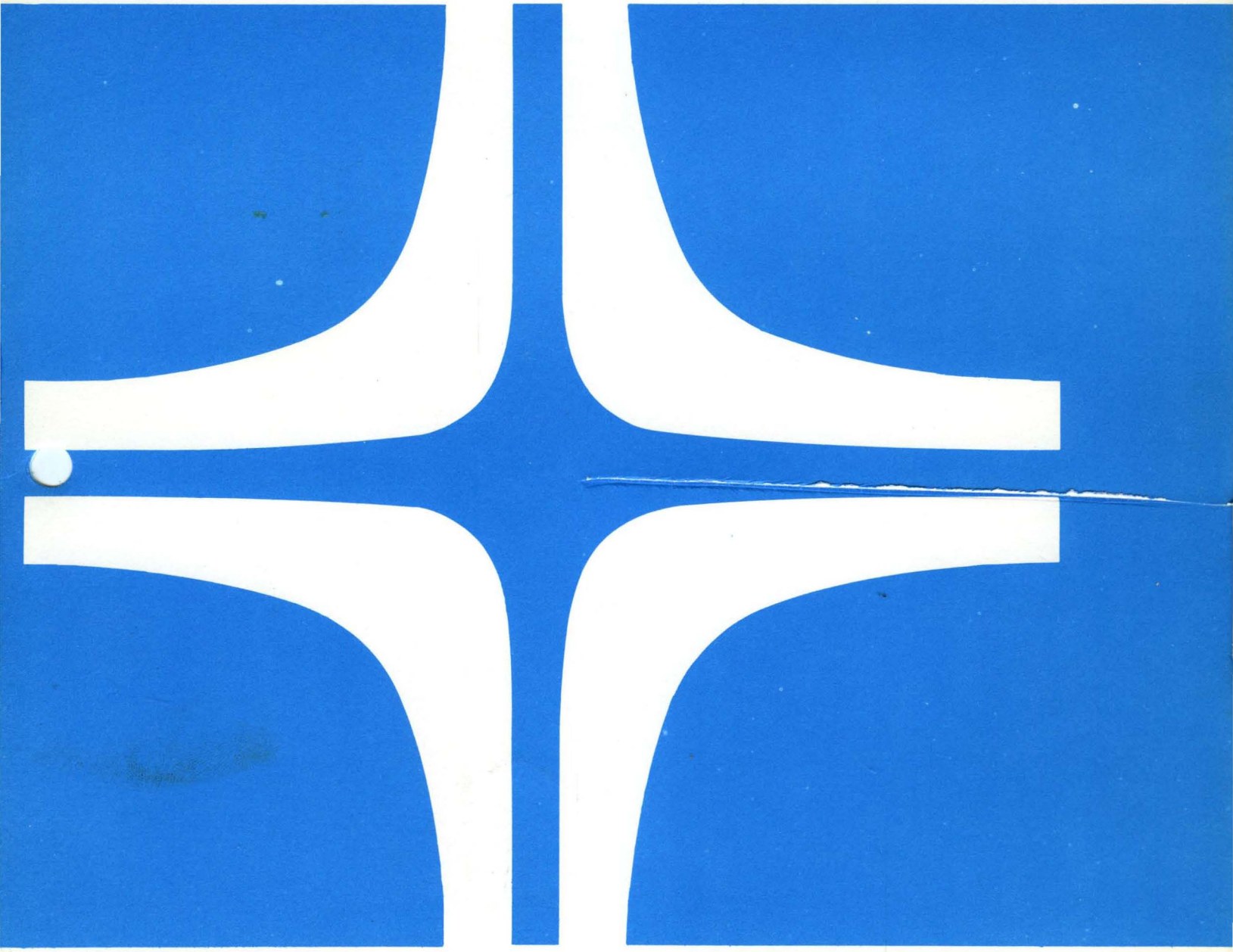
1100-C/SP SOFTWARE SYSTEM VISUALS

1100 SERIES

Preliminary



SPERRY UNIVAC
COMPUTER SYSTEMS

EDUCATION CENTER

UE-7026

CONTENTS

SECTION I

C/SP HARDWARE OVERVIEW

INDEX SECTION I C/SP HARDWARE

INTRODUCTION TO C/SP

The Communications/Symbiont Processor is a high performance internally programmed communications concentrator multiplexer intended for use as an on-site communications subsystem. The C/SP can control low and high speed communication lines and provide interfacing to a central computer system.

The primary purpose of the C/SP is to unburden the central computer software of the need for complex communication handlers and sub-routines, thus relieving the load on the central computer. The throughput of the system is thus increased by the reduction of central system overhead. The C/SP is utilized to increase speed and efficiency of on-line message switching, data manipulations, formatting, editing, translation, and verification.

As a remote, the C/SP effects reduced line costs and unburdens the central complex by performing message/data manipulations and exchanging all data with it over high speed lines.

The C/SP consists of three main sections:

Memory
Central Processing Unit
I/O Section

C/SP Software consists of:

Terminal Management Supervisor (TMS)
Message Control Program (MCP)
Terminal Management Control Routine (TMCR)
Communication Control Routines (CCR)
Host Handler
Symbiont Control Program (SCP)

Host Software consists of:

C/SP Symbiont
C/SP Handler
Initial Load Routine (non-resident)
Mass Storage Symbiont (non-resident)
Communication Routine (non-resident)
Assembler Support Processor
Collector - Support Processor
Simulator Support Processor

HARDWARE BLOCK DIAGRAM

All information transfers to and from the C/SP are handled by a maximum of seven channels designated as zero through six. Channels Ø and 6 are reserved for the General Purpose Communications and Special Device respectfully.

Channel Ø - GPCC: The link between the main storage and the communications line terminals (CLT's). The GPCC control's 8 CLT's (half-duplex) expandable to 64 CLT's (half-duplex) in increments of 8. GPCC expansion at this time may be 2. In the future 4 GPCC's may be allowed per C/SP.

Channel 6 - Special Device Channel (SDC): To provide the means for local program loading and maintenance using an 80 card/minute reader. In the future, a printer and keyboard may be added.

Channels 1 - 5 - May be used for the Adapter, Multiplexer, and Selector. The Multiplexer/Selector are alike in that they can drive up to 8 control units and each control unit may control up to 16 devices (9000 peripherals, discs). The MUX/SEL are different in respect that the Selector Channel locks on to a control unit and drives the device, whereas the Multiplexer multiplexes through the control units to any device. The MUX operates in excess of 85K bytes/second. The Selector operates in excess of 625K bytes/second.

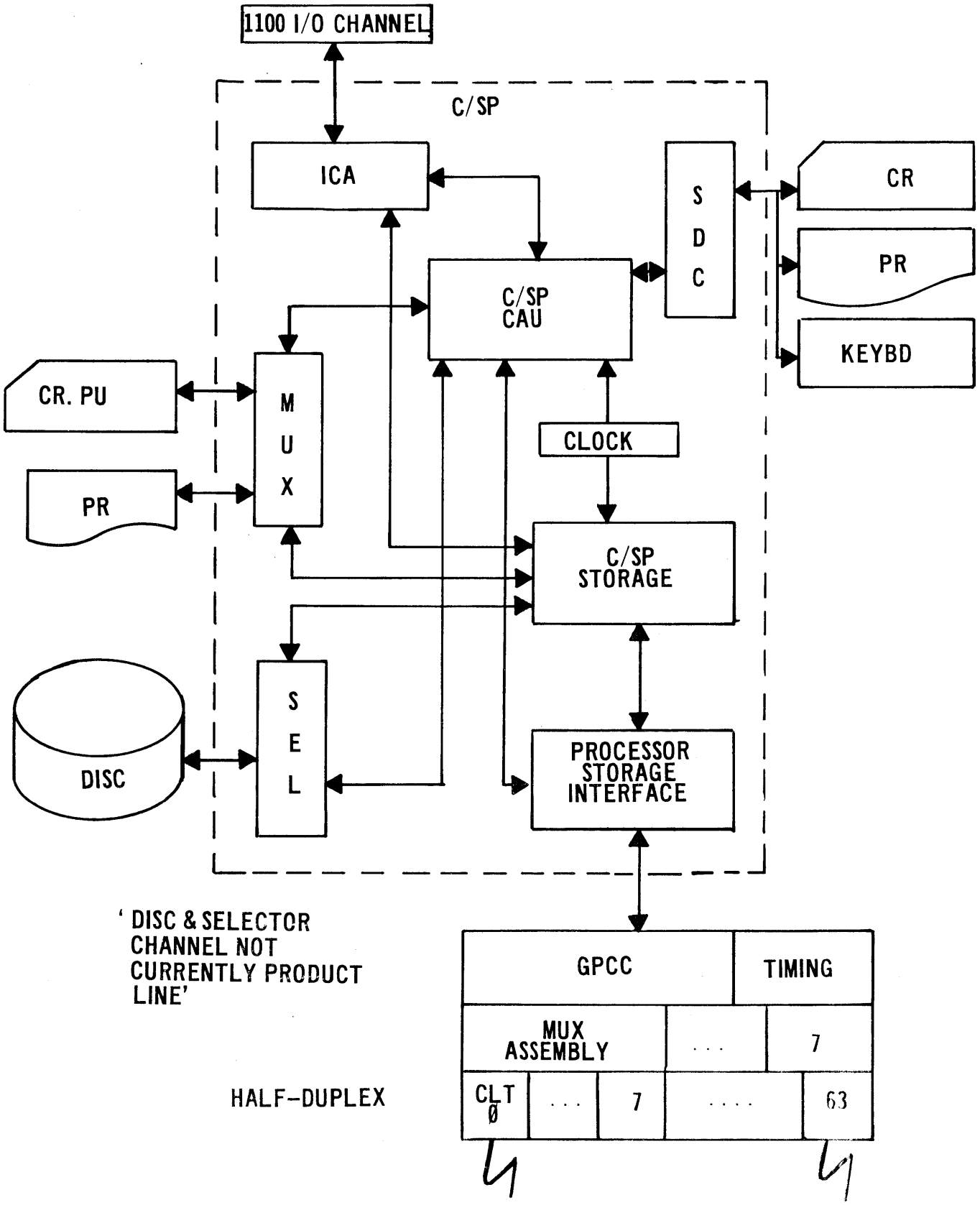
The Intercomputer Adapter (ICA) - Provides the means of communication between the C/SP and the Host Computer. The ICA operates at 300K 36-bit words per second on an Internally Specified Index (ISI) basis.

Special Purpose Communications Channel (SPCC) - Operates in 2 modes; either single duplex or full duplex.

Single Duplex - Consists of 1 full duplex CLT which operates at 57K bytes per second.

Full Duplex - Consists of 8 full duplex CLT's which operate at a combined maximum in excess of 300K bytes/second.

C/SP HARDWARE BLOCK DIAGRAM



FIXED MEMORY

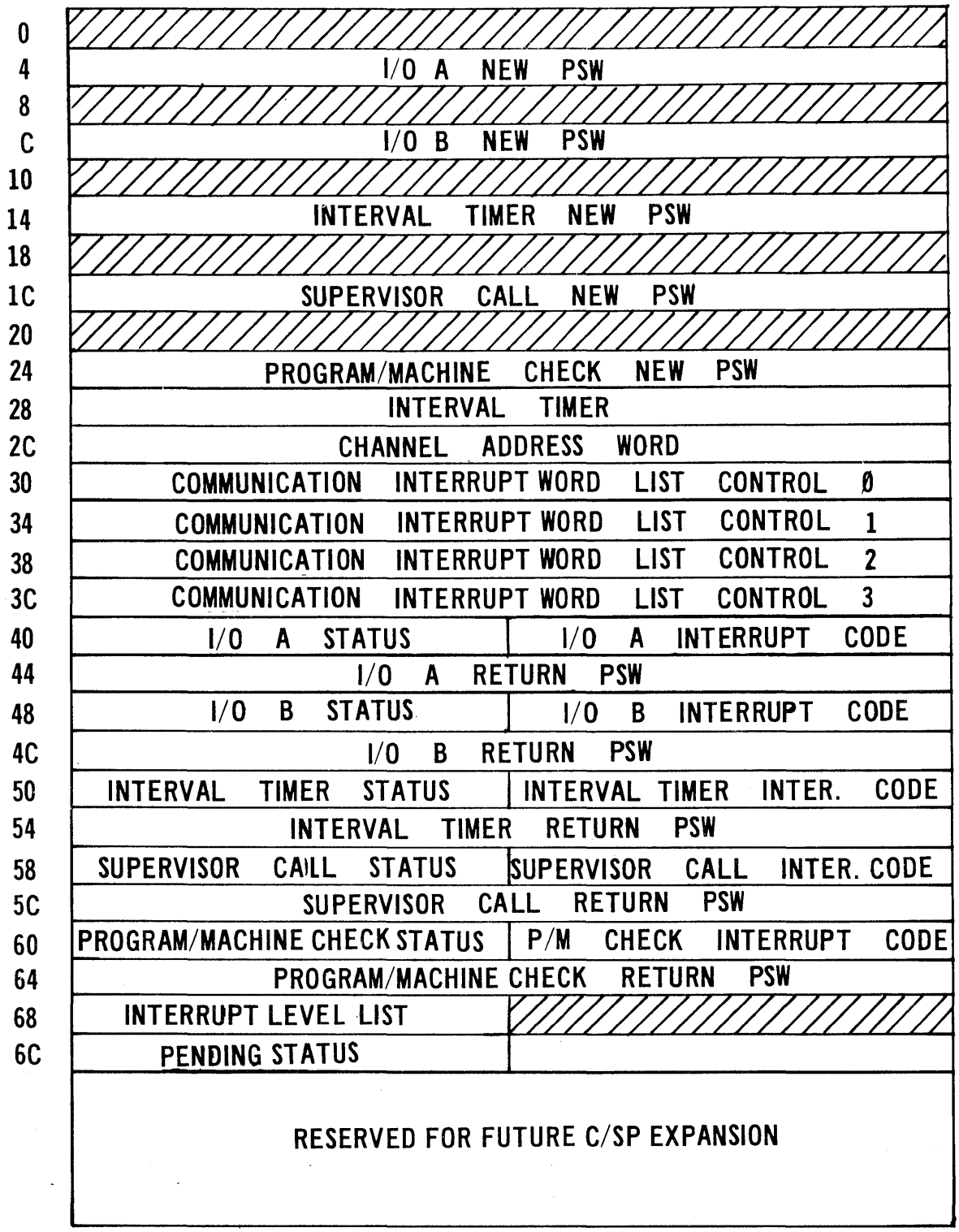
The low order memory locations are used by the C/SP in controlling the initiation of I/O commands, interrupt execution, and Internal Timer operation.

The area which is reserved for future C/SP expansion is used for four more CIW Lists when more than one GPCC is utilized in the C/SP System.

Address 68_{16} and 69_{16} are for the Interrupt Level List indicating a CIW was tabled by the GPCC. The foremost significant bits of address 69_{16} are set each time the GPCC tables an interrupt and each bit correspond to the list which was used. If more than one GPCC in the system, the same four bits are used.

Pending Status - Allows the software to pick up status information without requiring the acceptance of an interrupt.

FIXED MEMORY



UNASSIGNED MAY BE USED AS NORMAL STORAGE

PROGRAM STATUS WORD

PSW - Contains the information required for execution. The PSW is always zero upon initialization.

3 classes - current (Program in control)
- becomes return upon interrupt
- new is fetched corresponding to interrupt type

Upon interrupt the PSW for the intrrupted program is stored by type of interrupt in a certain location in fixed memory and a new PSW is placed in the PSWR.

Mask bits (M) - when set to zero's - inhibits all interrupts. If set to 1 - allows the interrupt.
I/O A Channel Ø. Channel Ø (GPCC) interrupts are tabled.
I/O B Channels 1-6. Interrupts are stacked until allowed, then requested.
Last interrupt (7) for the interval timer.

P - Ø - supervisor mode (allows execution of the priviledged instructions).
1 - Problem mode (or user mode)

CC - condition codes: (vary with the instruction, some use only 2 conditions)

ØØ - Equal
Ø1 - Less than
1Ø - Greater than
11 - Overflow

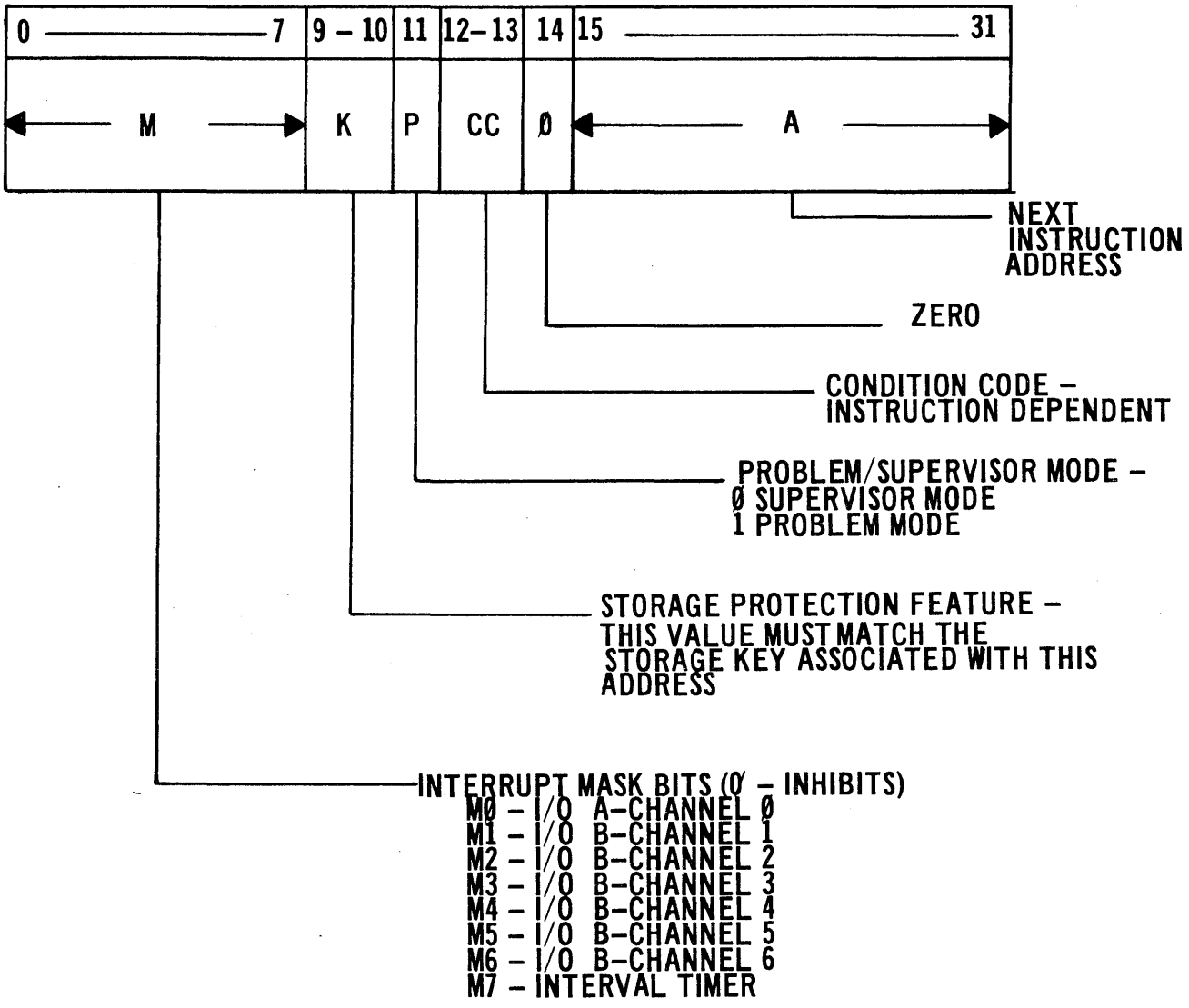
K - storage protect

Ø - Exec can reference anything no matter what the program key.
1 - Problem program
2-7 - Values for the user

Address = 18 bits

Upper 6 bits point to one of the hardware registers associated with each 2048 byte bank.
If the value in that register is equal to the value of the K-field in PSW, reference to the block of storage is allowed.
Otherwise an error occurs.

PSW


0 — 7
9 – 10
11
12–13
14
15 — 31
M
K
P
CC
Ø
A
NEXT INSTRUCTION ADDRESS
ZERO
CONDITION CODE – INSTRUCTION DEPENDENT
PROBLEM/SUPERVISOR MODE –
Ø SUPERVISOR MODE
1 PROBLEM MODE
STORAGE PROTECTION FEATURE – THIS VALUE MUST MATCH THE STORAGE KEY ASSOCIATED WITH THIS ADDRESS
INTERRUPT MASK BITS (Ø – INHIBITS)
MØ – I/O A–CHANNEL Ø
M1 – I/O B–CHANNEL 1
M2 – I/O B–CHANNEL 2
M3 – I/O B–CHANNEL 3
M4 – I/O B–CHANNEL 4
M5 – I/O B–CHANNEL 5
M6 – I/O B–CHANNEL 6
M7 – INTERVAL TIMER

C/SP INTERRUPT SEQUENCE

Interrupts

Automatic means to alert the C/SP processor to an exceptional or unexpected condition (end of I/O operation, program errors, machine errors, etc.) thus directing the processor to the appropriate program routine for handling the interrupt. The system provides for interrupting any task in order to take on a task of higher priority.

Interrupt Sequence: 1. Begins by transferring the current PSW to fixed location in storage ('Return' PSW) 2-3. Then the status half-word and interrupt code half-word is transferred to the assigned location. 4. 'New' PSW is fetched from the fixed location. 'New' becomes 'current' PSW and after processing, the processor can return to the interrupted program (load 'return' PSW) via the LPSW instruction. In certain cases of simultaneously presented interrupts, it is possible to execute more than one interrupt sequence before instruction processing is resumed.

Interrupt Class/Priority: There are 5 classes of interrupts; Supervisor Call, Program-Machine check (PMC), Interval Timer I/O B, I/O A. Within the classes may be several specific sources of interrupt requests.

I/O A, I/O B, and interval timer interrupt requests are controlled by the system mask bits in the 'current' PSW. If bits are set to zero the corresponding channel is not permitted to interrupt. No mask bit control exists over the supervisor call class and all interrupt requests are accepted (in this class). PMC interrupt requests are always accepted unless the processor is executing a PMC interrupt subroutine.

Interrupt Priority: It is possible to execute more than one interrupt sequence between instruction executions. The system mask bits (PSW) become effective immediately and therefore control subsequent interrupt sequences.

The order of executing the contiguous interrupt sequences varies and is a function of the pending instruction requests and mask bits.

The interrupt sequence selected is determined by the following priority:

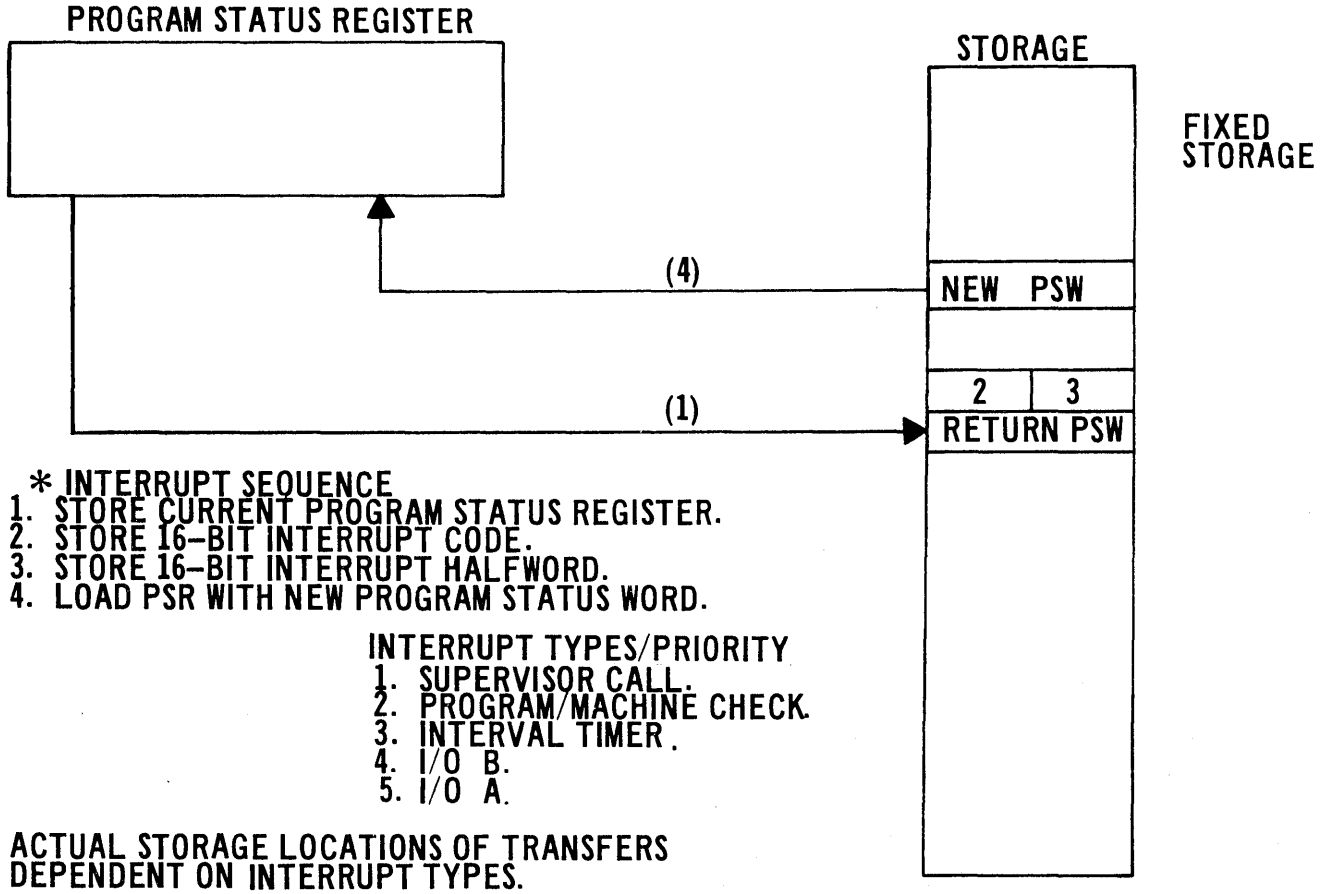
1. PMC (invalid address strg. protection, or parity, 2. Interval Timing, 3. I/O B, 4. I/O A, 5. PMC (power failure, invalid operation, etc.)

Supervisor Call is not included because the storing and fetching of PSW's is done in the execution phase of the instruction and should not be thought of as an interrupt sequence in the context described.

This does not determine the order, only indicates which interval sequence is selected when simultaneous interrupt requests are made. Example: If I/O A and I/O B interrupt requests are presented simultaneously, I/O B is processed first.

The final contents of PSW determine which interrupt subroutine is entered, so the order of executing these subroutines is reversed from the order of executing the Interrupt Sequence.

INTERRUPT SEQUENCE



* INTERRUPT SEQUENCE
1. STORE CURRENT PROGRAM STATUS REGISTER.
2. STORE 16-BIT INTERRUPT CODE.
3. STORE 16-BIT INTERRUPT HALFWORD.
4. LOAD PSR WITH NEW PROGRAM STATUS WORD.

INTERRUPT TYPES/PRIORITY
1. SUPERVISOR CALL.
2. PROGRAM/MACHINE CHECK.
3. INTERVAL TIMER.
4. I/O B.
5. I/O A.

* ACTUAL STORAGE LOCATIONS OF TRANSFERS DEPENDENT ON INTERRUPT TYPES.

INTERRUPT CLASS AND CODE

During the interrupt sequence the status halfword and interrupt code halfword are transferred to the assigned locations in fixed memory.

I/O A class interrupt is assigned to channel zero. Channel 0 is reserved for the GPCC and the bits identifying the channel ID, channel #, device address are all set to 0 by hardware as the channel ID and # equal zero and the device address is contained in the CIW.

I/O B class interrupts are assigned to channels 1-6. Bit 16 is set to 0, bits 17-19 are the channel identification code, bit 20 is set to 0, bits 21-23 are the channel number (1-6), and bits 24-31 are the device address.

The Interval Timer interrupt code halfword is always set to 0 by the hardware (timer).

PMC has a value for each type of interrupt within this class.

Supervisor Call - the 'R'bits are the bits from the SVC fields R1 & R2.

INTERRUPT CODE HALFWORD

INTERRUPT CLASS	INTERRUPT CODE
I/O A CHANNEL Ø	BITS 16 — 31 0 C C C 0 0 0 0 D D D D D D D D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	CHANNEL:
I/O B	I. D. NUMBER DEVICE ADDRESS
CHANNEL 1	0 C C C 0 0 0 1 D D D D D D D D
CHANNEL 2	0 C C C 0 0 1 0 D D D D D D D D
CHANNEL 3	0 C C C 0 0 1 1 D D D D D D D D
CHANNEL 4	0 C C C 0 1 0 0 D D D D D D D D
CHANNEL 5	0 C C C 0 1 0 1 D D D D D D D D
CHANNEL 6	0 C C C 0 1 1 0 D D D D D D D D
INTERVAL TIMER	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PROGRAM/MACHINE CHECK (PMC)	
POWER FAILURE	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
PARITY	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
INVALID OPERATION	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
PRIVILEGED OPERATION	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
STORAGE PROTECTION	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
INVALID ADDRESS	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
FORCE PMC SWITCH	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
MONITOR SWITCH	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
SUPERVISOR CALL	0 0 0 0 0 0 0 0 R R R R R R R R

C – CHANNEL IDENTIFICATION CODE
- 0 0 0 – GPCC
- 0 0 1 – SDC
- 0 1 0 – ICA
- 1 0 0 – SELECTOR
- 1 0 1 – MULTIPLEXER

INTERRUPT CLASS AND STATUS

The status halfword is stored during the interrupt sequence.

I/O A class interrupt - If bit 15 is set, indicates an error occurred on attempt to store a CIW and no CIW was stored.

I/O B class interrupt - The status is covered during the discussion of the individual channels.

Interval Timer status halfword - If bit 15 is set, indicates a parity error was detected when the Interval Timer read out one of the halfwords of the Interval Timer Word.

Program/Machine check and the Supervisor Call interrupts do not generate any status and these status halfwords are set to 0.

STATUS HALFWORD

INTERRUPT CLASS	STATUS HALFWORD
	BITS 0——————————15
I/O A	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 E_1
I/O B	REFER TO INDIVIDUAL I/O CHANNEL
INTERVAL TIMER	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 E_2
PROGRAM/MACHINE CHECK	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
SUPERVISOR CALL	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

E_1 -- ERROR OCCURRED ON ATTEMPT TO READ OUT CIW CONTROL LIST

E_2 -- ERROR OCCURRED ON ATTEMPT TO READ OUT INTERVAL TIMER

START I/O INSTRUCTION

All input/output operations are initiated by this instruction. The effect of the SIO varies and is a function of the channel to which it is directed.

The operation code is 9C hexadecimal.

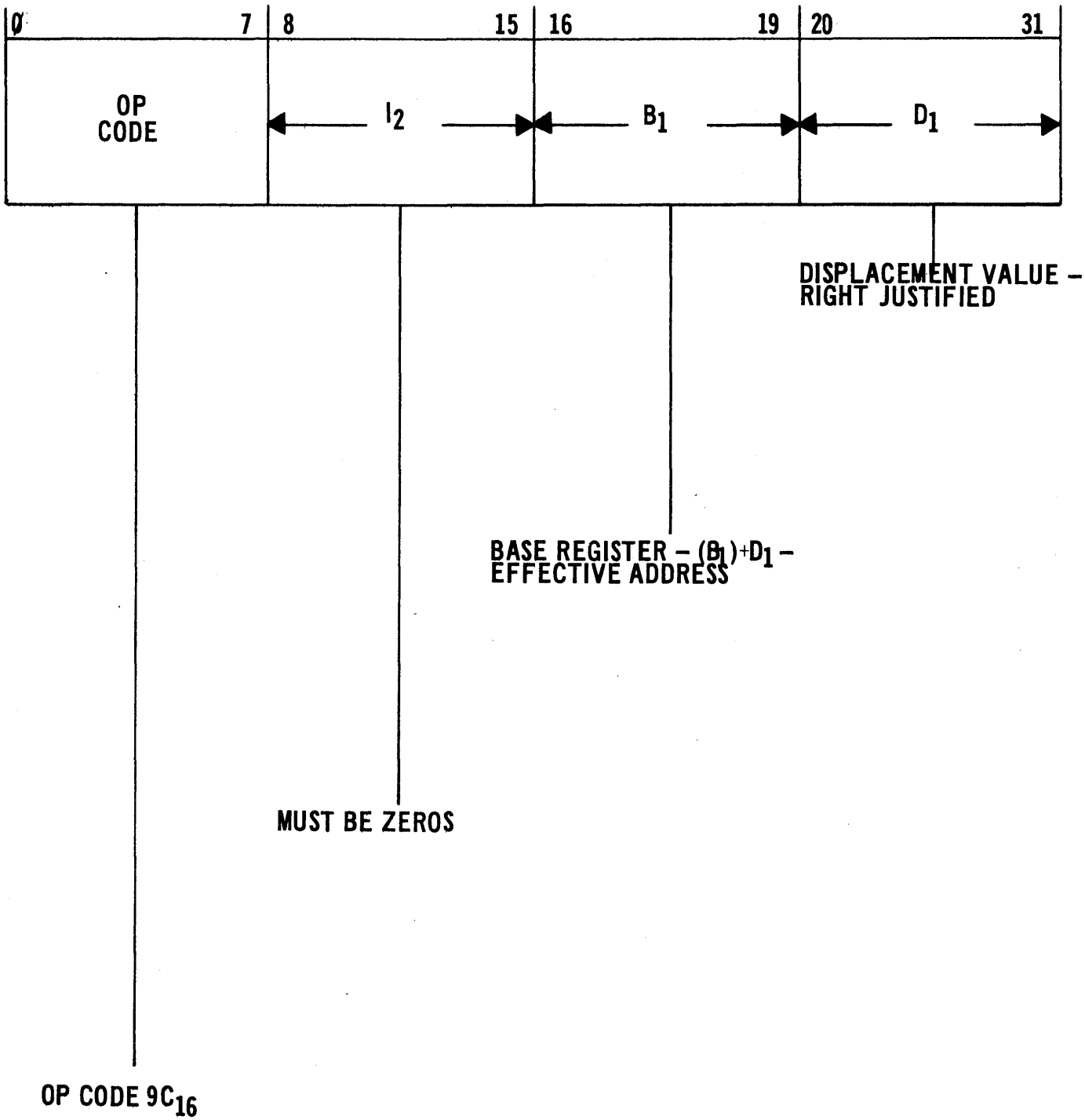
Bits 8-15 are not used and are set to zero.

