTY Overload Messages	AOTTY	AOVD1A00
Blocked Dial Tone Queue	WQUBDT	WQUE1A00
Outgoing Trunk Timing	OG4SEC	OGTC1A00
Maintenance Control Alarm Scan	MACP30	MACR1A00
Attendant Lamp Update	MILAMP	QSIF1A00
Ringing Current Scan	RRCNT6	RING1A00
Outgoing Revertive Digit Start-Pulsing Scan	OGRTIM	OGRV1A00
Incoming SXS PSPD Scan	PSXSIN	PSXS1A00
1-Second General-Purpose Noncall Register Timing	YPAB00	ECMP1A00
Auxiliary 500-ms Timetable	EC500	ECMP1A00

Note: Refer to PK-6A001 for data in other generics.

TABLE K

CLASS D JOBS (DTBL)

JOB NAME	GLOBAL/LOCAL	PIDENT
Class D Ordered Bits Buffer	ECDO00	ECMP1A00
Line Ferrod Scan Hopper	OGUNLH	CHGD1A00

Note: Refer to PK-6A001 for data in other generics.

TABLE L

CLASS D ORDERED BITS BUFFER JOBS (DJOBS)

JOB NAME	GLOBAL/LOCAL	PIDENT
6-Second General-Purpose Noncall Register Timing	YPAC00	ECMP1A00
Customer TOUCH-TONE Receiver Queue	WQUDR0	WQUE1A00
Customer Dial Pulse Receiver Queue	WQUDR1	WQUE1A00
Multifrequency Receiver Queue	WQUDR2	WQUE1A00
Revertive Pulse Receiver Queue	WQUDR3	WQUE1A00
AMA Completed Register Queue	AMQADE	AMAC1A00
Trunk Dial Pulse Receiver Queue	WQUDR4	WQUE1A00
Trunk TOUCH-TONE Receiver Queue	WQUTTT	WQUE1A00
Call Distributor Queue	MIUNLQ	QAPR1A00
Trunk Digit Receiver Queue	WQMFC4	WQUE1A00
Recent Change Processing (Main Entry)	RCFLGD	RSUB1A00
Auxiliary Trunk Maintenance List Processing	TNTMLA	TNKC1A00
Line Ferrod Scan	OGS200	OGLF1A00
Customer TOUCH-TONE Receiver Queue	WQUDR5	WQUE1A00
HL4W Multifrequency Queue	WQUHLM	WQUE1A00
Trunk Digit Receiver Queue	WQMFC2	WQUE1A00
Trunk-to-Trunk Register Queue	WQUTTM	WQUE1A00
HL4W Dial Pulse Transmitter Queue	WQUHLD	WQUE1A00
HL4W TOUCH-TONE Transmitter Queue	WQUHLT	WQUE1A00

Note: Refer to PK-6A001 for data in other generics.

execution is ECEO00; the OBB word and the activity bit word are E4EOBB and E4EACT, respectively; and the transfer table is EJOBS. Class E ordered bits buffer jobs and the associated globals and PIDENTs are listed in Table N.

5.21 When class E ordered bits buffer jobs have been completed, a chain of transfers is begun to five additional task dispensers, all of which are administered by ECMP:

EO01—This dispenser, which is responsible for the execution of 3-second jobs, is entered when ECEO00 has completed the class E ordered bits buffer jobs. Routine EC3SEC in ECMP uses the 3-second timetable pointer (E4NCNT) to select the proper entry from the timetable (N_3_SEC). This entry is ANDed with the activity bit word (E4YACT) to derive the ordered bits buffer word (E4NOBB)used by EO01 to index into the associated transfer table (NJOBS) via the rightmost 1 technique.

EO02—This dispenser, which is responsible for unloading the fixed quarter-hour ordered bits buffer jobs, is entered when EO01 has completed the 3-second jobs. Routine ADD in ECMP selects the proper entry from the

15-minute timetable (F_15_MIN), ANDs this entry with the proper entry from the 1-hour timetable (F_1_HR), and ANDs the result with the activity bit word (E4PACT) to derive the ordered bits buffer word (E4FQHW) used by EO02 to index into the associated transfer table (FJOBS) via the rightmost 1 technique.

EO05—This dispenser, which is responsible for unloading the variable quarter-hour ordered bits buffer jobs, is entered when EO02 has completed the fixed quarter-hour ordered bits buffer (OBB) jobs. Routine ADD in ECMP selects the proper entry from the associated day table (E4VSUN) for variable quarter-hour OBB jobs, ANDs this entry with the proper entry from the associated hour table (E4VMID), ANDs the result with the proper entry from the associated quarter-hour table (E4VQ00), and then ANDs the latter result with the activity bit word (E4QACT) to derive the OBB word (E4VQHW) used by EO05 to index into the associated transfer table (VJOBS) via the rightmost 1 technique.

EO06—This dispenser, which is responsible for unloading the second variable quarter-hour OBB jobs, is entered when EO05 has completed the variable quarter-hour OBB jobs. Routine ADD in ECMP selects the proper entry from the associated day table (E4WDAY) for the second variable quarter-hour OBB jobs, ANDs this entry with the proper entry from the associated hour table (E4WHR), ANDs the result with the proper entry from the associated quarter-hour table (E4W15M), and then ANDs the latter result with the activity bit word (E4WACT) to derive the OBB word (E4WOBB) used by EO06 to index into the associated transfer table (WJOBS) via the rightmost 1 technique.

ECEO08—This dispenser, which is responsible for unloading the deferrable class E OBB jobs, is entered when EO06 has completed the second variable quarter-hour OBB jobs. The contents of the OBB word (E4GOBB) is loaded into register K and E4GOBB is zeroed. The active programs flagged in K are executed, using the rightmost 1 technique to index into the associated transfer table (GJOBS).

When the chain of activity initiated by the execution of class E ordered bits buffer jobs has been completed, return is made to the third entry in the class E jobs table (Table M) and eventually to ECMPNXJB as explained in paragraph 5.19.

B. Interject Administration

5.22 One other frequency class exists on base level, namely, interject. A check for interject work is made at the completion of the processing of ***every*** hopper, queue, or timing list entry. If any interject work is found, all of it is performed before returning to the task dispenser presently in control. Thus, interject is at the top of the base-level frequency class hierarchy (Fig. 1) since interject may be executed many times during the execution of any other frequency class. When transfer is made to the interject class, the stack option is used to record the appropriate return address.

5.23 Interject-level programs will not allow another interject while they are processing. These programs may check for interject only if they use the stack mechanism to make the check. The interject-in-progress flip-flop will inhibit the interject check in the stack mechanism whenever an interject is in progress. The interject sources will be checked before control is returned to the base-level programs.

5.24 When an interject is detected, a fixed transfer of program control is made to the SIRE program unit associated with the particular interject level. There are four such interject levels (Section 231-310-300), one of which covers software-initiated interjects and is administered by ECMP. SIRE action for software-initiated interjects consists of transferring program control via PATT to global ECMPSOFT in ECMP. ECMPSOFT loads the flag word (E4IID) for interject jobs into register K, and the active jobs are executed using the rightmost 1 technique to index into the associated transfer table (IJOBS). When all active interject jobs have been completed, transfer is made to ECMPEXIT where the interject activity bit is reset (turned off) and a check is made for interject with return being made via the stack to the base-level program which was in execution at the time interject processing began. ECMPSOFT is also entered following each execution of class E jobs (see paragraph 5.19). Software-initiated interject jobs and the associated globals and PIDENTs are listed in Table O.

TABLE M

CLASS E JOBS (ETBL)

JOB NAME	GLOBAL/LOCAL	PIDENT
Completed Disk Request Task Dispenser	DKADANS	DKAD
Class E Ordered Bits Buffer	ECEO00	ECMP1A00
TTY Output Message Translation	IOCPXLAT	IOCPOMT1
TTY Input Message Translation	IOCPITRC	IOCPIMT1
Interpage Computation	PAGSEXNT	PAGSUPER
Supervisory Line Scan	SUSC_LI	CCLT0000
Routine Maintenance	MACRMAIN	MACAADMN
Cutover Lines Audit	SAELEV	SACT1A00

Note: Refer to PK-6A001 for data in other generics.

5.25 Two of the software-initiated interject routines (ECMINT and ECTK00) to which ECMPSOFT transfers are administered by ECMP. ECMINT, which is the second entry in Table O, unloads the miscellaneous interject hopper. An interject request for a particular task is loaded into the miscellaneous interject hopper by ECMP at global ECMT00 where the interject source bit is set and a two-word entry is made in the hopper. One word contains the data (contents of the X register) and the other, the address to be entered in interject. When ECMINT is entered via a transfer from ECMPSOFT, hopper slots are examined until one is found that contains an entry. At this point, the data word of the hopper entry is saved in the X register and transfer is made to the task program whose address was loaded in the hopper entry. Upon successful completion of the task, return is made to ECMINT to process the next hopper entry. When all hopper entries have been unloaded, return is made to ECMPSOFT via the stack return option.

5.26 ECTK00, which is the fourth entry in Table O, is flagged every 100 ms by ECIO. This 100-ms routine adds 100 ms to the system clock (E4KTOD) and administers a 100-ms timetable (K_100_MS) and the 100-ms noncall register timing list. Return is made to ECMPSOFT via the stack return option.

5.27 After completion of a task, many call processing programs transfer to ECMP via global entry RETURN. This routine checks for interject, with return being made via the stack to the task dispenser whose work was delayed because of interject work.

5.28 It is possible for an interject program to receive control at any real-time break. Any program which at some point expects to consume more than 3.5 ms between real-time breaks can take a real-time break to permit the execution of interject work. Transfer is made to ECMP at global ECBRAK where the interject sources are checked for an interject request. If interject work is waiting, transfer is made to the proper routine for processing the particular interject source that was flagged. Return is made to local REBK in ECBRAK and, subsequently, to the program that requested the break.

C. Supervisory Trunk and Line Scanning Flags (ECIOTABL)

5.29 The increased number of calls the No. 1A ESS can handle requires that a significant portion of real time must be spent on supervisory scanning of the lines and trunks. In order to make more time available for processing calls, trunk and line scan rates are decreased as traffic

TABLE N

CLASS E ORDERED BITS BUFFER JOBS (EJOBS)

JOB NAME	GLOBAL/LOCAL	PIDENT
TC18 MESSAGE: TMC AND EGO SEARCH	HCSRCH	VFHC1A00
Segment Collection of ACD Counts (100 Sec)	CTRETN	CTRF1A00
5-Minute Administration of Data for EADAS	TF5MIN	TFCL1A00
100-Second Weekly Traffic Segment	TCREMP	TFCL1A00
Network Management Recent Change Insertion	NMCBRC	NMGT1A00
Plant Measurement Segment (100-Sec Entry)	PPHSEG	PPMP1A00
Multifrequency Outpulse Customer Traffic Data	CTOUTP	CTOP1A00
Segment Collection of ACD Counts (15 Min)	CTBRAK	CTRF1A00
Update ACD2 B-Hive/Agent CW Lamps	QCSDLO	QCDL1A00
Verfication of Translation Data	TRVERI	TVMN1A00
Recent Change Processing (Auxiliary Entry)	RCMUFENT	RCMU0000
Centrex Maintenance Control	CXQUCL	CXMC1A00
10-Second Periodic Timing	ECRT00	ECMP1A00
Print Counts for Customer Traffic Data	CTRFPT	CTRF1A00
100-Second Traffic Segment	TCSEGM	TFCL1A00
Idle Trunk for Fabric Restore	NMFIDT	NMFA1A00

Note: Refer to PK-6A001 for data in other generics.

increases. This is accomplished by performing these scans on base level. All line scanning and nearly all trunk scanning functions are performed on base level, external to the A, B, C, D, and E frequency classes. There are three basic reasons for this:

(a) By setting scanning flags at lower priorities than the A, B, C, D, and E flags, scanning rates will start their decrease at traffic levels at which other call processing work is unaffected.

(b) The scanning jobs performed when honoring these low-priority flags effectively become time-filler work and, as such, minimize the main program cycle time variance. This is because, by definition of the algorithm, this low-priority scanning work is performed only when it is "not time" to do any of the frequency classes (all the A to E flags are zero).

(c) In the case of trunk scanning, by setting flags which are higher in priority than the frequency class flags, a minimum scanning rate may be maintained, independent of the traffic level. As a point of information, it should be mentioned that provisions have been made to also allow the execution of trunk scanning on J level. This provides for a minimum scanning rate during those periods of time that long audit segments are running on base level. A minimum rate is not as critical for lines as trunks, and thus a relative minimum rate for scanning lines is accomplished by performing some line scanning work in class E, whose maximum intervisit time will be controlled in overload situations.

TABLE O

INTERJECT JOBS (IJOBS)

JOB NAME	GLOBAL/LOCAL	PIDENT
AMA Midnight Update	AMNGHT	AMAC1A00
Miscellaneous Interject Hopper	ECMINT	ECMP1A00
Interject Priority POB Return	PQPOBI	QEPR1A00
Clock Adjustment and K-Flag Processing	ECTK00	ECMP1A00
Hopper Overflow	AOINJE	AOVD1A00
Centrex Lamp Order Queue	CNLPSQ	CNLP1A00
Long Segment Audit	SALGSM	SADT1A00
EA Interject Sanity Test	EAINTJ	MIRVRECV

Note: Refer to PK-6A001 for data in other generics.

Supervisory Trunk Scanning

5.30 In ECIOTABL (Fig. 7), three columns are used to set flags in EX2CONTROL for supervisory trunk scanning. Columns 2 and 3, labeled INBAND and HIGH_TK, are used to maintain a 50-ms scan rate for inband operator trunks and a 200-ms rate for all other supervisory trunk scanning. Column 15, labeled IN_TL, provides for additional trunk scan entries which effect a maximum of a 100-ms trunk scan rate in light traffic.

5.31 The INBAND flag is set every 50 ms. The HIGH_TK flag is set four times every 50 ms, evenly distributed between successive settings of the INBAND flag. The supervisory scan and audit program (CCLT) is responsible for segmenting trunk scanning work so that approximately 1/20 of the work will be done per entrance at global SUSC_TS. CCLT also has a routine (SUSC_IO) which is entered from ECMP every 10 ms to scan a segment of inband/toll scanners. (See Section 231-045-110.) Thus, to establish the minimum trunk scan rate, both SUSC_TS and SUSC_IO must be entered 20 times per 200 ms. Each time ECMP global ECMPNXJB finds the HIGH_TK flag set, a counter (CHITK) is incremented; the HIGH_TK flag (HITK) is zeroed; transfer is made to both SUSC_IO and SUSC_TS (each performs a segment of scan work), and return is made to ECMPNXJB (Fig. 11). Each time the INBAND flag is found set, counter CHITK is incremented. If the value of counter CHITK is less than five, transfer is made to both SUSC_IO and SUSC_TS with a return to ECMPNXJB. If CHITK equals five, CHITK is reinitialized, the INBAND flag (IBND) is zeroed, the long C timer (LONGC) is incremented, and a check is made to determine if C-to-C intervisit time has exceeded 500 ms. If not, the high-priority extra limit (EXLIM) is incremented by the amount of the overload control value (OV2CONTROL, paragraph 5.73) before transfer is made to both SUSC_IO and SUSC_TS with a return to ECMPNXJB.

