SWITCHING SYSTEMS MANAGEMENT
NO. 2 ELECTRONIC SWITCHING SYSTEM
OPERATIONAL FEATURES
PROGRAM ORGANIZATION

CONTENTS — PAGE

1. GENERAL

1.01 This section describes the organization of the generic program associated with the No. 2 Electronic Switching System (ESS). It provides the network administrator with the fundamentals of stored program control and its relationship to traffic data.

1.02 When this section is reissued this paragraph will contain the reason for reissue.

2. DESCRIPTION

2.01 The No. 2 ESS is under the control of a stored program. It differs radically from electromechanical switching systems in the devices that it uses as well as in the techniques that it employs. In electromechanical systems, each operation in the switching sequence triggers the operation that follows it. In the No. 2 ESS, the program is the trigger and it constitutes the operating intelligence of the system. The organization of the program is strongly influenced by the fact that it must operate in real time; that is, the program must respond promptly to signals and data submitted to it by customers and other switching systems. It must also respond to trouble-detection circuits designed into the hardware to ensure dependable operation. For the program to meet all these demands, it is necessary to establish a hierarchy of program tasks. Some tasks must be performed on a strict schedule; others may be delayed without significant adverse effects.

2.02 Virtually all of the actions of the system are determined by the sequences of instructions coded and stored in memory. Circuitry of eight priority levels can interrupt the continuity of program flow, but the interrupts seize control from one task only to pass it to another. Under certain circumstances the program must be manually interrupted; a restart or initialization process initiated at the maintenance and administration center is exercised in these cases.

2.03 The No. 2 ESS stored program can be divided into three parts: the normal call processing (base level) programs, programs which must execute on a regular basis to handle time-critical operations through a timed interrupt, and programs which are called in by unusual trouble conditions through a demand interrupt. The overall program control plan is shown in Figure 1. Figure 2 shows the basic information flow of the call processing system and indicates the relationship between the circuitry and the programs.

BASE LEVEL PROGRAMS

2.04 The base level includes all call processing programs plus deferrable or low-priority maintenance tasks which are called in at the end of the normal call processing loop. These programs function in a cycle called the base level program scan. Some of the tasks performed at the base level are as follows:

(a) ***Distribution of Supervision to Appropriate Transient Call Records (TCRs):*** All new supervisory reports are dispatched to either the TCR with which this report is associated or to a new TCR if the supervisory report represents a new call. (The TCR is an 8-word memory block in call store used in processing a call through the transient or nontalking state. When a call has reached a stable state, the TCR is released for use on another call). Supervisory reports are detected by the input monitor program which takes control at the beginning of each base level scan.

(b) ***Call Processing Scan:*** The call processing scan accesses the TCR associated with each nontalking call in the system. Any new information associated with a call is processed and an appropriate output response is generated.

(c) ***Maintenance Monitor:*** The maintenance monitor is entered after all TCRs have been accessed. The program performs essential maintenance and administrative tasks including recording of various traffic data, processing teletypewriter (TTY) messages, and performing the necessary audits to ensure that the system is operating normally. Other maintenance tasks which may be deferred are handled as time permits.

2.05 All base level tasks are scheduled by a base level maintenance monitor (BLMM) program that determines which additional tasks are to be performed after the normal call processing scan. The length of time required to complete a single base level scan is nominally 100 milliseconds. Upon completion of all nondeferrable work operations, a check is made to determine whether 100 milliseconds have elapsed. If not, deferrable work operations including audits (programs that locate inconsistencies between different memory records in call store and attempt to correct them) and maintenance activities are performed for the balance of the 100-millisecond base level cycle. The base level loop will exceed 100 milliseconds whenever there is more work in the nondeferrable area than can be completed in this time.

2.06 The responsibilities of the base level maintenance monitor include:

(a) The recognition of changes in the state of the system and the initiation of a proper response to a state change

(b) The control of almost all base level maintenance programs in the office

(c) The monitoring of all maintenance and utility type input messages from maintenance personnel

(d) The initiation of periodic work which must be done on a time-scheduled basis

(e) Miscellaneous functions such as timing and control of system alarm conditions and checking the integrity of the entire base level scan.