B_1 - The number of the general register which holds the base address for operand 1.

D_1 - The displacement value for the base address of Operand 1.

The addition of the contents of B_1 and the value of D_1 fields form the effective address which is the channel number and device address to start the I/O sequence.

START I/O INSTRUCTION



I/O FLOW

The overall I/O Flow involved in using the SIO instruction is as follows:

The program must first store a Channel Address Word (CAW) in fixed memory. The Channel Address Word, in general, contains a command to an input/output device or an address of such a command. In this case, the CAW indicates an address of a Channel Command Word (CCW-2 words). The CCW indicated the operation to be performed, the address of first byte of data to be transferred and the byte count.

1. Store the CAW in fixed memory.

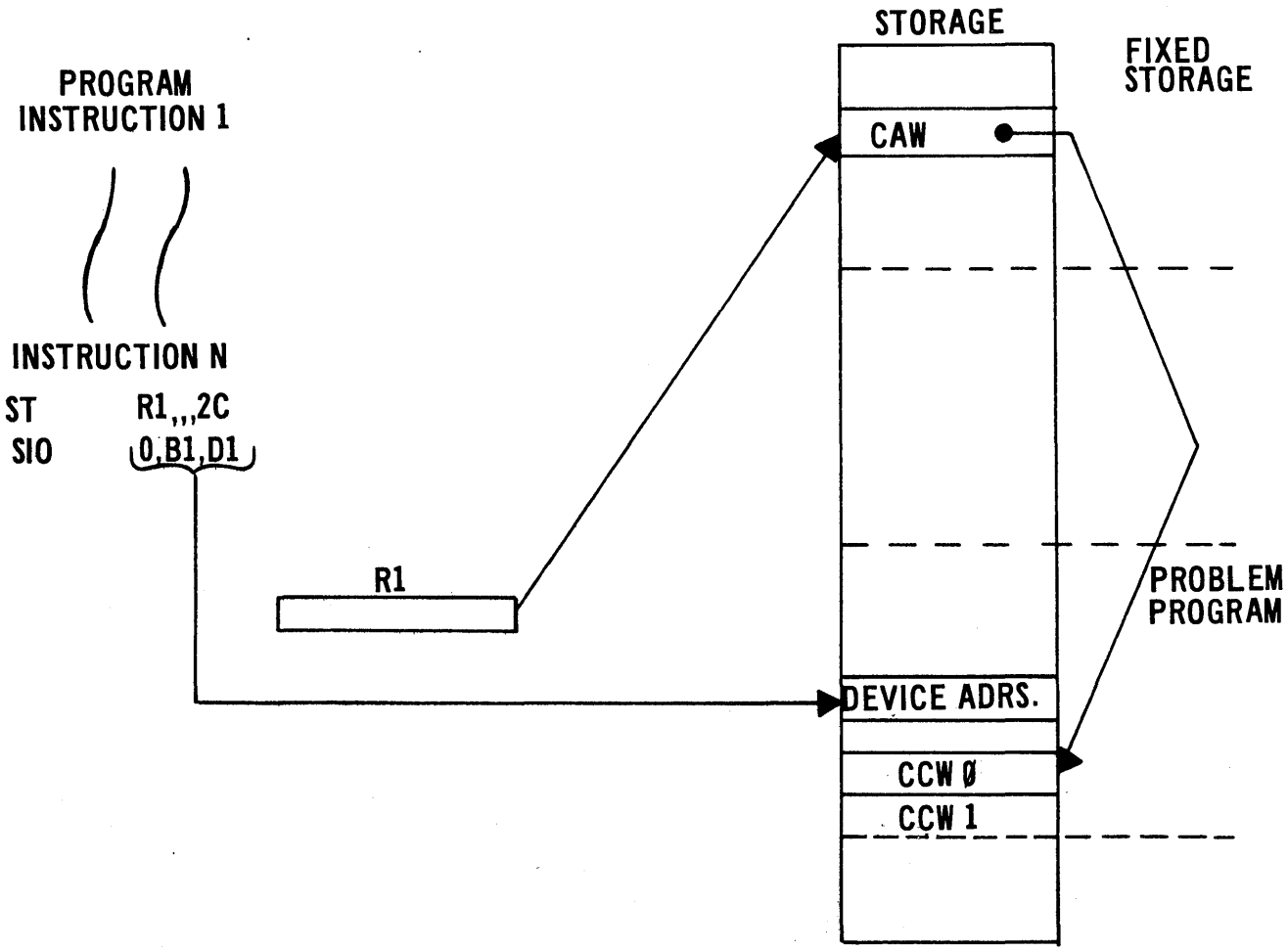
2. Issue SIO instruction.

    a. Before the channel responds with the condition code, it fetches the CAW and holds it in its hardware as required by the operation to be performed.

3. The channel gets the address of the Channel Command Word (CCW) from the CAW and the process is initiated.

NOTE: The operand field of the SIO instruction pointing to the device address does not mean the device address is in memory. This was done to show that the contents of B_1 register added to the value D_1 forms the device address.

I/O FLOW


STORAGE
FIXED STORAGE
PROGRAM INSTRUCTION 1
INSTRUCTION N
ST R1,,,2C
SIO 0,B1,D1
CAW
R1
PROBLEM PROGRAM
DEVICE ADRS.
CCW Ø
CCW 1

SECTION II

INTRODUCTION C/SP SOFTWARE

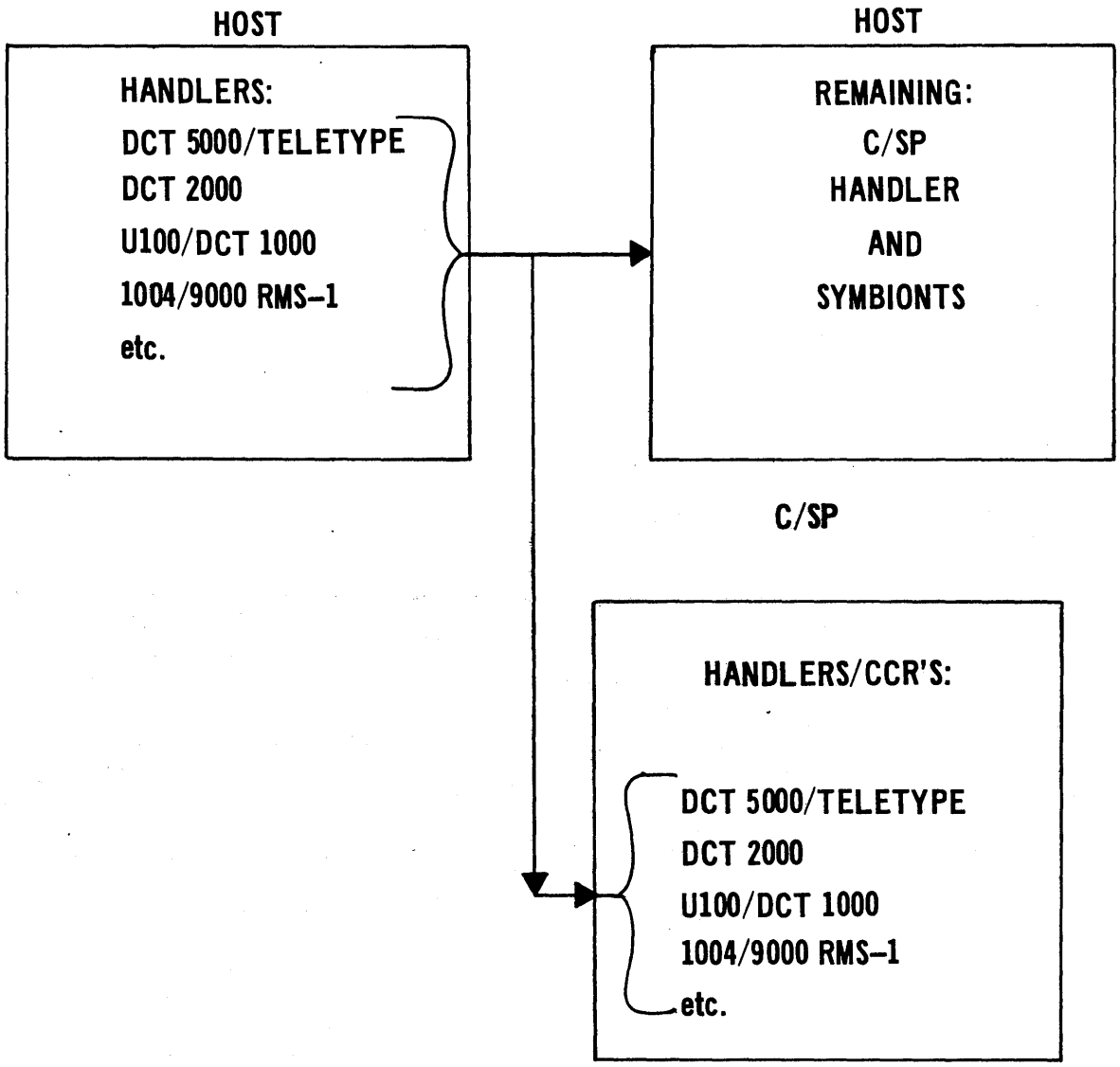
INDEX SECTION II - INTRODUCTION C/SP SOFTWARE

INTRODUCTION TO C/SP SOFTWARE

The software provided on the C/SP is designed to reduce the communications overhead in the host executive system and the user programs. The C/SP controls the remote communication devices, therefore core requirements are reduced. The device symbionts/handlers are not needed in the host computer minimizing critical timing situations and the interrupts required to gather data or scan for input. C/SP software also provides a simplified user interface for receiving or sending data to a remote device. The control characters for controlling a device and the standard code conversion is performed by a system message control program element.

Software Elements: written in modular elements to avoid fixed constraints and provide a system easily adaptable to new devices and specialized applications.

INTRODUCTION TO C/SP SOFTWARE



C/SP SOFTWARE IN HOST

Primary impace of the C/SP on the EXEC is the removal of device oriented symbionts and the substitution of a single C/SP symbiont. A C/SP handler is provided to interface the 11ØØ with the C/SP. Operator communication and the initial load function are provided for. Several elements in the EXEC require minor modifications.

C/SP SYMBIONT

The batch/demand function links to the EXEC through SYMICR as is currently being done with all input symbionts. The output function links directly to SMUPQ, by-passing the stacking of output through SMNXTP. The C/SP stacks output requests under its own stacking scheme because of C/SP control over the output terminals.

1. Input functions - set to handle batch and demand, along with special I/O requests.

2. Output functions - handles output of information from the PRINT and files for batch and demand.

3. Remote functions - to allow the remote user access to the Host computer.

SECONDARY C/SP SYMBIONTS

1. Program Load - allows the C/SP programs or the Host operator to request a program be found in Host storage and loaded into the C/SP memory for execution.

2. Console Communications - marking a terminal up/down in the assign table, allowing input/output to/from the console device.

3. Mass Storage - request access to mass storage file.

4. Logging - logging I/O errors along with other C/SP system malfunctions on the EXEC error log.

C/SP HANDLER

Consists of those routines in the Host EXEC that are concerned with support of the user interface with the C/SP, and the control of the Host and C/SP environment. The handler is designed to provide the user with a simple interface with the C/SP and its associated equipment, taking advantage of the capabilities of C/SP hardware. The user interface is similar to the interface provided by EXEC8 for a CTMC environment.

User Functions - control statements are provided to enable the user to assign and release remote devices. This enables the use of existing assignment and release linkage within the EXEC, with modifications to inform the C/SP of the operation performed.

11ØØ EXEC MODIFICATIONS

CONFIG, SMTAGS, SMLIST, DRIVE, INTI2, FIFEE, and others.

1100-C/SP SOFTWARE

A. PRIMARY C/SP SYMBIONTS
   1. INPUT
   2. OUTPUT
   3. REMOTE

B. SECONDARY C/SP SYMBIONTS
   1. PROGRAM LOAD
   2. CONSOLE COMMUNICATIONS
   3. MASS STORAGE
   4. LOGGING

C. C/SP HANDLER

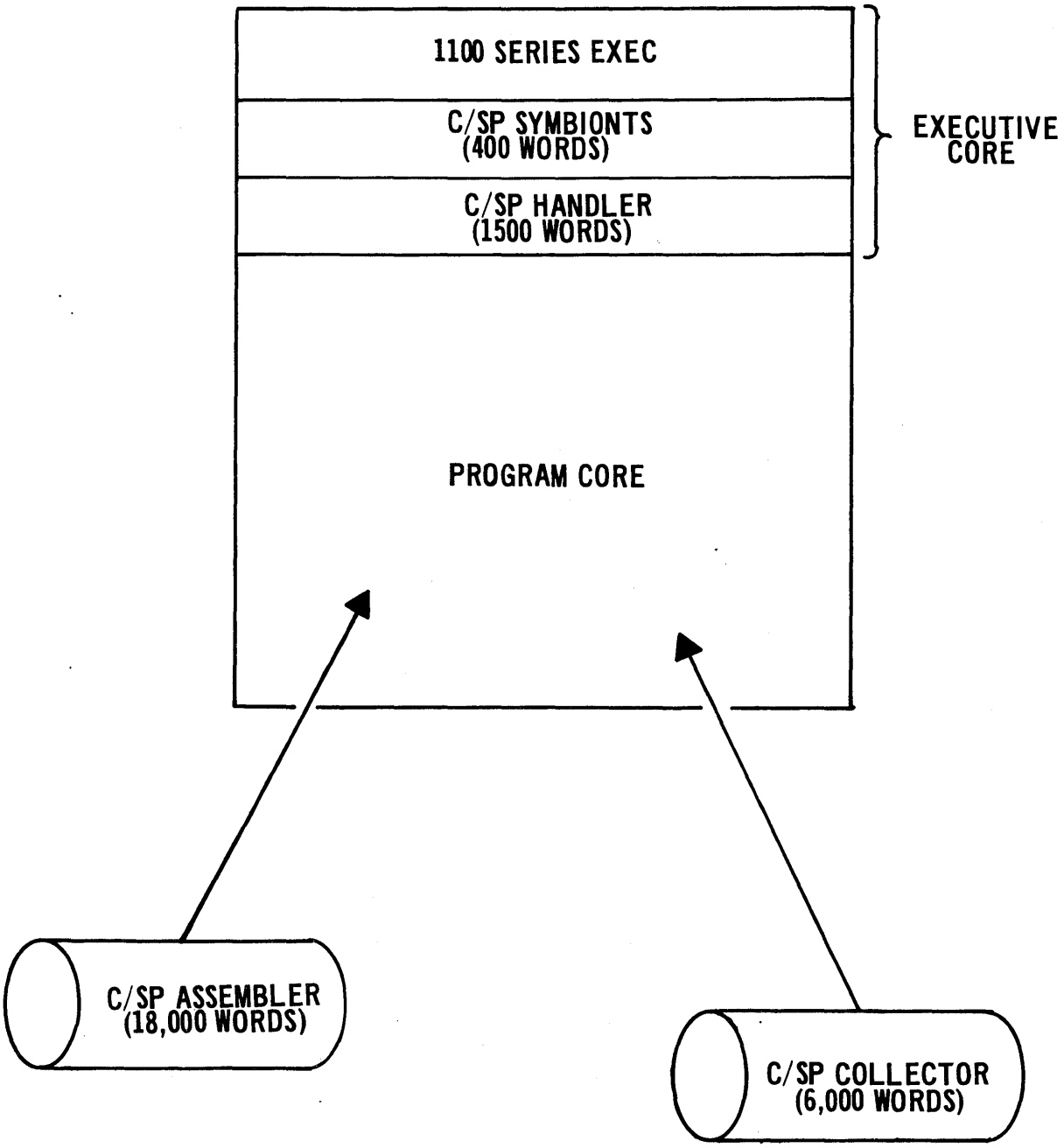
D. 1100 SERIES EXEC MODIFICATIONS

1100 STORAGE REQUIREMENTS

The C/SP Symbionts and C/SP Handler require approximately 400 words and 1500 words respectfully.

The C/SP Assembler requires approximately 18,000 words and the C/SP Collector approximately 6,000 words of 1100 memory. These Processors are loaded as needed.

1100 STORAGE REQUIREMENTS



C/SP SOFTWARE BLOCK

The C/SP software which is executed in the C/SP, may be logically divided into five main programs controlling the I/O flow between the various hardware components.

1. Terminal Management Supervisor (TMS).

2. Message Control Program (MCP).

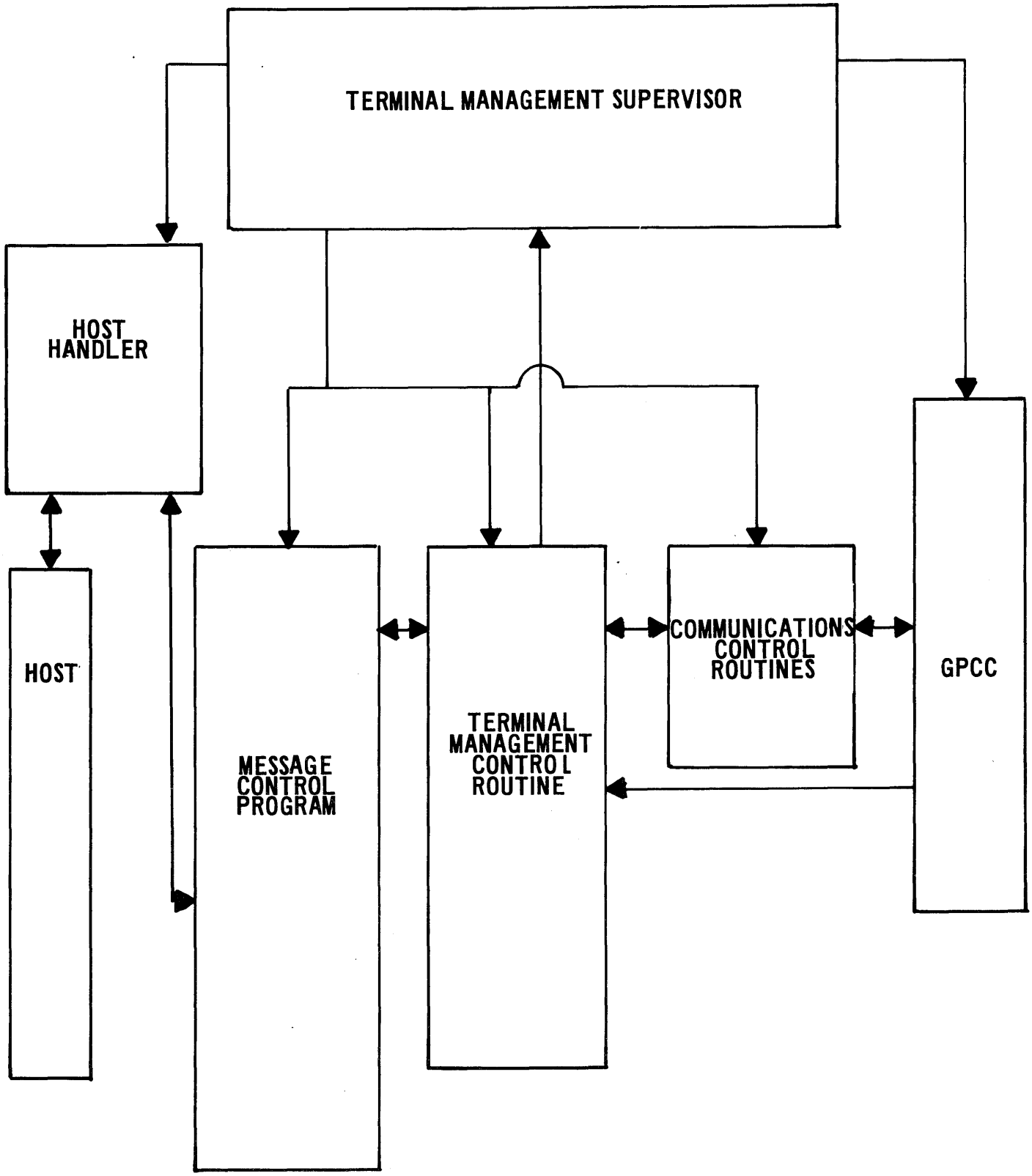
3. Terminal Management Control Routine (TMCR).

4. Communication Control Routines (CCR).

5. Host Handler.

6. Symbiont Control Program (SCP).

CISP SOFTWARE BLOCK



C/SP SOFTWARE ELEMENTS

Terminal Management Supervisor (TMS)

Provides the basic supervisory functions for the C/SP system. It issues time slices to the various control routines and provides a priority based interaction between these routines. Issues I/O commands, processes interrupts, and coordinates the control of Terminal Management Control Routine, System Message Control Program, and the users MCPs.

Terminal Management Control Routine (TMCR)

Consists of routines for controlling the communication to/from terminals on the C/SP system, performing functions which are common to all terminals.

Message Control Program (MCP)

MCP is a terminal driver program which receives input from various terminals. It routes the input from the terminals to, and accepts messages from, a host processor.

Communications Control Routines (CCR)

Compensates for the differences of a given communication discipline for different devices.

Host Handler

Resident within C/SP and controls all commands between the C/SP and the Host system.

C/SP SOFTWARE ELEMENTS

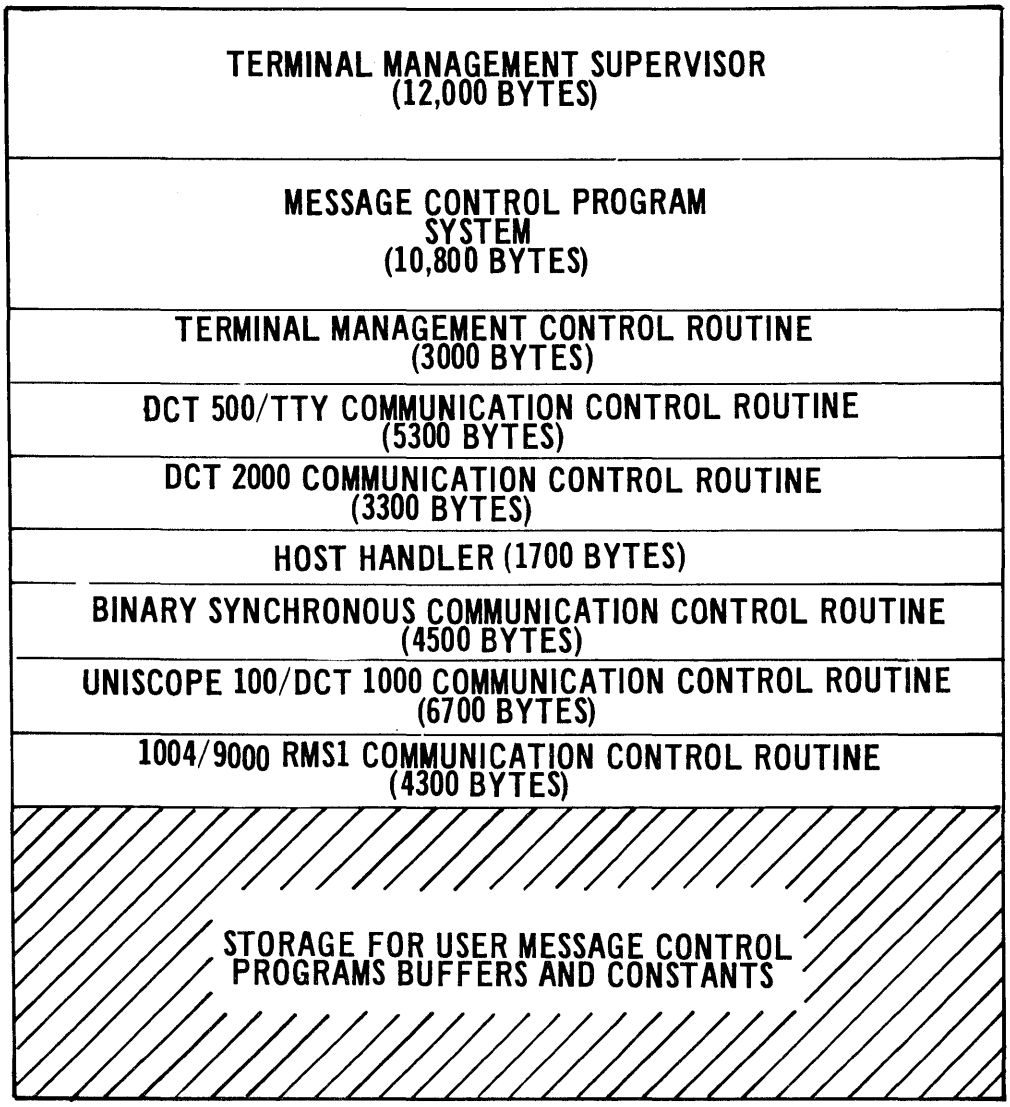
A. TERMINAL MANAGEMENT SUPERVISOR
B. TERMINAL MANAGEMENT CONTROL ROUTINE
C. MESSAGE CONTROL PROGRAM
D. COMMUNICATIONS CONTROL ROUTINES
    1. DCT 500/TELETYPE
    2. BINARY SYNCHRONOUS
    3. DCT 2000
    R UNIVAC 1004/9000 SERIES RMS-1
    5. UNISCOPE 100/DCT 1000
    6. REMOTE C/SP
E. HOST HANDLER

C/SP STORAGE REQUIREMENTS

The estimated storage requirements indicated on Section II, page 12 are subject to change as enhancements and debugging continues.

Reference UP-7942 (C/SP Installation Support) for a more detailed explanation as to the size of each module. The size will vary depending on the capabilities needed by the user.

C/SP STORAGE REQUIREMENTS

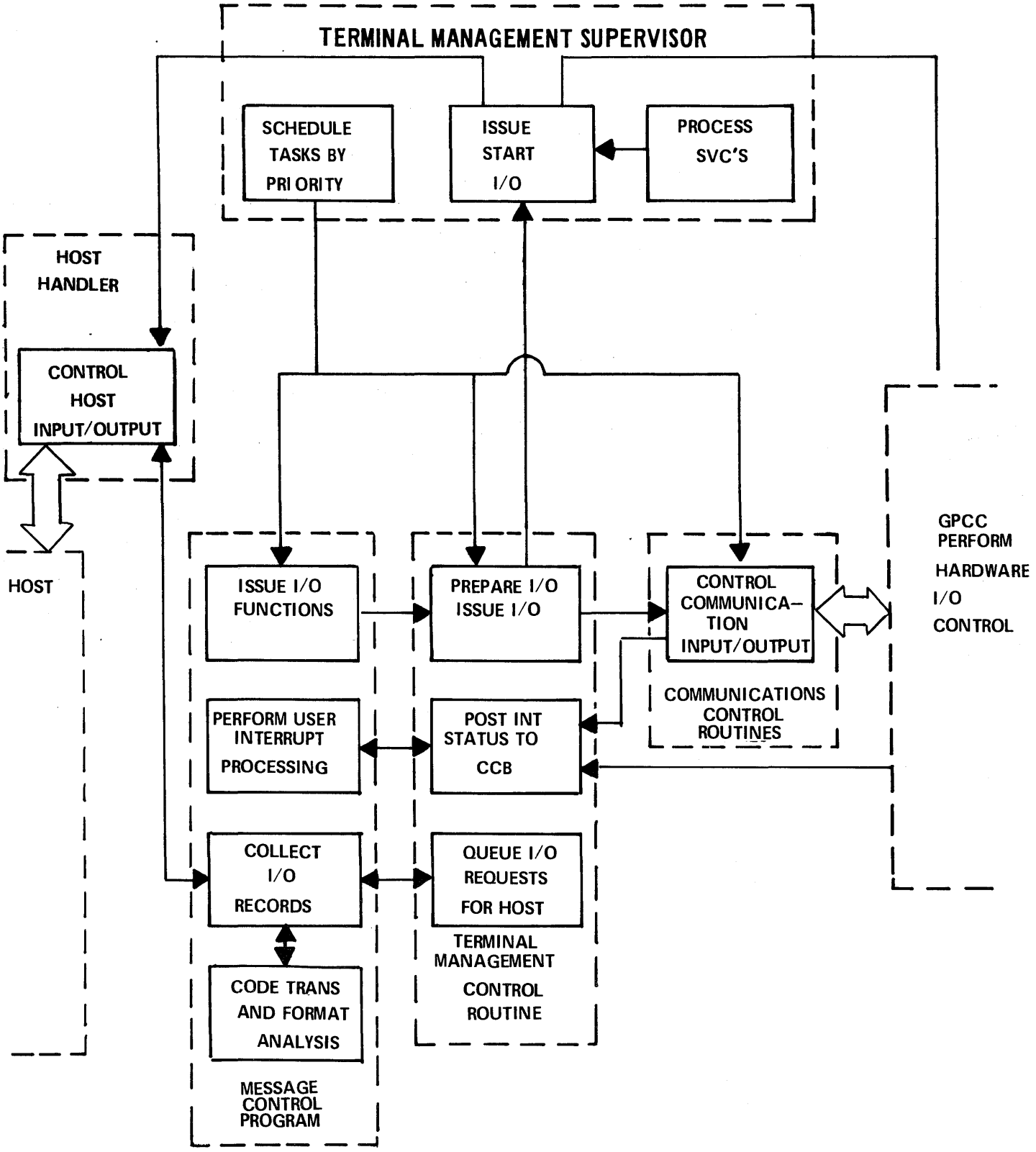


C/SP SOFTWARE BLOCK DIAGRAM

To illustrate the basic software flow, consider a message being routed from a terminal attached to the GPCC to the Host system.

Upon interrupt from the GPCC, the TMS is activated. TMS activates the TMCR to pre-process the interrupt. TMCR activates the appropriate CCR to complete processing of the interrupt. Control is passed to the MCP which issues the I/O function to the Host handler. The Host handler issues the I/O function to the Host.

C/SP SOFTWARE BLOCK DIAGRAM



C/SP - 1100 SOFTWARE ELEMENTS

(Symbiont Support Software)

A. The Terminal Management Supervisor performs the same functions with the symbiont addition to the system as in the communication environment.

B. Symbiont or Peripheral Control Program - provides the host computer with an efficient interface to onsite paper peripheral devices. The interface is established to input system run streams and output system display information.

C. Peripheral Control Routines - designed to control the I/O associated with devices using the multiplexer channel and the special device channel. The types of devices controlled through the PCR include: card reader/punch, paper tape, disc, printers, keyboard devices.

CISP-1100SOFTWARE ELEMENTS (SSS)

A. TERMINAL MANAGEMENT SUPERVISOR

B. SYMBIONT CONTROL PROGRAM

C. PERIPHERAL CONTROL ROUTINES

1. CARD READER
2. CARD PUNCH
3. PAPER TAPE
4. DISC
5. PRINTERS

11ØØ - C/SP SUPPORT SOFTWARE

C/SP Support Software - consists of a C/SP Assembler, MDW Compiler, Collector and Simulator. All are needed at program generation time only and not during communications processing.

C/SP Assembler (CSPASM) - a 1 phase, 2 pass/processor that operates under EXEC8. A modified version of the Procedure Definition Processor shall be provided to compile C/SP procedures in the EXEC library. An MDW compiler shall be provided that will process MDW source language.

C/SP Collector (MAPCC) - an 11ØØ processor to provide a means of collecting independent relative binary elements to produce an absolute program for execution in the C/SP.