5.32 During normal periods of operation, SUSC_TS will get four entries every 50 ms from the HIGH_TK flag plus one entry every 50 ms from the INBAND flag. However, in a situation where a single frequency class execution exceeds 10 ms, the HIGH_TK flag could get set on top of itself, thus losing one entry to SUSC_TS. Checking the counter (CHITK) at the INBAND entry solves this problem, since from one to five entrances will be made to SUSC_TS depending upon how many HIGH_TK entrances were missed in the last 50 ms. Thus, normally, this scheme not only distributes five scanning segments evenly over a 50-ms interval but also provides a "catching-up" mechanism once every 50 ms to allow for abnormally long single frequency class executions.

5.33 The IN_TL flag is set twice every 200 ms and can provide up to 20 additional entries to SUSC_TS per 200 ms, depending upon the traffic level, and up to 4 line scan entries per 200 ms. When the IN_TL flag is found set (Fig. 12), it is zeroed and bit positions 13 and 14 (corresponding to the blank columns labeled RP_TL and TL_CT in ECIOTABL Fig. 7) are set in EX2CONTROL. Routine RP_TL is entered, and a counter (CRPTL) is incremented and checked. If the value of CRPTL is less than six, transfer is made to SUSC_TS via transfer to SUONLY. If CRPTL is equal to six, it is reinitialized, the RP_TL flag is zeroed, transfer is made to the supervisory line scan program (SUSC_LI) if the overload-per-second traffic limit (OV2LIMIT, paragraph 5.73) is not negative, and return is made to ECMPNXJB. Thus, setting the RP_TL flag results in the execution of five trunk scan segments and one line scan segment.

5.34 When ECMPNXJB finds the TL_CT flag set, the RP_TL flag is set and TL_CT is zeroed. Transfer is made to routine RP_TL (paragraph 5.33). Thus, setting the IN_TL flag twice ultimately results in the execution of 20 trunk scan segments and 4 line scan segments. If this scanning work is completed in less than 200 ms, then trunk scanning is being performed at a 100-ms rate (a total of 40 entries to SUSC_TS per 200 ms). If completion of this work requires more than 200 ms because of time spent in higher priority work, then the trunk scan rate will have decreased to a point somewhere between 100 ms and 200 ms.

5.35 There are two reasons for administering the low-priority scanning flags in the manner just described. First, both the trunk scan and line scan are involved so that both rates decrease simultaneously as the point of system capacity is reached. System capacity is defined as the point where trunks are scanned at a 200-ms rate and lines at a 300-ms rate. Secondly, having more than one flag minimizes the probability of performing any MAC_FILL routine maintenance during a period of time when trunk and line scanning are not being performed at the maximum rates. This is because after honoring and zeroing the IN_TL flag (set twice every 200 ms), the 20 trunk and 4 line scan segments executed from the RP_TL and TL_CT flags must be completed in less than 200 ms or else the IN_TL flag will be set again. If a single flag (IN_TL) were used and zeroed at the completion of all 24 segments, the IN_TL flag might not be set again for up to 200 ms depending on the exact point of time of completion. This could allow

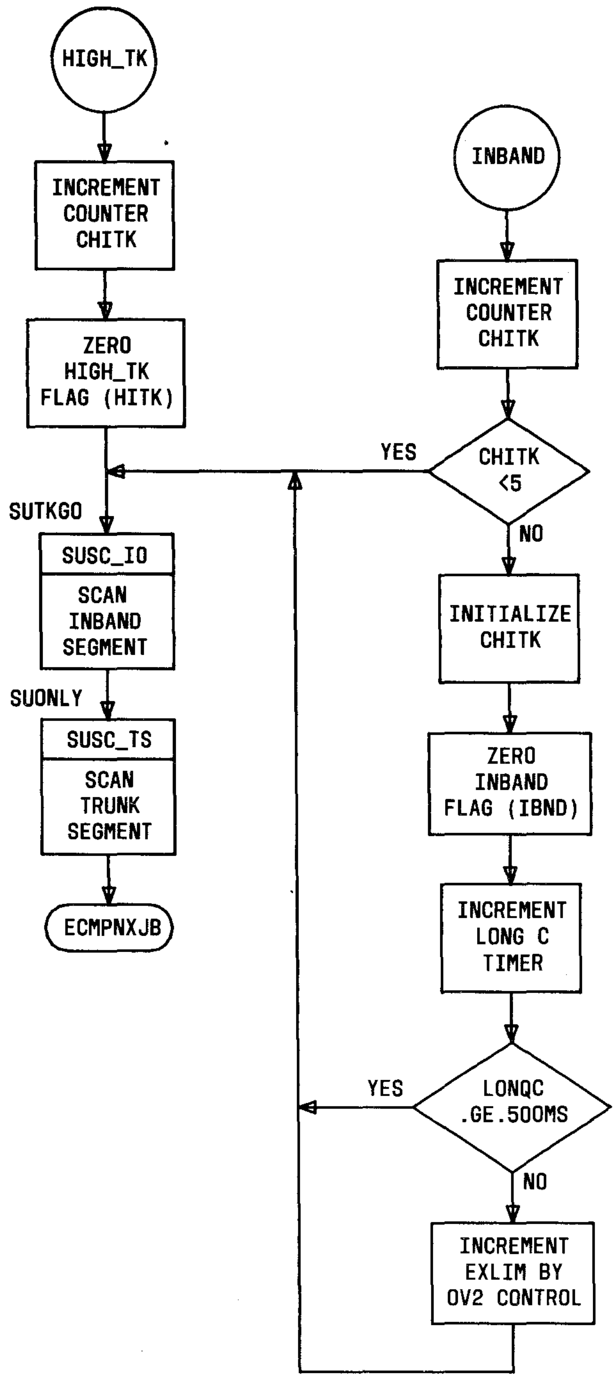


Fig. 11—Supervisory Trunk Scanning (High-Priority Flags)—Flow Diagram

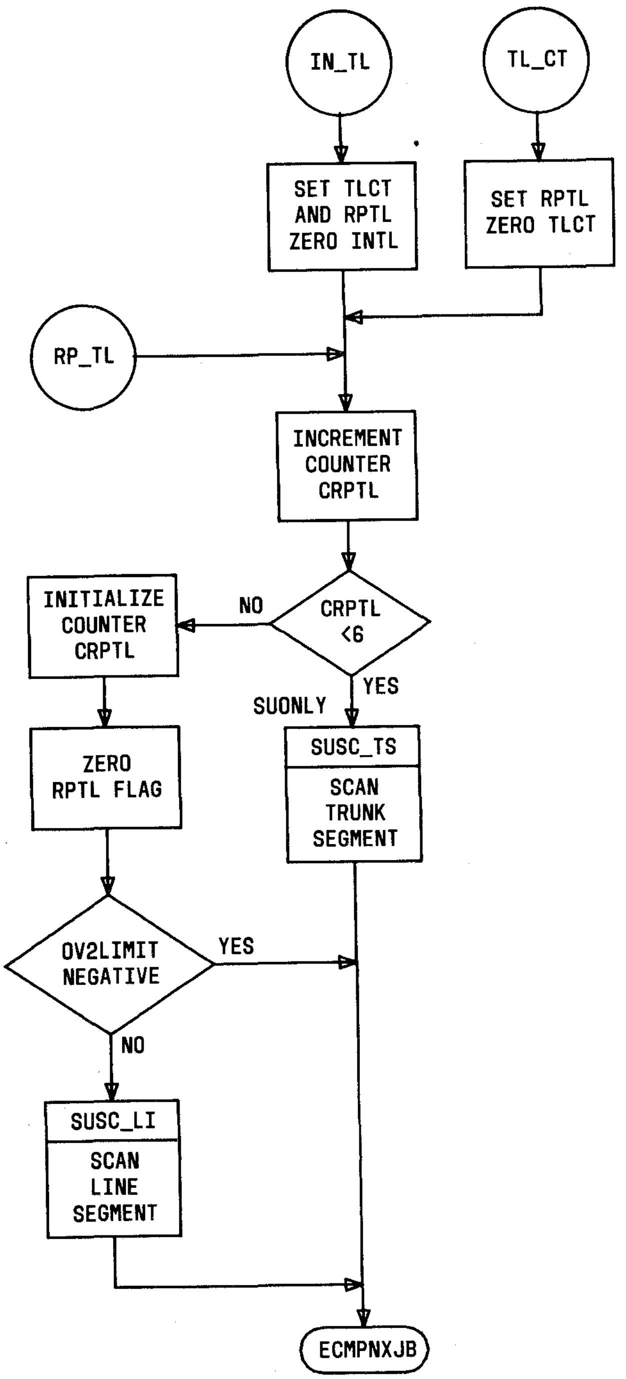


Fig. 12—Supervisory Trunk and Line Scanning (Low-Priority Flags)—Flow Diagram

time-fill routine maintenance to be performed even though the scans are not being done at the maximum rates.

Supervisory Line Scanning

5.36 Three columns in ECIOTABL (Fig. 7) are used to set flags in EX2CONTROL for entrances to supervisory line scanning. Column 8, the class E flag, is used to maintain a nominal minimum 1-second rate. Column 10, labeled IN_LS, provides for additional line scan entries as traffic warrants. Column 15, labeled IN_TL (paragraph 5.33), provides further line scan and trunk scan entries to allow for maximum scanning rates. No line scanning is done in class D as in No. 1 ESS.

5.37 The supervisory scan program (CCLT) is responsible for segmenting line scanning work so that 1/10 of the work will be done per entrance at global SUSC_LI. To establish the 1-second rate, SUSC_LI must be entered ten times per second out of class E or equivalently once per 100 ms, which is the rate at which the E flag is set. Thus, all that is required is that SUSC_LI be entered once per class E visit.

5.38 The IN_LS flag is set once every 200 ms and can provide up to four additional entries to SUSC_LI per 200 ms depending upon the traffic level. When the IN_LS flag is found set (Fig. 13), it is zeroed and bit position 10 (corresponding to the blank column labeled RP_LS in ECIOTABL) is set in EX2CONTROL. Routine RP_LS is entered and a counter (CRPLS) is incremented and checked. If the value of CRPLS is less than four, OV2EXTRA (paragraph 5.73) is set and transfer is made to SUSC_LI via LIMCHK providing the overload-per-second traffic limit (OV2LIMIT) is not negative. If CRPLS is equal to four, it is reinitialized, the RP_LS flag is zeroed, transfer is made to SUSC_LI in the manner just described and return is made to ECMPNXJB. Thus, setting the IN_LS flag ultimately results in the execution of four line scan segments. The reason for administering two flags instead of one is to minimize the probability of performing any lower priority work when the four IN_LS line scan segments are not completed in 200 ms.

5.39 In a 200-ms period, up to ten entries are made to SUCS_LI. Two are from class E and four each from the IN_LS and IN_TL routines. If both IN_LS and IN_TL are completed every 200 ms, the supervisory line scan is performed at its maximum rate of 200 ms (ten entries to SUSC_LI per 200 ms).

5.40 As traffic increases, the IN_TL flag is affected first because it is the lowest-priority scanning flag. As the rate of serving this flag decreases, the trunk and line scan rates fall off simultaneously from 100 ms to 200 ms and from 200 ms to 333 ms, respectively. The next scanning flag affected as traffic continues to increase is IN_LS. Honoring this flag less and less causes the line scan rate to further drop to 1 second (supported by the class E entrance) while the trunk scan rate remains constant at 200 ms because of high-priority scanning flags HIGH_TK and INBAND.

D. Other Job Flags (ECIOTABL)

5.41 The remaining assigned columns in ECIOTABL will be described briefly. These miscellaneous jobs futher illustrate the advantage of having an expanded scheduling universe on base level that includes a domain external to the five frequency classes.

5.42 Column 1, labeled IOCPSPEC, provides a 50-ms entry into the 1A Processor Input/Output Control Program (IOCP) at global IOCPIOH1 where all the input-output unit selectors are polled. Before transferring to IOCPIOH1, routine IOCPSPEC increments the long C timer. If the C-to-C intervisit time is less than 500 ms, the high-priority extra limit is incremented by the amount of the overload control value (OV2CONTROL). In any case, the IOCPSPEC flag (IOCP) is zeroed before transferring to IOCPIOH1. Return is made to ECMPNXJB.

5.43 Columns 0 and 9, labeled HI_OCC and TIMCALC, are flagged every 5-ms interval to provide for recalculating and updating time available at certain points within each interval. The HI_OCC routine is entered upon completion of a time-fill job that was started in an earlier 5-ms interval. The amount of time taken from the current 5-ms interval to complete the time-fill job started in an earlier interval is subtracted from the total time available. This must be done because total time available was previously calculated at the completion of the last J-level where all of the time remaining in the current 5-ms interval at that point was added to total time available. Thus, HI_OCC recalculates the total time available at the entrance to high-priority base-level occupancy jobs (columns 0 through 8). Return is made to ECMPNXJB.

5.44 The TIMCALC routine is entered following execution of the frequency classes (A through E). This routine updates the total time available by reading the millisecond clock, subtracting the clock value from 5 ms, and adding the result to total time available. The millisecond clock indicates the number of 700-nanosecond cycles that have elapsed since the last 5-ms interrupt. Return is made to ECMPNXJB.

5.45 Columns 12 and 18, labeled MAC_FILL and RCFILL, are time-fill jobs and are entered only when no other flags are set. MAC_FILL performs routine maintenance and RCFILL processes recent change messages.

E. General Purpose Timing

5.46 There are an assortment of timing requirements not fulflled by the timetables that are executed in interject and in certain of the frequency classes. To meet these requirements, ECMP provides two kinds of general-purpose timing facilities: call register timing and noncall register timing.

Call Register Timing

5.47 This type of timing facility is provided for call processing programs and is administered by ECMP via two types of call register timing linked lists: 1-way and 2-way. Timing linked lists are lists of call registers that have been linked together waiting for a time-out, release of equipment, release of call store memory, etc. One-way linkage begins at a head cell and points only forward to the next register, whereas 2-way linkage points in both forward and reverse directions.

5.48 A 1-way list is associated with a 2-word head cell; word 0 points to the first register on the list. The last register on the list points to the head cell whose second word contains the 0 end indication. A register on a 1-way list cannot be removed from the list until it has timed out. When a register is removed from a 1-way list, the program must trace through the list from the beginning to determine the information required to delete it from the list.

5.49 A hypothetical 200-ms 1-way linked list is shown in Fig. 14. The 2-word head cell designated TIM200 contains the address (22226) of the first register (ringing register) in word 0 and the 0 end indication in the second word. The digit 5 located in the second word of the ringing register specifies the number of 200-ms timing intervals required for this entry to time out. The number 30003 designates the address of the next register in the 200-ms 1-way linked list. Note that the last register linked to the list points to the head cell.

5.50 The timing for registers that have been placed on this 200-ms 1-way linked list is administered by a task dispenser program (OTMS) given control every 200 ms from the class B ordered bits buffer. This 200-ms general-purpose timing program goes through the linked list, reducing the timing count by 1 in the left part of the second word of each register. When this timing count is reduced to 0 (Fig. 14), the register is removed from the list and control is given to the appropriate task program via the RI-PT timing return entry YRIPTT. This routine determines the appropriate task program by using the register identification (RI) and program tag (PT) information located in the first word of each register. The RI selects a table of transfer addresses associated with the type of register, and the PT is used to select from this table the address of the proper task program. The queue indicator (QI) bit is zeroed to remove the timed-out register from the linked list and control is given to the task program.