SUBROUTINES

2.07 Subroutines are programs that specialize in frequently used functions that are not related exclusively to one type of call or to one phase of a call. The subroutines are called upon, when needed, by various base level programs to perform specialized tasks. Some of the functions of subroutines are as follows:

(a) Request a change in a network configuration or relay circuit states

(b) Obtain translation information (ie, determine the terminal equipment number that corresponds to a given directory number)

(c) Select or idle network paths and service circuits.

2.08 In No. 2 ESS the subroutine is the most effective and widely used technique to keep program size small. Greater efficiency of program storage is obtained through nesting of subroutines (subroutines calling other subroutines). The No. 2 ESS central processor is designed to make full use of the nesting concept. Higher-level subroutines have more complex tasks to perform, but these tasks are accomplished primarily by moving data appropriately and calling lower-level subroutines for further action. This structure leads to a basic call-handling program that consists mainly of calls to subroutines. In addition, data structures must be so arranged that information may be readily passed and returned from subroutines. Additional program efficiency is obtained by combining several different tasks into one program. Although these combinations lead to some superfluous operations at the cost of real time, they permit program savings.

INTERRUPT LEVEL PROGRAMS

2.09 Interrupt level programs break into the base level loop on a periodic timed basis (input/output interrupt) or on a demand basis caused by a trouble condition. When the interrupt level program has completed its allotted work, control is returned to base level at the the point of interruption.

2.10 Although the demand interrupt takes precedence over the timed interrupt, there are virtually no priorities within a given level. Instead, all tasks are handled as they are encountered and each program does as much work as possible on one job before proceeding to the next. The demand interrupt occurs only from an indication of machine failure or from a manual request. The timed interrupt is used only for those tasks where a small delay may cause an error or failure. All other tasks are handled on the main program (base) level.

2.11 There are facilities for interrupts of eight priority levels, numbered 0 through 7 from highest to lowest priority. Three of these levels are used: level 2 is the high-prority maintenance interrupt (HPMTC), level 5 is the input/output processing interrupt (IOPRO), and level 6 is the low-priority maintenance interrupt (LPMTC). In the absence of all interrupts, the system is said to operate at base level. An interrupt can seize control from base level or from an interrupt level of lower priority. Control can be in only one level at a time and the state of an interrupt status register keeps track of the active level. The program segment being executed at a given time does not necessarily identify the level since a few programs are executed in more than one level (at different times). However, most programs are executed on specific levels, with the bulk of them being base level programs. Figure 3 depicts the processing hierarchy of the various interrupts.

A. Input/Output Interrupt

2.12 The timed or input/output interrupt used by call processing for nearly all input and output functions is called the IØ25 interrupt. This interrupt which occurs every 25 milliseconds stops the base level program, stores its program location and all machine registers in call store, and initiates those tasks required at that time. Although all of the tasks performed by the interrupt need to be performed more often than could be done by the base level program loop, very few need to be done every 25 milliseconds. The various tasks are distributed among several interrupts at their required frequency as shown in Table A. This distribution of tasks attempts to equalize the amount of time required by each IØ25 interrupt.

2.13 Those functions performed by the IØ25 include the following:

(1) Scanning for supervisory changes in lines, trunks, and service circuits

(2) Distributing orders to peripheral circuits

(3) Administering originating registers in the sending and receiving of digits

(4) Scanning all TTYs for new inputs or outputting new characters to active TTYs

(5) Maintenance functions requiring critical timing.

2.14 The IØ25 interacts with the base level programs by means of buffered data in call store. The areas in call store used for passing data between base level programs and input/output programs are called buffers. Buffers used to pass input data to base level programs are called hoppers.