C/SP Simulator (SIMUL) - a user Program to run under the 11ØØ EXEC. It accepts C/SP object code, simulates execution on the C/SP and provides diagnostic printout to aid in the debugging of the C/SP program. Main purpose is to provide a means of developing the C/SP supervisor or operating system and testing it on an 11ØØ.

1100-C/SP SUPPORT SOFTWARE

A. C/SP ASSEMBLER

   1. PROCEDURE DEFINITION PROCESSOR

   2. MESSAGE DISCIPLINE WORD COMPILER

B. C/SP SIMULATOR

C. C/SP COLLECTOR

C/SP ASSEMBLER

C/SP Assembler is a subset of the 1100 Assembler and resides in the Host Computer. The formats and conventions are a subset of the 9ØØØ Assembler.

@ PROCESSOR, OPTIONS - F1.E1,F2.E2,F3,D3 - Same as the control card for 11ØØ series.

1. STATEMENT FORMAT - Is that of the 9ØØØ Assembler. Columns 1-71: Assembler statement and comments. Column 72: used to indicate continuation. Column 73-80: program identification and sequencing.

2. TERMS, OPERATORS, AND EXPRESSIONS:

   a. Self defining: B'101',X'FF,39,C'A'.
   b. Literals: = x'FF'
   c. Symbols: up to six alphanumeric characters of which the first must be alphabetic.
   d. Location counters: maintained by the assembler for the main control section and each dummy section created by the programmer.

3. OPERATORS and EVALUATION: Arithmetic, logical, relation, as: *(multiply), ++(OR), = (equal).

4. EXPRESSIONS: Consists of one or more terms connected by operators.

5. INSTRUCTIONS: 52 half-word and word instructions.

6. DIRECTIVES:

EQU	TITLE
START	SPACE
END	SET
ORG	DO
CNOP	ENDO
USING	GOTO
DROP	LABEL
ENTRY	AND OTHERS

C/SP ASSEMBLER

@CSPASM,OPTION F1. E1, F2. E2, F3. E3

1. STATEMENT FORMAT
2. TERMS, OPERATORS, EXPRESSIONS
    a. SELF DEFINING
    b. LITERALS
    c. SYMBOLS
    d. LOCATION COUNTERS
3. OPERATORS AND EVALUATION
4. EXPRESSIONS
5. INSTRUCTIONS
6. DIRECTIVES

SUMMARY OF PROCESSOR INSTRUCTIONS - I

The Binary Arithmetic instructions provide the capability of modifying and evaluating data. These instructions use full word memory, half-word memory and registers.

SUMMARY OF PROCESSOR INSTRUCTIONS-I

BINARY ARITHMETIC

INSTRUCTION	MNEMONIC	OP CODE	FORMAT
ADD	A	5A	RX
	AH	4A	RX
	AR	1A	RR
COMPARE	C	59	RX
	CH	49	RX
	CR	19	RR
DIVIDE HALF- WORD	DH	53	RX
LOAD	L	58	RX
	LH	48	RX
	LR	18	RR
MULTIPLY HALF- WORD	MH	52	RX
SHIFT LEFT SINGLE	SLA	8B	RS
SHIFT RIGHT SINGLE	SRA	8A	RS
SUBTRACT	S	5B	RX
	SH	4B	RX
	SR	1B	RR
STORE	ST	50	RX
	STH	40	RX

SUMMARY CONTINUED - II

The Branching instructions provide the capability of conditional transfer of control from one routine to another.

The Status Switching instructions are Supervisor instructions for controlling the multi-programming environment.

The Input/Output instructions (one) provides the data chennel transfer capability. It is a supervisor or priviledged instruction.

SUMMARY CONTINUED-II

BRANCHING

INSTRUCTION	MNEMONIC	OP CODE	FORMAT
BRANCH & LINK	BAL	45	RX
	BALR	Ø5	RR
BRANCH ON CONDITION	BC	47	RX
	BCR	Ø7	RR
BRANCH ON COUNT	BCT	46	RX
	BCTR	Ø6	RR
BRANCH ON INDEX HIGH	BXH	86	RS
BRANCH ON INDEX LOW OR EQUAL	BXLE	87	RS

STATUS SWITCHING

HALT & PROCEED	HPR	99	SI *
LOAD PSW	LPSW	82	SI *
SET STORAGE KEY	SSK	Ø8	RR *
SET SYSTEM MASK	SSM	8Ø	SI *
SUPERVISOR CALL	SVC	ØA	RR

INPUT/OUTPUT

START I/O	SIO	9C	SI *

* PRIVILEGED INSTRUCTIONS

SUMMARY CONTINUED - III

The Logical instructions provide the capability of testing and altering data by bit or groups of bits.

The Divide Polynominal is provided for cyclic redundancy check (sophisticated parity check).

SUMMARY CONTINUED - III

LOGICAL

INSTRUCTION	MNEMONIC	OP CODE	FORMAT
AND	N	54	RX
	NI	94	SI
	NR	14	RR
COMPARE LOGICAL	CL	55	RX
	CLI	95	SI
	CLR	15	RR
DIVIDE POLYNOMIAL	DP	81	RS
EXCLUSIVE OR	X	57	RX
	XI	97	SI
	XR	17	RR
INSERT CHARACTER	IC	43	RX
LOAD ADDRESS	LA	41	RX
MOVE	MVI	92	SI
OR	O	56	RX
	OI	96	SI
	OR	16	RR
SHIFT LEFT SINGLE LOGICAL	SLL	89	RS
SHIFT RIGHT SINGLE LOGICAL	SRL	88	RS
STORE	STC	42	RX
TEST UNDER MASK	TM	91	SI

C/SP COLLECTOR

The C/SP Collector is concerned with generating C/SP loadable (absolute) programs and resides in the Host Computer.

1. @MAPCC

2. Collector control statements or directives:
   a. LOADM: where the operand is FN.EN, Address FN is the file where the user wants the loadable program created. Address specifies the starting address.
   b. INCLUDE: specifies the relocatable program to be included in this collection.
   c. EQU: equates name to the value.
   d. RCORR: specify corrections to a relocatable program.
   e. LINKOP: collector options.
   f. SNAP: provides for dynamic memory dumps.
   g. ENDC: specifies the end of collector directives.
   h. ENTER: the location where the execution of the program starts.

CISP COLLECTOR

@MAPCC

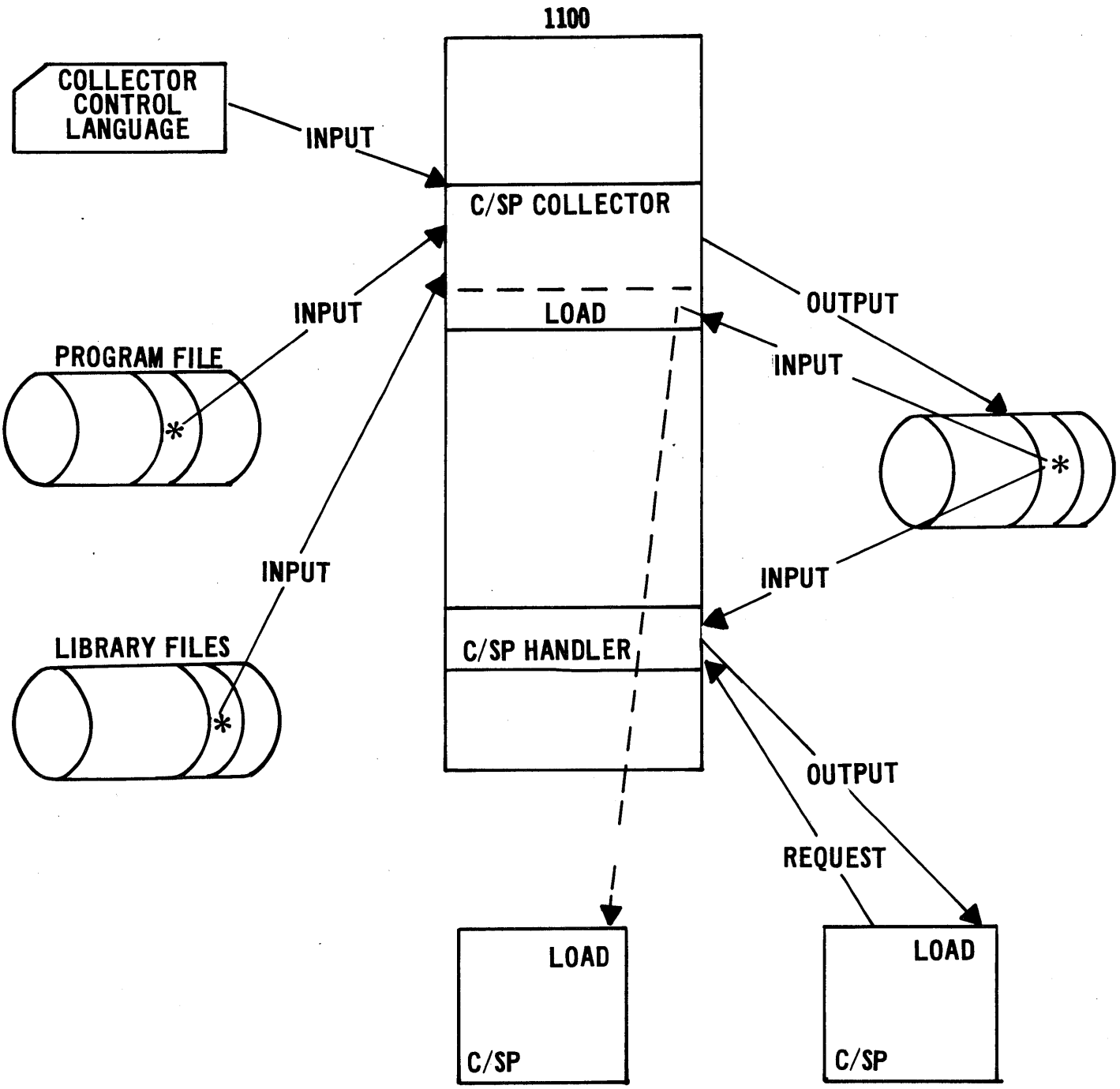
1. LOADM OPERAND
   FN. EN. ADDRESS
2. INCLUDE FN. EN
3. EQU NAME/VALUE
4. RCORR FN. EN
   ADDRESS/X'mn...n'
5. LINKOP OPTIONS
   (a) MAP
   (b) NOABORT
6. SNAP FN. EN, ADDRESS, DUMP SIZE
   COUNT, FREQUENCY, R.A
7. ENDC
8. ENTER LABEL

C/SP COLLECTOR DIAGRAM

Input to the Collector, under direction of the Collector control language, is from object programs produced by the C/SP Assembler. The object programs may be found in a program file (established by the user via a @ ASG), in TPF$., and the system library.

The C/SP collector creates the loadable program in a program file which may be TPF$., or a user-established program file.

C/SP COLLECTOR



C/SP DOCUMENTATION AND VIDEO TAPES

Reference material available:

C/SP System Description	UP-7850
C/SP Processor Languages	UP-7870
C/SP Languages 11ØØ Series Supplement	UP-7874
C/SP Simulator	UP-7862
C/SP Processor and Storage	UP-7866
C/SP Operating System PRM	UP-7900
C/SP Operators References	UP-7919
C/SP Operating System 11ØØ Supplement	UP-7919
C/SP OS Reference Booklet	UP-7912
C/SP Installation Support Manual	UP-7942

A C/SP Binder and Index Tabs are available under number UP-7880.

The video tape presentations are available on the C/SP Hardware. The software presentations are in the process of being prepared.

C/SP DOCUMENTATION AND VIDEO

A. PRODUCT SOFTWARE DESCRIPTIONS
   1. C/SP (CTMC REPLACEMENT)
   2. C/SP (SYMBIONT SUPPORT SUBSYSTEM)

B. C/SP USER DOCUMENTATION
   1. SYSTEM DESCRIPTION
   2. SIMULATOR
   3. PROCESSOR LANGUAGES
   4. LANGUAGES 1100 SERIES SUPPLEMENT
   5. PROGRAMMER REFERENCE
   6. OPERATING SYSTEM (PRM)
   7. OPERATING SYSTEM 1100 SUPPLEMENT
   8. OPERATOR REFERENCE
   9. INSTALLATION SUPPORT
   10. OS REFERENCE BOOKLET

C. VIDEO PRESENTATIONS
   1. INTRODUCTORY SEMINAR
   2. PROCESSOR AND STORAGE
   3. GENERAL PURPOSE COMMUNICATIONS CHANNEL

SECTION III

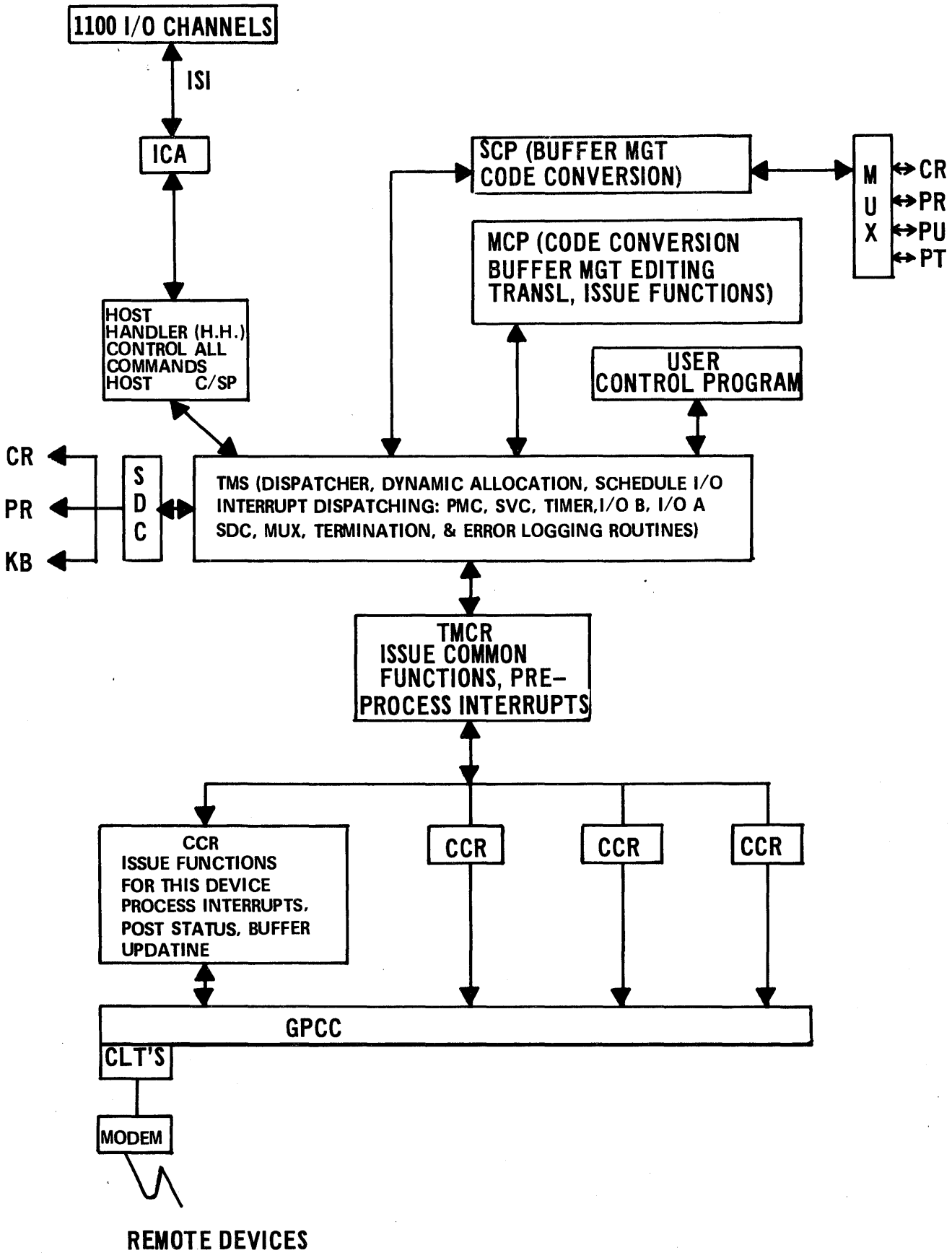
TERMINAL MANAGEMENT SUPERVISOR

SECTION III INDEX - TERMINAL MANAGEMENT SUPERVISOR

OPERATING SYSTEM OVERVIEW

The C/SP Operating System consists of a Terminal Management Supervisor (TMS), Terminal Management Control Routine (TMCR), Communications Control Routines (CCR), and Host Handler. In addition, two system control programs are provided: Message Control Program (MCP) and Symbiont Control Program (SCP) which are entered via Supervisor Call (SVC) instructions. The user is provided the capability to write control programs, to supplement control over the input/output process, and handlers (CCR) for non-supported devices.

OPERATING SYSTEM OVERVIEW



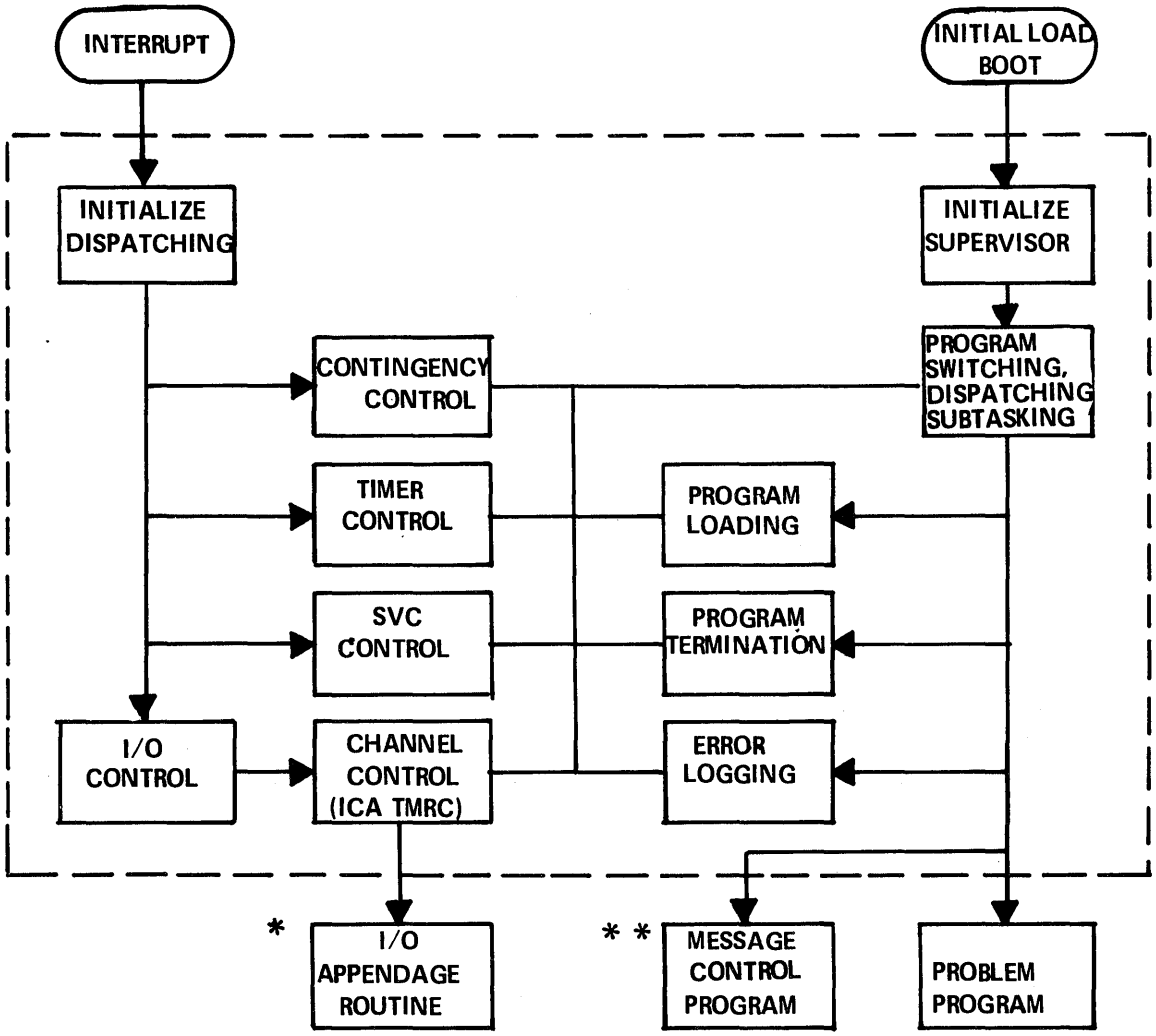
TERMINAL MANAGEMENT SUPERVISOR
(GENERAL DATA FLOW)

When the Terminal Management Supervisor has been initialized, any problem programs to be loaded into C/SP memory are loaded by the dynamic allocator module of the supervisor. The TMS dispatcher determines which programs, if any, are available for execution and control is dispatched to that program. If no programs are queued for execution, the dispatcher enters the idle mode with all interrupts allowed.

I/O appendages or user own-code routines are allowed, if specified at system generation, to provide additional control over I/O operations.

TERMINAL MANAGEMENT SUPERVISOR

(GENERAL DATA FLOW)



* ALLOWED FOR SPECIAL CASES ONLY

SPECIAL TYPE OF PROBLEM PROGRAM

TERMINAL MANAGEMENT SUPERVISOR (TMS)

Program switching, dispatching and subtasking switching accompanied by executing the highest priority program currently able to run. Programs of equal priority are given equal time slices. Up to 6 problem programs and the system MCP are supported on the C/SP. Within each problem program there are two levels of control or subtasking. The problem programmer has complete control of the activation and queue of secondary activities within his program.

Program loading and initialization - loading utilizes the concept of dynamic memory allocation, allocating an area of memory into which the program, originating from the host, is loaded. A relocation directory (produced by collector) is associated with each program loaded.

Program termination - problem programs may be terminated upon normal request by the user or occurrence of an error condition, as an unsolicited program check or unrecoverable I/O error without a contingency routine. The program terminator provides PMD's if requested by the user. Memory dumps and termination messages are sent to the host computer for output to an on-site device.

Error logging - error statistics are gathered and stored in a buffer which, when full, is transmitted to the host computer. The host stores the statistics on a file which is periodically listed on a printer. It is used mainly for hardware error checking and analysis.

Contingency Control - contingency routines may be supplied by the problem program to regain control upon occurrence of I/O, program check, or Interval Timer contingency.

Timer Control - timer control routine controls all access to the timer request queue. All timeouts requested by the problem program and all time slices requested by the dispatcher are interleaved in the timer request queue on a time-of-day basis.

Supervisor Call - all supervisor call processing routines are accessed by means of a jump table. SVC is the means by which a problem program requests a supervisor function including 1) timer control, 2) I/O request, and 3) contingency request.

I/O Control - I/O control routine determines the ownership of each I/O interrupt. Control then passed to respective channel control routine (including host handler and TMCR). The lowest level of I/O control is handled by the channel scheduling routine which controls loading of the CAW, issuing of the SIO, logical to physical device address translation, I/O queuing, and I/O appendage routing. I/O appendages are problem programs own-code routines providing additional control over I/O operation during channel program XQT at time I/O interrupt occurs.

TERMINAL MANAGEMENT SUPERVISOR (TMS)

1. PROGRAM SWITCHING, DISPATCHING
2. PROGRAM LOADING AND INITIALIZATION
3. PROGRAM TERMINATION
4. ERROR LOGGING
5. CONTINGENCY CONTROL
6. TIMER CONTROL
7. SUPERVISOR CALL
8. I/O CONTROL

TERMINAL MANAGEMENT SUPERVISOR

(General Entry Routine)

Functions:

1. Save the interrupt code, status PSW and registers.
2. Initialize the registers for the Supervisor.
3. Branch to the respective interrupt processing routine.

(General Exit Routine)

Function:

1. Restore registers and give control to the activity flagged in the respective switch list.
2. When exit from the Supervisor, no registers are restored.

Register 2: Contains the return address to which the General Entry Routine will branch upon completion of its function.

Register 3: Points to the switch list entry.

Register 4: Contains the address of the doubleword containing the interrupt code, status halfwords and the PSW for the respective interrupt.

Registers 5-6: Reserved for problem program.

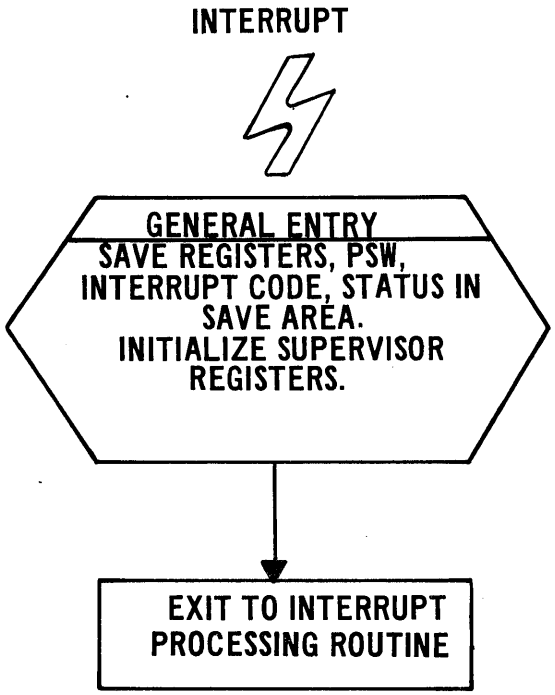
Registers 7-9: Supervisor base registers.

Registers 10-13: Supervisor work registers.

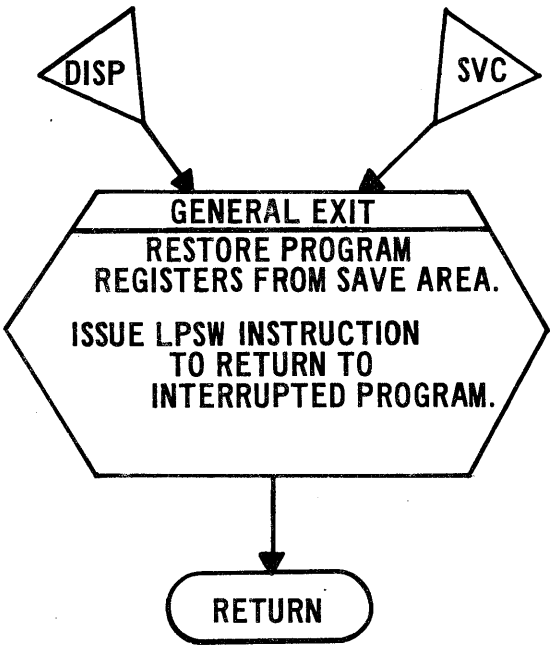
Registers 14-15: Branch and link registers.

Registers Ø-1: Parameter passing registers.

TERMINAL MANAGEMENT SUPERVISOR (GENERAL ENTRY AND EXIT)



1. SUPERVISOR CALL
2. PROGRAM/MACHINE CHECK
3. INTERVAL TIMER
4. I/O A–I/O B



REGISTER	DISPOSITION	SUPERVISOR USE
2 3 4	RESERVED FOR SUPERVISOR	RETURN ADDR. SWL ADDRESS STATUS,PSW
5 6	RESERVED FOR PROB. PRGM.	MAY NOT BE USED BY SUPERVISER
7 8 9		SUPERVISER BASE REGISTERS
10 11 12 13		SUPERVISER WORK REGISTERS
14 15		BRANCH AND LINK REGISTERS
Ø 1		PARAMETER PASSING REGISTERS

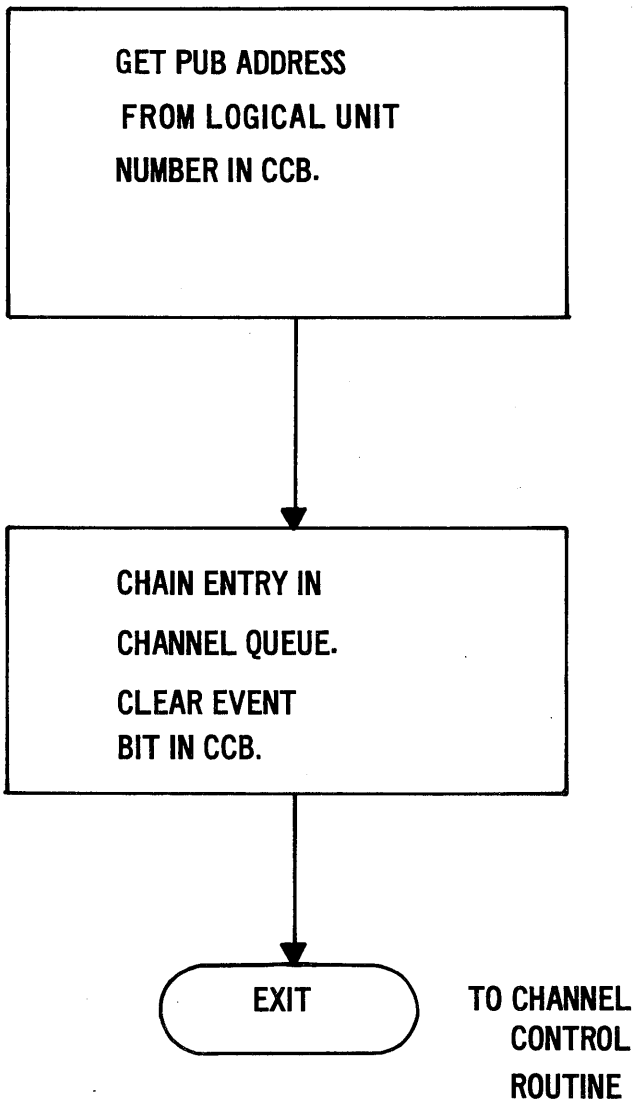
TERMINAL MANAGEMENT SUPERVISOR
(Channel Scheduler)

Functions:

1. Convert symbolic I/O references or logical numbers to physical device numbers.
2. Provide queuing of I/O requests for each device.
3. Forward control to the individual channel routines to handle the start I/O and other functions associated with each channel.
4. Maintains three tables which form the nucleus of the channel scheduler:
    a. LUB - Logical Unit Block
    b. PUB - Physical Unit Block
    c. CHQ - Channel QUEUE

All information necessary for the scheduling of each I/O request is stored in these tables.

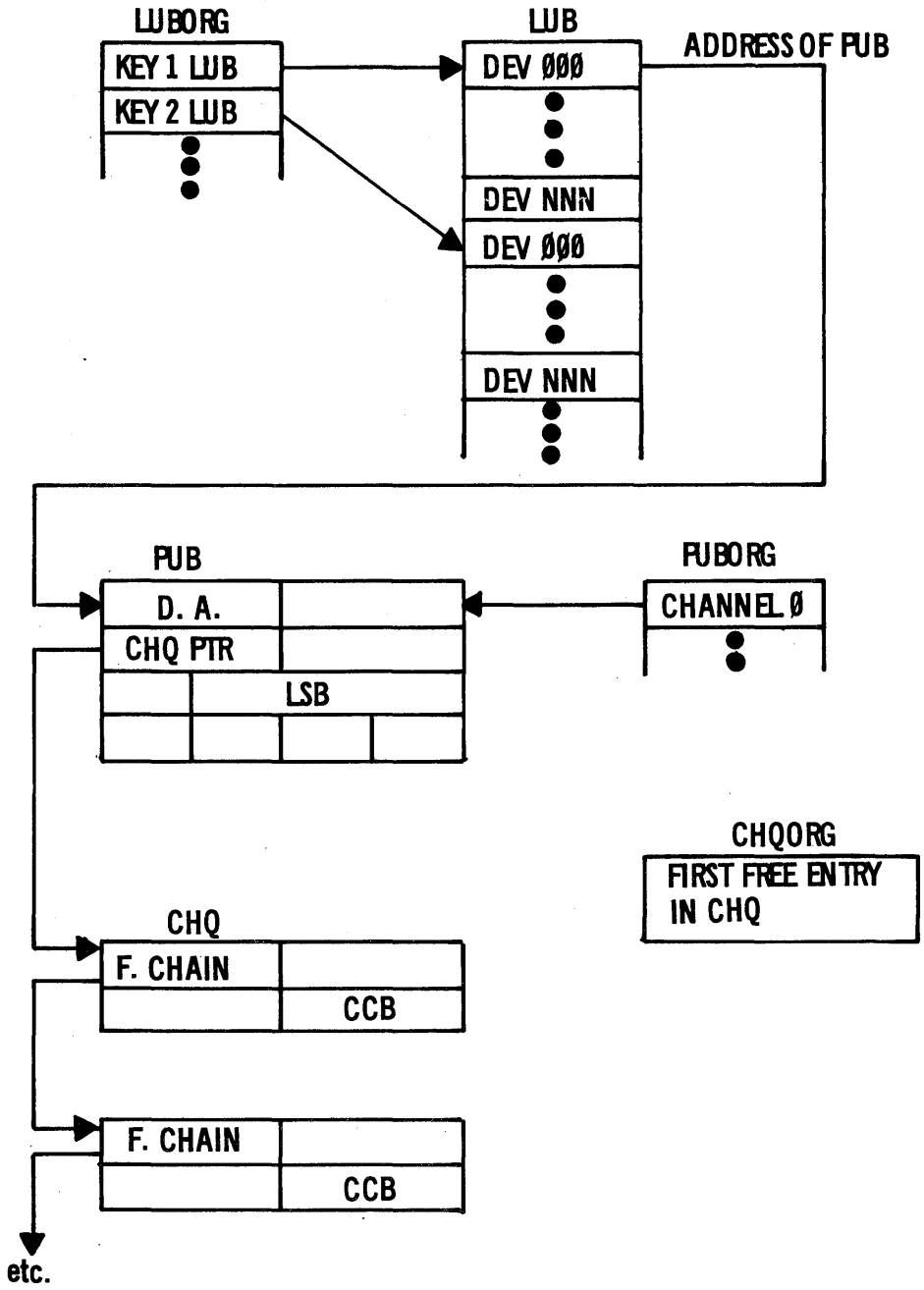
TMS
CHANNEL SCHEDULER



LUB, PUB, CHQ OVERVIEW

The Logical Unit Block (LUB) consists of halfword addresses of the Physical Unit Block (PUB) describing the physical device. The PUB contains a pointer to the Channel Request Queue (CHQ) which may consist of a chain of I/O requests for that device.