5.51 Programs request entry on 1-way linked timing lists via the expansion of macro ENTTMO (No. 1A ESS has only macro expansions). Timing is provided in units of 100 ms, 200 ms, 6 seconds, and 1 second (Table P). The rate of

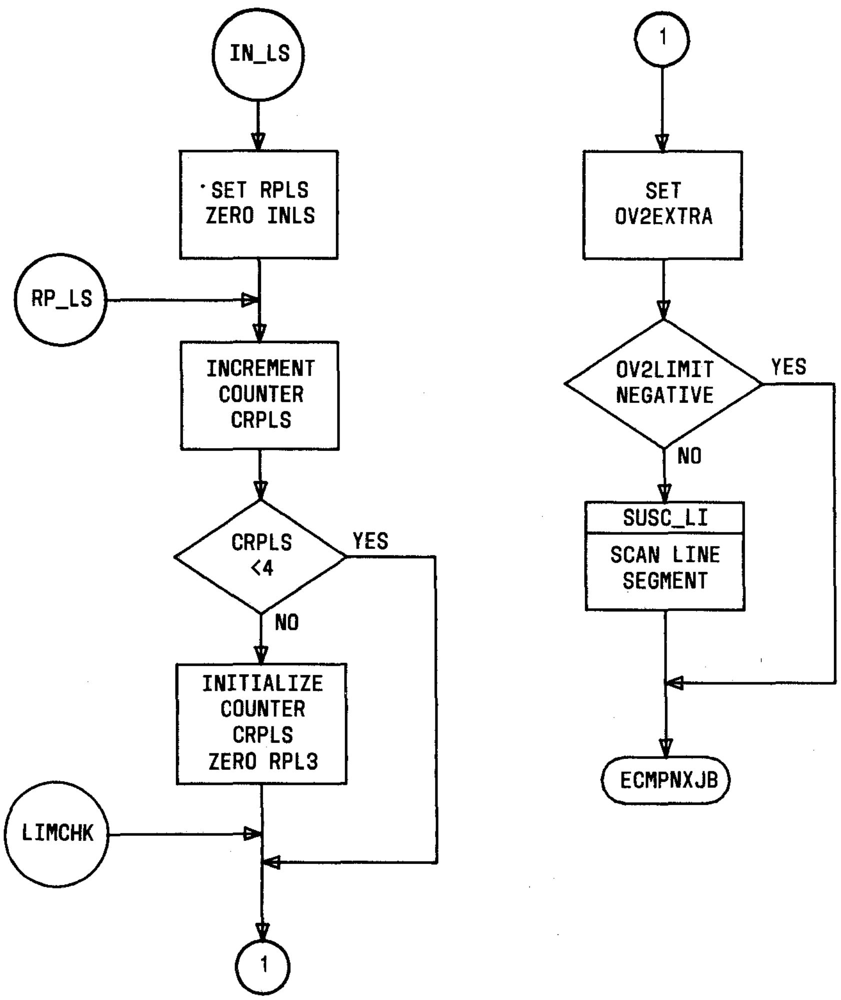


Fig. 13—Supervisory Line Scanning (Low-Priority Flags)—Flow Diagram

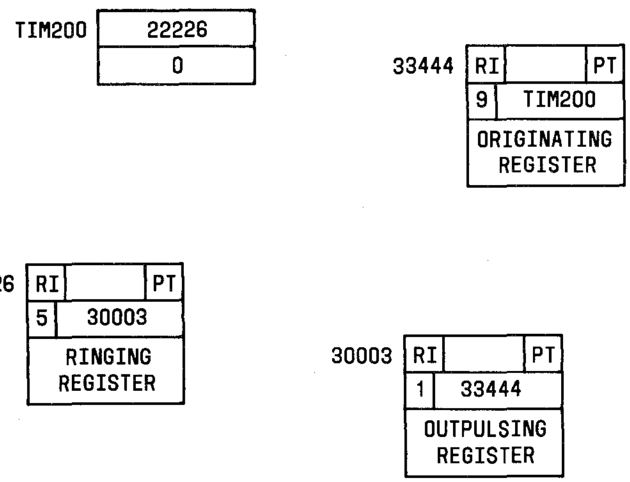


Fig. 14—Hypothetical 200-ms 1-Way Linked List for General-Purpose Timing

timing requested results in the register being entered on the proper list. The number of timing intervals or units specified is written into a counter which is stored in the left part of the register's queue word. The QI bit of the register is set to 1. For additional details on processing the various 1-way linked lists, see Table P and PR-6A004 (Executive Control Main Program Listing).

5.52 A 2-way linked list (Fig. 15) requires a 4-word head cell; word 0 points to the first register on the list. The last register points to the head cell whose second word contains the 0 end indication. The fourth word of the head cell contains the address of the last register on the list. Reverse linkage is found in the scan (fourth) word of each register and forward linkage, in the queue (second) word. The scan word of the first register contains the head cell address minus 1. A register on a 2-way linked list may be removed at any time via the expansion of macro ENDTMT (No. 1A ESS has only macro expansions) since the register contains enough information to allow the linkage to be mended.

5.53 Programs request entry on 2-way timing linked lists via globals YPWORD and YPCONS. Timing is provided in units of 200 ms, 1 second, and 6 seconds (Table Q). The rate of timing requested results in the register's being entered on the proper list. The number of timing intervals or units specified is written into a counter which is stored in the left part of the register's queue word. After forward and backward linkage have been established by storing the proper addresses in the register's queue and scan words as previously explained, the QI bit is set to 1. For additional details on processing the various 2-way timing linked lists, see Table Q and PR-6A004.

Noncall Register Timing

5.54 This type of timing facility is provided for noncall processing type programs (eg, audits, traffic measurements, maintenance) which perform functions without a standard call register. Three blocks of 2-word registers for noncall register timing are maintained by ECMP. These correspond to three different rates of timing: 100 ms, 6 seconds, and 1 second (Table R). The traffic measurements program is provided with two private registers in both the 100-ms and 6-second groups. In addition to other private registers (audit, ringing, etc), there are several common registers in each of the three timing categories.

5.55 Programs request noncall register timing via global ECNRS0. The address at which a requesting or client program is to be reentered is stored in the second word of the timing register. The first 18 bits of data in the X register, along with the timing counter, are stored in the first word of the timing register. As indicated in Table R, processing routines for all timing categories transfer to local ECNRT4 which is the time-out routine for noncall register timing. ECNRT4

TABLE P

GENERAL-PURPOSE TIMING FOR 1-WAY LINKED LIST OF CALL REGISTERS

TIMING UNITS	PROCESSING ROUTINE	COMMENTS
100 ms	YPOHMS	YPOHMS is entered from class B OBB jobs every 100 ms.
200 ms	OTMS	OTMS is entered from class B OBB jobs every 200 ms via entry into YPOTM3 (200-ms 2-way linked list timing routine) which stores the address of OTMS as the next timing list to be checked after 200-ms 2-way processing is complete.
6 sec	YPO6S	YPT6S (6-sec 2-way linked list timing routine) loads the address of YP06S in K and transfers to STRE which saves K as the next list to be checked after 6-sec 2-way processing is complete.
1 sec	YPO1S	YPT1S (1-sec 2-way linked list timing routine) loads the address of YPO1S in K and transfers to STRE which saves K as the next list to be checked after 1-sec 2-way processing is complete.

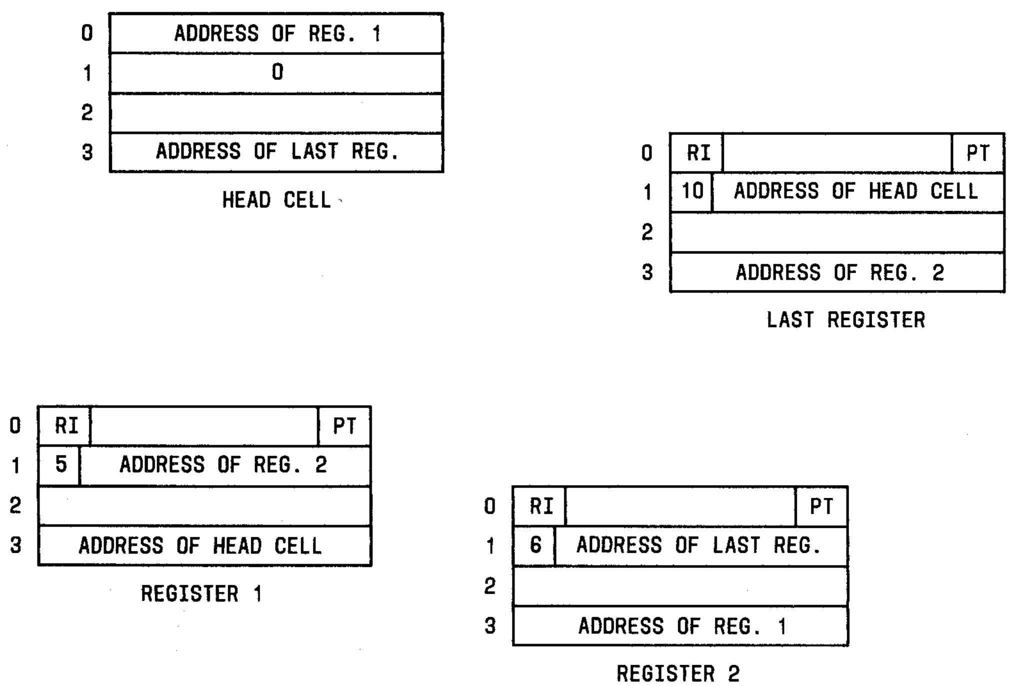


Fig. 15—Two-Way Linked List for General-Purpose Timing—Call Register Layout

TABLE Q

GENERAL-PURPOSE TIMING FOR 2-WAY LINKED LIST OF CALL REGISTERS

TIMING UNITS	PROCESSING ROUTINE	COMMENTS
200 ms	YPOTM3	YPOTM3 is entered from class B OBB jobs every 200 ms.
6 sec	YPT6S	YPT6S is entered via the 6-sec noncall register timing routine (YPAC00) when YPAC00 transfers to ECNRT4 with a stack return. If ECNRT4 finds none of the registers on the 6-sec noncall register timing list have timed out, then transfer is made to YPT6S (stack return address plus 4). YPT6S then transfers to STRE to complete processing.
1 sec	YPT1S	YPT1S is entered via the 1-sec noncall register timing routine (YPAB00) when YPAB00 transfers to ECNRT4 with a stack return. If ECNRT4 finds none of the registers on the 1-sec noncall register timing list have timed out, then transfer is made to YPT1S (stack return address plus 4). YPT1S then transfers to to STRE to complete processing.

periodically decrements the timing counter in each register and transfers to the client time-out address (in the second word of the register) when the timing counter for a register equals 0. The second word is zeroed, leaving the timing register idle.

5.56 Noncall register timing can be discontinued at any time before time-out via globals ECNRD0 and ECNDR1.

F. System Clock Maintenance and Periodic Timing

5.57 One of the task programs appearing in the 100-ms timetable is the ECMP 1-second control program (ECM11) which is executed in interject via local ECTK00. ECM11, which is activated every tenth 100-ms interval, is responsible for incrementing the time-of-day and date counters that are associated with system clock administration. These counters are stored in clock words E4DATE, E4PCNT, E4KTOD and E415MN (Fig. 16). Every second, ECM11 recycles the 100-ms portion of E4KTOD and adds 1 to the 1-second portion. After each 1-second addition, checks are made to determine which of the following counters must be incremented: 3-second, 10-second, 1-minute, 10-minute, 15-minute, 1-hour, 1-day, 1-week, 1-month, or 1-year. For example, every 10 seconds, the 1-second portion of E4KTOD is zeroed or recycled and the 10-second portion is incremented. Every 60 seconds,the 10-second portion is recycled and the 1-minute portion is incremented. Updating continues on all portions of E4KTOD for 24 hours. Words E4PCNT and E4DATE are updated when required to reflect any changes in day, month, or year.

5.58 At the 3-second, 10-second, and 15-minute points, the 1-second control program, either directly or based on the contents of timetables, sets flags in ordered bits buffers (OBBs) for the execution of various periodic programs. For example, every 10-seconds a flag is set in the class

TABLE R

GENERAL-PURPOSE NONCALL REGISTER TIMING

TIMING UNITS	PROCESSING ROUTINE	COMMENTS
100 ms	SNRT	SNRT is entered following the completion of KJOBS (100-ms timetable jobs administered by ECTK00 — the 100-ms control program executed in interject) and SJOBS (processed via entry at local 100NR from ECTK00 to execute those jobs that when turned on are entered every 100 ms). Routine 100NR transfers to SNRT which in turn transfers to ECNRT4 to start 100-ms noncall register timing.
6 sec	YPAC00	YPAC00 is entered from class D OBB jobs every 6 sec and transfers to ECNRT4 to start 6-sec noncall register timing.
1 sec	YPAB00	YPAB00 is entered from class C OBB jobs every second and transfers to ECNRT4 to start 1-sec noncall register timing.

E ordered bits buffer resulting in the execution of the 10-second timetable via local ECRT00 upon the next execution of class E.

5.59 In addition to the timing functions, the 1-second control progam controls entry into the permanent-signal partial-dial timing program based on the value of the permanent-signal partial-dial timing word E4PSPD when read through a 1-bit mask. The 1-second control program also updates five overload control words (OV2CNTR, OV2AVE, OV2EXTRA, OV2CONTROL, and OV2LIMIT) once per second in connection with implementing real-time overload control (paragraph 5.73) and calculates both the generic utility time limit (LU2GLIMIT) and system occupancy (EX2OCC) every 10 seconds. (The generic utility is an on-line, live-office debugging facility.)

Clock Setting

5.60 If it becomes necessary to manually set the system clock, TTY input message CLK-SET- (IM-6A001—Input Message Manual) is used. Global ECCSET in ECMP uses the parameter data provided via this message to set the correct date and time into E4DATE, E4PCNT, E4KTOD, and E415MN. The 4A timer register (E4ATMR) and the 100-ms correction register (E4A100) are zeroed, the new time of day is printed, and a time change label is made on the automatic message accounting (AMA) tape. An OK response to the input message is printed and return is made to the TTY program.

5.61 The system clock can be corrected (advanced) in units of seconds via TTY input message CLK-ADVANCE-aaa where aaa contains the number of seconds. The TTY program stores the number of correction units in a seconds correction register (E4ASEC), and the advance takes place when the ECMP clock correction routine (ECADDE) is entered in interject from the 100-ms timetable every 10 seconds after activation by the 10-second program (ECRT00). ECADDE checks the seconds correction register to determine if corrections in units of

23 — 0 BIT POSITIONS

	MONTH (22-19)	DATE (18-14)	YEAR (13-0)

(A) E4DATE

23 — 0

	DAY OF WEEK (14-7)	HOUR (6-2)	QH (1-0)

(B) E4PCNT

23 — 0

	HOURS (22-18)	10 MIN (17-15)	MIN (14-11)	10 SEC (10-8)	SEC (7-4)	100 MS (3-0)

(C) E4KTOD

23 — 0

	3-SEC CTR (10-2)	1-SEC CTR (1-0)

(D) E415MN

Fig. 16—No. 1A ESS System Clock Words—Call Store Layout

seconds are required. If so, the seconds correction register is decremented by 1 and the system clock is updated by 1 second. Return is made to ECADDE where the seconds correction procedure is repeated until the seconds correction register equals 0. At this point, the 100-ms correction register, the 4A timer register, and the H-level interrupt counter are zeroed. The new time of day is printed, and a time change label is made on the AMA tape.