B. High-Priority Maintenance Interrupt

2.15 The high-priority maintenance interrupt (level 2) seizes control from both base level programs and all lower priority interrupts. It is initiated by mismatches between the on-line and off-line call stores, by some peripheral unit and input/output errors, or by manual request. All three sources of interrupts have the same priority and, as mentioned earlier, block each other until their respective tasks are complete. These error signal interrupts immediately initiate trouble recovery programs and after the appropriate recovery actions (eg, switching the on-line/off-line configuration of the control units or a peripheral controller) the problem is passed on for further resolution to the lower-priority base level maintenance programs. This also allows the base level monitor to regain control of the system.

> ***Note:*** No call processing takes place during a maintenance interrupt.

C. Low-Priority Maintenance Interrupt

2.16 A low-priority maintenance interrupt (level 6) occurs only at the end of a high-priority maintenance interrupt that was caused by a mismatch. The high-priority maintenance interrupt, following a mismatch, calls the low-priority maintenance interrupt in order to allow the IØ25 interrupt to occur. The low-priority maintenance interrupt will not take place without first having a high-priority maintenance interrupt.

INITIALIZATION

2.17 Without manual intervention or the detection of trouble, both control units run simultaneously, executing the same instructions in synchronism, with the outputs of the off-line control unit to the periphery and maintenance administration center inhibited. If the control units switch their on-line/off-line configuration while the off-line control unit is out of synchronism or if they go into a multiple switching mode from which they cannot recover, the program is forced to a known fixed location and restarted to provide an orderly return to the beginning of the base level cycle. This is called a maintenance reset function (MRF). The initial source of trouble may be equipment (hardware) or stored program (software) irregularities or human error. The restart or initialization process reestablishes the continuity of program flow by selectively clearing areas of call store.

2.18 Those areas of call store which are cleared are determined by the degree of initialization or level count. Table B lists the types and level count of system initializations and the effects they have on call processing. The level count increases until the system has eliminated the difficulty and is able to restore the system to a working condition. The initialization is considered to have been manually caused if the ENABLE key is depressed at the emergency action panel.

A. Areas of Call Store Affected By An Initialization

2.19 With reference to Table B, certain areas of call store, defined below, are cleared by specified level counts.

(a) ***Traffic Work Table (TWT):*** The TWT, prepared by the network administrator, is an area in memory used to specify the time of day by clock-hour and quarter-hour and the day or days of the week at which certain traffic work operations are to be performed by the system. It is used to control the collect and the print times of traffic and plant measurement schedules. In addition, it is used by the maintenance forces to schedule automatic maintenance routines. There are two copies of the TWT in the system, one in call store and one in program store. In the event of a loss of call store data due to a transient initialization, an audit program will transfer the contents of the program store TWT to the call store TWT after the entry of specified TTY inputs by maintenance personnel (see 2.22), so that traffic and maintenance schedules will be continued.

(b) ***Originating Register (OR):*** Originating registers are used by the input/output interrupt programs for receiving and sending digits.

(c) ***Terminal Memory Record (TMR):*** A TMR is an area of call store assigned to each trunk, junctor, and service circuit used to record the current established paths through the switching network. Information is recorded in the TMR by path selection routines. A stable TMR indicates that a call associated with the particular circuit, junctor, or trunk is in a talking state while a transient TMR denotes a call in a nontalking state.

(d) ***Transient Call Record:*** A TCR is associated with each active nontalking (transient) call in the system. The TCR consists of eight call store words and controls the progress of the call from origination until an answer signal has been received and the connection is placed in a talking state. The TCR also controls the actions taken to disconnect a call.

(e) ***Peripheral Order Buffer (POB):*** A POB is an area of call store used to record data for changing the states of various trunk and service circuits. These data are transmitted to peripheral units (network frames, peripheral decoders, and scanners) to establish a particular connection or to cause a particular action. When a base level program has information the program desires to send to a peripheral unit, the POB loading program is called to select and load a POB. The POB execution program then controls the rate at which the information is sent from the POB to the peripheral units.

(f) ***Network Map:*** The network map is an area of call store that maintains a record of the busy-idle status of every link and junctor in the network. It also contains the line status bits.