TMS

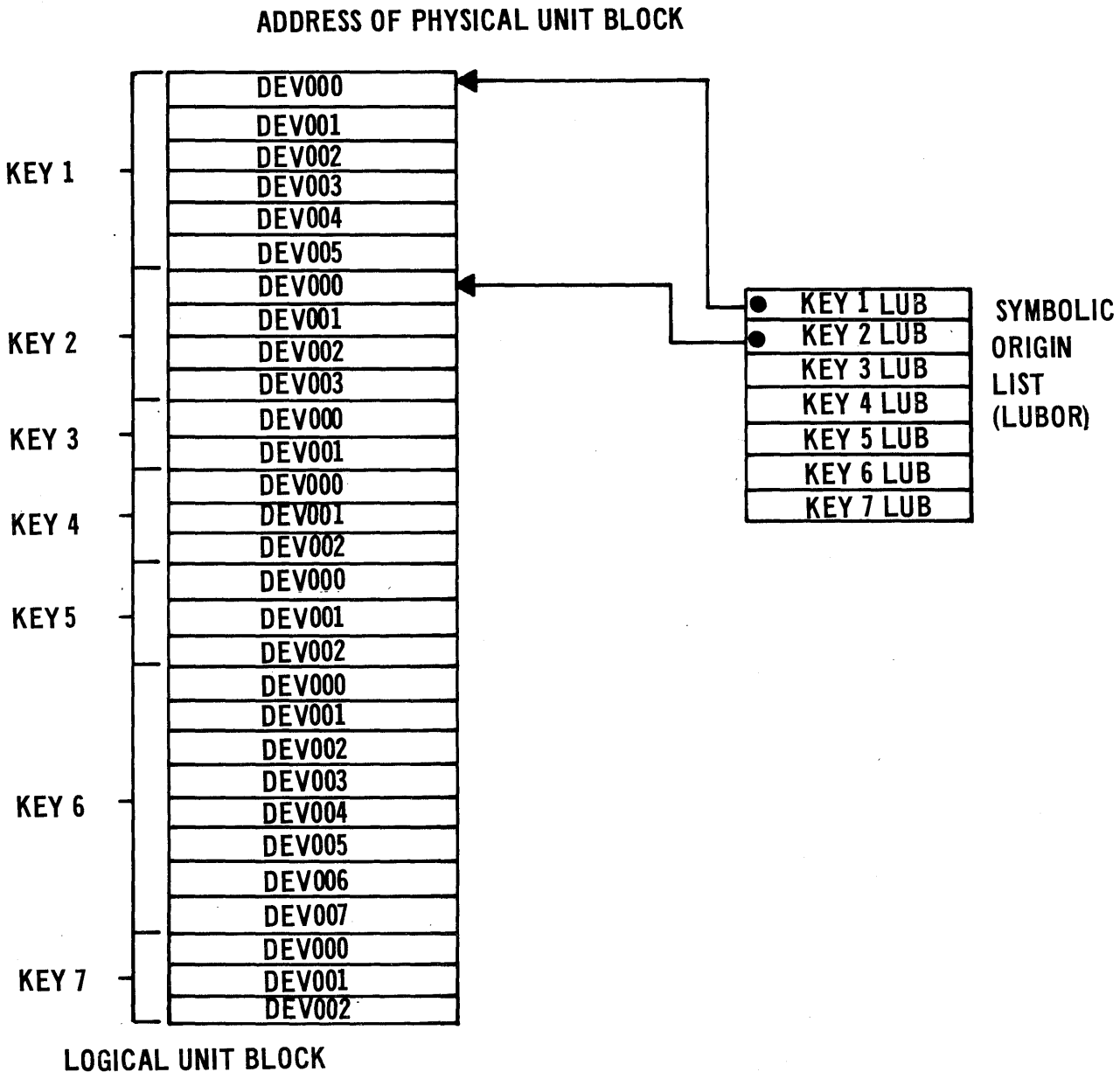

LUBORG
KEY 1 LUB
KEY 2 LUB
LUB
DEV ØØØ
DEV NNN
DEV ØØØ
DEV NNN
ADDRESS OF PUB
PUB
D. A.
CHQ PTR
LSB
PUBORG
CHANNEL Ø
CHQORG
FIRST FREE ENTRY IN CHQ
CHQ
F. CHAIN
CCB
F. CHAIN
CCB
etc.

CHANNEL SCHEDULER

(LOGICAL UNIT BLOCK)

The Logical Unit Block (LUB) consists of one halfword entry for each logical unit number up to the total allocated at SYSGEN Time. A list called The Logical Unit Block Origin (LUBORG), fixed at SYSGEN Time, designates the origin of each concurrently running program. The LUBORG entries for Keys 1-7 is the address of an origin point in the LUB table. Each entry in the LUB table is the address of the Physical Unit Block (PUB) table to which each LUB has been assigned.

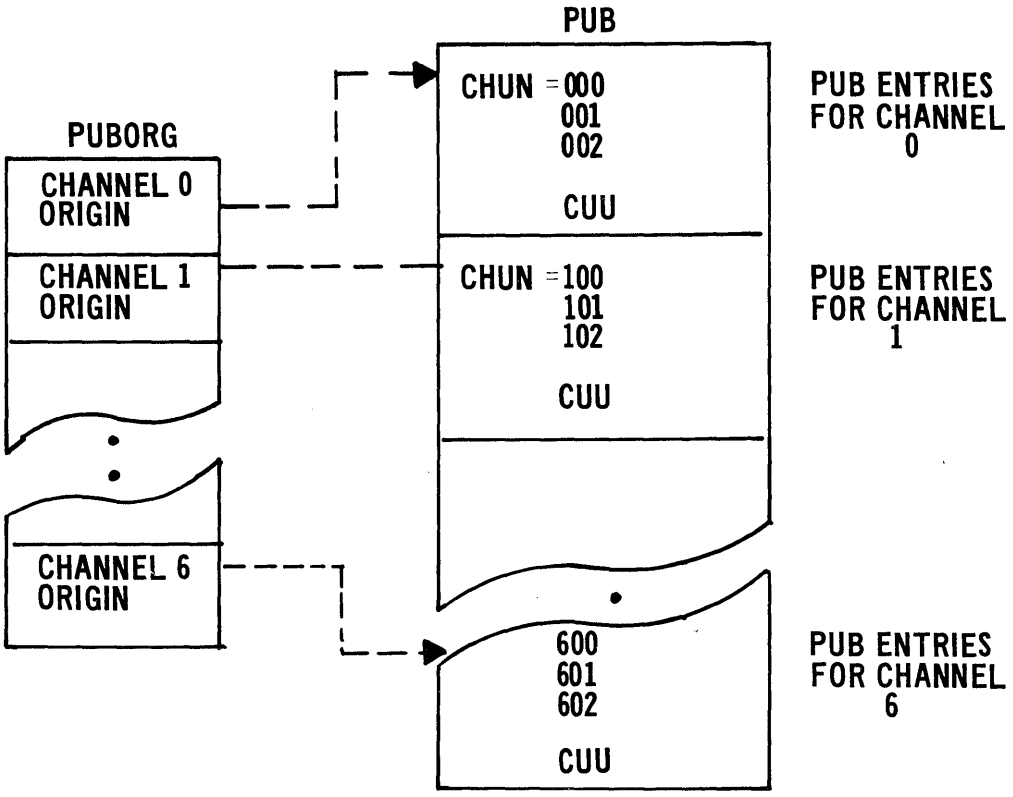
CHANNEL SCHEDULER (LOGICAL UNIT BLOCK)



CHANNEL SCHEDULER
(PHYSICAL UNIT BLOCK)

The Physical Unit Table (PUB) consists of one 16-byte for each physical device defined at SYSGEN Time. The designated origin of each channel group is kept in a list called the Physical Unit Block Origin List (PUBORG) which is fixed at SYSGEN Time. Each halfword entry in the PUBORG List is the address of a channel origin in the PUB table.

CHANNEL SCHEDULER (PHYSICAL UNIT BLOCK)



PUB

0	DEVICE ADDRESS		ALTERNATE DEVICE ADDRESS	CHANNEL 0 DEVICE 0
4	CHANNEL QUEUE POINTER		CONTROL FLAGS / PSW KEY	
8	DEVICE TYPE	LINE STATUS BLOCK ADDRESS		
C	OLD PSW KEY	OLD C/SP LOG. UNIT #	C/SP LOG UNIT NO. / HOST LOG UNIT NO.	
	• • •			CHANNEL N DEVICE N

CHANNEL SCHEDULER
(CHANNEL QUEUE TABLE)

The Channel Queue Table (CHQ) is used to chain all I/O requests awaiting execution for any one device. The requests are queued in the order requested. The size of the CHQ table is determined at SYSGEN Time and should be large enough to contain all current I/O requests plus space for all new requests. The CHQ entries are pointed to by bytes 4 and 5 of the PUB. Any free entries in the CHQ are also chained and are pointed to by the Channel Queue Origin (CHQORG).

TMS
(CHANNEL QUEUE TABLE)

CHQ ORG

FIRST FREE ENTRY IN CHQ

CHQ

Ø	ADDRESS OF NEXT ENTRY IN CHAIN	SWL ADDRESS OR PLACE – TO – GO ADDRESS
4	PSW KEY	COMMAND CONTROL BLOCK (CCB) ADDRESS

TERMINAL MANAGEMENT SUPERVISOR
(Dynamic Allocator)

Functions:

1. Allocation and release of main storage space.

2. Initial loading of programs.

3. Switch list initialization.

Free Memory Pool (FMP) - points to those areas which are not assigned to a program.

Used Memory Pool (UMP) - points to those areas which are assigned to programs. Each program assigned a 32-byte area.

DYNAMIC ALLOCATOR

FMP

Ø	Ø ——— Ø	LENGTH OF AREA IN NUMBER OF BYTES
4	ØØ OR FF IF LAST IN CHAIN	ADDRESS OF NEXT FREE AREA
8	Ø Ø OR FF IF FIRST IN CHAIN	ADDRESS OF PREVIOUS FREE AREA

Ø	Ø Ø	START ADDRESS OF AREA ALLOCATED TO PROGRAM MAIN TASK
4	Ø Ø	LENGTH OF AREA ALLOCATED TO PROGRAM MAIN TASK
8	Ø Ø	START ADDRESS OF AREA ALLOCATED TO PROGRAM REQUEST 1
C	Ø Ø	LENGTH OF AREA ALLOCATED TO PROGRAM REQUEST 1

✸ CONTINUES FOR EACH REQUEST UNTIL THE 32-BYTE AREA IS FULL.

TERMINAL MANAGEMENT SUPERVISOR
(Dispatcher)

Functions:

1. Switch list control and updating.

2. Asynchronous switching (attach macro).

3. Contingency switching (STXIT) macro).

4. Priority scheduling and time-slicing.

DISPATCHER

SWITCH LIST

SWL

Ø	PROGRAM CONTROL FLAGS	PROGRAM KEY	PROGRAM CONTROL FLAG	PSW KEY
4	Ø Ø	UNSOLICITED OPERATOR COMMUNICATION OCB ADDRESS		
8	ACTIVE TASK POINTER	MULTIPLE WAIT MCB ADDRESS		
C	TASK CONTROL FLAGS	MAIN TASK SAVE AREA ADDRESS		
1Ø	TASK CONTROL FLAGS	SUB–TASK SAVE AREA ADDRESS		
14	TASK CONTROL FLAGS	CONTINGENCY SAVE AREA ADDRESS		
18	MAIN TASK MWAIT OPTION	ATTACH MACRO ACTIVITY QUEUE PARAMETER LIST ADDRESS		
1C	SUB–TASK MWAIT OPTION	CONTINGENCY ENTRY ADDRESS		

TERMINAL MANAGEMENT SUPERVISOR

(Termination and Error Handling)

Termination Functions:

1. Set the switch list entry to initial condition and release memory allocated to the terminated program.

2. Initiate termination message to the host, including memory dump if the program was terminated due to an error and a dump option was requested.

3. Dequeue all pending I/O and timer requests for the terminated program.

Error Handling Functions:

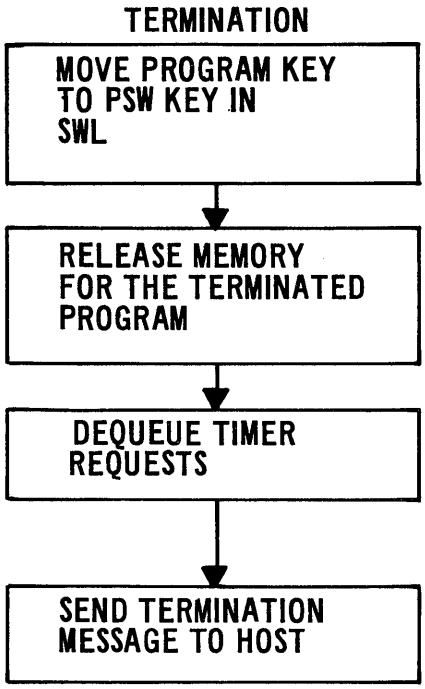
1. Identification of error.

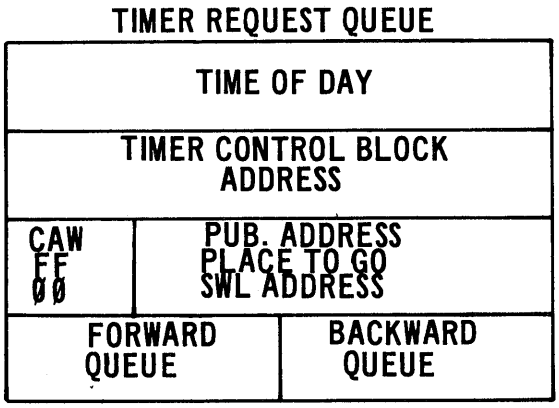
2. Compilation of error statistics.

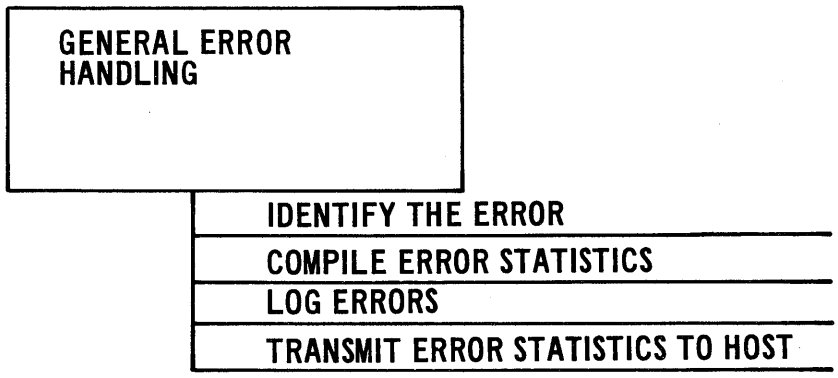
3. Initiation of I/O request to send error statistics and operator error messages to the host or some device chosen at system generation time for this purpose.

4. Logging of error conditions to be used by F. E. Diagnostic Routines when loaded into memory.

TERMINATION AND GENERAL ERROR HANDLING



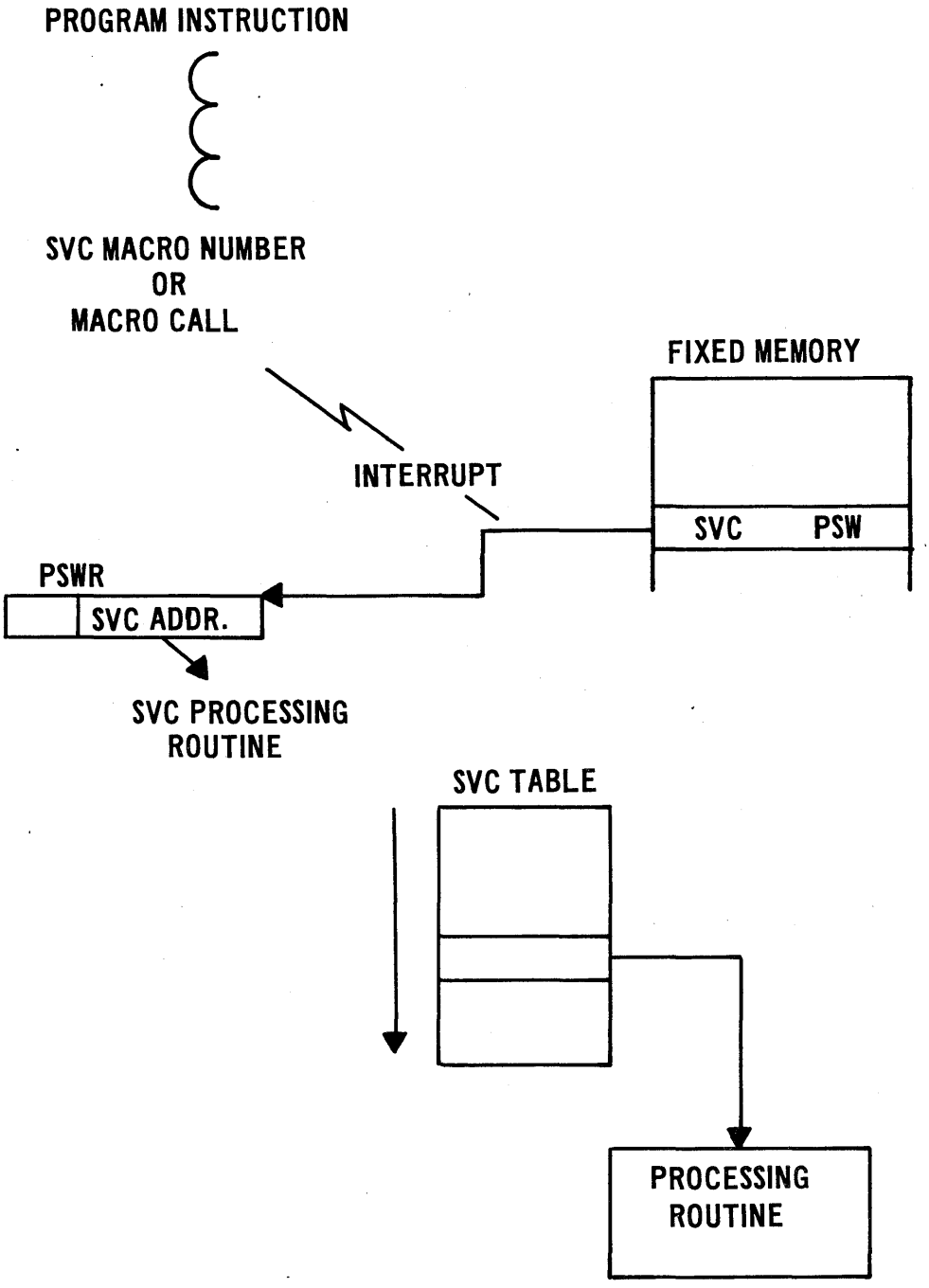




MACRO CALL SEQUENCE

Execution of a Supervisor Call (SVC) instruction with a macro reference in the operand field results in a SVC interrupt. The SVC Table is used as a jump table to the processing routine for processing.

TERMINAL MANAGEMENT SUPERVISOR



TERMINAL MANAGEMENT SUPERVISOR

(Supervisor and I/O Macros)

Macros:

CCB (Command Control Block) - The basic information link between the problem program and the executive supervisor.

EXCP (Execute Channel Program) - The means of performing I/O operations.

WAIT - Issued whenever the program requires that an I/O request be completed before the execution of the problem program continues.

ASSGN (Assign) - Used for dynamic assignment of symbolic or logical I/O references to physical hardware addresses.

OCB (Operator Communications Control Block) - The basic information link between the problem program and the supervisor for sending a message to the console operator.

WTO (Write to Operator) - Used to send a message to the host console operator. (ASCII or field data).

WTOR (Write to Operator with Reply) - Same as WTO, now requesting a reply.

MEMORY - Used to request additional memory or to release used memory. (2K bytes).

MCB (Multiple Control Block) - The information link between the problem program and supervisor for the purpose of posting traffic bits for multiple activities or requests.

SETMCB (Set Multiple Control Block) - The means of requesting or changing the status of the Multiple Wait facility.

MWAIT (Multiple Wait) - The means of setting a program in a wait condition until the first completion posting of any combination of events.

ATTACH - Used in a main program to initiate (attach) an asynchronous subtask and to establish the order of priority between the main and subtask.

ACTQ (Activity Queue) - Is referenced by the ATTACH macro and may be used to build a queue in which each entry is a separate ACTQ macro.

DETACH - Used by a main program to terminate all subtask activities and activity queue.

DEACT and REACT (Deactivate and Reactivate) - Used to deactivate and again reactivate a subtask from a main program.

STXIT (Set Exit) - Establishes linkage from the supervisor to a contingency routine for the purpose of processing abnormal I/O conditions, program checks, and interval timer interrupts.

TERMINAL MANAGEMENT SUPERVISOR (SUPERVISOR AND I/O MACROS)

[NAME]	CCB	[SYS=XXX / DVC=NNNN], [BCW=NAME / IOA=(NAME, LENGTH) / FCB=NAME] , [APPNDG=NAME] , [SENSE=NAME] , [ERRCV=YES] , [STATUS=REGS]
[NAME]	EXCP	[CCB-NAME]
[NAME]	WAIT	[CCB-NAME]
[NAME]	ASSGN	DVC=NNN,OPT=X'CUU'
NAME	OCB	MSG=(NAME,LENGTH), [REPLY=(NAME,LENGTH)] , [APPNDG=NAME]
[NAME]	WTO	[OCB NAME]
[NAME]	WTOR	[OCB NAME]
[NAME]	MEMORY	[GET / RELS], [SIZE] , [ADDRESS]
NAME	MCB	NOT USED
[NAME]	SETMCB	[MCB-NAME] , [MWAIT ON]
[NAME]	MWAIT	[MCB-NAME] , X'ii'
[NAME]	ATTACH	Q,ACTQNAME,PRIORITY
NAME	ACTQ	SUBTASK,SAVE-AREA, [PRIORITY] , [NEXT ACTQ]
NAME	DETACH	NOT USED
[NAME]	DEACT and REACT	NOT USED
[NAME]	STXIT	IO,PC,IT, [ENTRY] , [SAVE-AREA] , [OC=OCB-NAME]

(Supervisor and I/O Macros, cont'd.)

CANCFL - Used to abnormally terminate a problem without generating a dump.

EXIT - Used to terminate either a main program, a subtask, or a contingency routine.

TCB (Timer Control Block) - Information link between the problem program and the supervisor, for passing the number of timer intervals required before an interrupt to the supervisor.

SETIME (Set Interval Timer) - Used to set the interval timer to the value that is specified in the operand.

DQTIMR (Dequeue Interval Timer Request) - Used to remove or dequeue a previously issued SETIME macro.

GETIME - Used to obtain the current time of day at any time during the program execution.

GETDAT - To move the date to a two-word area set up by the problem program.

PUTCOM - Used to store information in the communication region.

GETCOM - Causes the seven-word (28 byte) communication region to be moved to save area set up by the problem program.

OPEN - Open a communication line or data file.

CLOSE - Close a communication line or data file.

LCB (Load Control Block) - The basic information line between the user program and the executive supervisor.

LOAD - Returns control to the calling sequence following a SVC instruction.

FETCH - Giver control to the program which was loaded. Both LOAD and FETCH macros are used to load user programs from the host mass storage.

CCW - Used to generate an 8-byte Channel Command Word.

DUMP - Dump memory assigned to user and cancel.

CDUMP - Dump all C/SP memory and return.

PDUMP - Partial dump.

TERMINAL MANAGEMENT SUPERVISOR

(SUPERVISOR AND I/O MACROS)

[NAME]	CANCEL	NOT USED
[NAME]	EXIT	NOT USED
NAME	TCB	INTERVAL NUMBER
[NAME]	SETIME	[TCB-NAME] , [SAVE AREA]
[NAME]	DQTIME	[SAVE AREA]
[NAME]	GETIME	NOT USED
[NAME]	GETDAT	[SYMBOLIC ADDRESS OF DATE SAVE AREA]
[NAME]	PUTCOM	[SYMBOLIC INFORMATION ADDRESS]
[NAME]	GETCOM	[SYMBOLIC ADDRESS OF SAVE AREA]
[NAME]	OPEN or CLOSE	[CCB-NAME]
BLOCKNAME	LCB	PROG-NAME, [LOAD-NAME] , [EXEC-N]
[NAME]	LOAD or FETCH	[LCB-NAME]
[NAME]	CCW	COMMAND, [DATA-ADDRESS] , [FLAGS] , [COUNT]
[NAME]	DUMP or CDUMP	NOT USED
[NAME]	PDUMP	[STARTING ADDRESS] , [ENDING ADDRESS]

TERMINAL MANAGEMENT SUPERVISOR

(System Generation)

NUP=N Number of user priorities. Range is 1 or 2. Default is zero.

NPP=N Number of probelm programs allowed to run in addition to MCP. Default is zero.

ITWR=N Interval Timer word replacement value. This is the maximum number of milliseconds between which times interrupts can occur. Default is zero.

TRQSIZ=N The number of entries in the timer request queue. Default is zero.

TIMER=(OPTION) Allows the use of the timer feature. Option is either YES or NO. Default is NO.

RDF=(OPTION) Allows for the recoverable delay flop feature. Option is either YES or NO. Default is NO.

CNTGCY=(OPTION) Allows the attachment of contingency routines. Option is either YES or NO. Default is YES.

APPNDG=(OPTION) Allows the use of appendage routines in the problem programs. Option is either YES or NO. Default is NO.

DATE=(OPTION) Allows the use of the GETDAT macro. Option is either YES or NO. Default is NO.

DUMP=(OPTION) Allows for the automatic jump support. Option is either YES or NO. Default is NO.

LUBS=(N1,N2,...NI) The number of LUBS to be assigned to the MCP and each program. Default is zero.

PUBS=N The total number of PUBS to be generated. 'N' is the number of devices to be connected to the system. Default is zero.

CHANLS=(AØ,A1,...A6) The types of channels connected to the system.

TSLICE=N The time slice (in milliseconds) used for program switching. Default is zero.

MEMSIZ=N The total size of memory. Default is 32K.

CHANRQ=N The total number of entries in the Channel Request Queue. Maximum is 255 and default is zero.

FXDMEM=(AØ,A1,...AI) The amount of fixed memory allocated to the MCP and each problem program.

DYNMEM=(OPTION) Allows for dynamic memory allocation. Option is either YES or NO. Default is NO.

LIST=(OPTION) Allows the user to turn the assembler print feature on or off. Option is either ON or OFF. Default is 'ON'.

SYSDVC=(AØ,A1,...A8) Request support for system devices.

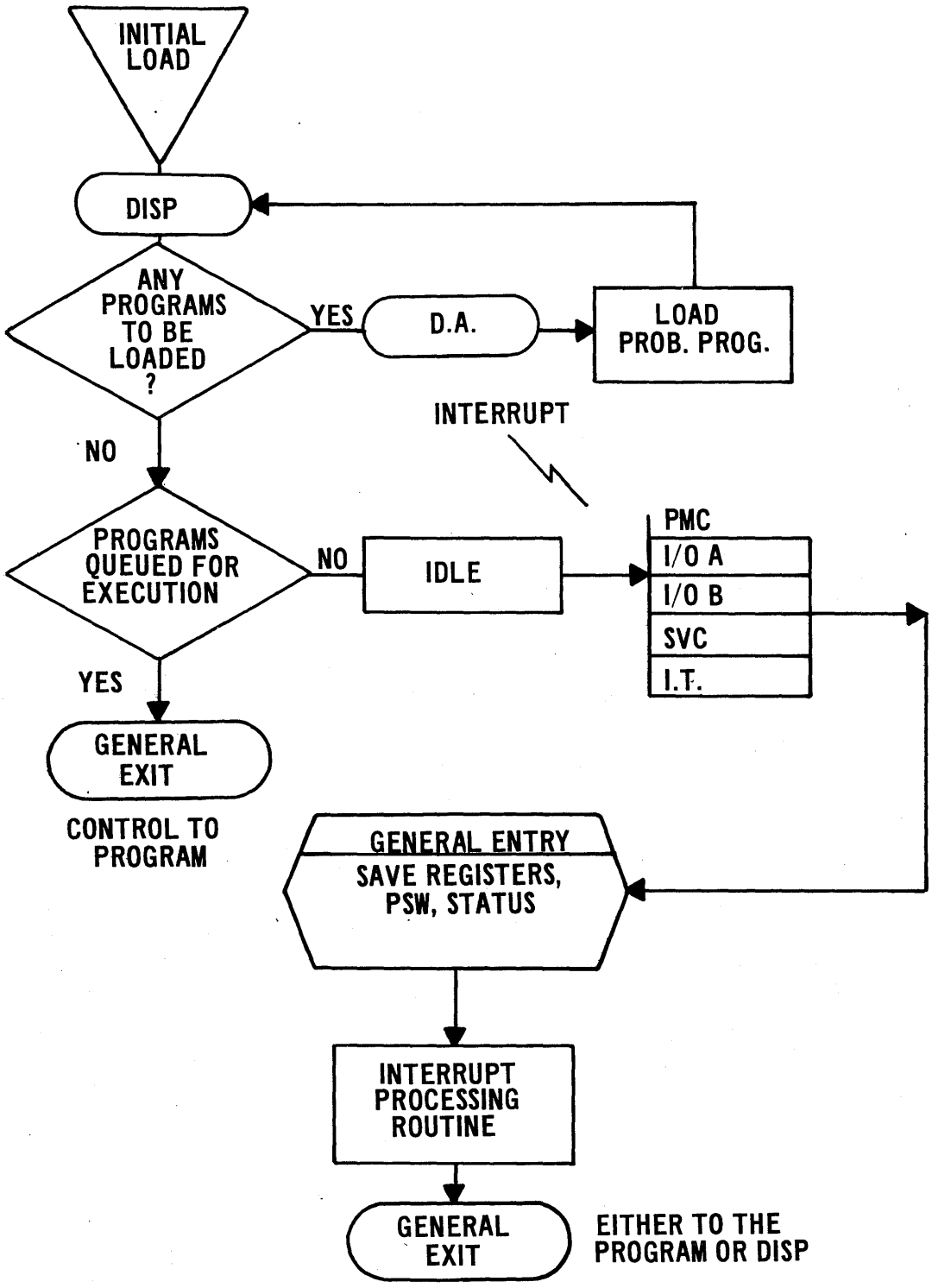
TERMINAL MANAGEMENT SUPERVISOR (SYSGEN MACRO)

NAME	OP CODE	OPERAND
	SYSGEN	[NUP =N] [,NPP =N] [,ITWR =N]
		[,TRQSIZ =N] [,TIMER =<OPTION>]
		[,RDF =<OPTION>] [,CNTGY =<OPTION>]
		[,APPNDG =<OPTION>] [,DATE =<OPTION>]
		[,DUMP =<OPTION>] [,LUBS =(NØ,N1...NI)]
		[,PUBS =N] [,CHANLS =(AØ,A1,...AI)]
		[,TSLICE =N] [,MEMSIZ = N] [,CHANRQ =N]
		[,FXDMEM =(AØ,A1...,AI)] [,DYNMEM =<OPTION>]
		[,LIST =<OPTION>] [,SYSDVC =(AØ,A1,...A8)]

TERMINAL MANAGEMENT SUPERVISOR (GENERAL FLOW)

The Terminal Management Supervisor (TMS) receives control after initial load of the operating system. The TMS dispatcher determines if there are problem programs to be loaded. When all programs have been loaded into C/SP Memory, the dispatcher determines if there are any programs queued for execution and passes control to that program. If there are no programs queued for execution, the dispatcher enables all interrupts and enters a wait mode. The occurance of an interrupt causes control to be passed to the routine to handle that class of interrupt. The interrupt processing routine uses the subroutine 'GENERAL ENTRY' to save the registers, etc., in the programs save area. When the processing of the interrupt is complete, the 'GENERAL EXIT' subroutine is used to restore the registers. Control is returned to the program or to TMS dispatcher.