5.62 When ECADDE is entered, the system clock can also be corrected in units of 100 ms. The 100-ms correction register (E4A100) is checked to determine if corrections in units of 100 ms are required. A 100-ms correction can be added to or subtracted from the system clock. If E4A100 is positive, the clock correction routine (ECADDE) subtracts 100 ms from the system clock and decrements E4A100 by 1; otherwise, the 100-ms correction is added to the system clock and E4A100 is incremented by 1. When the 100-ms corrections are completed, ECADDE is deactivated.

5.63 There are three sources from which a 100-ms correction can be generated:

(a) Normal correction of system clock—Due to an inherent error which causes the system clock to be fast, a 100-ms clock correction is required every 250 seconds.

(b) 4A timer—The system clock is compared with the 4A timer (a hardware timer) every 6 seconds via global ECLOCK in ECMP (paragraph 5.64). Each time the system clock and the 4A timer disagree within certain limits, a 100-ms correction is generated.

(c) H-level interrupts—Each time an H-level interrupt occurs, a delay of 5 ms is introduced. When 20 H-level interrupts have occurred (a count is kept in E4HINT), a correction of 100 ms is made in the system clock.

Clock Accuracy

5.64 The accuracy of the system clock (E4KTOD) is checked every 6 seconds via global ECLOCK in ECMP by comparing the second and 100-ms portions of the clock (converted to binary) with a hardware timer (Western Electric 4A timer). The 4A timer is equipped with two master scanner points (even and odd) which saturate every 6 seconds. Every change from unsaturated to saturated is reported to ECMP via the scan point change director program (CHGD). Only one scan point report is used to check clock accuracy. When the even scan point reports, 6 seconds are added to a 4A timer register (E4ATMR), and the timer register and the clock are compared. If the clock and timer disagree by less than 500 ms, no corrective action is taken. If the disagreement is greater than 500 ms, but less than 2.9 seconds, a correction (either positive or negative) is initiated by either incrementing or decrementing the 100-ms correction register. The correction actually takes place when the ECMP clock correction routine (ECADDE) is entered. If the clock and timer difference is 2.9 seconds or greater, the 4A timer register is set to agree with the clock. Large corrections must be made via the TTY.

REAL-TIME OVERLOAD CONTROL STRATEGY

5.65 The No. 1A ESS real-time overload control strategy includes the following basic departures from the No. 1 ESS overload control plan: incorporation of a new base-level main program scheduling algorithm, the use of routine maintenance work as a time filler as opposed to main program cycling, implementation of variable supervisory trunk and line scanning rates on base level, elimination of the line service request hopper, and incorporation of a new plan for controlling originating traffic in coordination with the control of incoming traffic. The explicit control logic of the No. 1A ESS strategy (paragraph 5.73) is highly interrelated with the implicit control logic provided by the new scheduling algorithm (paragraph 5.06). Each of the functions performed in this logic is essential to the satisfaction of the various overload objectives.

A. Real-Time Overload Objectives

5.66 These objectives are most logically categorized according to levels of traffic, starting with an idle system and progressing to an overload state.

Idle System

5.67 During periods with virtually no traffic, all nonessential tasks are performed and line and trunk scanning are performed at their maximum rates. The traffic acceptance rate is kept at a sufficient minimum value to allow for any reasonable surge or change in traffic level without introducing processing delays.

Low Traffic

5.68 This situation is characterized by a plentiful supply of available real time. Thus all nonessential tasks are performed as closely to the maximum rates as available real time allows. The acceptance rate should be very high so that the probability of delaying any traffic into the system is virtually zero. Sufficient real time is available to handle the processing of any unexpected surge or change in level of traffic.

High Traffic

5.69 During periods of high traffic, real time is at a premium and nonessential tasks are suspended. Trunk and line scanning are performed as time allows with the provision that the trunk scanning rate be maintained at least at a 200-ms minimum. The acceptance rate is decreased to provide a firm upper limit on traffic into the system and the ability to react quickly in the event of an overload. However, the acceptance rate is still maintained at a sufficiently high value to keep the probability of imposing delays on traffic into the system at a very low level.

Real-Time Overload

5.70 At the point where (by overload control definition) available real time is exhausted, an immediate reaction is required to limit traffic. A feedback control loop becomes operational, allowing more traffic as real time becomes available and less traffic as real time is exhausted. Further control is provided to prohibit an excessive amount of traffic into the system during short intervals of time when it appears that adequate real time is available due to statistical occurrences of finding little work in the hoppers.

5.71 Incoming traffic is given priority over originating traffic. All incoming traffic is accepted to a point where originating traffic has been limited to some minimum value. At this juncture, limiting also begins to include incoming traffic. This coordination operates on virtually an instantaneous basis and implies that the overload control is generic and adaptive in the sense that it reacts appropriately regardless of the traffic mix.

Decreasing Traffic

5.72 As the system moves from the overload state to the high traffic state or from high traffic to low traffic, etc, the stated objectives of the new traffic level are rapidly met. For example, as an overload subsides an immediate reaction is to allow all traffic into the system and thereby eliminate processing delays.

B. Real-Time Overload Control Logic

5.73 Five call store locations are employed in the implementation of the real-time overload control logic:

(a) OV2CNTR—This word is initialized to a value of 1 at the beginning of each 1-second time interval (ECMP local OVLD_1SEC, Fig. 17). For the remainder of the 1-second interval, OV2CNTR is incremented by 1 every time a line origination (Fig. 18) or a trunk seizure (Fig. 19) from the trunk seizure and answer hopper is allowed into the system. At the end of each 1-second interval, the value of OV2CNTR is 1 plus the total number of originations and seizures allowed into the system during that second.

(b) OV2AVE—This word is used as a running average of the traffic allowed into the system. At the end of each 1-second interval (Fig. 17), the contents of OV2AVE are read and 15/16 of this value is added to 1/16 of the value of OV2CNTR. This sum is then stored in OV2AVE and becomes its new value.

(c) OV2EXTRA—This flag is set whenever the supervisory line scanning program (SUSC_LI) is entered via ECMP local RP_LS (Fig. 18) and is reset (or zeroed) at the start of each 1-second interval (Fig. 17). If line scanning is being entered from RP_LS, then the line scan is being performed at least at a 300-ms rate; the trunk scan, at least at a 200-ms rate; and the definition point of system capacity (paragraph 5.35) has not been reached. If line scanning is not being entered from RP_LS, then defined system capacity has been exceeded. Thus, the state of OV2EXTRA at the end of a 1-second interval is an indication of extra real-time availability or lack thereof.

(d) OV2CONTROL—This word is used in the control of traffic into the system and is updated once per second. If traffic is high, 15/16 of the value of OV2CONTROL is added to 5/64 the value of OV2AVE. This sum is then stored in OV2CONTROL and becomes its new value. If traffic is low, 1/4 is added to the value of OV2CONTROL to obtain the new value of OV2CONTROL. In either case, further checks are made to determine whether this value is within certain bounds or if additional adjustments are required (Fig. 17).

(e) OV2LIMIT—This is the word whose value is directly used to limit traffic into the system. The value of OV2LIMIT is increased by 1/64 of the sum of OV2CONTROL and EXLIM (25-ms high-priority limit increment—incremented via ECMP locals IOCPSPEC and INBAND when the C-to-C intervisit time is less than 500 ms) every class C execution, just prior to unloading the trunk seizure and answer hopper via global AOUTKS (Fig. 19) in AOVD (No. 1A ESS Automatic Overload Control). OV2LIMIT is decremented by 1 every time a line origination or trunk seizure is allowed into the system. If OV2LIMIT becomes negative, no more originations or seizures are accepted until it becomes positive again. If a 1-second check (Fig. 17) finds that the value of OV2LIMIT has exceeded that of OV2CONTROL, then OV2LIMIT is set equal to OV2CONTROL.

5.74 As indicated in the flowchart for processing line originations (Fig. 18), line scanning may be entered from ECMP locals RP_LS, RP_TL, or EINC (class E). OV2EXTRA, which is used as an indicator of available real time, is set only when line scanning is entered from RP_LS.

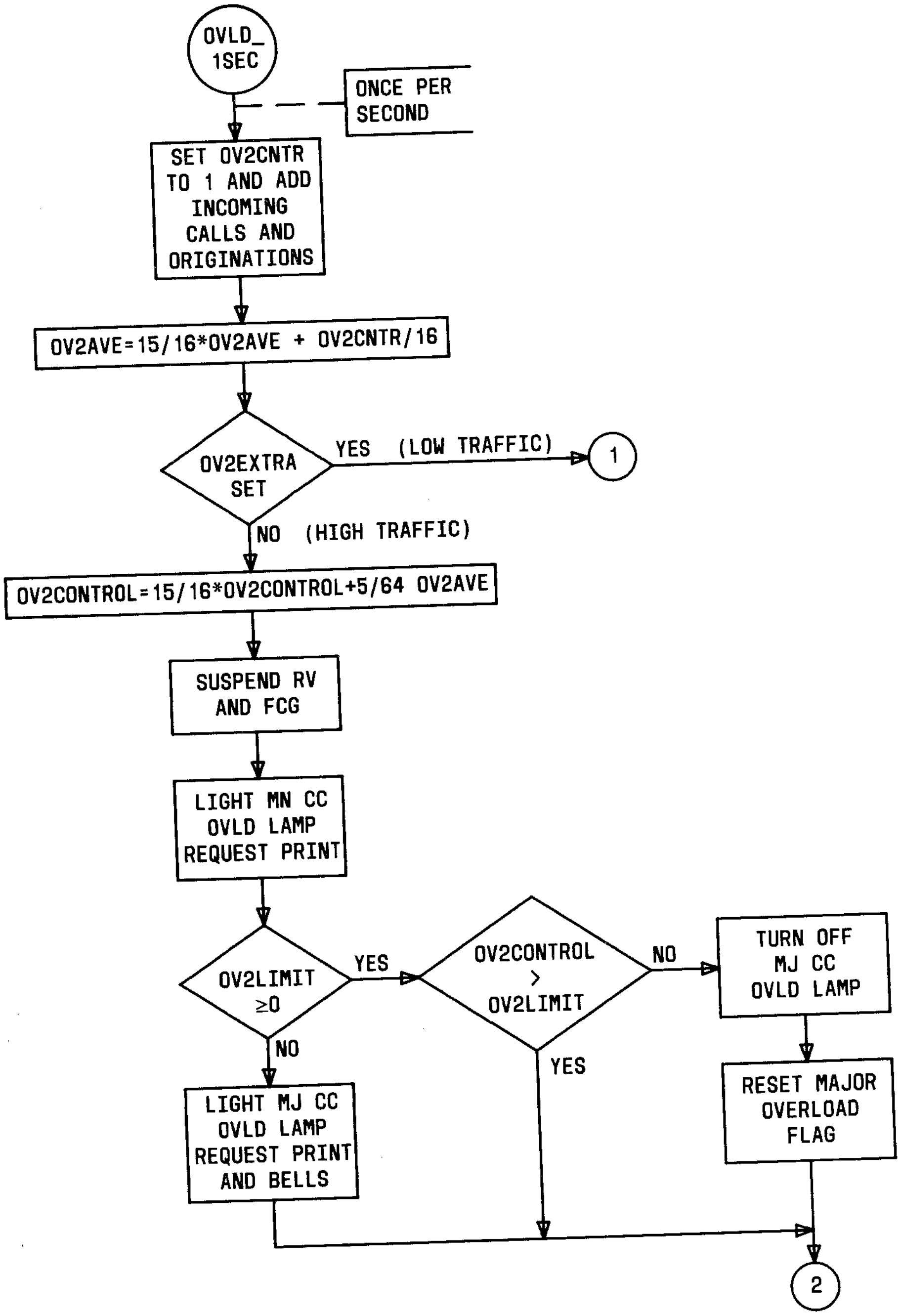
5.75 Both the RP_LS and RP_TL entries check OV2LIMIT, and no line scanning is done nor any originations accepted if its value is negative. The EINC or class E entry does not initially check OV2LIMIT and consequently will accept at least one line origination per class E visit regardless of the value of OV2LIMIT.

5.76 If an origination is found while scanning the lines, scanning is temporarily halted and the origination is processed up to activation of the POB for a receiver (no line service request hopper in No. 1A ESS). After the processing is completed, OV2CNTR is incremented by 1 to record the fact that another call has entered the system. OV2LIMIT is then decremented by 1 and a check is made to see if it has gone negative. If so, no more originations will be accepted this visit and line scanning is exited with a transfer to ECMPNXJB. If not, line scanning is resumed and continued until (a) another origination is found, with subsequent actions taken as just described, or (b) this segment of scanning work is completed, and either an exit is made (if the last line scan segment terminated normally) or another segment is begun (if the last line scan segment was terminated by OV2LIMIT going negative).

5.77 As shown in the flowchart for processing trunk seizures (Fig. 19), AOVD global AOUTKS is entered once per class C execution to unload the trunk seizure and answer hopper (TSAH) and to serve the incoming overload control queue. The first action taken is to increase the value of OV2LIMIT by 1/64 of the sum of OV2CONTROL and EXLIM. This is the only time that OV2LIMIT is increased. The fact that it is done just prior to unloading the TSAH indicates that seizures have priority over originations. With the exception of the one origination always accepted every class E execution, the number of originations accepted is determined by that portion of the value of OV2LIMIT which was "unused" during the latest unloading of the TSAH.

5.78 The fact that OV2LIMIT is increased by 1/64 of the sum of OV2CONTROL and EXLIM is extremely important. Class C intervisit time is 40 ms in low traffic and increases with no theoretical upper bound as traffic increases and exhausts real time. Thus, there are a maximum of 25 class C visits per second in low traffic and a smaller number in high traffic as C-to-C intervisit times increase. This means that 1/64 of the sum of OV2CONTROL and EXLIM is added to OV2LIMIT a maximum of 25 times per second in low traffic and a lesser number of times in high traffic, thereby providing an immediate reaction in limiting traffic in an overload situation which is causing C-to-C intervisit times to increase. As the overload becomes more and more severe, the value of OV2LIMIT is automatically increased less and less frequently. Conversely, the maximum number of 25 entries per second provides protection against increasing the value of OV2LIMIT at an excessive rate during temporary slack periods of work.