2.20 Because some levels of initialization are lengthy, it is important that personnel know that the initialization program is in progress. Upon entry to every initialization, all lamps are lighted on the display buffer located on the maintenance center control and display panel.

2.21 The network administrator is informed via the network administration TTY that an initialization has occurred. Each time an initialization takes place, the TTY prints out that a clearing of call store may have occurred and indicates the type of initialization in progess. This is a supplemental output message; a complete initialization report is printed on the network maintenance TTY. In all cases, consult the output message manual, OM-2H200, for the specific message.

2.22 Of particular importance to the network administrator is the extent to which an initialization affects office data. There are three different types of system initialization which will affect the traffic and/or plant measurements. They are as follows.

(a) ***Manually Requested Stable Initialization:*** Many traffic and plant registers are declared ***stable*** in call store memory. This means that these registers can only be cleared by a manually requested stable initialization. Refer to Bell System Practices Section 232-113-301, System Initialization. Whenever a stable initialization occurs and clears all traffic and plant registers, a clear (CLR) message is printed on the network administration and network maintenance TTYs to indicate that the action has taken place.

(b) ***Automatically or Manually Induced Transient Initialization:*** All service circuit, trunk, PBX, and junctor group measurements are transient in call store memory. Thus, the registers for these measurements will be cleared by a ***transient*** initialization. The call store copy of the TWT is also cleared by a transient initialization. Whenever this initialization occurs the load service measurements (LSM) schedule is printed on the network administration and network maintenance TTYs. The LSM schedule contains the same measurements as the Q-schedule except that the measurements contain data collected from the last clock quarter-hour to the time of the printout. Following restoration of the date and time clock by maintenance personnel, the call store copy of the TWT is updated by the program store copy.

(c) ***Recent Change Initialization:*** The recent change area measurements of the plant (PLT) schedule can only be cleared by a manuallly requested ***recent change*** initialization. Whenever a recent change initialization is requested a transient initialization is also performed.

B. Dynamic Service Protection

2.23 Following any stable clear initialization, the status of dynamic service protection (DSP) is dependent upon the generic program in use. For offices with programs preceding LO-1, Issue 4.6, and EF-1, Issue 3.4, DSP is allowed. For offices with LO-1 Issue 4.6, and EF-1, Issue 3.4, DSP is prevented from being implemented. (Due to the unique operational features of DSP, ***it is recommended that this feature be denied in attended or monitored offices.***) If the system was in DSP at the clock quarter-hour, the Q-schedule will be printed out on both the network administration and network maintenance TTYs. For further information concerning DSP refer to Dial Facilities

Management Practices, Division H, Section 10d(1), Dynamic Service Protection.

2.24 The base level maintenance monitor program zeros the level count after a specified number of program scans have successfully elapsed since the last initialization. An initialization sequence, by definition, is the length of time from the first initialization until the level count is zeroed by the base level maintenance monitor.

2.25 Once the level count reaches the highest permissible level and the program has taken the action at that branch, the level count is reset to start again from zero. This prevents overloading the system with a flood of data resulting from a major level count.

2.26 A complete description of the initialization process is contained in Dial Facilities Management Practices, Division H, Section 10d(6), Initializations.

3. REFERENCES

3.01 Additional information can be obtained by consulting the following sources:

SECTION	TITLE
232-103-101	Call Processing Description (Bell System Practices)
232-113-301	No. 2 ESS System Initialization Procedures (Bell System Practices)
966-200-100	No. 2 ESS General Description (Bell System Practices)
10d(6)	Initializations, (Division H, Dial Facilities Management Practices)
TG-2H	Translation Guide, Division 10, Traffic Measurements
PD-2H010	Common Systems Maintenance Interrupt Monitor
PD-2H011	Common Systems Initialization Program
PD-2H013	Common Systems Base Level Maintenance Monitor Program
PD-2H016	Common Systems Teletype Base Level and Interrupt Program
PD-2H107	Input/Output 25-Millisecond Interrupt Program
PD-2H124	Application Portion of Initialization Program
PD-2H125	Application Portion of the Base Level Maintenance Monitor Program
PD-2H135	Overload Control Program
PD-2H201	Program System Description
PD-2H202	Call Processing Monitor Program
IM-2H200	Input Message Manual
OM-2H200	Output Message Manual