TERMINAL MANAGEMENT SUPERVISOR



SECTION IV

TERMINAL MANAGEMENT CONTROL ROUTINE

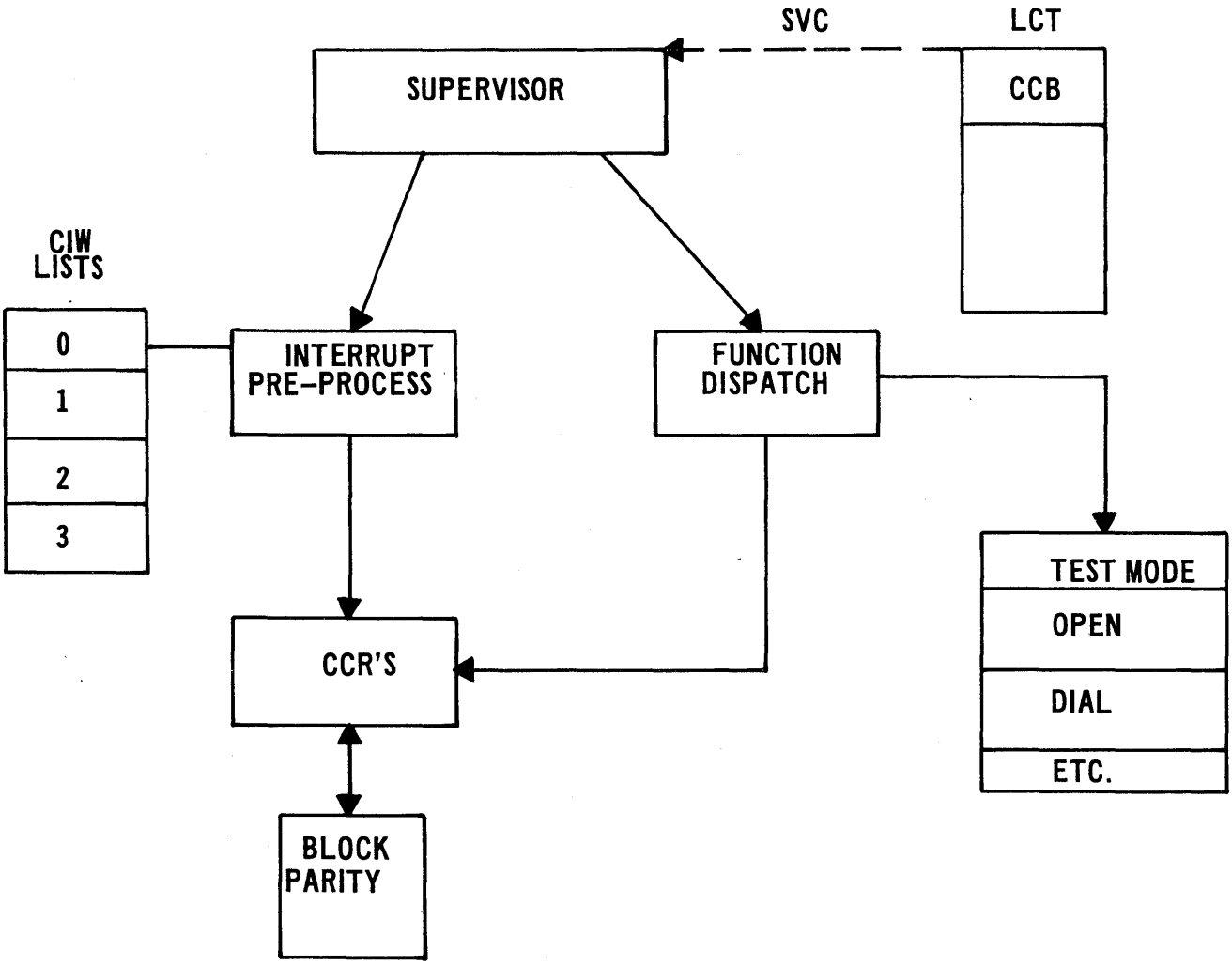
SECTION IV INDEX - TERMINAL MANAGEMENT CONTROL ROUTINE

TERMINAL MANAGEMENT CONTROL ROUTINE

(General Flow)

There are two entry points to TMCR. The function entry via SVC execution and a GPCC interrupt. In either case, TMCR determines the CCR ownership and passes the information to that CCR. Functions which are common to all devices are executed by TMCR.

TERMINAL MANAGEMENT CONTROL ROUTINE (GENERAL FLOW)



TERMINAL MANAGEMENT CONTROL ROUTINE (TMCR)

Routines for controlling the communication terminals on the C/SP divided into two general areas: 1. general handler and the individual, 2.communications control routines. TMCR is a series of routines performing functions which are common to all CCR's. (Example: dial and test mode, CIW pre-processing, interface with MCP and the supervisor). TMCR maintains the CIW lists and receives control from the supervisor whenever a GPCC interrupt occurs. After initial processing of the interrupt, it is passed to the appropriate CCR. A master status block (MSB), one for each type of device, is maintained by TMCR and contains accounting information concerning each device. In normal operation of the communication device attached to the system, any MCP executes I/O requests in the form of 'SVC's. The supervisor passes control to the TMCR which determines which CCR to activate. The CCR executes the desired I/O functions and returns status information.

Hardware: Supports communications through a GPCC channel on the C/SP.

Devices:

Teletype/DCT 500	RMS1
DCT 1000/UNISCOPE 100	Dialer
DCT 2000	
Binary Synchronous	

Software:

1. Receives all function requests for the GPCC.
   a. Locates needed tables.
   b. Validation.
   c. Passes its information to the appropriate CCR.

2. Receives all GPCC interrupts.
   a. Processes each CIW list.
   b. Routes each CIW to the appropriate CCR.

3. Contains utility subroutines used by the CCR's to post status to the user CCB and to calculate block parity.

4. Performs device-independent communications functions:
   a. OPEN - CLT's are reset from any previous condition and the line is marked open.
   b. DIAL - Controls the operation of an automatic dialer.
   c. TEST MODE ON - Provides a way to place a pair of CLT's into test mode for using the hardware back-to-back test. (lines must be closed)
   d. TEST MODE OFF - Resets the CLT to a normal state of operation.
   e. BEGIN TEST - Provides a way to perform three types of test on a line placed in test mode:
      1. CLT back-to-back on the line.
      2. Mode back-to-back.
      3. Continuous.
   f. END TEST - Stop the continuous test without taking the line out of test mode.

TERMINAL MANAGEMENT CONTROL ROUTINE (TMCR)

1. FUNCTIONS COMMON TO ALL CCR's
   - A. DIAL
   - B. TEST MODE ON/OFF
   - C. OPEN
   - D. BEGIN TEST
   - E. END TEST
2. CIW LIST PRE-PROCESSING
3. ROUTES FUNCTION CALLS TO HANDLERS (CCR's)
4. COMMON ENTRY FOR GPCC FUNCTIONS

(MSB) MASTER STATUS BLOCK

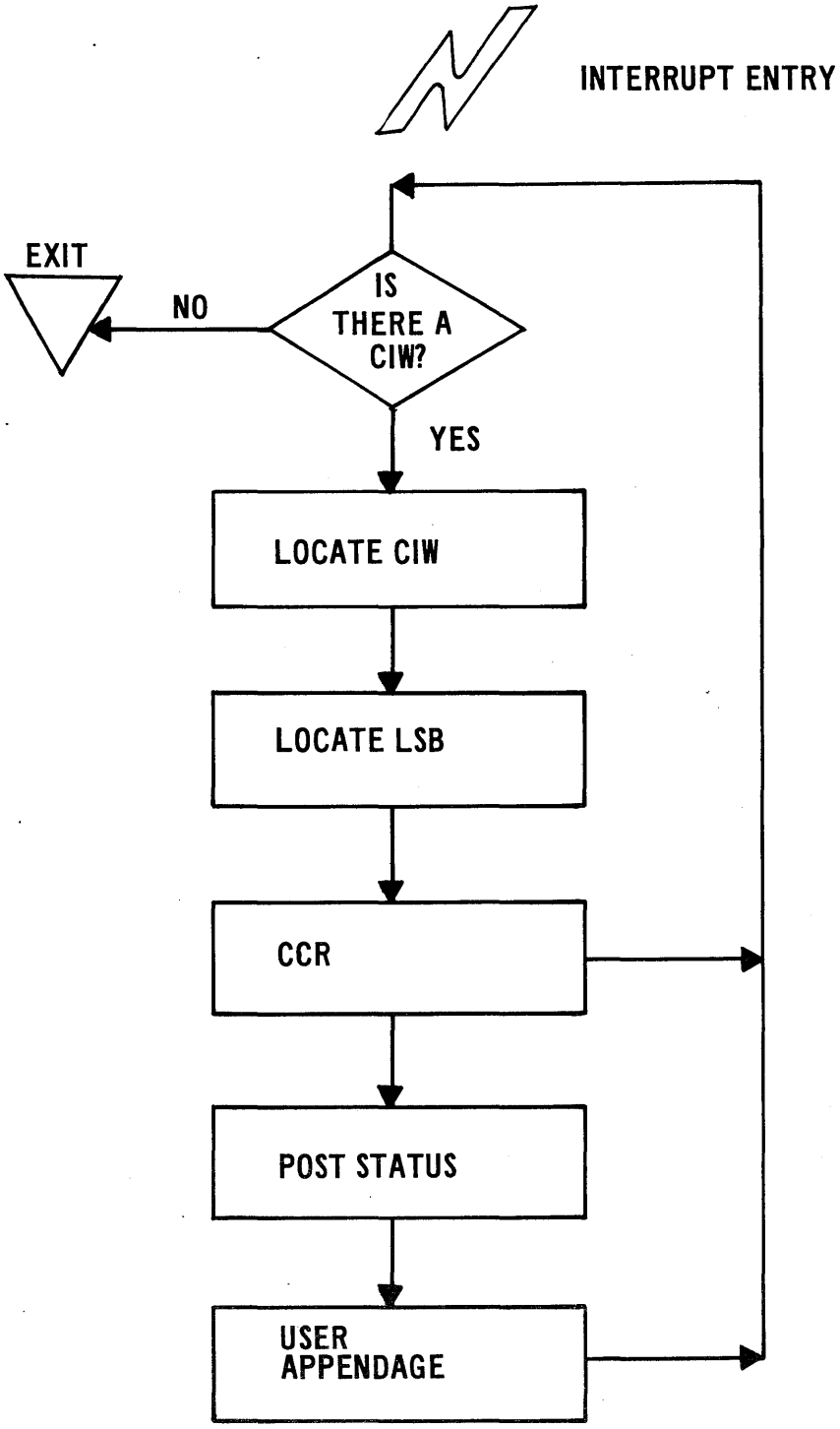
TOTAL MESSAGES OUT	TOTAL BAD OUT
TOTAL MESSAGES IN	TOTAL BAD IN

TERMINAL MANAGEMENT CONTROL ROUTINE (TMCR)

INTERRUPTS

GPCC interrupts cause CIW's to be stored in four lists. When TMCR receives control, it processes all outstanding CIW's in each list. The first CIW in the highest priority list is selected and routed to the appropriate CCR for processing. When the CCR returns status, a new selection is made until all the lists are cleared.

TERMINAL MANAGEMENT CONTROL ROUTINE (INTERRUPTS)



TERMINAL MANAGEMENT CONTROL ROUTINE (TMCR)

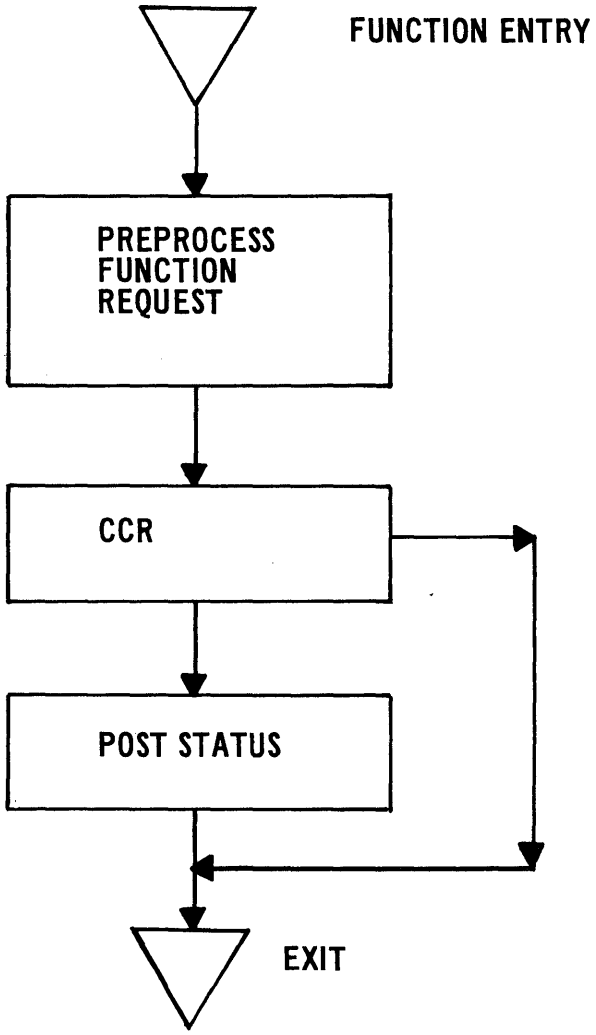
FUNCTION PROCESSING

When function is requested, TMCR validates it before routing it to the appropriate CCR;

1. Open is always accepted.
2. Test functions are accepted only if the line is closed.
3. All other functions are accepted if the line is open.
4. If the line is down, it must be placed in Test mode before issuing any other function.
5. To place half-duplex line in test mode, the mate must be closed or missing.
6. For normal dialing, the DIAL function must be issued to the line, not the dialer.
7. To test the dialer, place it in test mode, then issue the DIAL function to the dialer.

When a function request is rejected, TMCR posts status and returns to the user without passing the request to the CCR.

TERMINAL MANAGEMENT CONTROL ROUTINE (FUNCTION PROCESSING)



TERMINAL MANAGEMENT CONTROL ROUTINE

(GPCC GENERATION)

The GPCGEN macro is required for the generation of the TMCR and CCR's, allowing the user to initialize the GPCC tables to match his configuration.

GPCA = N, GPCB = N: At least one of these parameters are required, as the BCW's are generated from these parameters. GPCA or GPCB refers to one or two GPCC's. 'N' indicates the highest numbered multiplexer position on this GPCC. BCW's are generated from zero to this value (Ø - 63).

AØ = N,...A3 = N: Optional parameters to generate the CIW's and CIW List Control Words for the GPCC. AØ...A3 represent the CIW Lists Ø...3 for GPCA. 'N' is any number from Ø - 64 specifying the number of words to be generated for a CIW List.

BØ = N,...B3 = N: Same as AØ - A3, only for GPCB.

DVCA + X'CUU',; DVCB = X 'CUU': At least one of these parameters are required and are used to load the GPCC registers 1, 2, and 3. X 'CUU' refers to the I/O Channel (C) and line number (UU) for the GPCC.

TERMINAL MANAGEMENT CONTROL ROUTINE (GPCC GENERATION)

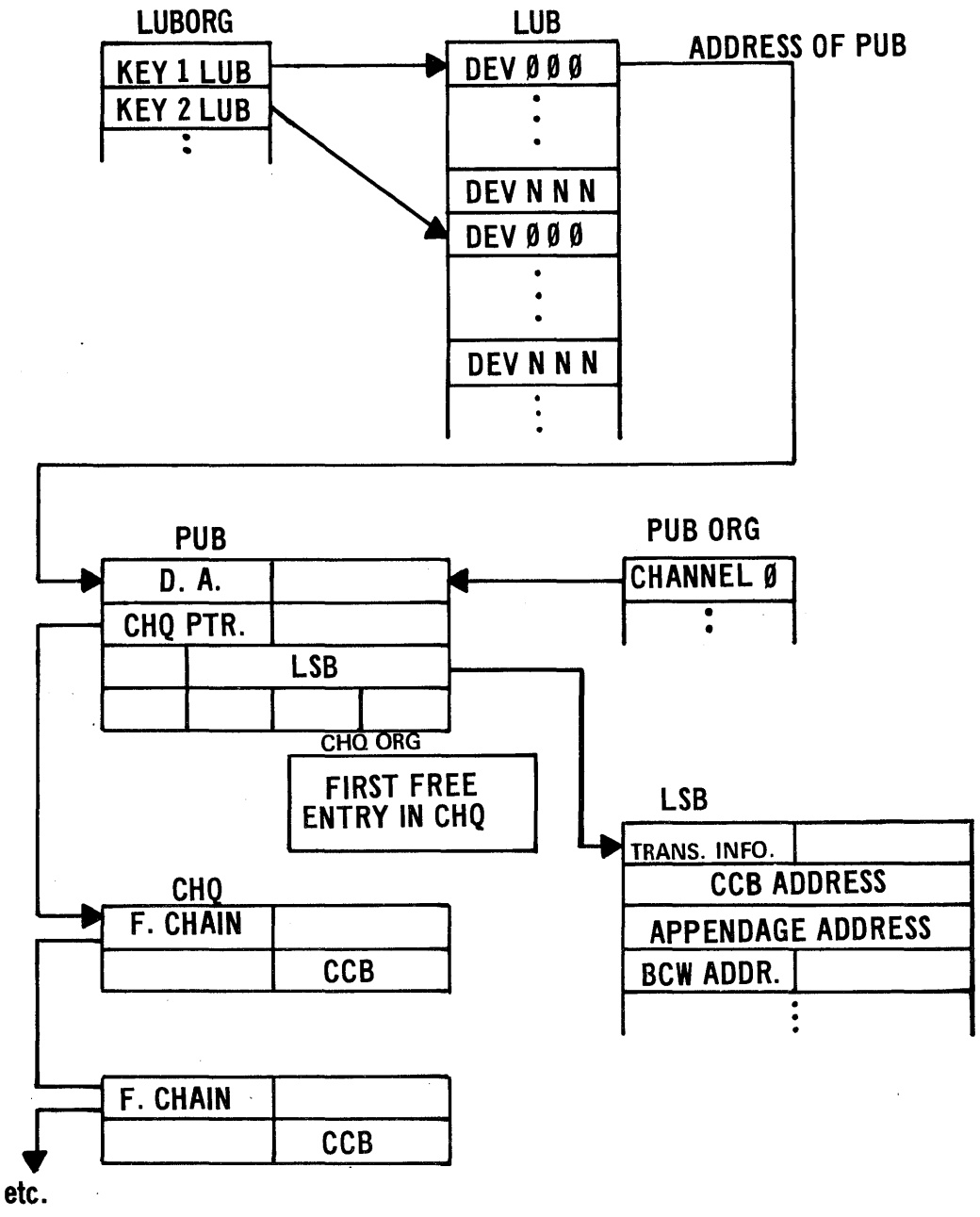
NAME	OP CODE	OPERAND
	GPCGEN	GPCA =N, GPCB =N
		[,AØ =N,A1 =N, A2 =N,A3 =N]
		[,BØ =N,B1 =N, B2 =N,B3 =N] ,
		DVCA =x'CUU',DVCB =x'CUU'

TERMINAL MANAGEMENT CONTROL ROUTINE

(Line Status Block - Table Relationship)

Each PUB (device description) has a pointer to the Line Status Block (LSB) for the line which the device is attached. The LSB contains line information required by TMCR and the CCR's.

TMCR LINE STATUS BLOCK (TABLE RELATIONSHIP)



TERMINAL MANAGEMENT CONTROL ROUTINE

(Line Status Block)

The Line Status Block contains line information required by TMCR and the CCR's. There is an LSB generated for each line supported by a CCR.

For a detailed description of the fields of the LSB reference the Technical Documents pertaining to each CCR.

TERMINAL MANAGEMENT CONTROL ROUTINE (LINE STATUS BLOCK)

LSB

Ø	TRANSMISSION INFORMATION	
4	CCB ADDRESS	
8	APPENDAGE ADDRESS	
C	BCW ADDRESS	DIALER PUB
1 Ø	I/O WAIT SERVICE BIT ADDR.	CLT PUB
1 4	OUTPUT PLACE-TO-GO	

TERMINAL MANAGEMENT CONTROL ROUTINE (LINE STATUS BLOCK GENERATION)

The LSBGEN macro generates a Line Status Block (LSB), which provides certain line indepent information required by the TMCR and CCR. A LSBGEN macro must be coded for each line to be supported by a CCR.

LNTYP=(Option): Specifies the number of multiplexer positions in use for this line where the option is either F (two multiplexer positions) or H (one multiplexer position). Default is H.

DIAL=(Option): Specifies whether this line has an associated automatic dialer. Options are SW, TWX, NO. Default is NO (neither a dialer associated with the line nor a dialer LSB only).

CNFG=(Option): Specifies the configuration of this line. Options are SW (switched line), MTPM (multipoint line with C/SP as master station), MTPT (multipoint line with C/SP as tributary station), PTP (point-to-point line). Default is PTP.

LNSP=N: Specifies the speed in bauds for asynchronous lines. 'N' = 300, 150, 110. Default is 110.

SPFTR=(Option): Special features on this line. Options are ASYNC (asynchronous U100/DCT1000), TTYMDE (TTY or DCT 500 in teletype mode), CRC16 (dinary synchronous lines and VRC/CRC16 error checking is to be used), NO (no special features). Default is NO.

DVC=X'CUU': Refers to the channel and multiplexer position of this line.

DVCASC=X'CUU': Mandatory if there is a dialer CLT position. If DVC specifies the dialer, the DVCASC is associated with the non-dial position, and conversly.

GPCC=(Option): Mandatory if more than one GPCC is available. Must correspond to the GPCA=NN parameter of the GPCGEN procedure. Default is GPCA.

LSBTYP=(Option): Specifies the type of CLT. Option is DIAL or NORM. Default is NORM.

MODE=(Option): Mode of circuit operation. Options are FD (full duplex), SO (simplex output), SI (simplex input), HD (half duplex). Default is HD.

DCTARS=X'XXXX': Specifies the RID SID for an automatic operation DCT 500 on a point to point line. X'XXXX' is the hexadecimal representation of the ASCII RID SID characters.

TERMINAL MANAGEMENT CONTROL ROUTINE (LINE STATUS BLOCK GENERATION)

NAME	OP CODE	OPERAND
	LSBGEN	[LNTYP=<OPTION>] [,DIAL=<OPTION>]
		[,CNFG=OPTION>] [,LNSPD=N]
		[,SPFTR =<OPTION >][,DVC =x'CUU']
		DVCASC=X'CUU' [,GPCC =<OPTION >]
		[,LSBTYB =< OPTION >][,MODE =<OPTION >]
		[,DCTARS =X'XXXX']

SECTION V

COMMUNICATIONS CONTROL ROUTINES

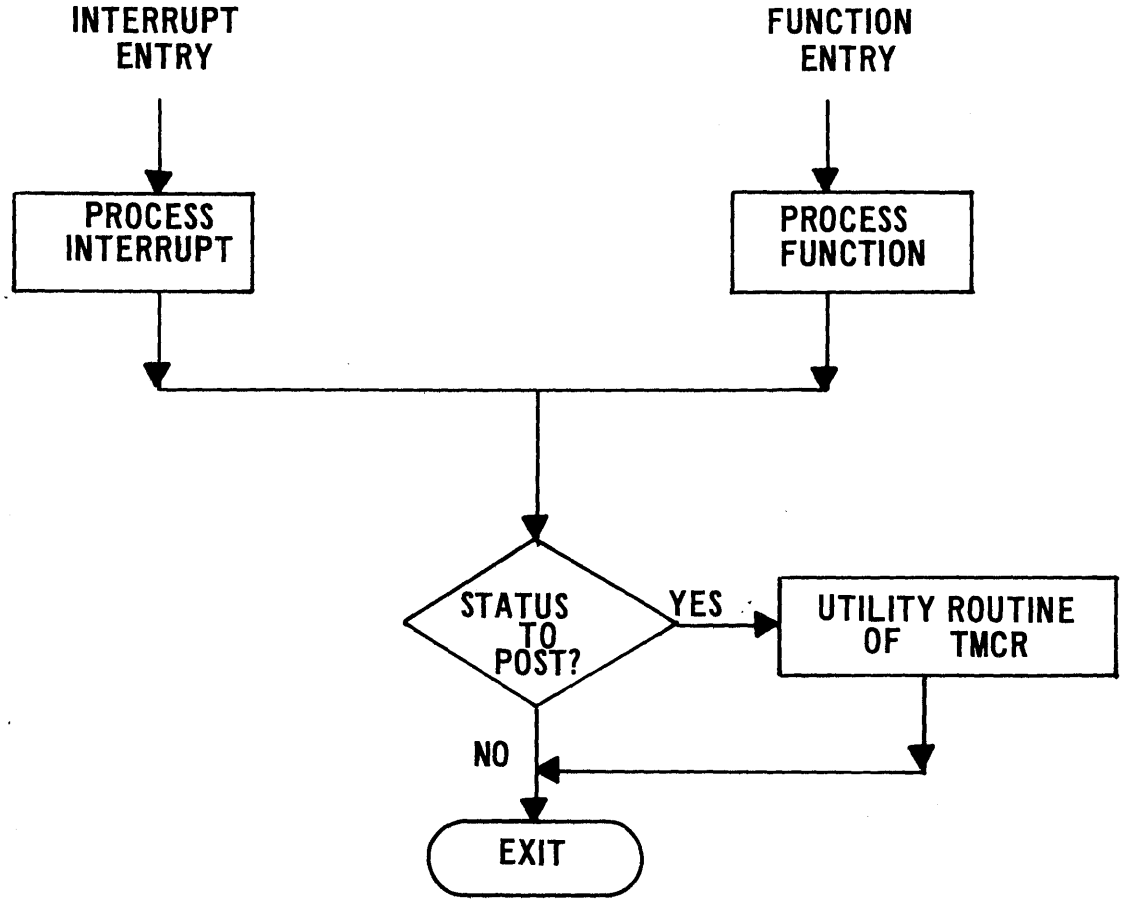
SECTION V INDEX - COMMUNICATIONS CONTROL ROUTINES

COMMUNICATIONS CONTROL ROUTINES

(GENERAL FLOW)

There are two main entry points to each of the CCR's, function and interrupt entry. The function is passed to the CCR for processing as is the interrupt information. After processing, in either case, the CCR determines if there is status to be posted. If status is to be posted, an appropriate utility routine of TMCR gets control, else the CCR just exits.

COMMUNICATIONS CONTROL ROUTINES
(GENERAL FLOW)



COMMUNICATIONS CONTROL ROUTINES (CCR)

CCR's are designed to compensate for the pecularities of a given communication discipline. For devices which have enough in common, a particular CCR may support more than 1 type of etrminal (XS-3 and ASCII code DCT's). All CCR's are re-entrant in that 1 CCR will support all like devices on the system. CCR's control the BCW's and MDW's for each line. BCW, MDW, and CIW's are initialized at sysgen time. Initiation and continutation of I/O is controlled by the CCR's and status information returned via TMCR to the MCP (which issued the function). A Line Status Block (LSB) is established at sysgen time,for each line, by a sysgen procedure. The LSB contains certain accounting and line characteristic information. For devices which require a message parity check, the CCR generates block parity for output messages and checks block parity for input messages. Some character parity is generated and checked by the hardware. CCR's provide the capability for terminal operation communication with the system through the use of control messages.

These control messages permit the terminal operator to inform the system of certain operating contingencies that have developed.

Functions performed by the CCR's and TMCR:

Note: certain types of terminals do not require the use of all these functions. All entry to the CCR's is via TMCR and return is via TMCR or the dispatcher.

1. Open - line marked open.
2. Input - allow data to be input on the line.
3. Output - allow data to be output on the line.
4. Close - terminate the line.
5. Halt - halt a current operation in an orderly manner.
6. Hang up - place the line 'on hook'.
7. Dial - place the line 'off hook'.
8. Test Mode on/off - condition the line for back to back testing.
9. Post - used after input or output functions and supplies buffers to the CCR to permit continuation of the message without interruption.

COMMUNICATIONS CONTROL ROUTINES (CCR)

1. CONTROL BCW's AND MDW's
2. RETURN STATUS TO USER PROGRAM
3. GENERATE BLOCK PARITY
4. PROCESS INTERRUPTS
5. FUNCTIONS
   - A. OPEN
   - B. INPUT
   - C. OUTPUT
   - D. CLOSE
   - E. HALT
   - F. HANG UP
   - G. DIAL
   - H. TEST MODE ON/OFF
   - I. POST

COMMUNICATION CONTROL ROUTINES (CCR)

GENERAL

1. Process functions requested by the MCP after they have been processed by TMCR. Process CIW's and provide status information to MCP through TMCR.
2. Use subroutines of TMCR and TMS.
3. Operate in privileged mode with all interrupts locked out.
4. Re-entrant.
5. Handle variable length input/output buffers.
    a. buffers may be large enough to contain an entire message or fragraments of a message, in which case a function may be issued to provide additional buffers.
6. FUNCTIONS:
    a. IPTIN - initializing batch input or soliciting conversational input.
    b. OPTIN - initializing batch output or sending conversational output.
    c. IPTCT - continuing batch input.
    d. REPEAT - solicit the retransmission of previous input for either conversational or batch input.
    e. OPTCT - continuing batch output.
    f. POST - to present additional buffers after transmission has started.
    g. HNGUP - disconnect a dialed line.
    h. HALT - to halt input operation normally.
    i. CLOSE - to discontinue use of a line.

CCR
(GENERAL)

1. PRIVILEGE MODE

2. RE-ENTRANT

3. VARIABLE LENGTH I/O BUFFERS

4. FUNCTIONS
    a. IPTIN
    b. OPTIN
    c. IPTCT
    d. REPEAT
    e. OPTCT
    f. POST
    g. HNGUP
    h. HALT
    i. CLOSE

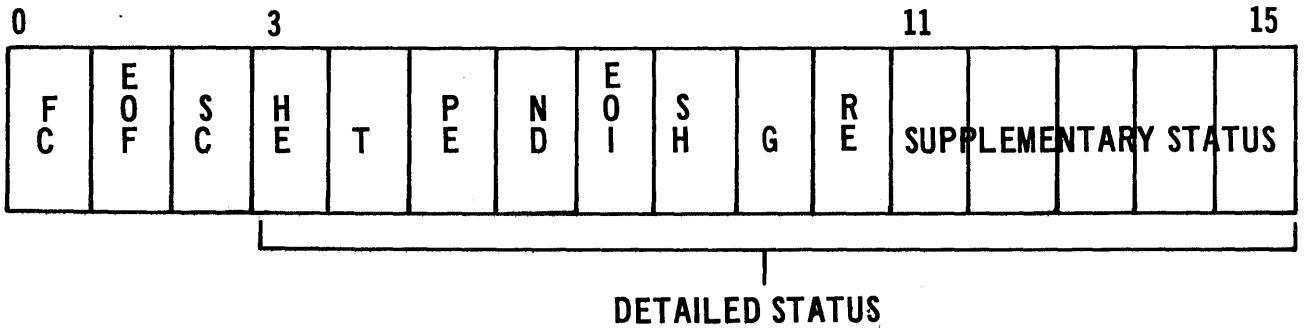
COMMUNICATION CONTROL ROUTINES (CCR)

GENERAL

STATUS:

1. Function complete.
2. End-of-Input file.
3. Status check (detailed):
   a. Hardware errors
   b. Time out
   c. Parity/Format error
   d. No data
   e. End-of-Input image
   f. Software errors
   g. Loss of carrier
   h. Retry exceeded
4. Supplementary status.
   a. Invalid function
   b. Ready
   c. Wait
   d. Suspend current operation
   e. BEL
   f. Hang-up request received
   g. Abort
   h. Terminate
   i. Halt-go-voice
   j. Toggle devices
   k. Who-are-you?
   i. ACK/WABT in response

CCR STATUS


0
3
11
15
FC
EOF
SC
HE
T
PE
ND
EOI
SH
G
RE
SUPPLEMENTARY STATUS
DETAILED STATUS

COMMUNICATION CONTROL ROUTINE (CCR)

GENERAL

MDWGEN - Generates all MDW's required for the user configuration with just one MDWGEN call.

CBAMDW = (<OPTION>, X'WW', X'SS', X'PP'): Specifies the MDW's for controlling a Binary Synchronous ASCII line. Option is YES or NO. If YES, generate the MDW chain. Default is NO. X'WW' is the hexadecimal value of the wait acknowledge character. Default is X'3B'. X'SS' specifies the hexadecimal value of the first character of the C/SP selection address. X'PP' specifies the hexadecimal value of the first character of the C/SP Poll Address. Default is zero.

CBEMDW = (<OPTION>, X'WW', X'SS', X'PP'): Specifies the MDW's for controlling a Binary Synchronous EBCDIC line. The same as CBAMDW except the default for X'WW' is X'6B'.

CTMDW = (<OPTION1>, <OPTION2>): MDW's for Teletype or DCT 500. Option 1 is YES or NO. Default is NO. MDW's to be generated. Option 2 is TLYMDW, DCTMDW, or BOTH. Default is BOTH.