5.79 The remainder of the control logic for trunk seizures is similar to that for line originations. After each trunk seizure is processed, OV2INCO


OVLD_
1SEC
ONCE PER
SECOND
SET OV2CNTR
TO 1 AND ADD
INCOMING
CALLS AND
ORIGINATIONS
OV2AVE=15/16*OV2AVE + OV2CNTR/16
OV2EXTRA
SET
YES (LOW TRAFFIC)
1
NO (HIGH TRAFFIC)
OV2CONTROL=15/16*OV2CONTROL+5/64 OV2AVE
SUSPEND RV
AND FCG
LIGHT MN CC
OVLD LAMP
REQUEST PRINT
OV2LIMIT
≥0
YES
OV2CONTROL
>
OV2LIMIT
NO
TURN OFF
MJ CC
OVLD LAMP
NO
LIGHT MJ CC
OVLD LAMP
REQUEST PRINT
AND BELLS
YES
RESET MAJOR
OVERLOAD
FLAG
2

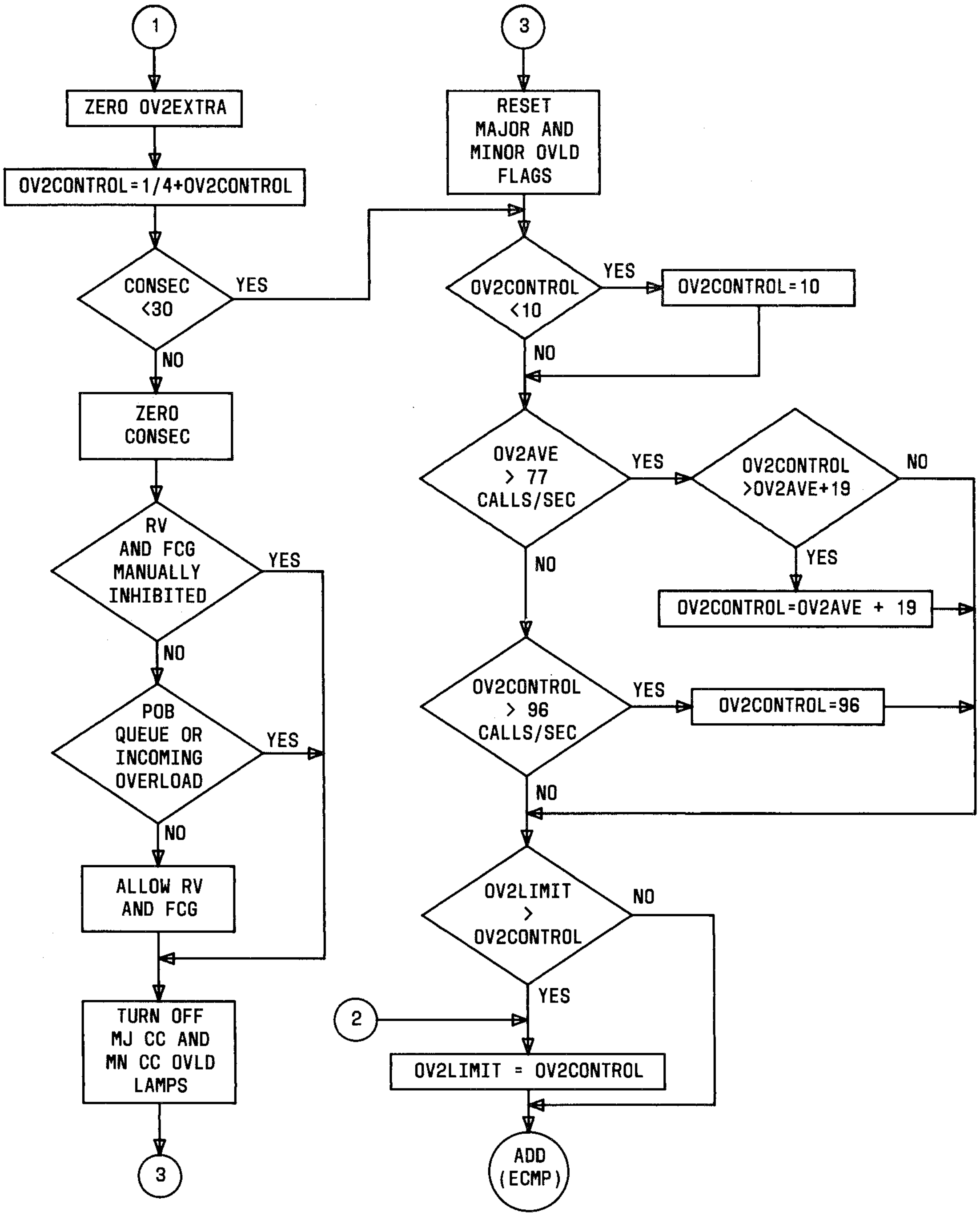


Fig. 17—One-Second Real-Time Overload Control—Program Flow

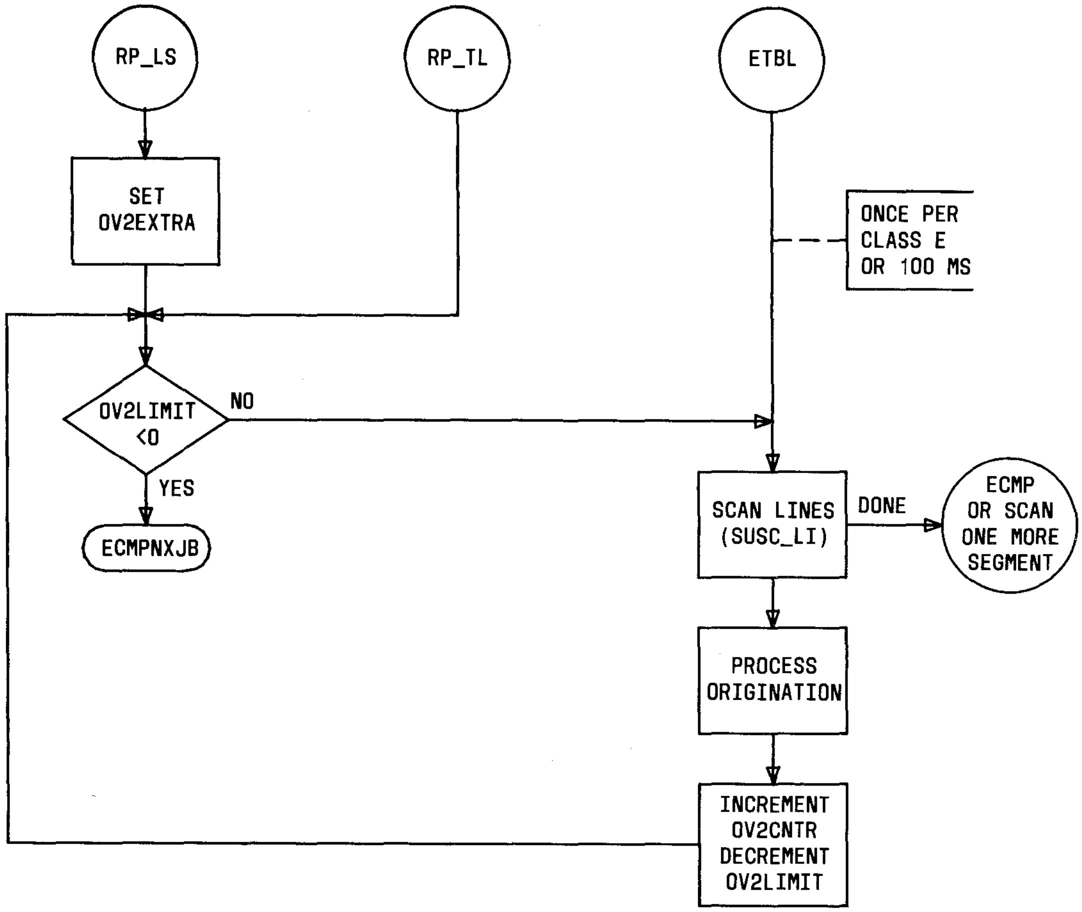


Fig. 18—Processing Line Originations—Flow Diagram

is incremented by 1 to record the fact that another call has entered the system. (At the beginning of each 1-second interval, incoming calls [OV2INCO] are added to originations [OV2CNTR] and the sum is used in calculating the running average [OV2AVE] of traffic allowed into the system.) OV2LIMIT is then decremented by 1 and a check is made to see if it has gone negative. If so, no more trunk seizures are accepted this visit and excess TSAH entries are queued until time becomes available for further processing. If OV2LIMIT is not negative, another trunk seizure (if any remain) is processed with subsequent actions taken as just described.

5.80 The OVLD_1SEC routine (Fig. 17) is a part of the ECMP 1-second control program and, as such, is entered once per second. The first function after loading the current value of OV2CNTR in a register for later use in calculating OV2AVE is to reinitialize OV2CNTR to 1. This guarantees that in very light traffic or when the system is idle, OV2AVE never has a value less than 1 and the equilibrium value of OV2CONTROL is never less than 8. Thus, setting OV2CNTR to 1 has the effect of maintaining a reasonable minimum control level in light traffic while having a negligible effect in heavy traffic.

5.81 The next function performed is the calculation of OV2AVE as described in paragraph 5.73(b). The use of 15/16 of the old value of OV2AVE provides inertia and stability so that any unsually high or low amount of traffic in a 1-second interval does not drastically affect OV2AVE.

5.82 Next, the OV2EXTRA flag is checked to see if it is set. If so, this means that scanning was entered from RP_LS (Fig. 18) at least once in the last second which is an indication that some extra real time is available. In this

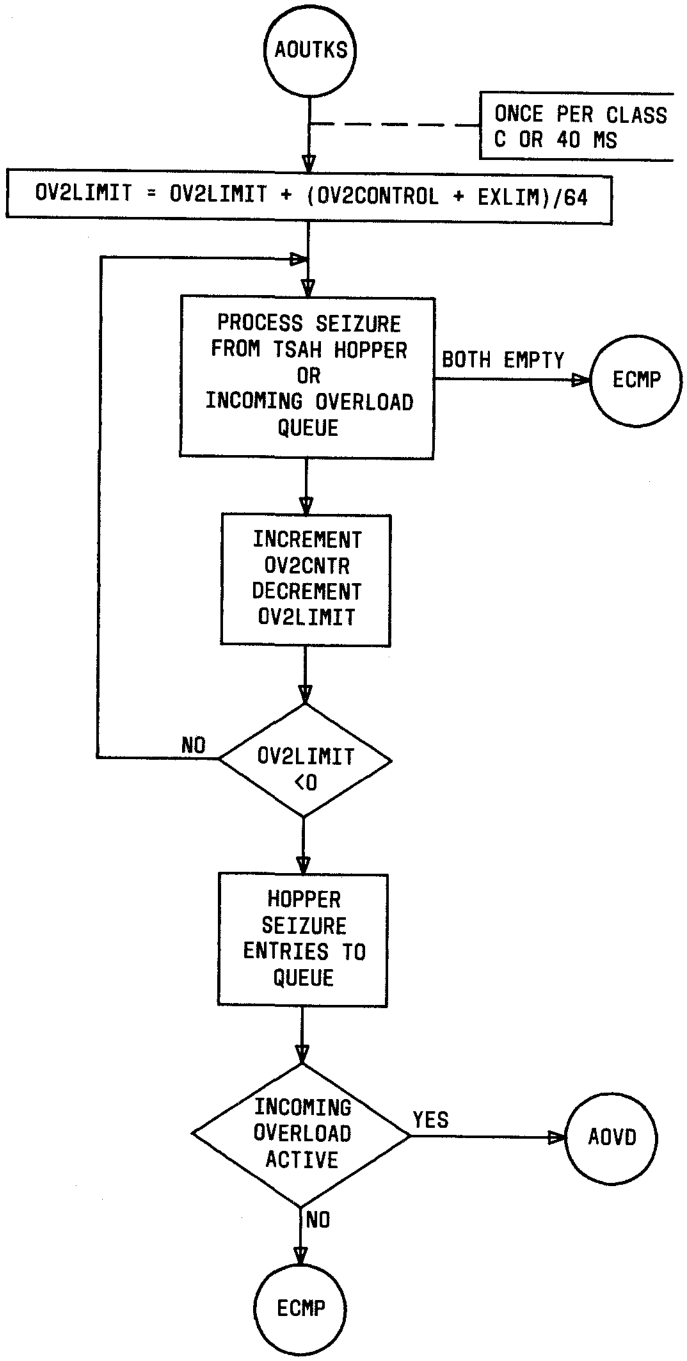


Fig. 19—Processing Trunk Seizures—Flow Diagram

case, OV2EXTRA is zeroed in preparation for the next 1-second interval and OV2CONTROL is increased as described for low traffic in paragraph 5.73(d). When extra real time is available, it is desirable to have OV2CONTROL at a much higher value than OV2AVE so that all traffic is allowed into the system without delay. The equilibrium value for OV2CONTROL in this process is 8 times the value of OV2AVE. Under certain conditions, available real time provides for two nonessential jobs, restore-verify (RV) and false cross and ground (FCG) tests, to be turned on and for all overloads to be terminated. Major and minor CC overload lamps are extinguished, and major and minor overload flags are reset. (See Fig. 17.)

5.83 If the check of OV2EXTRA finds it to be 0, scarcity of real time is indicated. In this situation it is prudent to establish a firm upper bound on traffic into the system. OV2CONTROL is recalculated as described for high traffic in paragraph 5.73(d). Any single occurrence of failing to set OV2EXTRA in a 1-second interval will cause this action to be taken. Thus, at the first indication of a lack of real time, OV2CONTROL is immediately and drastically reduced, and any subsequent indication holds it at this low level. Under these conditions the two nonessential jobs (RV and FCG tests) are suspended and the minor CC overload lamp is lighted. If OV2LIMIT has gone negative, the major CC overload lamp is lighted.

5.84 The last function performed is a check to see if OV2LIMIT exceeds OV2CONTROL. If not, no action is taken; if so, OV2LIMIT is set equal to OV2CONTROL. This action is required in normal office traffic to stop OV2LIMIT from growing without bound. By setting OV2LIMIT equal to OV2CONTROL rather than 0, the capability is acquired to handle temporary surges of "1 second's worth" of traffic without delay, ***near*** overload. This, however, is not the case in overload because by definition OV2LIMIT is forced by traffic to remain near 0.

COMMUNICATION OF OTHER PROGRAMS WITH ECMP

5.85 Programs communicate with ECMP by setting and resetting bits in various OBBs, activity bit words, etc, used by ECMP task dispensers. The bits are changed via macros which locate the proper word and bit. Included in these macros are MPFLAGON, MPONACT, MPOFFACT, MPTIMEON, and MPTIMOFF.

MAINTENANCE OR EMERGENCY ACTION RECOVERY

5.86 An interrupt may occur at any time during system operation. Dependent upon the severity of the interrupt, certain options are available in the determination of a return point from which to restart. The No. 1A ESS Maintenance Restart and Recovery Program (MARS) selects one of three basic returns to normal call processing (Fig. 1):

(a) *Unwinding to the interrupted instruction.* This option is used only for hardwire D- , E- , F- , and G-level interrupts.

(b) *Rolling back to the point in the program that an H- or J-level interrupt occurred.* This return can be used only if an H or J level has program control at the time of the interrupt.

(c) *Returning to a reference point in ECMP.* This return is used when a nonrecoverable interrupt occurs. The reference point is global ECRSUM in ECMP. In ECRSUM, the H- and J-level interrupts are uninhibited; several main program flags are initialized including the control flag word (EX2CONTROL), the frequency class pointer (EX2NEXT), and the ECIOTABL timetable activity bit word (EX2ACT); a control pulse is generated to initialize the 5-ms interrupt timing; and transfer is made to global SARCOV in the No. 1A Call Register Audit Program (SARG). Return is made via the stack followed by a transfer to ECMPNXJB.

6. NO. 1A ESS AUTOMATIC OVERLOAD CONTROL (AOVD)

GENERAL

6.01 System overload occurs when offered traffic produces excessive demands on any of the available system resources (hardware, software, and real time). The objective of automatic overload control is to identify overload conditions as they occur and then to initiate appropriate control strategies.

6.02 There are fundamental differences in the basic strategies employed for the various types of overload; however, overload conditions are not independent of each other. The basic strategies for hardware and software overloads are (a) try another way, (b) queue, (c) try again later, and (d) do not serve the call. For real-time overloads, the strategies are to delay and/or eliminate work.