TABLE A

FREQUENCY OF TASKS PERFORMED DURING TIMED INTERRUPTS

TASK	INPUT/OUTPUT 25-MS INTERRUPTS TIME IN MS																		
	0	25	50	75	100	125	150	175	200	225	250	275	300	325	350	375	400	425	450
Line scanner check	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Line hit list rescan	X		X		X		X		X		X		X		X		X		X
Fast trunk scan		X		X		X		X		X		X		X		X		X	
Normal trunk scan	X				X				X				X				X		
Trunk hit list rescan			X				X				X				X				X
Tone digit check		X		X		X		X		X		X		X		X		X	
Dial pulse digit check	X					X					X					X			
Sending check		X		X		X		X		X		X		X		X		X	
MF digit turn-off	X		X		X		X		X		X		X		X		X		X
Peripheral order distribution		X		X		X		X		X		X		X		X		X	
TTY input/output	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AMA output	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Maintenance functions	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bylink trunk scan	X		X		X		X		X		X		X		X		X		X
Set new digit flag	X					X					X					X			

Note: Task is performed during interrupts marked with an X.

TABLE B

SYSTEM INITIALIZATIONS

TYPE INITIALIZATION	SOURCE REQUESTED	LEVEL COUNT	EFFECT ON CALL PROCESSING
Nominal (no memory cleared)	Automatic	1	Fails all sending originating registers (ORs)
Partial clear	Automatic	2	Partially clears areas of call store
		3	Same as level count 2 except more drastic
Emergency audit	Automatic	4	• Idles all peripheral order buffers (POBs) • Idles all transient call records (TCRs) except designated TCRs which are checked for correctness • Idles ORs* • Idles all transient terminal memory records (TMRs)* • Restores all line ferrods to enable customer origination • Restores system network map *Except those associated with designated TCRs
	Automatic and manual	5	Same as level count 4; occurs only if 4 does not complete
Transient clear	Automatic and manual	6	• Zeros all transient data except TCRs and transient TMRs • Forms a list of customer lines involved in TCRs • Idles all TCRs • Sets up a special POB to restore customer line ferrods upon resumption of call processing on those lines that were involved with a TCR • Calls in a TMR audit to zero transient TMRs, a line status bit (LSB) audit to reconstruct line status bits for stable calls and a network audit to rebuild the network map based on stable calls • Call store copy of the traffic work table (TWT) is cleared • LSM schedule printed on traffic and maintenance TTYs (call store copy of TWT restored using program store copy)
Recent change clear	Manual		• Clears recent change data such as service orders, customer-originated changes to custom calling features, traffic and plant measurements • Disables calls in transient state causing an emergency audit • Performs transient clear initialization *Note:* A backup tape of service order information should be available when a recent change clear is initiated to restore data following the initialization.

TABLE B (Cont)

SYSTEM INITIALIZATIONS

TYPE INITIALIZATION	SOURCE REQUESTED	LEVEL COUNT	EFFECT ON CALL PROCESSING
Stable clear	Manual	6	• Clears all of call store except recent change area • Idles all trunks and service circuits • Restores all idle line ferrods after resumption of call processing • Causes network administration and maintenance TTYs to print out that all traffic, plant, and performance measurement have been lost • With LO-1, Iss. 4.6, and EF-1, Iss. 3.4, after any stable clear initialization, dynamic service protection (DSP) is denied. (Prior to LO-1, Iss. 4.6 and EF-1, Iss. 3.4 DSP is allowed.)
Recent change and stable clear	Manual	6	• All calls are disconnected • Zeros all transient data • Zeros all recent change data in call store • Zeros all stable data in call store and initializes the physical equipment