CRMDW = <OPTION>: MDW's for RMS-1 (1004/9000) line. Option is YES or NO. Default is NO.

CDMDW = <OPTION>: MDW's for DCT 2000 ASCII line. Option is YES or NO. Default is NO.

CUSMDW = <OPTION>: MDW's for U100/DCT 1000 Synchronous ASCII line. Option is YES or NO. Default is NO.

CUAMDW = <OPTION>: MDW's for U100/DCT 1000 Asynchronous ASCII line. Option is YES or NO. Default is NO.

CSMDW = <OPTION>: MDW's for Remote/Local C/SP line at the Remote C/SP site. Option is YES or NO. Default is NO.

CLMDW = <OPTION>: MDW's for Remote/Local C/SP line at the Local C/SP site. Option is YES or NO. Default is NO.

COMMUNICATIONS CONTROL ROUTINE
MDWGEN MACRO

LABEL	OP CODE	OPERAND
NAME	MDWGEN	[CBAMDW =(<OPTION>,X'WW',X'SS',X'PP')]
		[,CBEMDW =(<OPTION>,X'WW',X'SS',X'PP')]
		[,CTMDW=(<OPTION> ,<OPTION 2>]
		[,CRMDW=<OPTION>] [,CDMDW =<OPTION>]
		[,CUSMDW =<OPTION>] [, CUAMDW =<OPTION>]
		[, CSMDW =<OPTION>] [,CLMDW =<OPTION>]

CCR

TELETYPE AND DCT 500

HARDWARE: Teletype - must operate in ASCII code.
KSR
ASR
RO

DCT 500 - Teletype mode: Requires no selection sequence.

Automatic mode: selection sequence used on all messages.
a. Fast selection.
b. Slow selection (status poll).
c. Broadcast (status poll prohibited).

One-way mode: DCT 500 cannot transmit.

All device types operate in the asynchronous mode using ASCII eight level (even parity).

Four or two wire, private, or dialed lines may be used.

Multi-point lines supported only for DCT 500 operating in the automatic mode or for one-way broadcast applications to Teletypes and DCT 500 in teletype mode. Line speeds supported: 110, 150, 300 Bands.

Full duplex is not supported except for allowing the terminal to interrupt transmission from the C/SP.

SOFTWARE: CCR is designed to be compatible with the 1108 Teletype and DCT 500 symbionts.

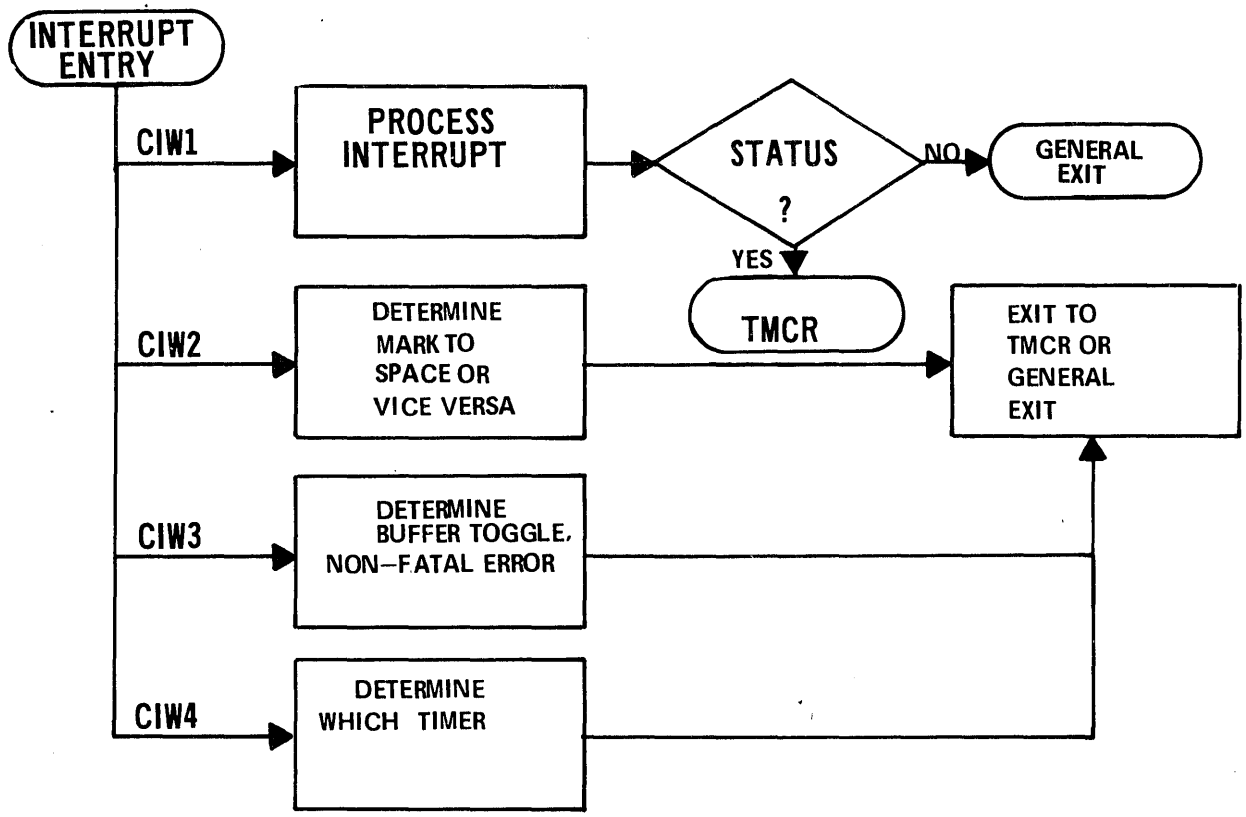
Sysgen - DTYGEN procedure applied at sysgen time.

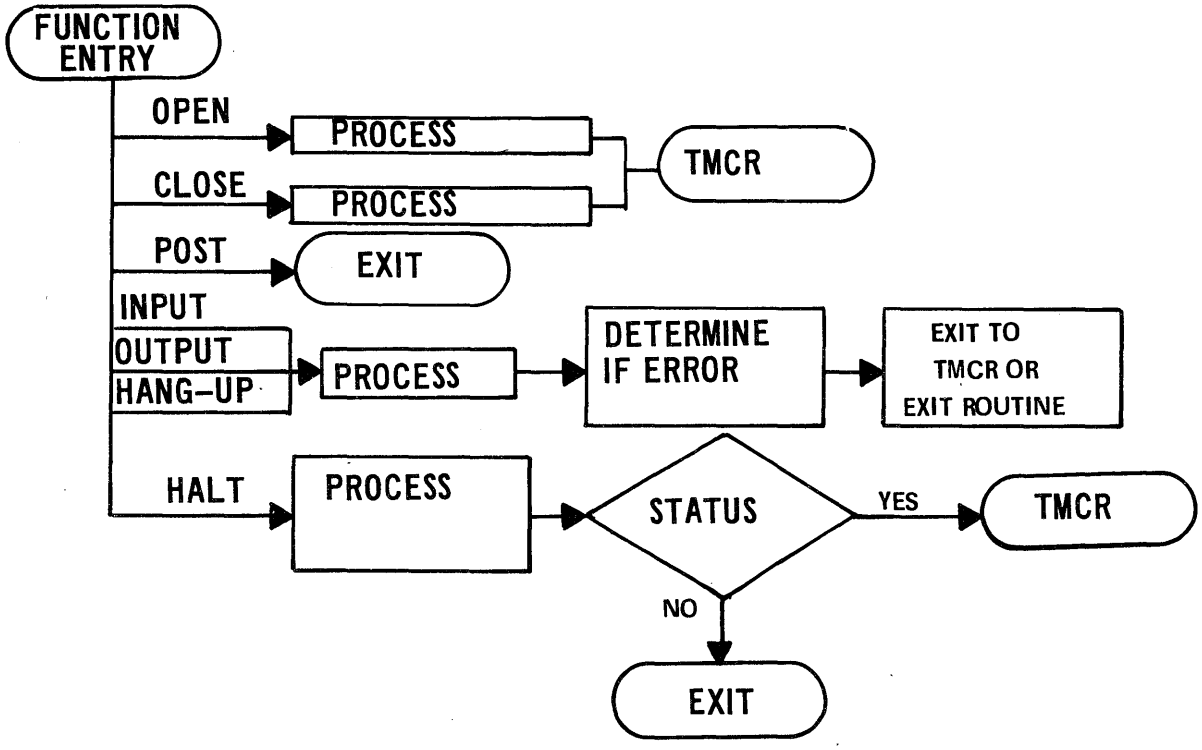
Error Processing:

Non-Fatal errors - occur during message I/O.
1. Parity errors.
2. Input data overrun.
3. Input buffer wrap around.

Fatal errors - cause termination of I/O.
1. Output buffer wraparound.
2. Parity error on access of MDW or BCW.
3. Storage invalid address error.
4. Data carrier lead has gone from active to inactive.
5. CAW rejected.

CCR
TELETYPE AND DCT 500





CCR

UNISCOPE 100 AND DCT 1000

Hardware: Private dialed, multipoint with or without multiplexer) configurations are supported.

Lines may be either synchronous or asynchronous transmission in ASCII code. Full or half duplex operation is provided.

Software: Polling and Selection lists.

Input.
Output.
Block Parity.
Acknowledgements, Errors, and Retransmissions.

Functions:

IPTIN

HALT

IPTCT

REPEAT

OPTIN

OPTCT

POST

HNGUP

CLOSE

Sysgen Parameter - UCDGEN procedure.

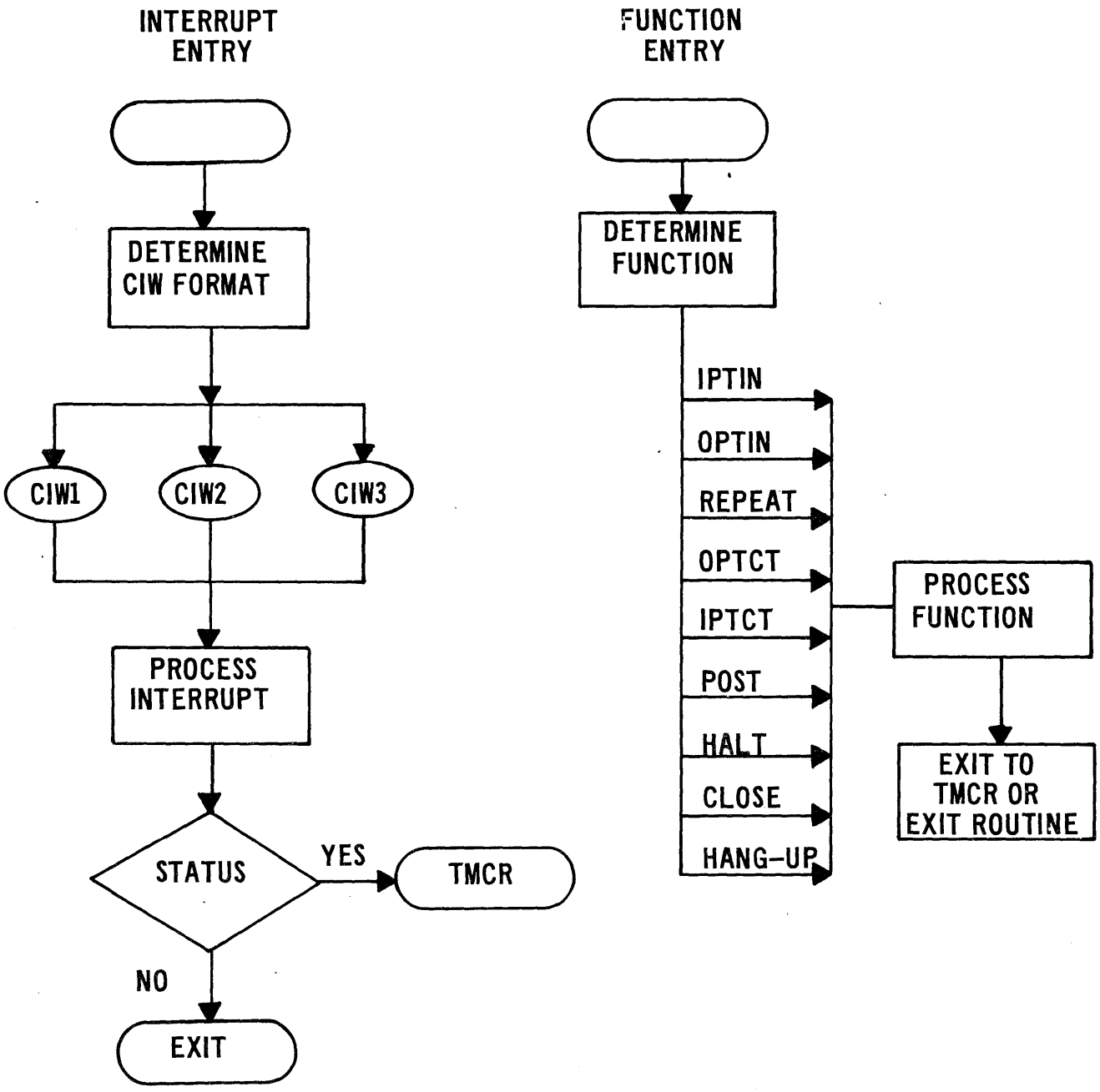
Error Processing:

Non-fatal: Parity.
Polling Timer time outs.
Input Buffer wraparound.

Fatal Errors:

Output buffer wraparound.
Parity error on access of MDW or BCW.
Storage invalid address error.
Data carrier head has gone from active to inactive.
A spacing time-out has occurred (ASYNCH).

CCR
UNISCOPE 100 AND DCT 1000



CCR

DCT 2000

Hardware: Point-to-point (Private) and dialed access lines using ASCII code.

All terminals to be supported are:

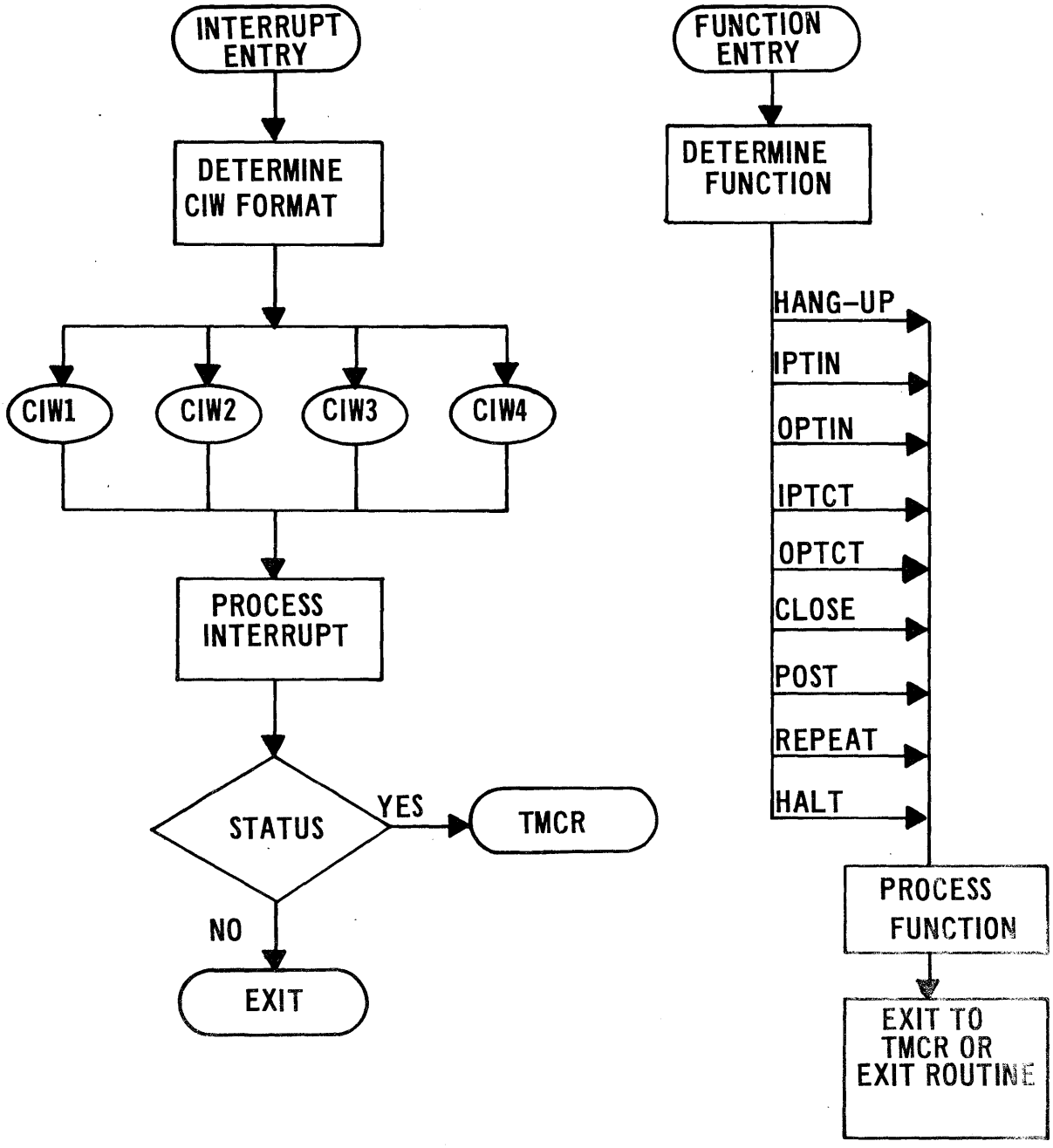
DCT 2000 ASCII
Read/Punch
Punch/Check
128 Print Positions
Short Block
Select Character
Telephone Alert
Error Detection
Forms Control
ASCII Print
Unattended Answering Feature

Software:

Functions:
IPTIN - Solicit Initial Input
OPTIN - Send Initial Output
IPTCT - Send acknowledgement sequence and receive a new message in the accompanying buffer
REPEAT - Send negative acknowledge
OPTCT - Send text from buffer
POST - Present additional buffers after the data transmission has started
HAFIP - Hang up the dialed line
HALT - Halt the input operation
CLOSE - Mark the line closed

Interface: Enter and exit via TMCR. Uses subroutines of the Supervisor for setting and dequeuing system timers and the issuance of all I/O commands.

CCR
DCT 2000


INTERRUPT ENTRY
DETERMINE CIW FORMAT
CIW1
CIW2
CIW3
CIW4
PROCESS INTERRUPT
STATUS
YES
TMCR
NO
EXIT
FUNCTION ENTRY
DETERMINE FUNCTION
HANG-UP
IPTIN
OPTIN
IPTCT
OPTCT
CLOSE
POST
REPEAT
HALT
PROCESS FUNCTION
EXIT TO TMCR OR EXIT ROUTINE

CCR

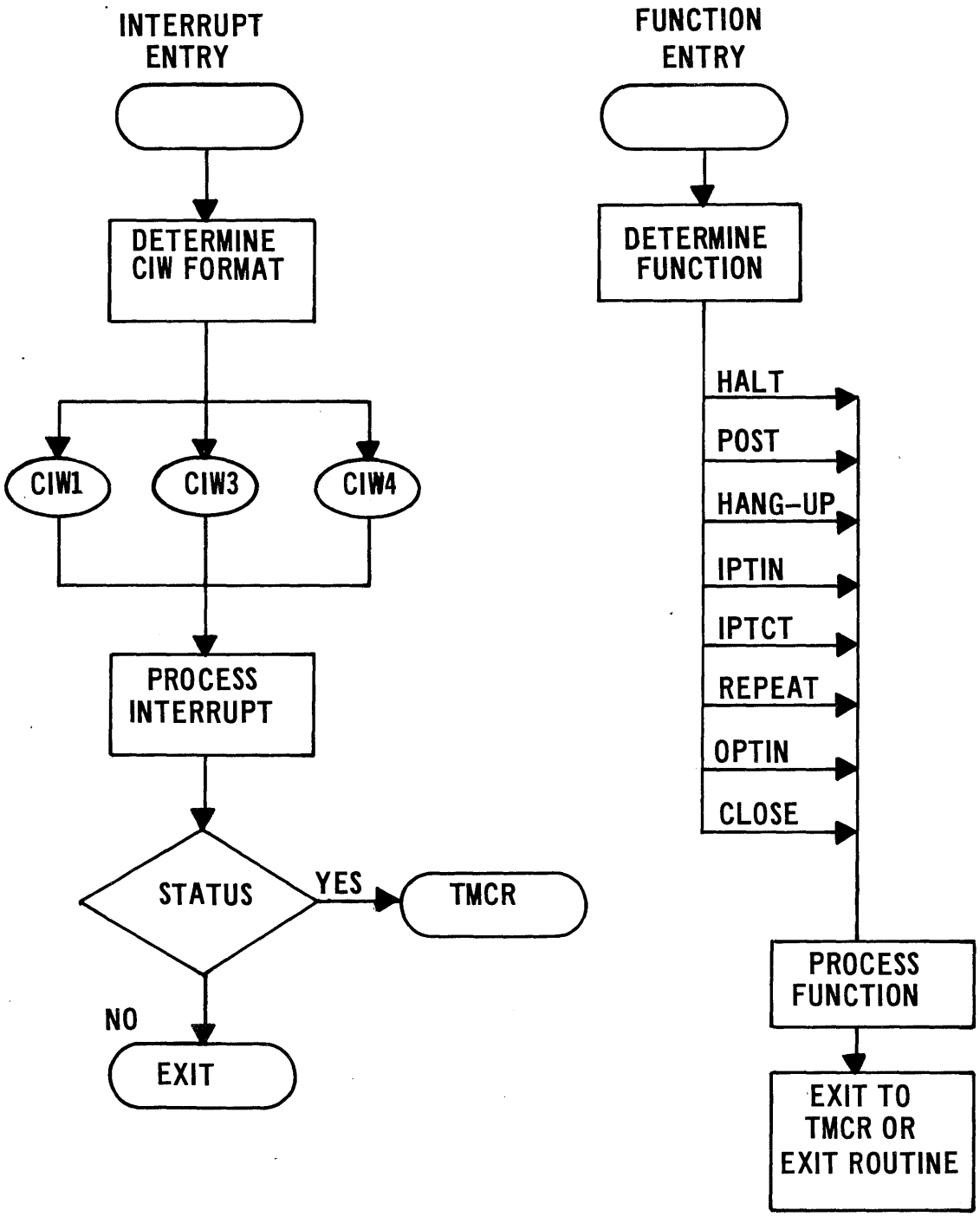
1004/9000

Hardware:
1. Synchronous mode using odd parity and XS3 code (7 level including parity).
2. Half or full duplex, however RMS1 configuration requires half-duplex environment, the full duplex mode is not used.
3. Two line configurations used - switched point-to-point, and non-switched point-to-point.

Software:
1. Variable length I/O buffers maximum at 330 characters plus control characters.
2. Message format of 2 categories:
   a. Control messages - dictate various commands.
   b. Data messages - data being transmitted.

   1. Both - sssss / yyyyo / nnnnm / cccc -- EL / OR / MC
   2. Functions for control - Probe, Output, Input.
   3. Maximum of 330 XS3 data characters for data messages.

CCR
1004/9000



CCR

BINARY SYNCHRONOUS COMMUNICATIONS

Hardware: Private (point-to-point), dialed access, centralized Multi-point Master, and centralized Multi-point Tributary lines are supported.
These lines use either EBCDIC or ASCII code and transparent or non-transparent transmission method.

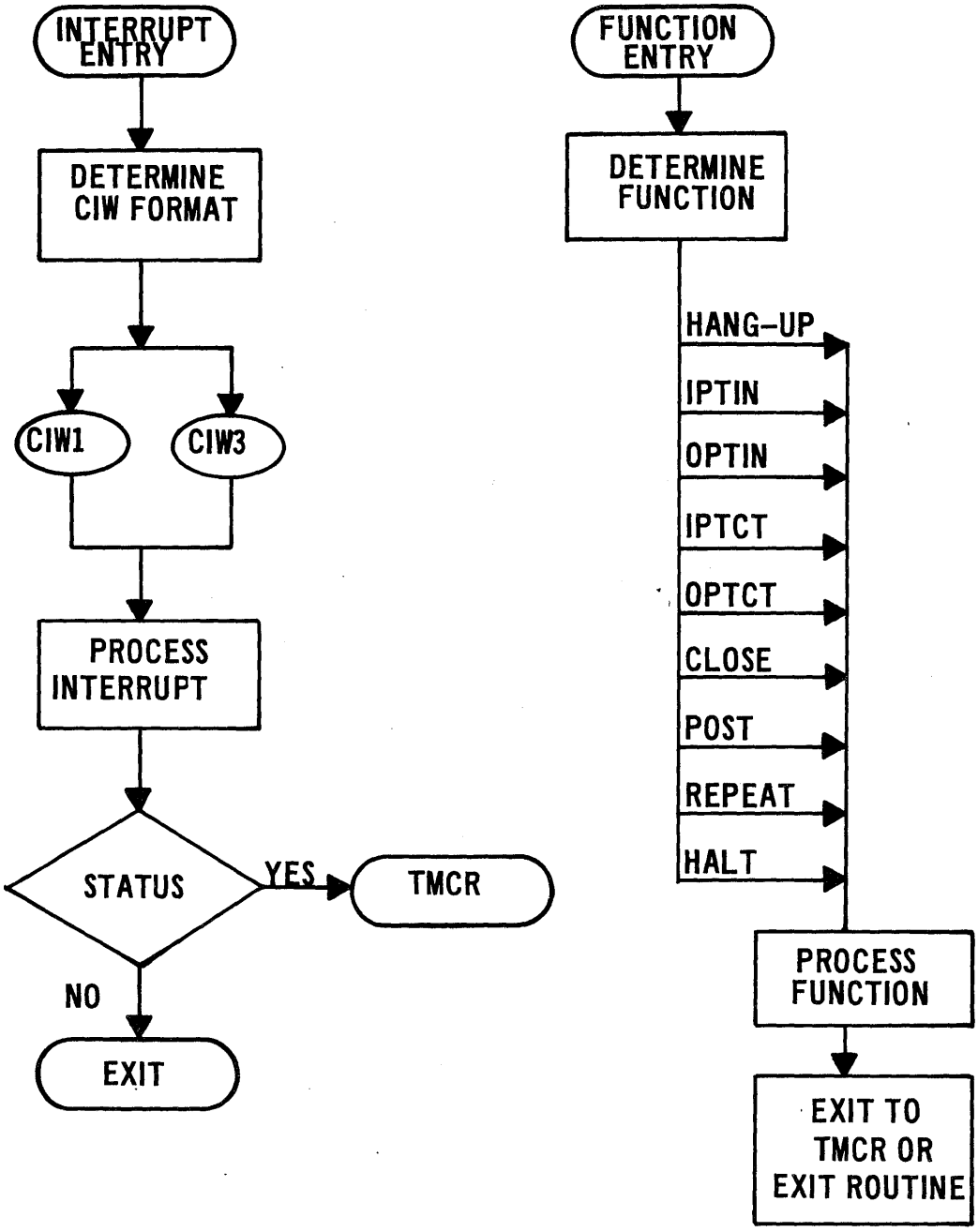
Not supported - Conversational mode
Intermediate Block check
Leading graphics
Six-bit transcode

Software:

Functions:
IPTIN - Solicit Initial Input
OPTIN - Send Initial Output
IPTCT - Send Alternating Acknowledgement and Receive New Message in Accompanying buffer
REPEAT - Send Negative Acknowledgement
OPTCT - Send Text from Buffer
POST - Present Additional Buffers After Transmission Has Started
HNGUP - Hang Up the Dial Line
HALT - Halt Input Operation
Close - Mark Line Closed

Interface: Enter and exit via TMCR. Uses subroutines of the Supervisor for setting and dequeuing system timers.

CCR
BINARY SYNCHRONOUS COMMUNICATIONS



SECTION VI

MESSAGE CONTROL PROGRAM

SECTION VI INDEX - MESSAGE CONTROL PROGRAM

MESSAGE CONTROL PROGRAM (SYSTEM MCP)

The MCP is a C/SP resident program operating in supervisor mode (different protect key). It is a high priority task, gaining control from the supervisor at a level as high or higher than any user task, but at a lower level than the supervisory functions such as I/O queuing, interrupt processing, timer management, etc. MCP is a terminal driver program which receives input from various terminals, routes the input from the terminals and accepts messages from a host processor for transmission to remote terminals. MCP supports capabilities of 1108 EXEC8 such as demand, batch, and real time. MCP handles message translation, editing, remote device acquire, sign-on procedures for demand and remote batch operation and job stream-job control statement sequence checking. MCP is modular in design, allowing the user to generate a program controlling a general class of terminals consistent with the users needs. A means of dynamic buffering is used to optimize the memory requirements for communication lines. The total package for dynamic buffering uses less than 50 words of memory plus three words of memory to describe each pool of buffers.

Service to a communication line is on a first in, first out basis except where more than one service sequence is stacked on a line; then a priority scheme is used. In the case of service required at the same (i.e., simultaneous interrupt requests) a hardware priority is used for processing interrupts. MCP then services the various lines in the order that interrupts occur.

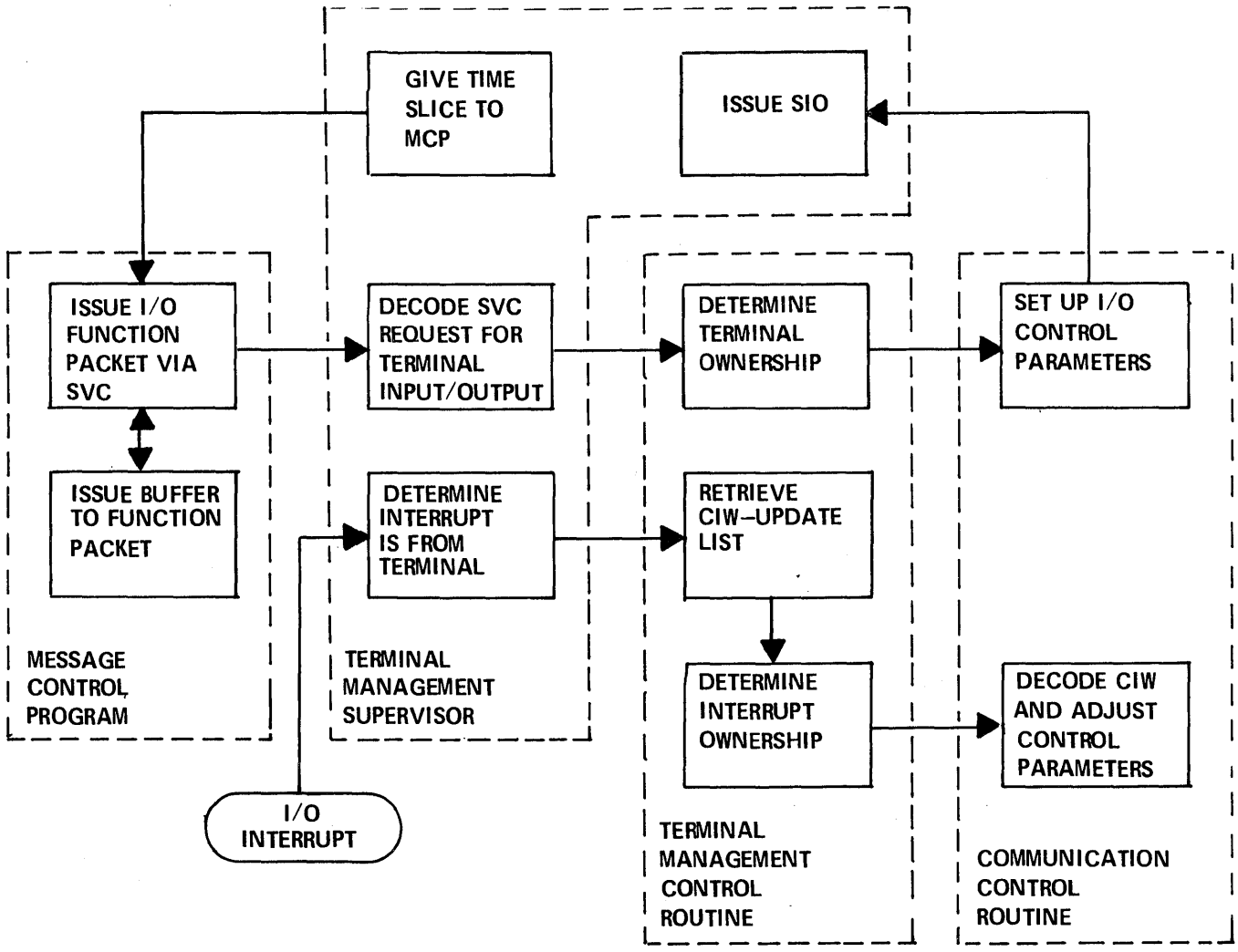
MESSAGE CONTROL PROGRAM (MCP)

1. ACCEPTS INPUT MESSAGES
2. MESSAGE TRANSLATION
3. EDITING
4. REMOTE DEVICE ACQUIRE AND SIGN-ON PROCEDURES
5. CONTROL BUFFER POOLS
6. INITIATES POLLING

MESSAGE CONTROL PROGRAM
(Terminal I/O Flow)

The system Message Control Program (MCP) is the highest priority problem program in the C/SP. Control is passed to the MCP via appendage routines. The MCP acquires or releases buffers, sets up the I/O packet with the proper function code and issues the I/O request by executing a SVC instruction. The Terminal Management Supervisor determines the call is for terminal I/O and control passed to Terminal Management Control Routine and the Communications Control Routine to handle the I/O.