HARDWARE OVERLOAD

6.03 When the quantities of engineered hardware items are insufficient to meet existing traffic demands, hardware overload occurs. Hardware overload conditions are generally classified into three areas:

(a) Service circuit overload

(b) Outgoing and 2-way trunk overload

(c) Network overload.

The detection of busy and/or blocking conditions occurs in the call processing client programs of the system.

A. Service Circuit Overload

6.04 When a call encounters a service circuit busy condition, the queue administration program (WQUE) is notified. If a queue does not already exist for this particular circuit, one is established with this call being the first entry. Otherwise, the call is loaded into the existing queue. The ECMP job flag associated with the queue is set to 1. This causes ECMP to relinquish control to the queue administration program (WQUE) on a regular basis for the purpose of unloading entries from the queue. Unloading occurs whenever a service circuit is available. When all entries have been unloaded, the job flag associated with the queue is set to 0 to eliminate further ECMP entries. During the period of time that the queue exists, the sign bit is set to 1 in the service circuit idle list head cell. This inhibits seizures of idle equipment by calls not on the queue and ensures that the "first come, first served" principle is maintained.

6.05 When there is a shortage of receivers, certain special actions occur. The first entry on a receiver queue causes a shortening of permanent signal partial-dial (PSPD) timing. Decreased PSPD timing causes the maximum receiver holding time for incoming trunks to be reduced. The queue administration program (WQUE) enters AOVD via global AOFERQ where the conditional print and shortened PSPD timing flags are activated. The receiver queue type (A6CODE), the SAHO audit (A6ONE), the register busy (A6BUSY), and the queue active (A6QACT) bits are set in the receiver queue overload register and the minor receiver (MN RCVR) overload lamp is lighted.

6.06 If a call fails to seize a customer dial pulse receiver or a customer TOUCH-TONE® receiver, the call is placed on queue and a receiver preemption request is made. This request will free a receiver which is being used only for supervision of an outgoing call during outpulsing. The call associated with the preempted receiver is then supervised by a special line ferrod scan.

6.07 When a receiver queue is established, a 100-second timer is started. If the queue overflows, a check is made every 10 seconds via an ECMP entry into AOVD at global AOARQ0 to determine if the queue has existed for at least 100 seconds. If so, a receiver queue overload exists and the 100-second flag (A6AFLG) in the associated receiver queue overload register is set.

6.08 The queue administration program (WQUE) enters AOVD via global AOVFIQ to process the receiver queue whose overload register has timed out. The appropriate queue overload flag for traffic and overload control (TOC) print is set as well as the conditional print and bell flags. If not already requested by a previous queue overload, the TC15 quarter-hour traffic print flag for receiver queue overflow (B6BFLG) is set in the traffic register along with the indication of receiver queue type. The 100-second flag is zeroed, the receiver queue overload register is released by zeroing the register busy-idle bit (A6BUSY), and the major receiver (MJ RCVR) overload lamp is lighted. A request is made to run the hopper and fixed length queue audit (audit 40) via global entry SAHFLQ in PIDENT SAHO. This audit verifies that the load and unload pointers for hoppers and fixed length queues are within valid limits and corrects those that are not. Return is made to WQUE.

6.09 When a receiver queue has been emptied, WQUE enters AOVD via global AOEMTQ where the appropriate queue overload flag for TOC print is zeroed. If no more receiver queues have overflowed, flags are set to extinguish the MJ RCVR and MN RCVR lamps; otherwise, the lamps are left on and the appropriate queue overload register is released by zeroing the busy-idle (A6BUSY) and queue active (A6QACT) bits.

6.10 Every quarter hour, the traffic measurements program (TFCT) enters AOVD at global AORFTR to determine if a receiver queue overload exists. If no receiver queue overflow flags are set, the quarter-hour traffic print flag for receiver queue overflow (B6BFLG) and the flag for type of receiver queue (B6TORQ) are zeroed. Code for blanks is set up in B6BFLG to ensure no

15-minute traffic printout. Whether or not receiver queue overflow flags are set, the conditional print flag is set for a TOC02 message if any control flags are set.

Traffic and Overload Control Status

6.11 Every 4 seconds, ECMP enters AOVD via global AOTTY where checks are made to determine if a traffic and overload control status message (TOC01 or TOC02) should be printed on the TTYs. A mask of the bits set in the overload and traffic conditions flag words (A6FLG1 and A6FLG3) is compared with flags set on the last entrance into AOTTY. Bits changing from 0 to 1 are saved for use in message processing, and all currently set bits are stored for comparison on the next entry into AOTTY.

6.12 If no TTY print was requested, return is made to ECMP. If an unconditional print was requested or if any new flags were set (0 to 1) since the last print and a conditional print was requested, TTY codes for the various conditions to be reported in the TOC message are saved in a scratch area (T scratch) for later use in printing the TOC message. If a request was made to ring the TTY bell and major alarm, this request is honored followed by the printing of the TOC01 message on the maintenance, traffic, and network management TTYs. If no bell was requested, the TOC02 message is printed on the TTYs. (The TOC01 is used to report serious conditions while the TOC02 is used to report less serious or routine conditions.) If the TTY program fails to honor the print request, the unconditional print flag (PRNT) is set so that an attempt will be made to print on the next 4-second entrance. Return is made to ECMP.

6.13 A print of the traffic and overload control status may be requested via TTY input message TOC-STATUS-_. When this message is entered at the TTY, the TTY program (TTIA) transfers to AOVD global AOSTAT where the unconditional print flag is set. This causes the TOC02 message to be printed on the next 4-second entrance to AOTTY. Return is made to the TTY program.

6.14 The 15-minute traffic data is transmitted to the maintenance and traffic TTYs on any clock quarter hour when one or more of the following conditions exist:

(a) Receiver queue overload has existed for some interval during the last 15 minutes.

(b) Line load control is on.

(c) The maintenance TTY has requested the data.

(d) The traffic TTY has requested the data.

6.15 Maintenance and traffic personnel may turn the printing of the traffic data on or off by means of the following TTY input request:

(a) LS-QUARTER-M,ON. (Maintenance on request)

(b) LS-QUARTER-M,OF. (Maintenance off request)

(c) LS-QUARTER-T,ON. (Traffic on request)

(d) LS-QUARTER-T,OF. (Traffic off request).

When any one of the above requests is entered at the TTY, AOVD is entered via global AOTTYR where the request is analyzed for processing. If either the traffic on request or maintenance on request is entered, the traffic data for the complete quarter-hour interval preceding the request is printed at the appropriate TTY immediately and on subsequent quarter hours until cancelled by the traffic off or maintenance off request. When traffic data printout is requested, the manual on control for dial tone speed tests (DTST) is also activated. When the print request is turned off, the manual on control for DTST is deactivated. Following processing of the on/off traffic print request, transfer is made to global RETURN in ECMP.

B. Outgoing and 2-Way Trunk Overload

6.16 If an outgoing call fails to find an idle trunk in the primary group, successive alternate groups are tried until either an idle trunk is found or until all alternate groups are exhausted in which case overflow tone is given to the subscriber.

6.17 Certain trunk groups in an office can be designated as toll protected, and certain lines can be designated as toll essential. If the toll network protection program is activated via a TTY input message, only toll essential lines are permitted to access toll protected trunks. An attempted seizure of a toll protected trunk by a nontoll essential line will result in overflow tone being given to the subcriber.

C. Network Overload

6.18 On an incoming call, if blocking on all possible paths occurs on the network path hunt from the incoming trunk to the terminating line, the trunk is put up to overflow. On an outgoing call, if an idle trunk is found but the network path hunt blocks, another idle trunk and path are hunted. If blocking occurs a second time, the procedure is repeated once more. A third failure results in overflow being given to the originating line. If the system is in a real-time overload state, blocking retrials of trunk hunts are eliminated.

SOFTWARE OVERLOAD

6.19 Various software items in the system are engineered according to the expected traffic load in the office during the average busy hour. A software overload exists when the demand on one or more of these items exceeds the supply or engineered capability. Engineered software items include the various system hoppers, call registers, peripheral order buffers (POBs), junior registers, etc. For purposes of discussion, software is classified either as hopper type or nonhopper type.

A. Hopper Overload

General

6.20 Hoppers are engineered to hold various numbers of entries. When an input-output (I/O) program loads an entry into a given hopper, a check is made to see if the hopper is full. If the hopper is full, overload exists and control is passed to AOVD. The last entry made in the hopper is zeroed and the memory associated with this entry is restored to the previous state. These actions are taken because I/O programs always load an entry into a hopper before checking to see if the hopper is full. This requires that a vacant slot be available in the hopper.

6.21 In order to prevent a potential bottleneck caused by a full hopper, the responsible I/O program enters AOVD via the associated global entry where the last entry in the full hopper is zeroed, the hopper load pointer is updated, the associated hopper overflow flag bit is loaded in the L register, and a transfer is made to interject request routine INJRQ in AOVD. This routine uses the contents of the L register to mark the appropriate flag in interject flag word A6INTJ for later use in hopper unloading. In addition to setting hopper overflow flags for SAHO and SYPI (system performance) and requesting junior register and hopper audits, INJRQ requests hopper unloading in interject by setting the proper interject job flag and the interject source bit (IN1SINJS). Return is made to the hopper loading program.

6.22 When it is time to process interject jobs on base level and the interject job flag for hopper unloading is found set, transfer is made to global AOINJE in AOVD to empty any hoppers that have requested unloading in interject. If more than one hopper is to be unloaded, the requests are processed on a priority basis as indicated in Table S. AOINJE reads interject flag word A6INIJ to determine which hoppers are to be unloaded and, using the rightmost 1 technique, processes all requests and returns to base level (ECMP).

6.23 If the I/O program that caused the hopper overflow looks for only one type of report and loads only one hopper, the remainder of the I/O job is skipped. If the I/O program looks for more than one type of report or loads more than one hopper, the I/O job is continued.

TSAH and Incoming Overload Control Queue Administration

6.24 The automatic overload control program (AOVD) contains the hopper unloading routine for unloading the trunk seizure and answer hopper (TSAH). ECMP enters AOVD via global AOUTKS every class C execution to unload the TSAH and to serve the incoming overload control queue. AOUTKS calculates the current per-visit limit (OV2LIMIT) of trunk seizures that can be processed during this class C visit (Fig. 19). Each time the TSAH is unloaded as a result of a class C visit, a count is kept of the number of trunk seizures processed. When the count equals the current per-visit limit, the remaining entries in the TSAH are placed in the incoming overload control queue. Thus, during periods of heavy incoming traffic,

TABLE S

HOPPER OVERFLOWS ADMINISTERED BY AUTOMATIC OVERLOAD

HOPPER	INTERJECT UNLOADING PRIORITY	PROCESSING PIDENT	GLOBAL ENTRY
Abandon Interdigital Timeout	6	ORDL1A00	ORAINT
Centrex Key	9	CXKY1A00	CXAOKY
Dial Pulse Transmission	11	DPOP1A00	DPHULD
Hit Scan Result	13	CHGD1A00	DIHSRR
Line Ferrod Disconnect	15	CHGD1A00	OGUNL1
Miscellaneous Scan (TSJR)	10	YMRG1A00	YRDINT
Multifrequency	1	ICAL1A00	ICCONV
Multifrequency Compelled Signalling	16	(International)	MCSS_IUL
Release Dial Tone	7	ORDL1A00	ORDINT
Revertive Digit Reception	2	ICRV1A00	ICAOEN
Revertive Digit Transmission	8	OGRV1A00	OGAOCE
Ring Trip	12	ECMP1A00	ECTRIP
Step-by-Step Reorder AIT	3	ISXS1A00	ISXSOV
TOUCH-TONE	4	ORDL1A00	ORTINT
Trunk Dial Pulse Reception	5	ICAL1A00	ICAOTD
Trunk Seizure and Answer	14	CHGD1A00	DCUTK1

the incoming overload control queue helps to maintain a high call-completion rate by reducing the possibility of a TSAH overflow.

6.25 After calculating the current per-visit limit of trunk seizures, AOUTKS checks the length of the incoming overload control queue to determine whether to serve (unload) the queue or the TSAH first. When the length of the queue is short (less than 64 times the average maximum number of trunk seizures this C visit), the queue is served before the hopper via a transfer to global WQUIQS in the queue administration program (WQUE). This forces more calls through the queue. A "first-in, first-out" order is maintained and, because the queue is relatively short, no incoming calls delayed excessively. Prior to queue unloading, the incoming overload print flag is zeroed, and the major incoming (MJ INC) overload lamp is extinguished. Following queue unloading, return is made to AOVD via global AOUTSA to unload the TSAH.

6.26 When the incoming overload control queue reaches a length that is 64 times the average maximum number of incoming trunk seizures this C visit, the hopper is served before the queue. When the queue is long, the time an incoming call remains on the queue becomes excessive and results in receiver time-outs and customer abandons. By serving the hopper first, those calls which are processed have a shorter receiver holding time. This results in reduced receiver time-outs and customer abandons and, thus, increases the call completion rate. When the queue length causes a reversal of the order of service from queue first to hopper first, the incoming overload, conditional TOC print, and bell flags are set and the MJ INC overload lamp is lighted. As each hopper entry is unloaded via routine AOUTSA, transfer is made to DCUTK1 in the scan point change director program (CHGD) to process the entry. When all entries have been processed, a check is made for incoming overload via AOVD local TRYQ. If an overload is in process, transfer is made to global WQUIOC in WQUE to process the incoming overload control queue. When processing is complete, a check is made for interject with a return to the top of the stack.

6.27 If the TSAH overflows, the supervisory universal trunk scans and master scans are temporarily suspended. A request is made to unload the hopper in interject. When the hopper is empty, the scans are resumed.

B. Nonhopper Overload

6.28 Some types of registers are engineered on a one-to-one basis with hardware; thus, if the hardware is available, so is the register. If nondedicated types of registers are not available, the call that made the request for the register is generally restored to a previous state and placed on a queue for the required software.

6.29 When an attempt is made to seize a junior register and there are no idle junior registers of the type required, that type of junior register is considered to have overflowed. If the supervisory scan and audit program (CCLT) finds a step-by-step junior register (SXJR) overflow, AOVD is entered via global AONSXJ. A check is made to determine if an audit (SAHO) of hoppers and junior registers has been previously requested by AOVD during the current audit cycle. If so, AONSXJ simply returns to the calling program. This audit check prevents AOVD from consuming extra real time by repeating its actions frequently. If the SAHO audit has not been requested, AONSXJ attempts to speed up the release of SXJRs by requesting that the step-by-step hopper be unloaded in interject.

6.30 If an attempt to seize a timed scan junior register (TSJR) fails, the supervisory universal trunk scan, master scan, junctor scan, and line side step-by-step scan are all turned off. This is necessary because the programs that administer these scans use TSJRs for hit timing on disconnect reports. In an attempt to release TSJRs more quickly, the timed scan junior register processing program (DITS) enters AOVD via global AOHSRO where a request is made to unload the hit scan result hopper in interject. When an idle TSJR becomes available, the scans are turned on again.

REAL-TIME OVERLOAD

6.31 System real time is allocated to various tasks according to priority requirements, frequency of execution requirements, maximum tolerable delay requirements, etc. The central control acts as the single server in this multiqueue environment. Regardless of the effectiveness of the system's interrupt schemes or scheduling algorithms, service request delays can never be completely eliminated. If two service requests occur simultaneously, one must be delayed because there is only one server to process requests.