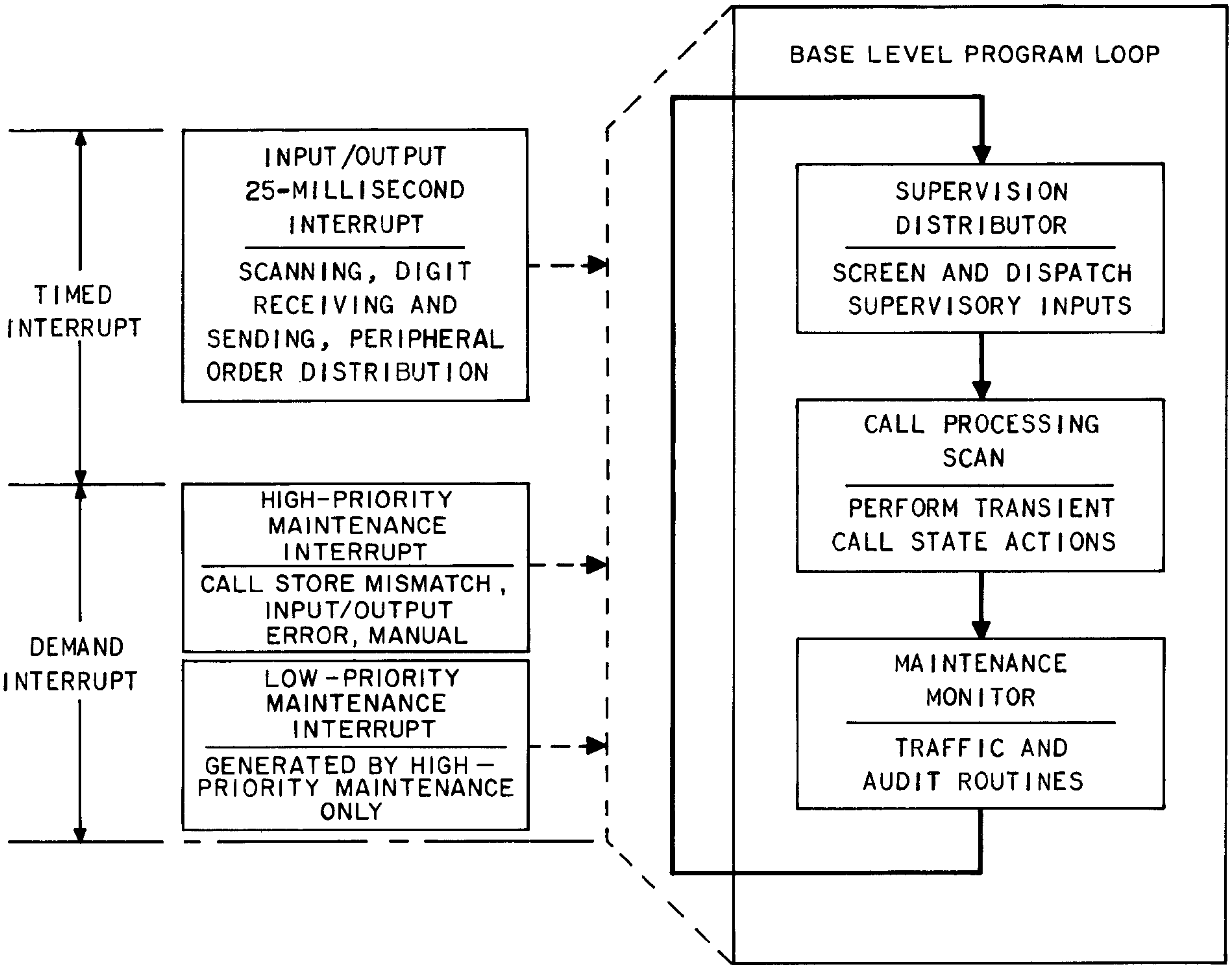