TERMINAL MANAGEMENT CONTROL



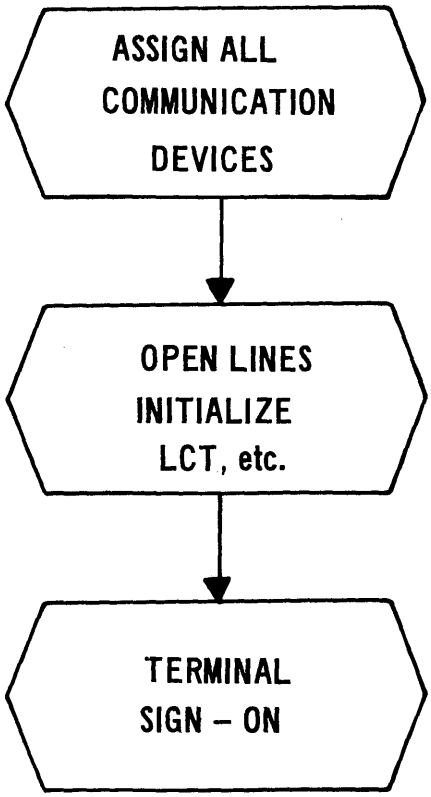
TERMINAL I/O CONTROL FLOW

MESSAGE CONTROL PROGRAM (MCP)
INITIALIZATION

A. Initialization component of MCP used for a variety of initialization tasks.
   1. Assign all communications devices.
   2. Open all communication lines controlled by the MCP.
   3. Initialize the line control tables associated with the various communication lines.
   4. Initialize the service chain.
   5. Initialize linkage from the supervisor to a dummy I/O control block to be posted on any interrupt belonging to the MCP.

B. Terminal Sign-On.
   1. Non-switched lines initialized to receive/solicit a sign-on message from a remote terminal.
   2. Switched lines initialized to receive/solicit when ring-in occurs.
   3. Input message received, sent to host for validation of site I.D.
   4. Valid I.D. indication received, accompanied by a line initialization function containing all tables and entries for line control. The line is flagged 'initialized' and the station is flagged 'sign-on'. Message 'UNIVAC 1108 EXEC8 READY' sent to the station. Control given to the normal I/O routines to handle further transmission. For multi-point lines the same procedure is followed for each station. Under a real time program all stations are considered signed-on and no sign-on procedures are followed.

MESSAGE CONTROL PROGRAM (MCP) (INITIALIZATION)



MESSAGE CONTROL PROGRAM (MCP)

Continuous Input Routine (CIR)

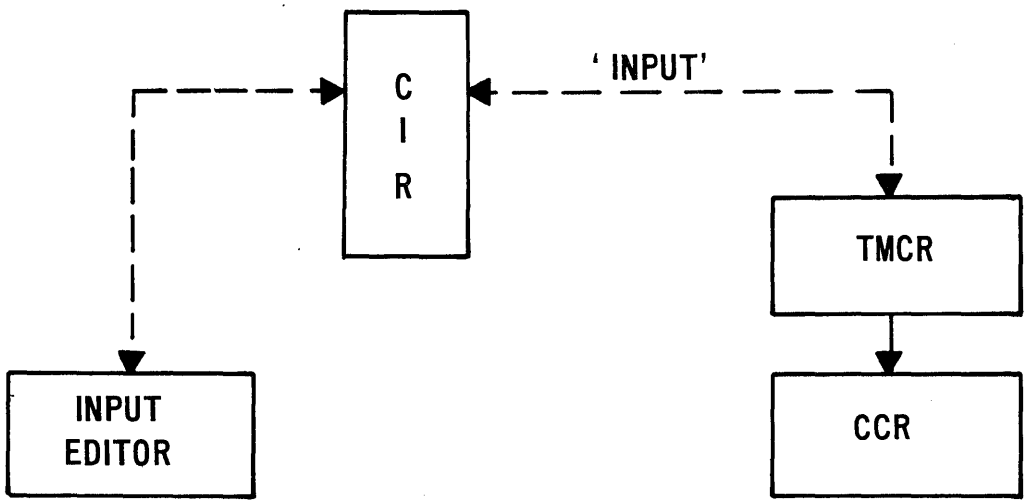
Issues an 'INPUT INITIAL' or 'INPUT CONTINUOUS' function through TMCR to the CCR routines and waits for the input image to complete. Upon completion status is posted at the CCR and appendage routine. Control given back to the CIR via the activity chain through the line activity dispatcher. CIR determines from the status bits if the image was transmitted correctly. If not, CIR issues a REPEAT function to the CCR. If good, CIR gets a host buffer and goes to INEDIT which edits and sends the image to the host.

Continuous Output Routine (COR)

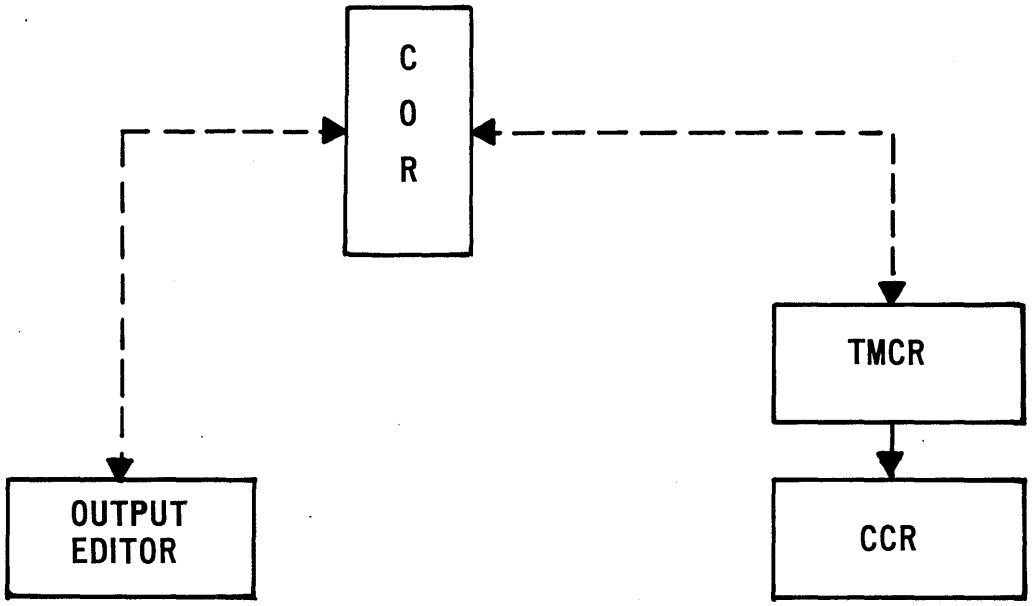
Initiates output to the remote site, analyzes line status, and continues posting line buffers as required for transmission to the remote site. COR is primarily concerned with batch output, however, may include demand or real-time modes. In the demand or real-time modes, control is given to CIR to check for input. If in the batch mode, the MCP notifies the host handler with a send-next-record status that it is ready for the next image. Transmission to remote site continues until an EOT is received from the host. Control is then given to CIR to check for input.

MESSAGE CONTROL PROGRAM (MCP)

(CIR)



(COR)



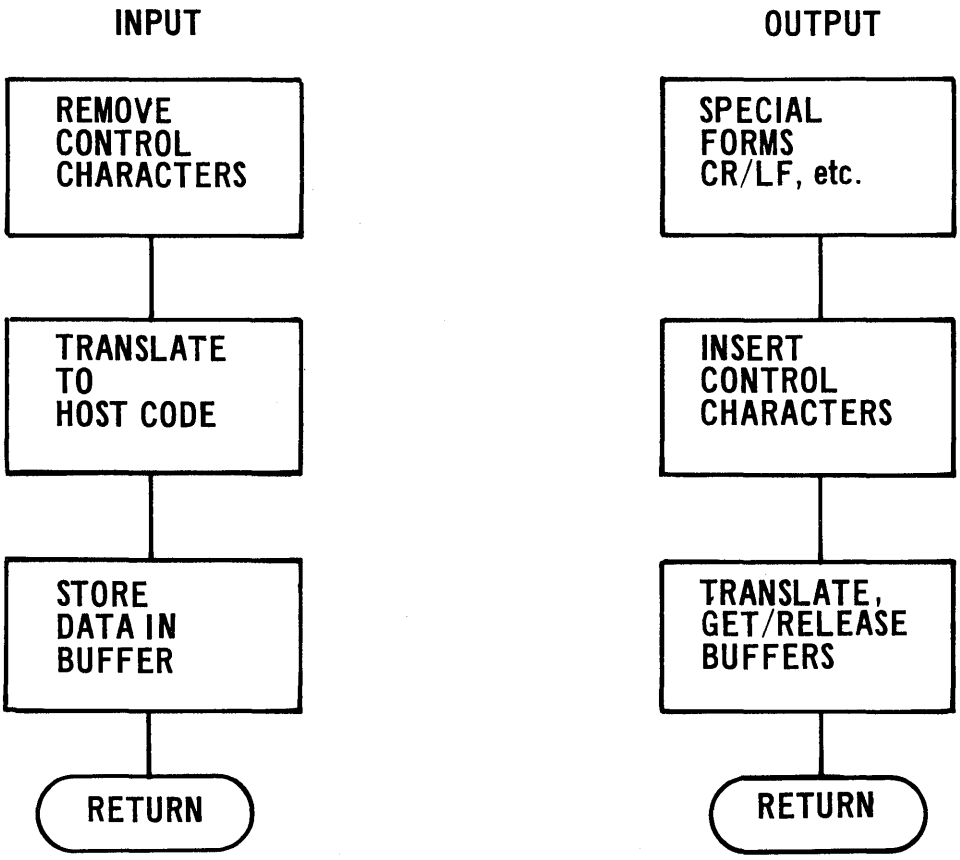
MESSAGE CONTROL PROGRAM (MCP)

EDITING

1. Input Editing - the degree of input editing is controlled by the user through initialization parameters passed to the MCP by the host. Editing of characters is optional such that no editing is performed or editing of all special and control characters is performed such as the removal of CR/LF, LF, SOH, etc. If editing is used, the data is also translated from terminal code to a host code of either ASCII or Fieldata. When editing occurs only the data is transmitted to the host. Four additional data words are given as control information to the inter-computer communications routines. The user has the option of writing his own program to manipulate the data. The user routine receives control after the input data has been stored in a buffer and is ready to send to the host.

2. Output Edit - Purpose is to attach data from the host to an output line descriptor and initiate output to the remote site. Performs all special features for output transmission of messages such as editing special form control sequences (CR/LF, LF, ESC, etc.) in the output message formatting the message through the use of special control characters (SO, SI, EM, ETX, etc.) performing any necessary translation of data, acquiring line buffers, releasing host buffers, chaining line buffers, and compressing data for output. In certain cases output to a remote device is unique, as: RUS1, the output message is compressed, edited, translated and sent in 330 character blocks to the terminal.

MESSAGE CONTROL PROGRAM (MCP)
(EDITING)



MESSAGE CONTROL PROGRAM (MCP)

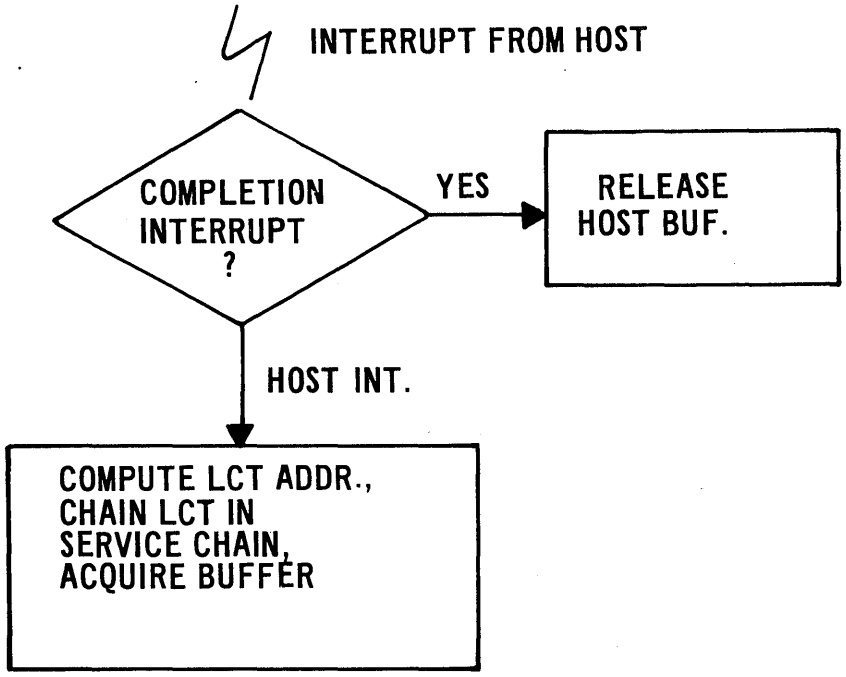
Interrupt Appendage Routines

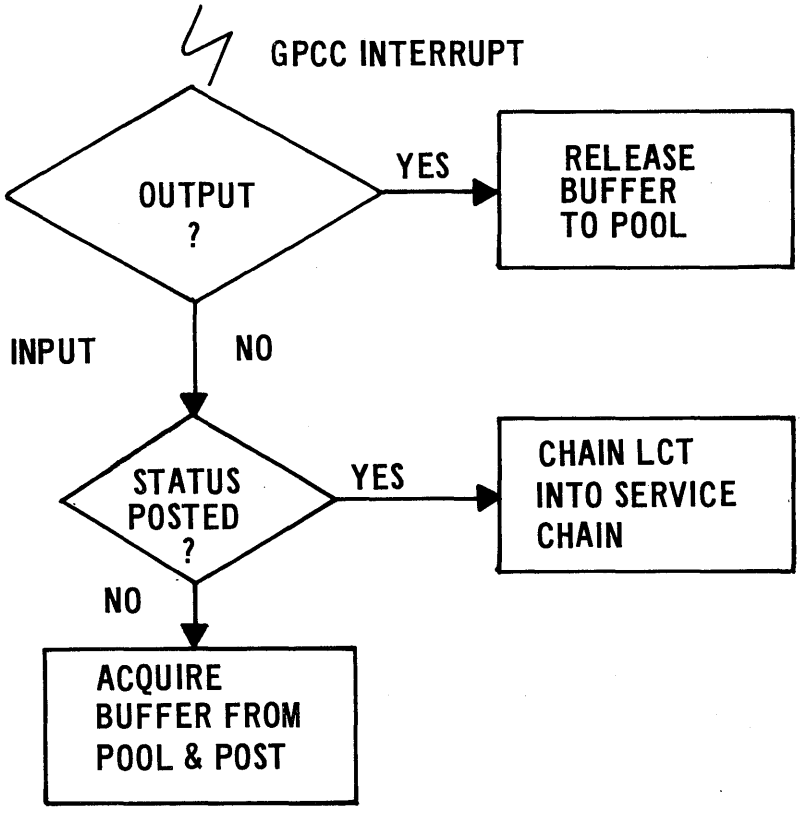
A. Host - Two routines associated with host I/O functions.
   1. Completion Interrupt Routine - Releases the host buffer after the data has been transmitted to the host.
   2. From host interrupt - Routine gets control when the host buffer has filled with data from the host. Purpose is to compute a line control table address and chain the line control table in the service chain. After completion, another host buffer is acquired and posted to the host for the next output data.

B. GPCC - One routine, two logical paths; one for input and one for output (data coming from the host).
   1. Input - Checks to see if status has been posted. If status has been posted by the CCR, the line control table is chained into the service chain. If no status posted, a line buffer is acquired from the buffer pool and posted with the CCR after it has been chained into the line buffer chain.
   2. Output - Releases the line buffer (just transmitted) back to the buffer pool. If status posted by the CCR, the line control table is chained into the service chain.

MESSAGE CONTROL PROGRAM (MCP)

(INTERRUPT APPENDAGE)





MESSAGE CONTROL PROGRAM (MCP)

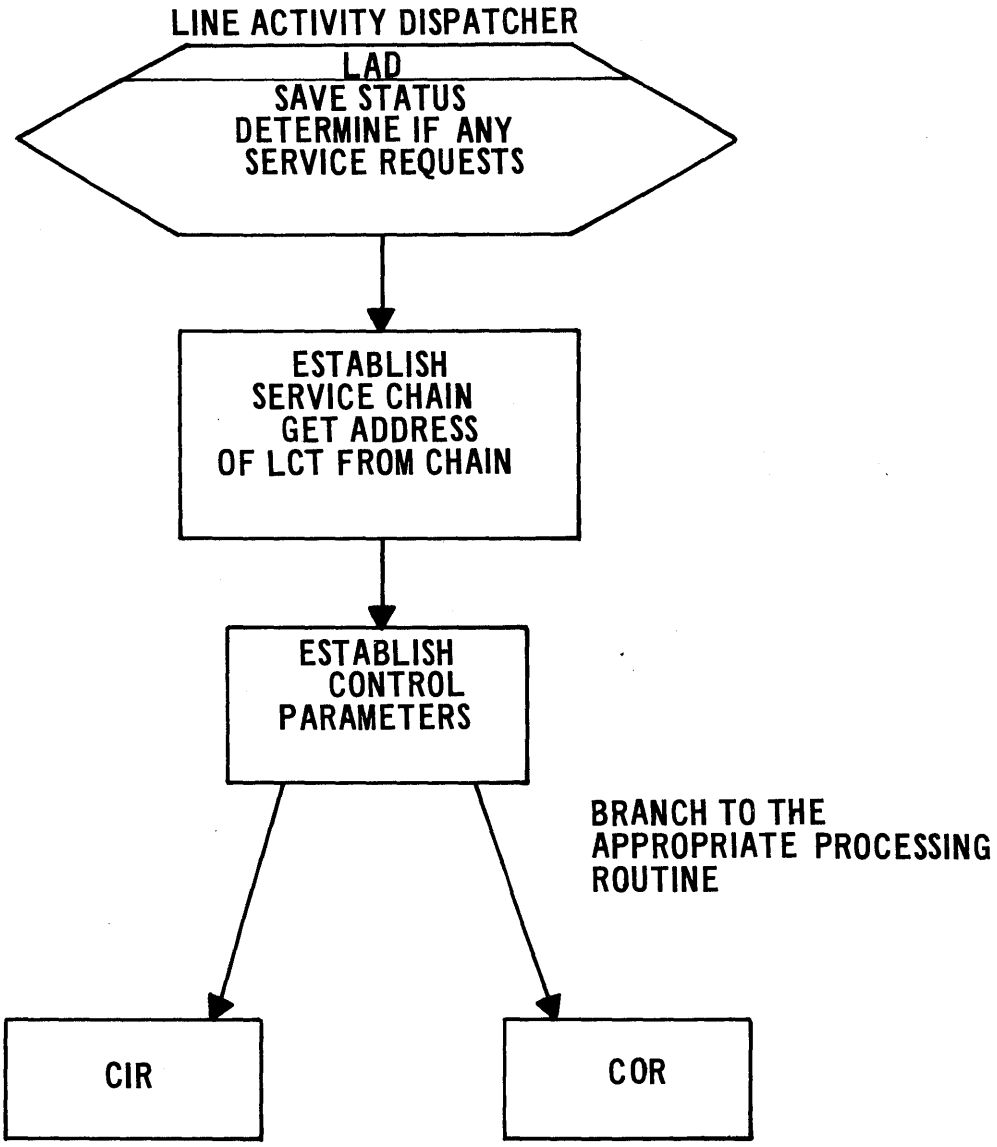
LINE ACTIVITY CHAINING

Line Activity Dispatcher (LAD) - Looks for things to do and dispatches accordingly to the appropriate processing point.

Dispatching for communication line service of I/O interrupts on a first-in first-out basis within a given priority. Priority dispatching occurs when more than one data service sequence is stacked for a given line. Also provides dispatching based on a time-out condition or as directed by an activity completion routine. Upon entry to LAD from an I/O service routine, the status of the line to be serviced is saved and the line is considered in the wait state. LAD then determines if there is any service requests pending (any value other than 255_{10} in the LCT indicates request is pending). If none pending in the interrupt service chain, a wait is issued for the MCP through the supervisor on a dummy CCB which gets posted with a completion on any interrupt for the MCP. Control returned to the MCP after an interrupt occurs and the supervisor dispatcher determines MCP currently has the highest priority on the system. The service chain is established during interrupt processing so the address of the LCT describing the interrupted device is stored in the start of the interrupt chain. LAD removes the address from the start of the chain, places the value in a general prupose register which serves as a base register in addressing any component of the LCT which describes the line requiring service. An interrupt count is associated with each device. If the count is zero (or one for the priority chain) the link to the next LCT requiring service is placed at the head of the chain. If the count is two or more, the head of the chain is not changed. During this time interrupts are masked to prevent the interrupt appendage from restructuring the chain. Before exit from LAD, control parameters for the line are established in various general purpose registers. Transfer of control to an entry point of the appropriate processing routine. The return point may be altered, upon occasion, by interrupt processing or other routines when special requirements for processing are encountered.

MESSAGE CONTROL PROGRAM (MCP)

(LINE ACTIVITY CHAINING)

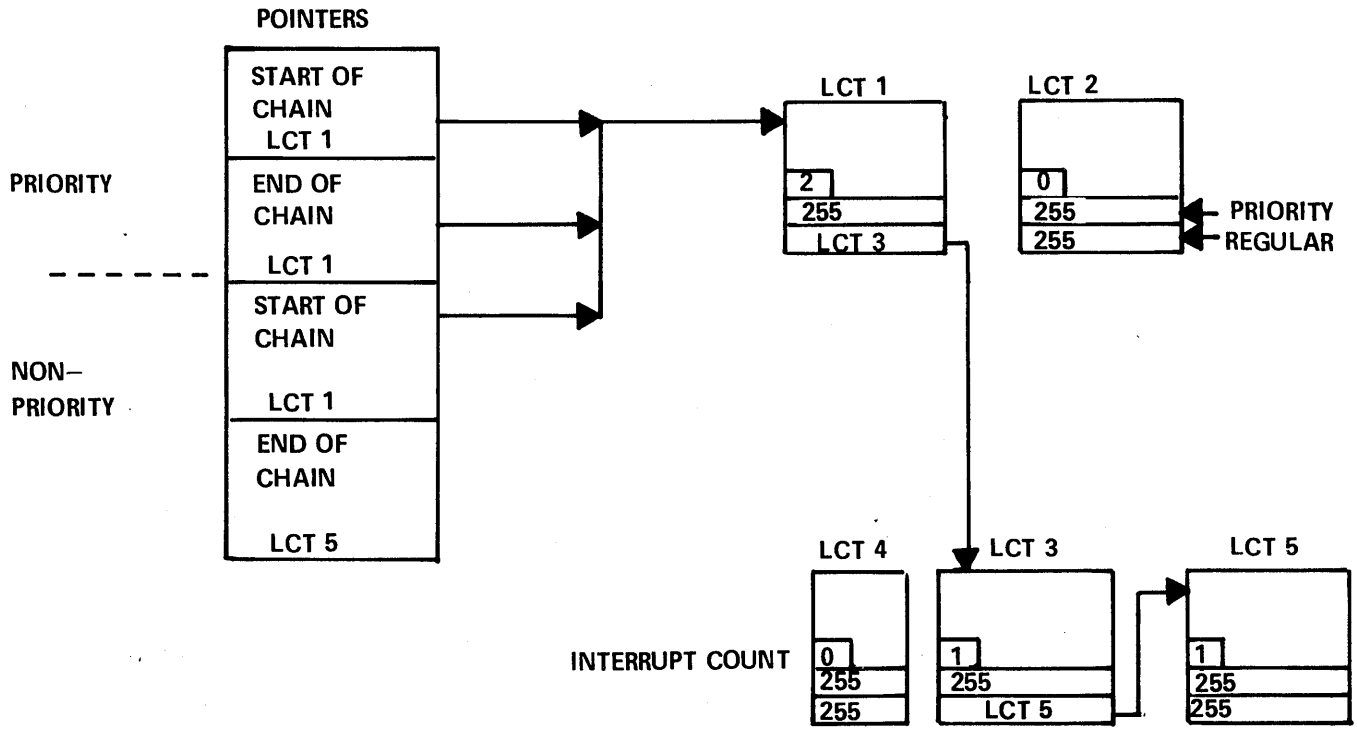


MESSAGE CONTROL PROGRAM (MCP)

LINE ACTIVITY CHAINING

Line Service Dispatcher Sequencer (LSDS) - The operating system has the ability to extend interrupt processing to a user program, allowing the user to create a dispatching scheme through a chain which has been created by a user written interrupt appendage routine. Two constants are used to control each interrupt chain. One, the start of the chain, is manipulated by the MCP. The second is managed by the interrupt appendage routine and points to the final LCT in the chain. This allows quick linking at the end of the chain when an interrupt occurs. Each time the interrupt appendage routine is entered the interrupt count for that LCT is increased by 1. If the value is 2, the link is placed in the priority chain. If the count is 3 or more no chain manipulation takes place as only two levels of priority are maintained. This allows for a dispatch for the same LCT, once it has become the head of chain, until it no longer has priority processing to be performed.

MESSAGE CONTROL PROGRAM (MCP)
LINE ACTIVITY CHAINING



MESSAGE CONTROL PROGRAM (MCP)

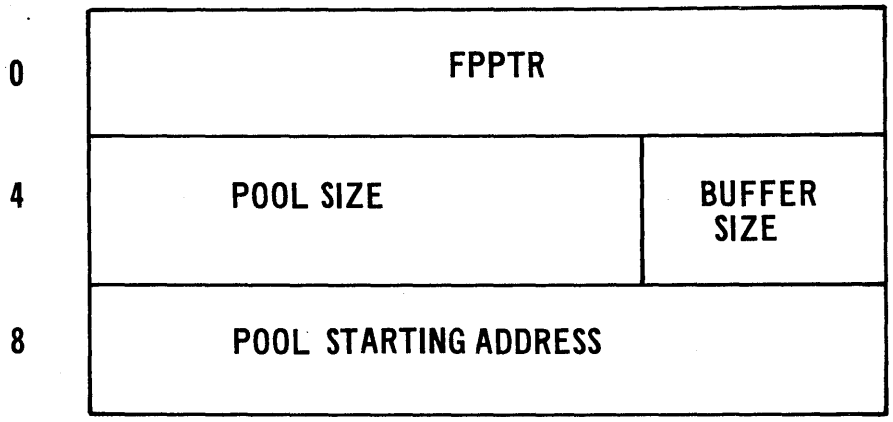
BUFFER MANAGEMENT

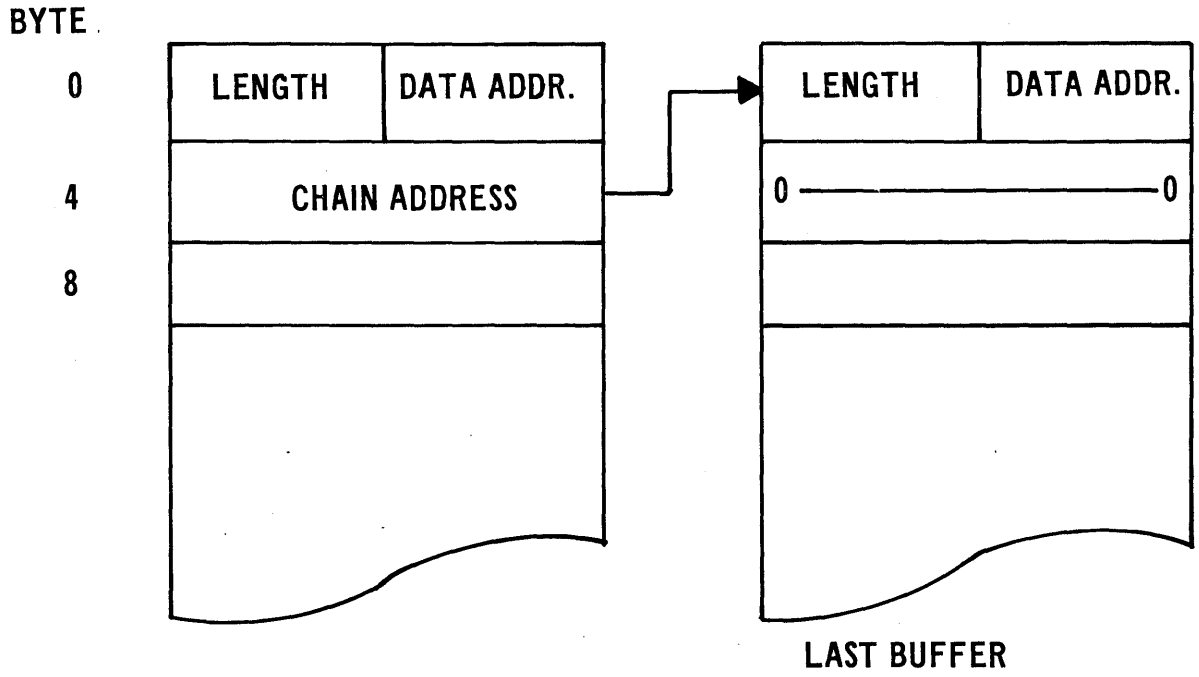
Fixed Size Dynamic Buffering - A method to acquire and release buffers to/from a buffer pool. The advantage provides for a fixed time to acquire/release a buffer of given size. The disadvantage is the rigidity of buffer size as compared to variable buffer size. May have variation by having pools of various buffer size. There is a Pool Descripter Block (PDB) for each size buffer in the system and are located sequentially. A Free Pool Pointer (pointer to the first free buffer in the pool) is the first word in the PDB. The first word of each buffer is a chain pointer to the next available buffer in the pool. The last buffer in the pool is flagged with x'FF' in the first byte of the buffer. A request for a buffer always causes a buffer of at least 8 bytes larger to be given as the first word is established as a buffer control word and the second is reserved as a link to the next buffer of a chain. A buffer is generally of fixed size for a given class of remote devices, therefore the buffer size should be optimized for the size of commonly requested buffers.

Buffer Chaining - A system of chaining I/O buffers is required in order to minimize storage requirements, so that as buffers are transmitted they may be released or acquired from the buffer pool as needed.

A 'GET BUFFER' routine allocates the buffer requested. It builds word 1 (Buffer Control Word) of the allocated buffer. The length value in the BCW is the length supplied in the 'GET BUFFER' calling parameter. The address in the BCW will point to byte 8 of the buffer. The second word is reserved as a forward chain address to the next buffer in the chain.

MESSAGE CONTROL PROGRAM (MCP)
POOL DESCRIPTOR BLOCK (PDB)





MESSAGE CONTROL PROGRAM (MCP)

BUFFER ACQUISITION

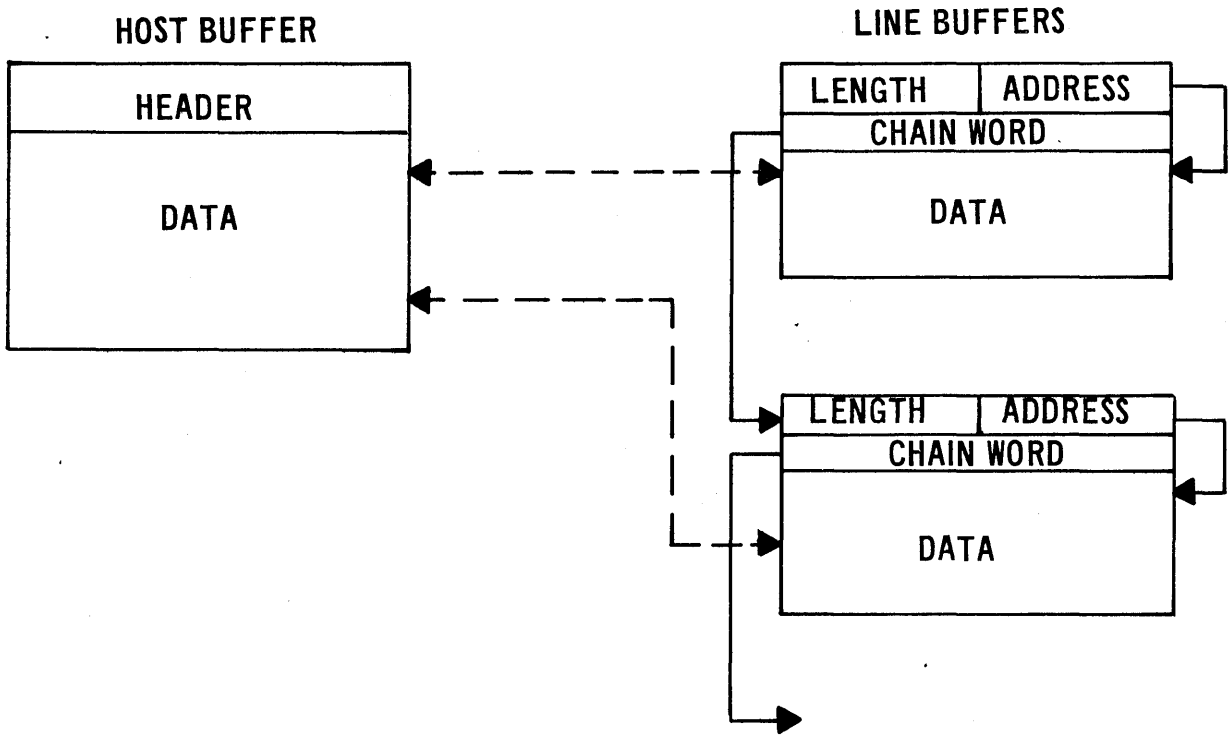
Allows for reading data from host at some optimum size, then editing and moving the output data to a number of smaller buffers chained together. The larger buffer returned to pool and the smaller released as they are transmitted. On input buffers are acquired as they are needed. When enough data has been collected to warrant transmission to host a single large buffer is acquired and the data from the smaller is moved into the large buffer. The smaller buffer released as they are emptied.