6.32 Certain system delay characteristics are dependent upon such things as scheduling, overhead, nature, and level of traffic. However, delays tend to increase as available real time decreases. The concern about excessive delays with regard to overload control is twofold. First, maximum delay requirements exist for the various call processing tasks. These requirements must be satisfied if service objectives are to be met. Second, long delays produce processing inefficiencies which decrease maximum throughput.

6.33 In the absence of overload controls as the system experiences a shortage of real time, the following succession of events could occur, resulting in inability to process calls:

(a) A real-time shortage

(b) Delays in call processing

(c) Longer holding times for engineered hardware and software items

(d) Queueing and retrials

(e) More real time to process the given load of traffic.

This chain of events produces an even more critical real-time shortage and the cycle starts again with (a). The process continues until the emergency action program interprets the extreme delays as a system failure and invokes some form of corrective action. This action means that all call processing is halted for a period of time. When it is resumed, the huge backlog of traffic ensures that another emergency action will occur very quickly. Obviously, controls must be exercised during real-time overloads if catastrophic results are to be avoided. The system scheduling algorithm, which is of great importance during a real-time overload, is described in Part 5 of this section, as are real-time overload control strategy and real-time overload control logic.

7. NO. 1A ESS LINE LOAD CONTROL AND TOLL NETWORK PROTECTION (LLOD)

GENERAL

7.01 During an extreme situation such as an emergency or a disaster, traffic may increase to the point that the automatic overload controls cannot alleviate the overload condition; consequently, service is severely degraded. Line load control (LLC) provides a means of assuring acceptable grades of originating service to lines considered essential in an emergency by temporarily denying originating service to nonessential lines.

7.02 Unlike the automatic overload controls which are inherent in system operation, LLC is selected by maintenance and/or traffic personnel in compliance with local procedures established by the telephone company. The consequences of selecting LLC must be carefully weighed, and all system indications (lamp indicators at the master control console, TTY output messages, etc) must be considered when evaluating an overload condition to determine the need for LLC.

7.03 Two conditions are required for LLC to be active: (1) LLC ***must*** be enabled and (2) a measured overload (based on dial tone speed test failures or network overload) ***must*** be occurring. The degree of control provided by LLC is determined by the extent and persistence of the overload.

PROGRAM INTERFACE

7.04 The LLOD program interfaces with the following PIDENTs (Fig. 20):

(a) Executive Control Main Program (ECMP)

(b) Input-Output Control Program (IOCP).

7.05 ECMP transfers to LLOD every 4 seconds to check the LLC mode of operation, the status of DTST, and the percentage of network overload. IOCP provides for controlling the LLC mode of operation via TTY input and output messages. Refer to the Input/Output Message Manuals (IM-6A001 and OM-6A001) for a more detailed description of the following messages:

(a) LLC-ALLOW-ON. (Manually enables LLC)

(b) LLC-ALLOW-AU. (Automatically activates LLC)

(c) LLC-INH-. (Deactivates LCC)

(d) LLC-MASK-PRNT. (Generates LC02 printout of current LLC scan mask—used to determine the number of line groups affected)

(e) TOC01 (Provides traffic and overload control status—printed as result of input message LLC-ALLOW-ON)

(f) TOC02 (Provides traffic and overload control status—printed as a result of input message LLC-ALLOW-AU)

(g) LC01 (Indicates change in line status).

STRUCTURE AND OPERATION

7.06 The LLOD program consists of four global routines:

(a) LLCCSM—Entered every 4 seconds from ECMP

(b) LLCINP—Entered via TTY input message LLC-MASK-PRNT.

(c) LLCLKO—Entered via TTY input message LLC-ALLOW-aa.

(d) LLCLKN—Entered via TTY input message LLC-INH-.

A. LLC Mode Selection

7.07 The mode of operation of LLC may be automatic, manually activated, or manually deactivated. The desired mode is selected by entering the appropriate input message (Table T) at either the maintenance or traffic TTYs.

7.08 No line groups are denied originating service (LLC active) unless LLC is enabled in either

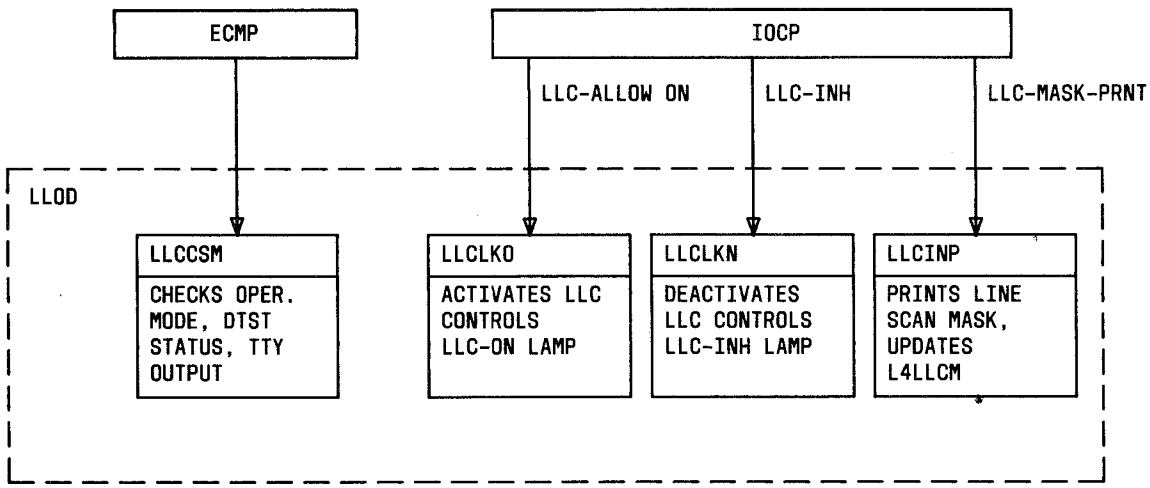


Fig. 20—Line Load Control (LLOD)—Program Interface

the ON or AUTOMATIC mode and a measured overload is occurring. LLC may be placed in the manual ON mode (manually enabled) by entering input message LLC-ALLOW-ON at the TTY or in the AUTOMATIC mode by entering input message LLC-ALLOW-AU. Note in Table T that in the AUTOMATIC mode, LLC is enabled and disabled automatically by the system according to extended DTST (11-sec) failures. LLC may be placed in the OFF mode via TTY input message LLC-INH.

B. LLC ON Mode

7.09 When the LLC-ALLOW-ON input message is entered, the TTY program (TTIA) enters LLOD via global LLCLKO. The LLC AUTOMATIC flag (LAU) is zeroed, the LLC ON flag (LON) is set, output message TOC01 is printed, and appropriate bells and alarms are sounded. The LLC INH lamp is turned off and return is made to TTIA.

7.10 After LLC is enabled, the system administers LLC according to calculations based on the results of 3-second DTSTs (Section 231-045-245) and the extent of network path blockage occurring on incoming calls. Every 4 seconds ECMP transfers to LLOD at global entry LLCCSM to determine if LLC should be applied. If DTSTs are inhibited, the 15-minute traffic printout (TC15) is turned off, the LLC ON lamp is extinguished, and the appropriate flags (the DTST failure count and LLC active last time flags, as well as the print flags for LLC denying service and for trunk link network overload) are zeroed. If the LLC denying service flag was previously on, the LC01-AGR output message is printed on the TTYs to indicate that all groups of nonessential lines have been restored to service. Return is made to ECMP.

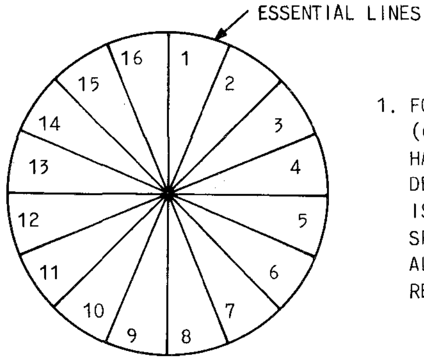
7.11 If DTSTs have not been inhibited and LLC is enabled, the 15-minute traffic printout (TC15) is turned on and the LLC ON lamp is lighted. LLC either allows or denies originating service to one or more nonessential line groups based on the DTST and network blockage results. Three successive 3-second DTST failures or 10 percent network blockage for a 3-minute interval causes LLC to remove dial tone service from one-half of the nonessential line groups currently able to receive dial tone service. Three more successive 3-second DTST failures or another 3-minute period with 10 percent or greater network blockage results in another 50 percent decrease in the number of nonessential lines served. After four such decreases in service, all nonessential lines are denied dial tone service. Items 1 through 5 of Fig. 21 depict the quantity of nonessential line groups denied orginating service based on DTST failures. Items 7 and 8 depict the reduction of nonessential line groups based on network blockage.

7.12 When LLC is active, nonessential line groups are denied originating service by means of a 16-bit scan mask (L4LLCM). The bit positions correspond to the line groups (verticals) in the office. A 1 in a particular bit position allows that corresponding vertical to be scanned, whereas a 0 inhibits scanning of that vertical. The 16-bit scan

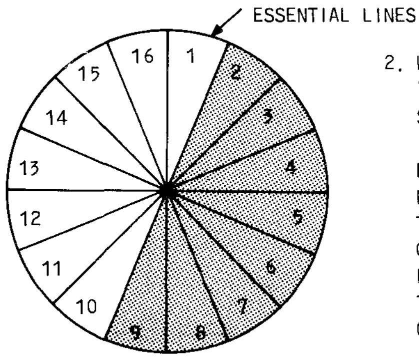
TABLE T

LLC SUMMARY

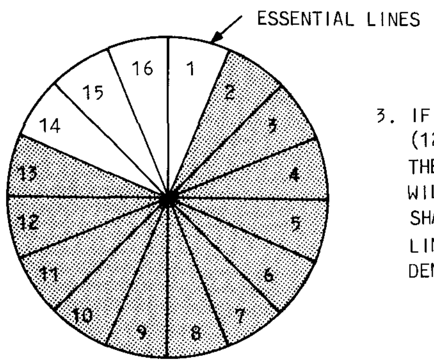
MODE SELECT TTY INPUT MESSAGE	LLC MODE	DIAL TONE SPEED TEST (DTST) FAILURES	LLC STATUS	INDICATOR LAMPS			RELATED TTY OUTPUT MESSAGES	REMARKS
				MN DT DLY	LLC ON	LLC INH		
LLC-INH	Off	No Affect	Inhibited	Off	Off	On	TOC02	LLC not enabled or active; all lines have originating service.
LLC-ALLOW-ON	ON	3-sec DTST< 3	Enabled	Off	On	Off	TOC01	LLC enabled but not active; no service overloads based on DTST and network blockage. All lines have originating service.
		3-sec. DTST< 3	Enabled and Active	On	On	Off	TOC01, LC01, TC15	LLC actively denying originating service to one or more nonessential line groups based on the degree and persistence of the service overload.
LLC-ALLOW-AU	AUTOMATIC (OFF)	11-sec. DTST< 3	Not Enabled or Active	Off	Off	On	TOC02	In AUTOMATIC mode, LLC is either OFF or enabled (ON) automatically based on extended DTST failures.
	AUTOMATIC (ON)	11-sec. DTST> 3	Enabled, not Active	On	On	Off	TOC01	LLC is automatically enabled after three successive extended DTST failures; however, LLC is not active until three successive 3-second DTST failures occur or network blocking exceeds certain limits.
		3-sec. DTST> 3	Enabled and Active	On	On	Off	TOC01, LC01, TC15	



1. FOR EXAMPLE, AN OFFICE WITH 1/16 (6.25%) OF THE OFFICE LINES HAVING ESSENTIAL SERVICE IS DEPICTED AT LEFT. WHEN THE OFFICE IS OPERATED WITHOUT DIAL TONE SPEED DELAY OR NETWORK BLOCKAGE, ALL SIXTEEN LINE GROUPS CAN RECEIVE DIAL TONE.



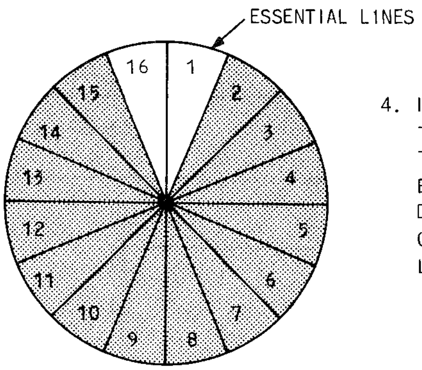
2. WHEN THE LLC FEATURE IS IN THE "ON" MODE, THREE SUCCESSIVE 3-SECOND DTS TEST FAILURES (11 SECONDS ELAPSED TIME) WILL REMOVE DIAL TONE FROM 50% OF THE NON-ESSENTIAL LINES ROUNDED UP TO THE NEXT 1/16 (6.25%) OF THE OFFICE. IN THIS EXAMPLE EIGHT LINE GROUPS ARE BEING DENIED DIAL TONE. SHADED AREA REPRESENTS LINE GROUPS DENIED DIAL TONE.



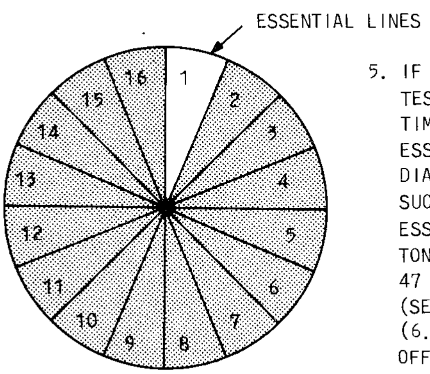
3. IF THE NEXT THREE DTS TESTS FAIL, (12 SECONDS ELAPSED TIME) 50% OF THE REMAINING NONESSENTIAL LINES WILL BE DENIED DIAL TONE. THE SHADED AREA (75% OF THE OFFICE LINES) REPRESENTS LINE GROUPS DENIED DIAL TONE.

NOTE:

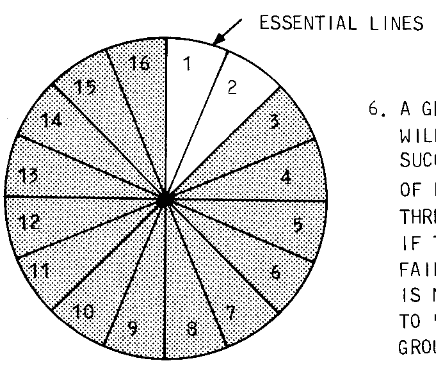
THE NUMBERS ONE THROUGH SIXTEEN REPRESENT THE LINE GROUPS ASSIGNED TO THE SIXTEEN VERTICALS IN AN OFFICE. THE LINE GROUP NUMBERS ASSIGNED DO NOT REPRESENT OR CORRESPOND TO THE PARTICULAR OFFICE VERTICALS. FOR THIS FIG., ONE LINE GROUP (1) IS CLASSED ESSENTIAL; THE REMAINING ARE NONESSENTIAL.



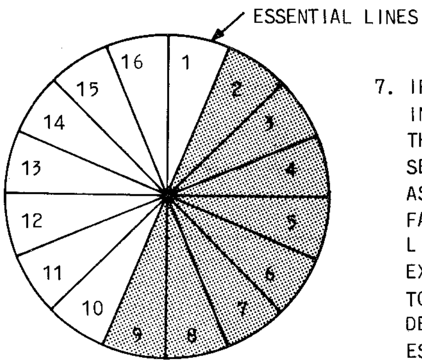
4. IF THE NEXT THREE SUCCESSIVE DTS TESTS FAIL, (12 SECONDS ELAPSED TIME) 50% OF THE REMAINING NON-ESSENTIAL LINES WILL BE DENIED DIAL TONE. THE SHADED AREA (87.5% OF THE OFFICE LINES) REPRESENT LINE GROUPS DENIED DIAL TONE.