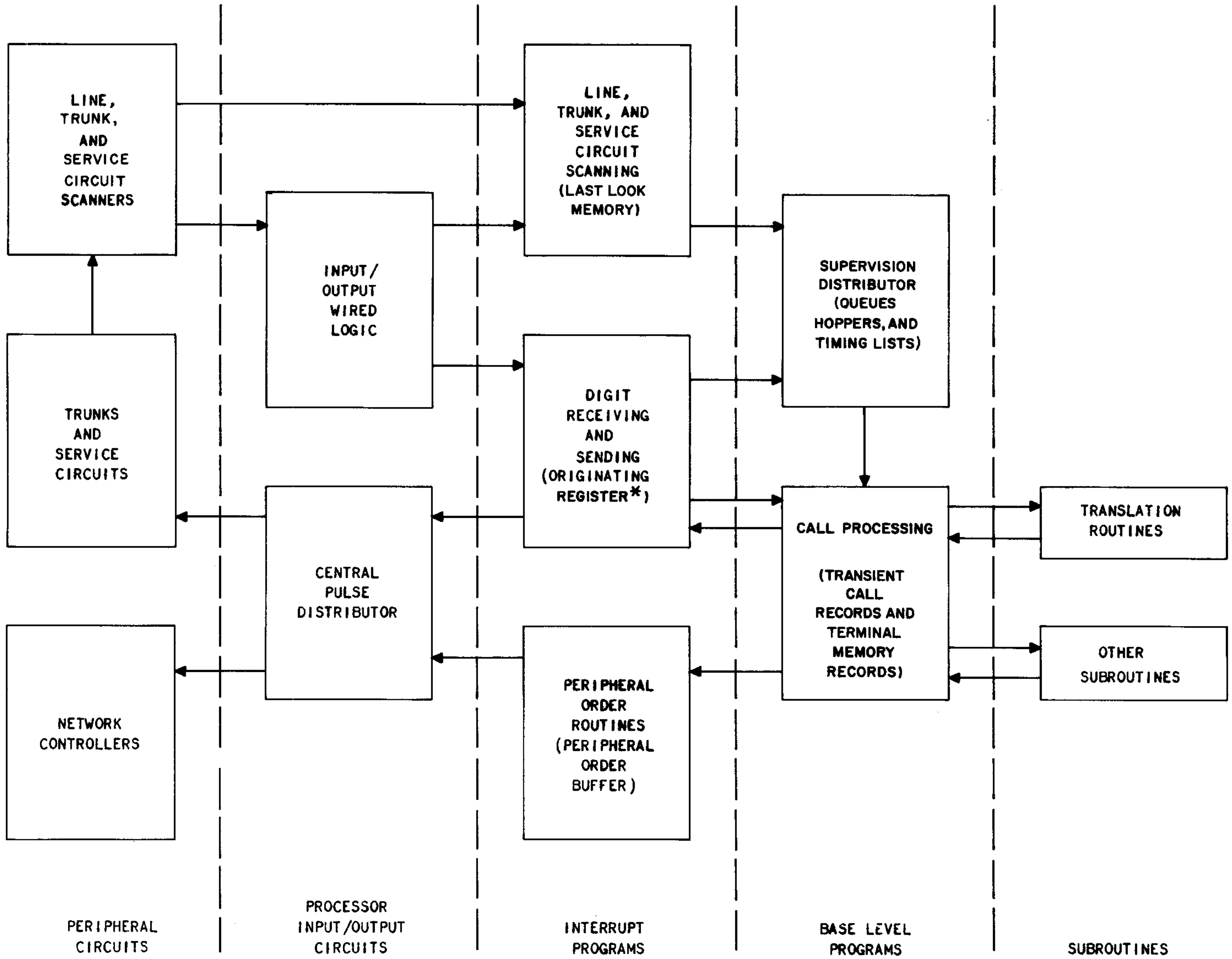


Fig. 1 —Program Control Plan



( ) CALL STORE RECORD USED BY ASSOCIATED PROGRAM
* AND ORIGINATING REGISTER EXTENSION IN EF-I

Fig. 2—Call Processing Information Flow

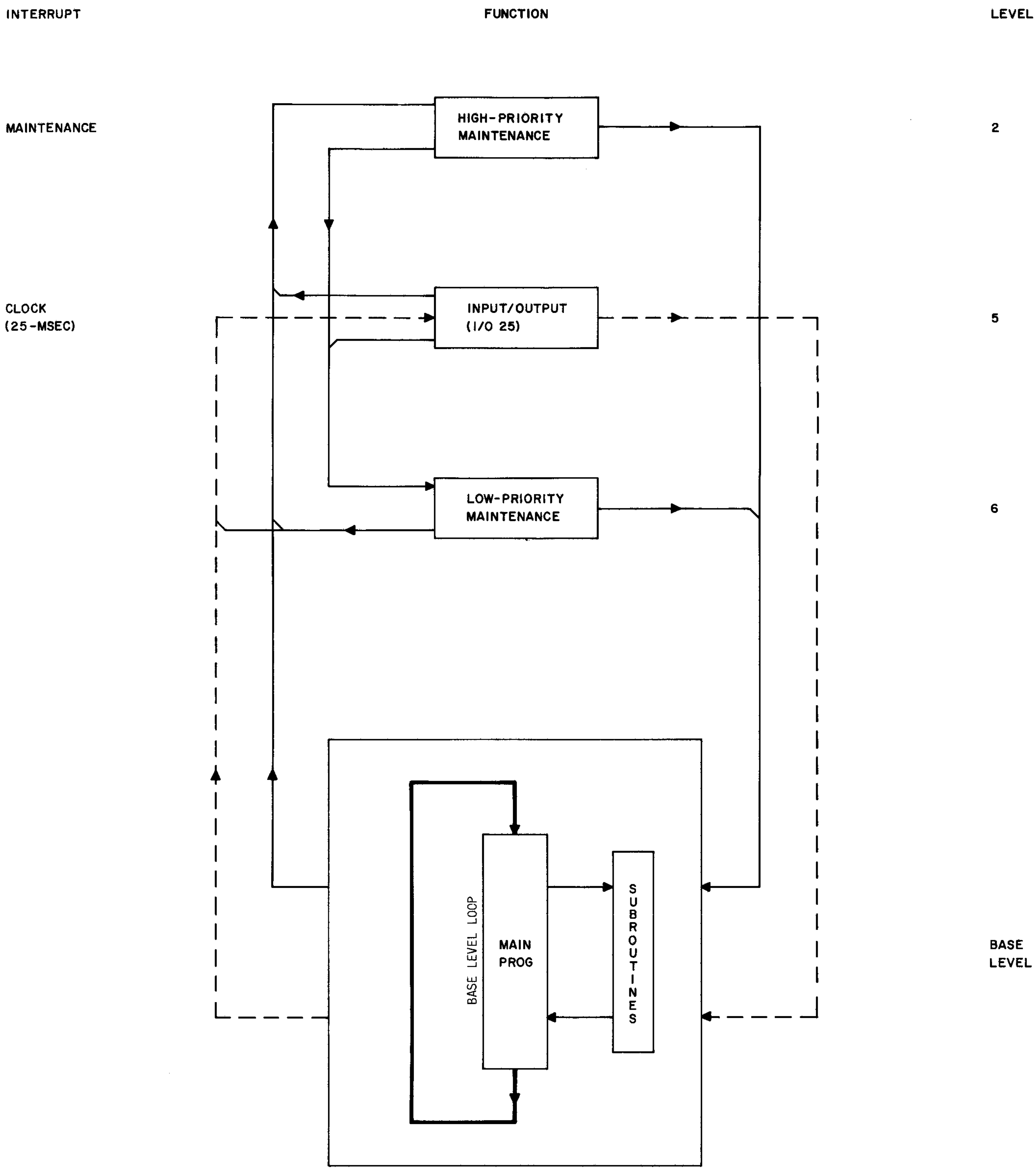


Fig. 3—Program Hierarchy