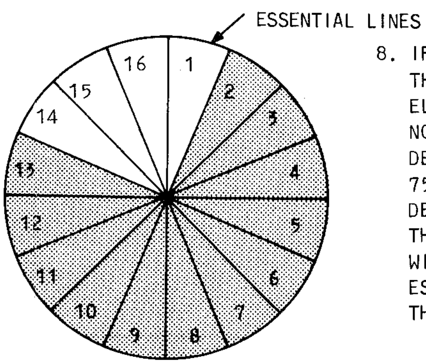
5. IF THE NEXT THREE SUCCESSIVE DTS TESTS FAIL, (12 SECONDS ELAPSED TIME) THE LAST GROUP OF NON-ESSENTIAL LINES IS ALSO DENIED DIAL TONE. THUS, IN FOUR SUCCESSIVE DECREASES, ALL NON-ESSENTIAL LINES ARE DENIED DIAL TONE IN A TOTAL ELAPSED TIME OF 47 SECONDS. ONLY ESSENTIAL LINES (SEE NON-SHADED AREA) OR 1/16 (6.25%) OF THE OFFICE IS BEING OFFERED DIAL TONE.



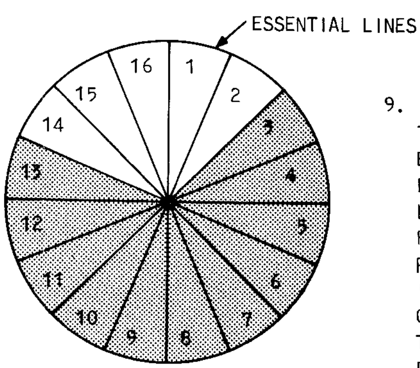
6. A GROUP OF NONESSENTIAL LINES WILL BE RESTORED WITH EACH SUCCESSFUL DTS TEST. WHEN A GROUP OF LINES IS RESTORED, THE NEXT THREE DTS TESTS ARE NONRELEVANT IF THEY FAIL. THUS, SIX CONSECUTIVE FAILURES, FOLLOWING A RESTORAL IS NECESSARY TO DENY DIAL TONE TO 50% OF THE NONESSENTIAL GROUPS IN SERVICE. THE UNSHADED AREA "2" DEPICTS THE FIRST LINE GROUP RESTORED TO SERVICE FOLLOWING THE FIRST SUCCESSFUL DTS TEST.



7. IF 10% NETWORK BLOCKAGE OCCURRED IN AN OFFICE WITH 1/16 (6.25%) OF THE OFFICE LINES HAVING ESSENTIAL SERVICE, THE RESULT IS THE SAME AS THREE SUCCESSIVE DTS TEST FAILURES. 50% OF THE NONESSENTIAL LINES, (8 LINE GROUPS IN THIS EXAMPLE) ARE BEING DENIED DIAL TONE SERVICE. WHEN DIAL TONE IS DENIED BASED ON BLOCKAGE, NON-ESSENTIAL LINE GROUPS WILL NOT BE RESTORED DURING THE THREE MINUTE INTERVAL.



8. IF THE NETWORK BLOCKAGE IS GREATER THAN 10% AFTER THREE MINUTES OF ELAPSED TIME, 50% OF THE REMAINING NONESSENTIAL GROUPS WILL BE DENIED DIAL TONE SERVICE. THUS, 75% OF THE OFFICE LINES ARE NOW DENIED DIAL TONE AS DEPICTED BY THE SHADED AREA. AGAIN, DIAL TONE WILL NOT BE RESTORED TO ANY NON-ESSENTIAL LINE GROUP DURING THE THREE MINUTE INTERVAL.



9. IF THE NETWORK BLOCKAGE IS LESS THAN 10% AFTER THREE MINUTES OF ELAPSED TIME, THE NONESSENTIAL LINE GROUP DENIED DIAL TONE THE LONGEST WILL BE RESTORED WHEN THE FIRST DTS TEST IS SUCCESSFUL. THE REMAINING GROUPS WILL BE RESTORED IN A LIKE MANNER WITH EACH OCCURRENCE OF A SUCCESSFUL DTS TEST. WHEN THE FIRST GROUP IS RESTORED A MAXIMUM OF FIVE DTS TESTS MAY FAIL, BUT THE SIXTH DTS TEST IS USED TO DETERMINE WHETHER TO DENY OR RESTORE. THE UNSHADED AREA "2" DEPICTS THE FIRST LINE GROUP RESTORED.

Fig. 21—Line Load Control Results

mask is updated as the measured overload varies. When LLC is active, LLC determines every 4 seconds if the scan mask should be updated. The type of updating depends on the originating traffic load in the office and the time elapsed since the last update. The scan mask is updated for the following reasons:

(a) To reduce the number of nonessential line groups served

(b) To rotate the nonessential line groups served

(c) To increase the number of nonessential line groups served.

7.13 A printout of the line scan mask, which indicates the verticals currently being allowed and denied originating service, can be obtained at the maintenance or traffic TTY by typing input message LLC-MASK-PRNT. When this message is entered, TTIA transfers to LLOD via global entry LLCINP. An LC02 output message containing a 16-bit representation of the present state of the line scan mask is printed, and return is made to TTIA.

Denying and Restoring Nonessential Line Groups Served

7.14 When LLC starts denying service, an LC01-FGD output message is printed to indicate that the first groups of nonessential lines have been denied. As the overload subsides, LLC gradually restores originating service to all lines. When dial tone is denied on the basis of network blockage, LLC will not restore any line groups during the 3-minute interval but will deny nonessential line groups based on 3-second DTST failures. If the network blockage is less than 10 percent after 3 minutes, the line group denied dial tone for the longest time is restored following a successful DTST. For each successful DTST thereafter, LLC restores service to another line group. When dial tone is denied on the basis of 3-second DTST failures, LLC restores service to the nonessential line group that has been denied service for the longest period, each time a DTST is successful.

7.15 In either case (DTST failure or network blockage), the first three DTSTs are nonrevelant if they fail. Thus, when a nonessential line group is restored, five consecutive DTST failures are possible without a further reduction in the nonessential line groups served. If the sixth DTST is successful, originating service is restored to another nonessential line group. If the sixth DTST fails, 50 percent of the nonessential line groups in service will be denied service. Items 6 and 9 of Fig. 21 depict nonessential line groups being restored. Originating service is gradually restored in this manner until all nonessential line groups have originating service. When LLC is denying service or when a 3-second DTST failure occurs, the LLC denying service flag (LLD) is set and the MN DT DLY lamp is lighted. When originating service has been restored to all lines, an LC01-AGR message is printed, LLD is zeroed, and LLC is no longer active. However, LLC remains enabled in the ON mode until another mode is selected. Return is made to ECMP.

Rotating Nonessential Line Groups Served

7.16 Once per minute a check is made to determine if some, but not all, of the nonessential line groups are receiving originating service. If so, the two nonessential line groups that have had originating service for the longest period of time are denied originating service, and the two nonessential line groups that have been denied originating service for the longest period of time are given originating service. If only one line group is being denied, then that line group is either rotated or restored. Since line group rotation is accomplished once per minute and one line group may be restored for each successful DTST (accomplished every four seconds), the probability is that a single line group being denied service will be restored to service before the time occurs for rotation. Rotating the nonessential line groups served is accomplished by updating the line scan mask.

7.17 When line groups are rotated, the number of nonrelevant DTSTs is six. On this basis, nine successive DTST failures would have to occur before additional nonessential line groups would be denied orignating service immediately following rotation of line groups. Rotation actions always apply when at least two nonessential line groups are being served and at least two nonessential line groups are being denied. If this is not the case, certain actions are modified as mentioned in paragraph 7.16 for one line group being denied. If all nonessential line groups are either served or denied, it is not necessary to rotate any nonessential line group.

7.18 Some of the conditions and actions resulting from possible combinations of network blockage measurements and DTST results are summarized in Table U. The number of nonessential lines denied and restored columns indicate the update actions required for the 16-bit scan mask word. The numbers in these two columns do not indicate the absolute number of nonessential lines being served or denied at any given time.

C. LLC AUTOMATIC Mode

7.19 When the LLC-ALLOW-AU input message is entered, TTIA enters LLOD via global LLCLKO. When this mode is selected, the LLC AUTOMATIC flag (LAU) is set, and output message TOC02 is printed. The LLC INH lamp is turned off and return is made to TTIA.

7.20 If LLC is in AUTOMATIC mode and three successive extended DTSTs have occurred when ECMP makes its regular 4-second entry into LLOD via LLCCSM, LLC is automatically enabled (LON is set). (When a regular 3-second DTST fails, the test on that line is extended an additional 8 seconds. Thus the total maximum time for an extended DTST is 11 seconds.) When LLC is enabled, the LLC ON lamp is lighted and the appropriate alarms, bells, and TOC01 output message are generated. In the AUTOMATIC mode, LLC is administered using the 3-second DTSTs and network blocking measurements in the same manner as that described earlier for the LLC ON mode (paragraphs 7.11-7.13). (Since the extended tests start as regular 3-second DTSTs, the extended tests occur at the same rate—one every 4 seconds.) As the service load subsides, originating service is restored to nonessential line groups as explained in paragraphs 7.14-7.18 until LLC is no longer active.

7.21 In the AUTOMATIC mode, LLC is disabled (OFF) when all nonessential line groups have been restored to service and a service overload no longer exists. LLC remains in the AUTOMATIC mode until another mode is selected via the appropriate TTY message: LLC-INH (OFF) or LLC-ALLOW-ON (manual ON).

D. LLC Deactivation

7.22 When input message LLC-INH is entered at the TTY, TTIA transfers to LLOD via global entry LLCLKN. The LLC INH lamp is lighted and output message TOC02 is printed. Return is made to TTIA.

8. ABBREVIATIONS AND ACRONYMS

ACD	Automatic Call Distribution
ADS	Auxiliary Data System
AIT	Abandon Interdigital Time-out
AMA	Automatic Message Accounting
AOVD	Automatic Overload Control Program
CC	Central Control
CCLT	Supervisory Scan and Audit Program
CHGD	Scan Point Change Director Program
CW	Call Waiting
DP	Dial Pulse
DTS	Dial Tone Speed
DTST	Dial Tone Speed Test
EADAS	Engineering Administration and Data Acquisition System
ECIO	Executive Control Input-Output Program
ECMP	Executive Control Main Program
EGO	Equipment Group or Office Count Number
ESS	Electronic Switching System
FCG	False Cross and Ground
IOCP	Input-Output Control Program (1A Processor)
IOU	Input-Output Unit
IREC	Interrupt Level Audit Interface Program

TABLE U

LINE LOAD CONTROL ACTIONS

TIME FOR NETWORK OVERLOAD CALCULATION	TIME FOR ROTATION	(NOTE 1) NETWORK OVERLOADED	(NOTE 2) RESULT OF DTST	DTST FAILURE RESULTED IN CONTROL OVERLOAD (NOTE 3)	NUMBER OF NONESSENTIAL LINES DENIED AND/OR ROTATED	NUMBER OF NONESSENTIAL LINES RESTORED OR ROTATED (NOTE 4)	CORRECTION TO NONRELEVANT DTST COUNT (NOTE 5)
ONCE/3 MIN	ONCE/MIN		ONCE/4 SEC		N=NUMBER SERVED		
Yes	Necessarily Yes	Yes	#	#	N/2* + 2	2	+6
Yes	Necessarily Yes	No	Failure	Yes	N/2* + 2	2	+6
Yes	Necessarily Yes	No	Failure	No	2	2	+6
Yes	Necessarily Yes	No	Success	Can't Occur	1	2	Zero, Then +6
No	Yes	#	Failure	Yes	N/2*+2	2	+6
No	Yes	#	Failure	No	2	2	+6
No	Yes	Yes	Success	Can't Occur	2	2	+6
No	Yes	No	Success	Can't Occur	1	2	Zero, Then +6
No	No	#	Failure	Yes	N/2*	0	None
No	No	#	Failure	No	0	0	None
No	No	Yes	Success	Can't Occur	0	0	None
No	No	No	Success	Can't Occur	0	1	Zero, Then +3

Legend

* Fractional results are rounded up to next integer.

Answer can be yes or no without affecting resulting actions due to other conditions.

Notes:

1. Network Overload: Every 3 minutes (the average holding time of a call) a calculation is made to see if 10 percent or more of the incoming calls to an office experienced network blocking on the talking path during the last 3-minute period. If they have, the office is considered to have a network overload for the purpose of LLC during the next 3-minute period.

2. If dial tone delay exceeds 3 seconds during a dial tone speed test (DTST), the DTST is a failure. An indicator, which tells whether or not the last DTST succeeded, is updated every 4 seconds. This indicator is examined by line load control (LLC) every 4 seconds.

3. If three consecutive relevant DTST failures occur, LLC decreases the number of nonessential line groups served by one-half. Four such reductions result in temporary denial of orginating service for all nonessential line groups.

4. Each successful relevant DTST causes originating service to be restored to the (one) nonessential line group which has been denied for the longest period of time. If the network is overloaded, no line group is restored. Every minute, the two nonessential line groups denied the longest are rotated with the two served the longest.

5. Nonrelevant DTST: While a line group is denied originating service, a backlog of originations may build up in the group. Restoration of the group can produce a surge of originations causing dial tone delay. This surge does not represent the steady state load of the office, so the first three DTSTs following restoration of a group are considered nonrelevant for LLC purposes. (The nonrelevant DTST count is kept in L4NRDT. If nonzero, the count is decremented by one following each DTST.)

I/O	Input-Output
LLC	Line Load Control
LLOD	Line Load Control and Toll Network Protection Program
MARP	1A Processor Maintenance Restart Program
MARS	No. 1A ESS Maintenance Restart Program
MSN	Master Scanner Number
OBB	Ordered Bits Buffer
PATT	Processor Application Transfer Table
PBX	Private Branch Exchange
PIDENT	Program Identification
POB	Peripheral Order Buffer
PROCON	Programmable Controller
PSPD	Permanent Signal and Partial Dial
PT	Program Tag
QI	Queue Indicator
RAM	Random Access Memory
RI	Register Identification
RV	Restore-Verify
SAHO	Hopper and Fixed Length Queue Audit
SARG	Call Register Audit
SIRE	1A Processor System Interrupt Recovery Program
SXJR	Step-by-Step Junior Register
SXS	Step-by-Step
TFCT	Traffic Measurements Program
TMC	Traffic Measurement Code
TOC	Traffic and Overload Control
TSAH	Trunk Seizure and Answer Hopper
TSJR	Timed Scan Junior Register
TTIA	TTY Program
TTY	Teletypewriter
WQUE	Queue Administration Program

9. REFERENCES

A. IM-6A001—Input Message Manual

B. OM-6A001—Output Message Manual

C. PK-6A001—Attributes and Symbols

D. PR-6A004—Executive Control Main Program Listing

E. PR-6A075—Line Load Control and Toll Network Protection Program Listing

F. PR-6A076—Automatic Overload Control Program Listing

G. PR-6A100—Executive Control Input—Output Program Listing

H. Section 231-045-110—Scanning

I. Section 231-045-215—Audits

J. Section 231-045-245—System Performance.