When a buffer becomes full during a read operation, a buffer swap is undertaken through hardward control, an interrupt word is stored, interrupt processing eventually takes place and the MCP gets control at a logical re-entry point of the read routines. The read routine determines no error has taken place, then places the 'Buffer Address' valve from the LCT in the second word of the buffer just filled. The valve buffer address is then moved to 'Current Buffer Address'.

A request for another buffer is made and the address and length of the buffer in the form of a BCW is issued to the appropriate CCR. The data length is accumulated and compared to an optimum length (specified by the host at line initialization time) and if the optimum length is surpassed or an end of record interrupt is given, a large buffer is requested (at least 16 bytes larger than the data record length) from the buffer pool. The 16 bytes are used as a temporary area for host handler status data, a chain to the next item to the host, and the LCT address. Editing and translation of the input data is started, moving data from the chained string of buffers to the larger buffer as it is edited.

MESSAGE CONTROL PROGRAM (MCP)

(BUFFER ACQUISITION)

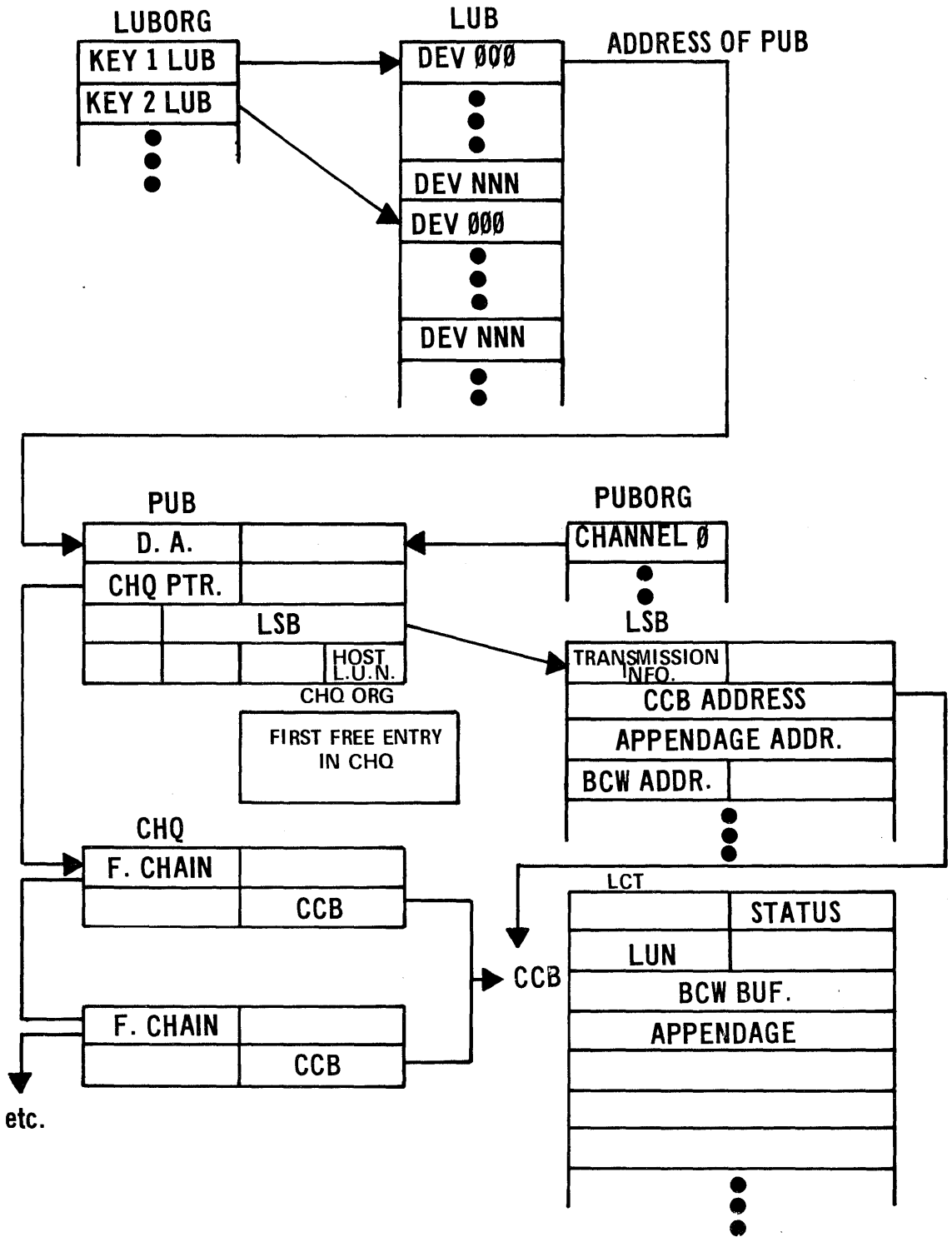


MESSAGE CONTROL PROGRAM

(LCT - Table Relationship)

Information used during the operation of the terminals on the line is kept in the Line Control Table (LCT). The passing of information to/from the MCP is done with one of the general purpose registers pointing to the Command Control Block (CCB) which is the first 28 bytes of the LCT.

MESSAGE CONTROL PROGRAM (LCT-TABLE RELATIONSHIP)



MESSAGE CONTROL PROGRAM (MCP)

TABLES

Line Control Table (LCT) - A 140 byte table containing parameters used in operating the perminals on the line. The first 28 bytes are the Command Control Block (CCB). Reference the Technical Document for the Message Control Program to obtain the detailed description of the fields of the LCT.

LINE CONTROL TABLE

Ø	TRANSMISSION INFORMATION		STATUS HALFWORD		CCB
4	LOGICAL UNIT NUMBER		RESIDUAL COUNT		
8		BCW BUFFER ADDRESS			
12	CONT. OPTIONS	APPENDAGE ROUTINE ADDRESS			
16	DEVICE TYPE	COMM. OPTIONS	PHYSICAL UNIT NUMBER		
20		POLL LIST ADDRESS			
24		SELECTION LIST ADDRESS			
28		SERVICE CHAIN POINTER			
32	I/O PLACE – TO – GO ADDRESS				
36	RETURN ADDRESS				
40	TIMER	IE FLAG	SYSTEM MESSAGE		
44		START INPUT BUFFER CHAIN			
48		ACTIVE INPUT BUFFER			
52		POSTED INPUT BUFFER			
56	ERROR COUNT	HOST FNCT	APPENDAGE	TRANSLATE	
60	LINE STATUS	LINE FLAGS	INTERRUPT	HOST STATUS	
64	INPUT CHARACTER COUNT		HOST BUFFER SIZE		
68	LGSF 1	STATION CONTROL ADDRESS			
72		HOST INPUT BUFFER ADDRESS			
76	INPUT CHARACTER COUNT		DID	STATION#	
80	INPUT OPTION	INPUT SOL	LINE SPEED	OUTPUT OPTION	
84		HOST OUTPUT BUFFER ADDRESS			
88	HOST RESIDUAL COUNT		HOST COUNT		
92		START OUTPUT BUFFER CHAIN			
96		ACTIVE OUTPUT BUFFER			
100		DATA BUFFER			
104	INPUT SAVE AREA				
108	SAVE AREA				
112	SAVE AREA				
116	SAVE AREA				
120	SAVE AREA		LINE BUFFER SIZE		
124		POLL LIST ADDRESS			
128		SELECTION LIST ADDRESS POINTER			
132	ENTRY SIZE		ENTRY CNT	RID	
136	SID	DID	FLAG		

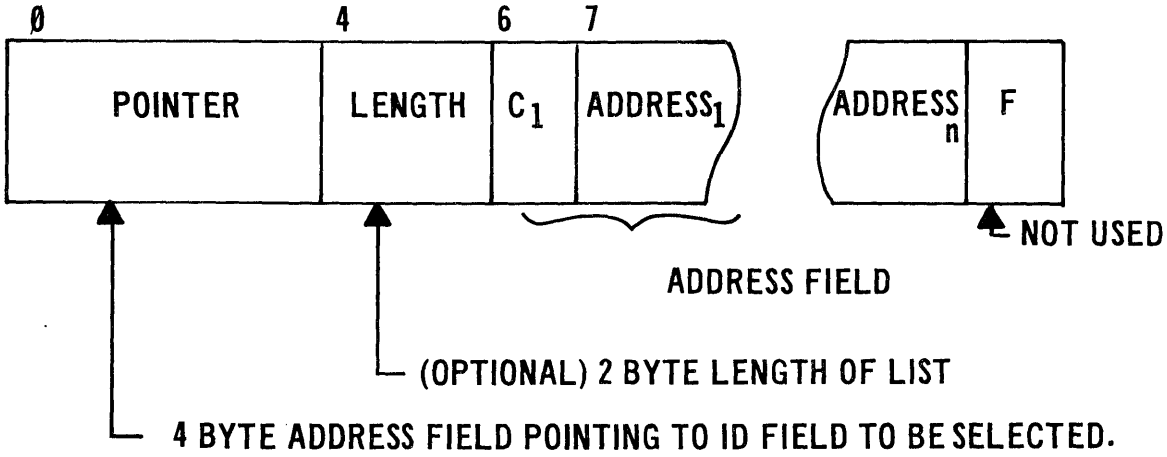
MESSAGE CONTROL PROGRAM (MCP)

TABLES

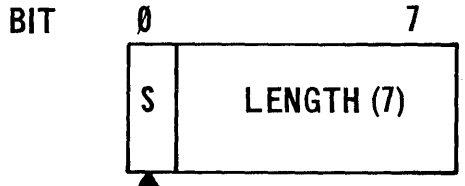
Selection List Table - A one-entry poll list with the length field and Timer Control Block field absent. Used to pass the terminal address characters to the CCR to output data to the terminal.

SELECTION LIST

SELECTION LIST FORMAT



C_1: ADDRESS FIELD CONTROL BYTE.



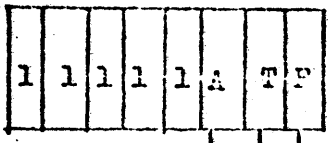
SKIP BIT: MEANING PER CCR:
0100/TTY/DCT 2000 DOES NOT APPLY
DCT 1000 –S=0 CONVERSATIONAL MOLD
S=1 BATCH MODE
BSC/DCT 500 – S=0 SEND HEADER
S=1 DO NOT SEND HEADER

MESSAGE CONTROL PROGRAM (MCP)

TABLES

Poll List Table - Used to pass the terminal addresses to the CCR so it can solicit data from the terminals on the line.

F: Flag Byte

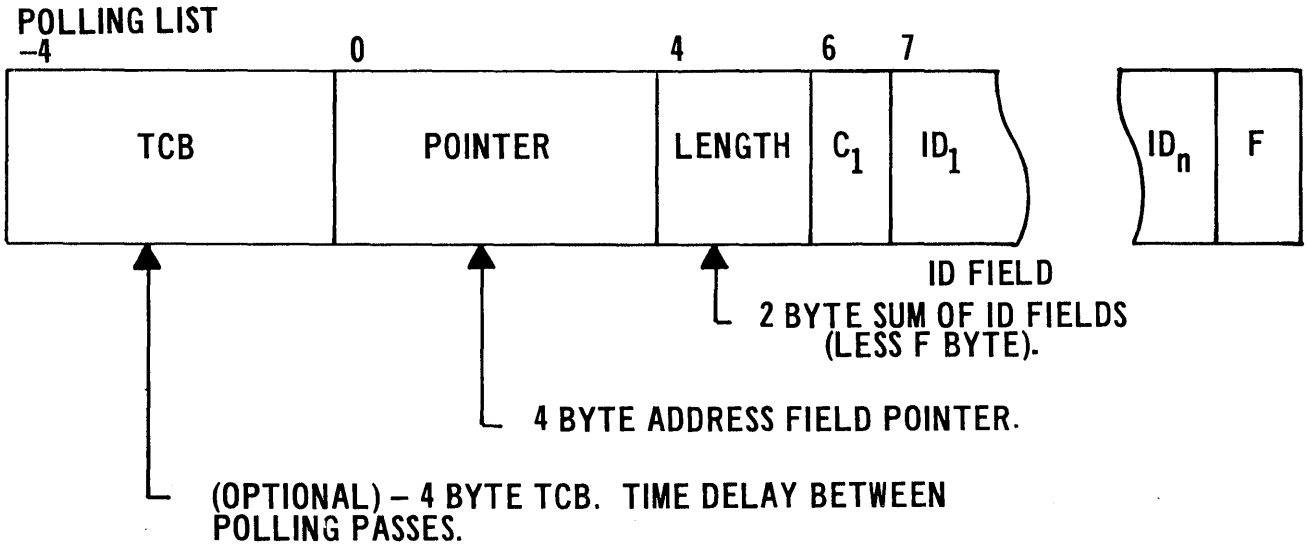


F - Continuation Flag - If set, restart poll.

T - Time Delay Flag - If set and F set, restart poll after time-out (TCB interval).

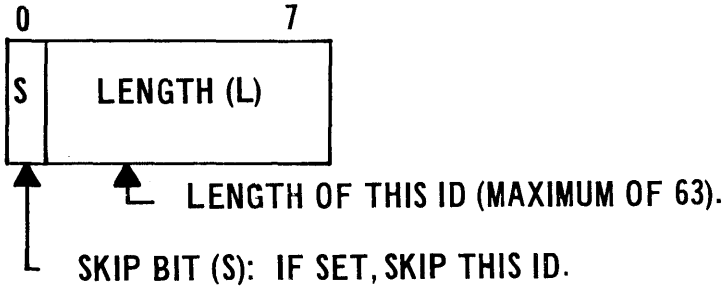
A - Activity Flag - If set and "F" set, polling will be restarted immediately notwithstanding the condition of "T". This flag is set by the CCR when any response to poll is received other than no traffic. The "A" flag is set by the MCP when an output message is sent. The "A" flag is cleared by the CCR when the pointer is repositioned to the first of the list.

POLL LIST



FOR THE CURRENT CCR'S, THE ID FIELD TO APPEAR AS FOLLOWS:

C : CONTROL BYTE



MESSAGE CONTROL PROGRAM (MCP)

TABLES

Station Control Table - Used to control individual terminals on a line. If the line is point-to-point the first word is not used, and the entry is contained in the LCT. If the line is multi-point, a buffer is acquired and the table placed in it.

Number of terminals - on the network.

Flags - bits to indicate the condition of the terminal.

Address - the characters that address the terminal.

STATION CONTROL TABLE

0	NOT USED		NUMBER OF TERMINALS	
4	B FLAGS	FLAGS	ADDRESS	ADDRESS

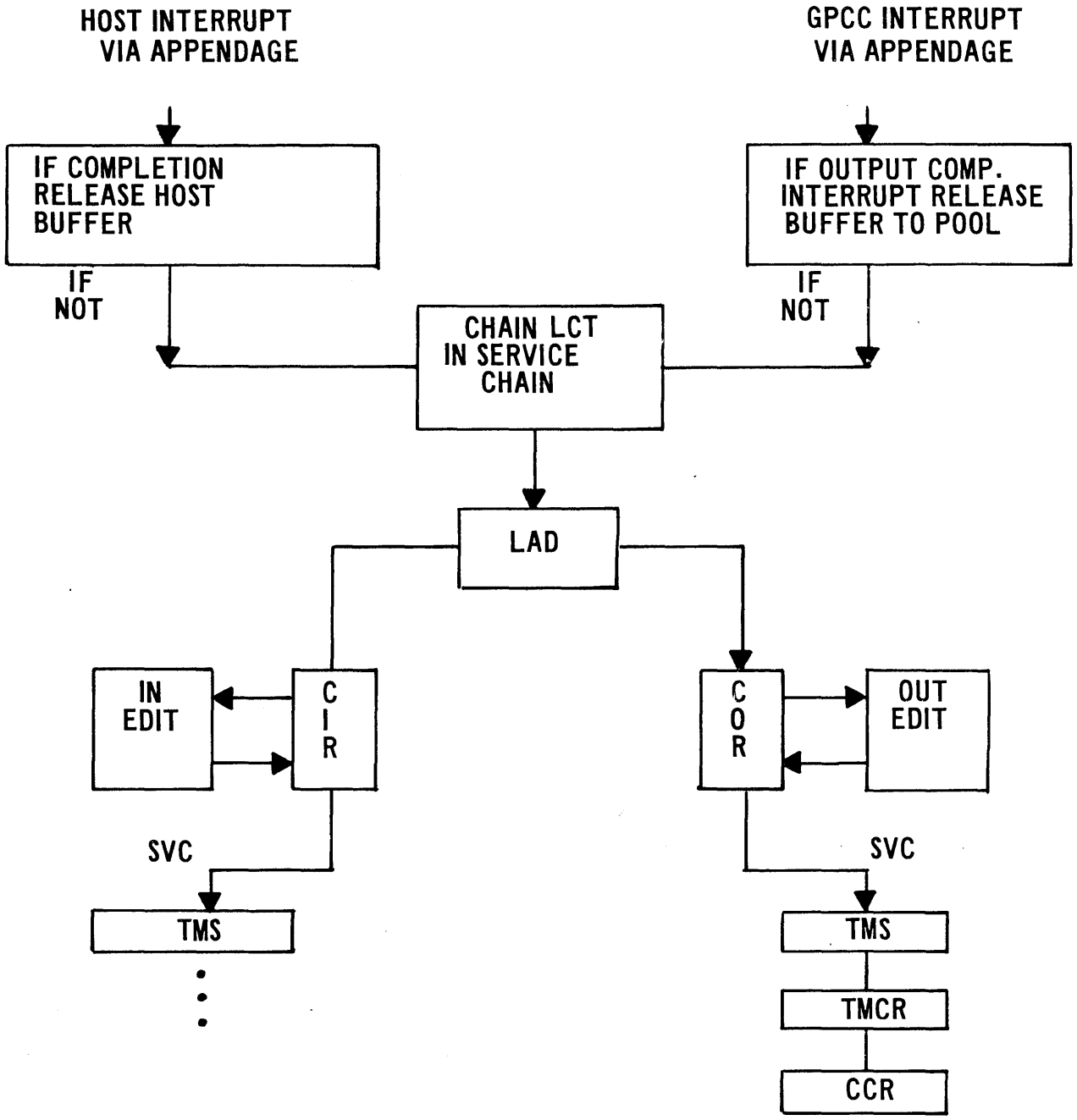
MESSAGE CONTROL PROGRAM

(BLOCK OVERVIEW)

Control is passed to the system MCP via appendage routines with a pointer to the Line Control Table for that line. The LCT's are chained in a service request chain on a first-in, first-out, basis. The line activity despatcher module of the MCP determines if the service is for input or output and passes control to either the continuous input or continuous output routines. If any editing needs to be done, the respective editor is given control. The function to be performed is determined, the parameter passing packets are set up and a SVC instruction is executed for that function.

MESSAGE CONTROL PROGRAM

(BLOCK OVERVIEW)



MESSAGE CONTROL PROGRAM (MCP)

(GENERATION)

The MCP macro is called at system generation time. It sets all global symbols to generate the non-terminal dependent code and sets parameters for reference by the MCP routines.

HOSBUF = N: The number of bytes the MCP buffers will contain for ICA data transfers. Sizes are adjusted to a multiple of 12. Default, HOSBUF = 160.

CSPNO = N: The logical C/SP number. Zero is the onsite number. Default is zero.

LCTAB = N: The maximum number of lines the MCP is to support. Used to allocate memory space for a table. Default, LCTAB = 8.

HCODE =< OPTION >: The character code supported as Host computer codes, ASCII, FD, or BOTH. Default is BOTH.

ITIMER =<OPTION>: Indicates whether the MCP gross timer is to be used to detect errors. Option is YES or NO. If YES, the supervisor timer feature must also be selected. Default is YES.

MESSAGE CONTROL PROGRAM (MCP) (GENERATION)

LABEL	OP CODE	OPERAND
	MCPGEN	[HOSBUF =N] [,CSPNO =N] [,LCTAB =N] [,HCODE =<OPTION>] [,ITIMER =N <OPTION>]

MESSAGE CONTROL PROGRAM (MCP)

(BUFFER GENERATION)

Used to generate the buffer pools used by MCP to handle I/O. A maximum of 10 separate buffer pools may be generated.

P1: Specifies the buffer size in bytes of the pool of smallest buffers.
P2: Specifies the number of buffers in the pool.

These parameters are used in pairs. The odd numbered parameter indicating the size of the buffer and the next even numbered parameter indicating the number of buffers in the pool.

The buffer size must be a multiple of 4.

A pool should also be specified for Host buffers. The buffer size should be 8 bytes greater than the HOSBUF specification in the MCPGEN macro, allowing for the larger header of the Host buffers.

MESSAGE CONTROL PROGRAM (MCP)

(BUFFER GENERATION)

LABEL	OP CODE	OPERAND
	BUFGEN	P1, P2, ...,P20

MESSAGE CONTROL PROGRAM (MCP)

(LINE CONTROL TABLE GENERATION)

The LCT macro builds a line control table containing the necessary information to control a line for the initial input to the MCP and also sets global symbols used to generate the MCP terminal dependent code. The LCT macro is called once for each line the MCP is to control.

DEVC = X'CUU': The physical channel and the CLT position of the line.

DEVICE =<OPTION> : The type of terminal to be on that line where option is:

TTYA - Teletype
TTY5 - Teletype or DCT500 in teletype mode.
DCT5 - DCT500 in semi-automatic mode.
U100 - U100/DCT1000.
DCT1 - U100/DCT1000, column binary capability.
RMS1 - 1004/9000 with a RMS1 program.
DT2A - DCT2000 with ASCII code.
BSCA - BSC with ASCII code.
BSCE - BSC with EBCDIC code.

LSPD = N: The line speed in BAUDS, i.e.bits per second. N may equal 110, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, or 9600.

Defaults:
Teletype - 110
DCT500 - 300
U100/DCT1000 - 2000
DCT2000 - 2000
RMS1 - 2000

DUBSIZ = N: The size of the data portion of the buffers used in terminal I/O operations. 'N' must be a multiple of 4. Defaults:

Line Speed	N
110-1800	20
2000-4800	44
9600	80

SWNT =<OPTION >: Switch line. Option either YES or NO. Default is YES.

MESSAGE CONTROL PROGRAM (MCP)
(LINE CONTROL TABLE GENERATION)

LABEL	OP CODE	OPERAND
	LCT	DEVC =X'CUU',
		DEVICE =< OPTION >
		[,LSPD =N] [,BUFSIZ =N]
		[,SWNT =< OPTION >]

SECTION VII

SYMBIONT CONTROL PROGRAM

SECTION VII INDEX - SYMBIONT CONTROL PROGRAM

SYMBIONT CONTROL PROGRAM

The SCP is a complex of resident C/SP routines which provide an interface between the C/SP - 11ØØ configuration and any supported on-site I/O device.

The main function of the SCP is to reduce overhead from the 11ØØ by assuming the functions of data collection and distribution. Provides a standard interface between the 11ØØ and the I/O devices connected to the C/SP eliminating the need for the number of peripheral control routines on the 11ØØ and the overhead associated with the gathering and distribution of data.

SYMBIONT CONTROL PROGRAM

DEVICES SUPPORTED

Ø716 CARD READER

Ø6Ø4 CARD PUNCH

Ø786 PRINTER

Ø920 PAPER TAPE DEVICE

CONTROL STATEMENTS

(SCP RECOGNIZED)

Various control statements are recognized by SCP and are processed dependent on the statement. The FILE, REPL, and SNAP require immediate attention of the Host and are transferred on an image basis. All other images are blocked in SDF Format and transferred to the Host.

CONTROL STATEMENTS (SCP RECOGNIZED)

@ RUN

@ FIN

@ DATA

@ REPL

@ ELT

@ COR

@ FILE

@ SNAP

@ END

@ COL

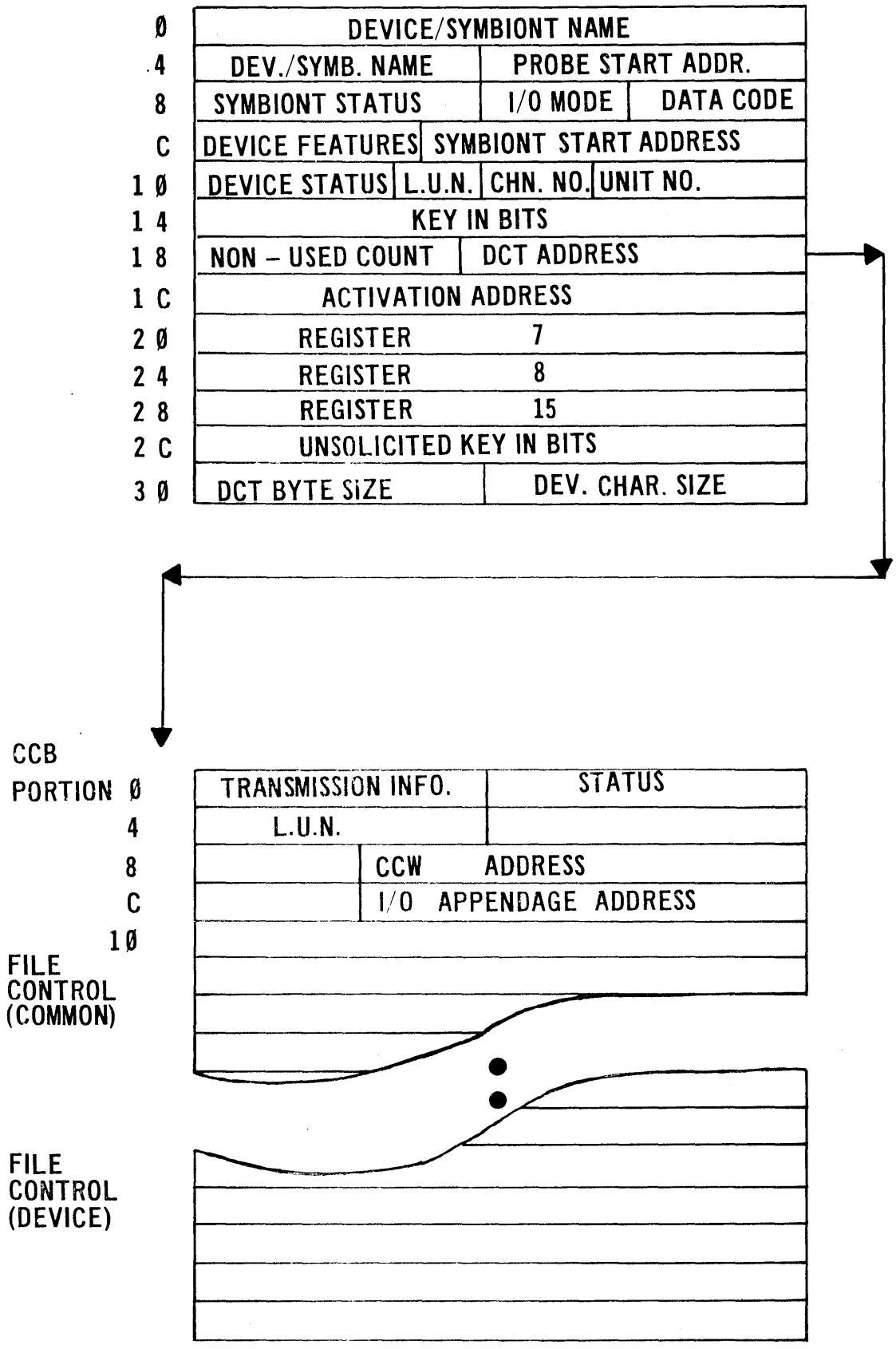
@ ENDCL

PERIPHERAL CONTROL TABLE (PCT)

Each device assigned to the C/SP has a peripheral control table where device status and code links are maintained. The PCT of each device is probed periodically to determine if the device needs to be activated. If so, control is passed to a routine which assigns, reserves and ups the device. Then control is passed to the routine to handle the input or output depending on the device.

DEVICE CONTROL TABLE (DCT)

PERIPHERAL CONTROL TABLE

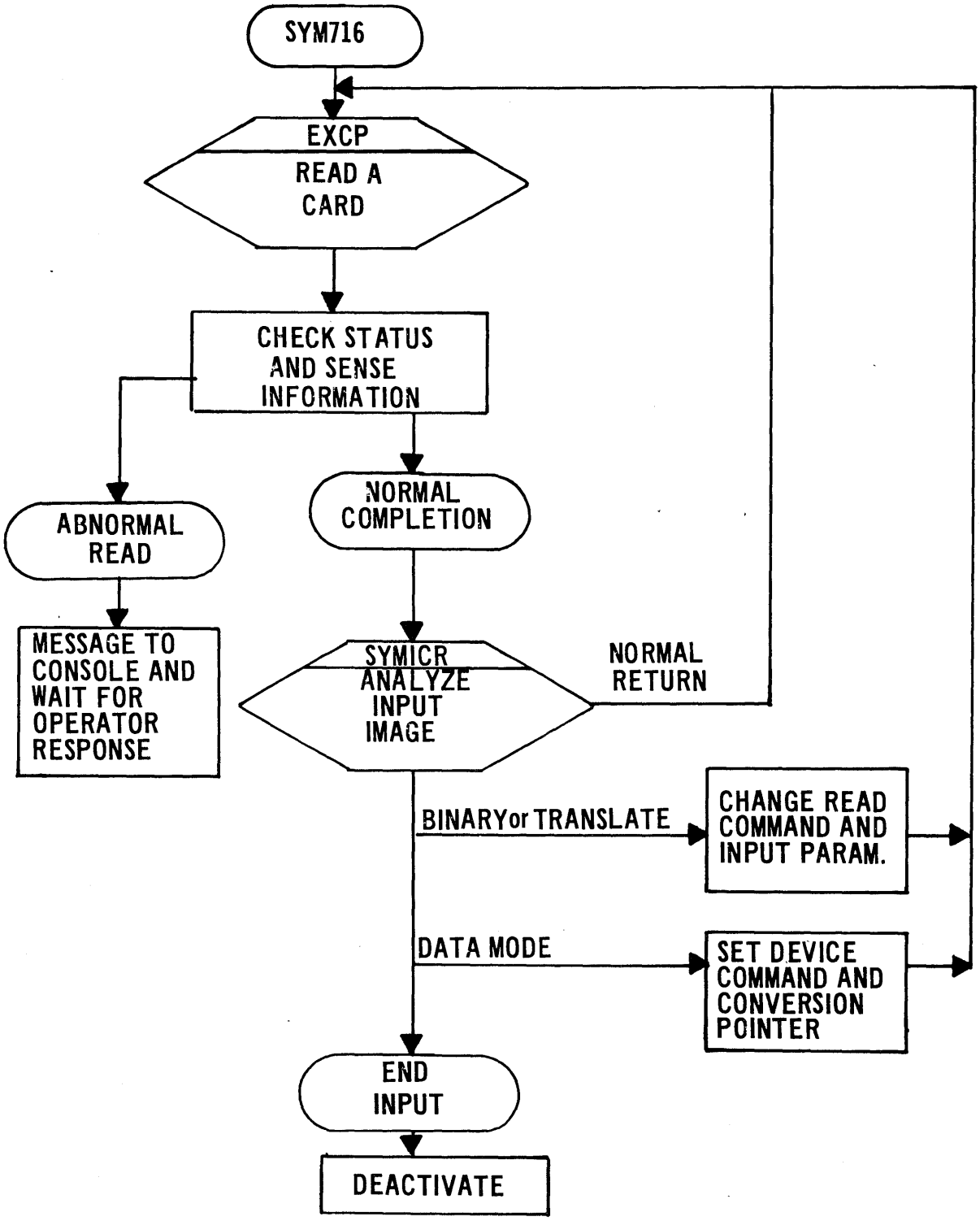


SCP ROUTINES

(716 Card Reader)

Issues the start I/O (SIO) via macro call 'EXCP (Execute Channel Program)' to read a card. Checks status sense information and takes appropriate action.

SCP ROUTINES

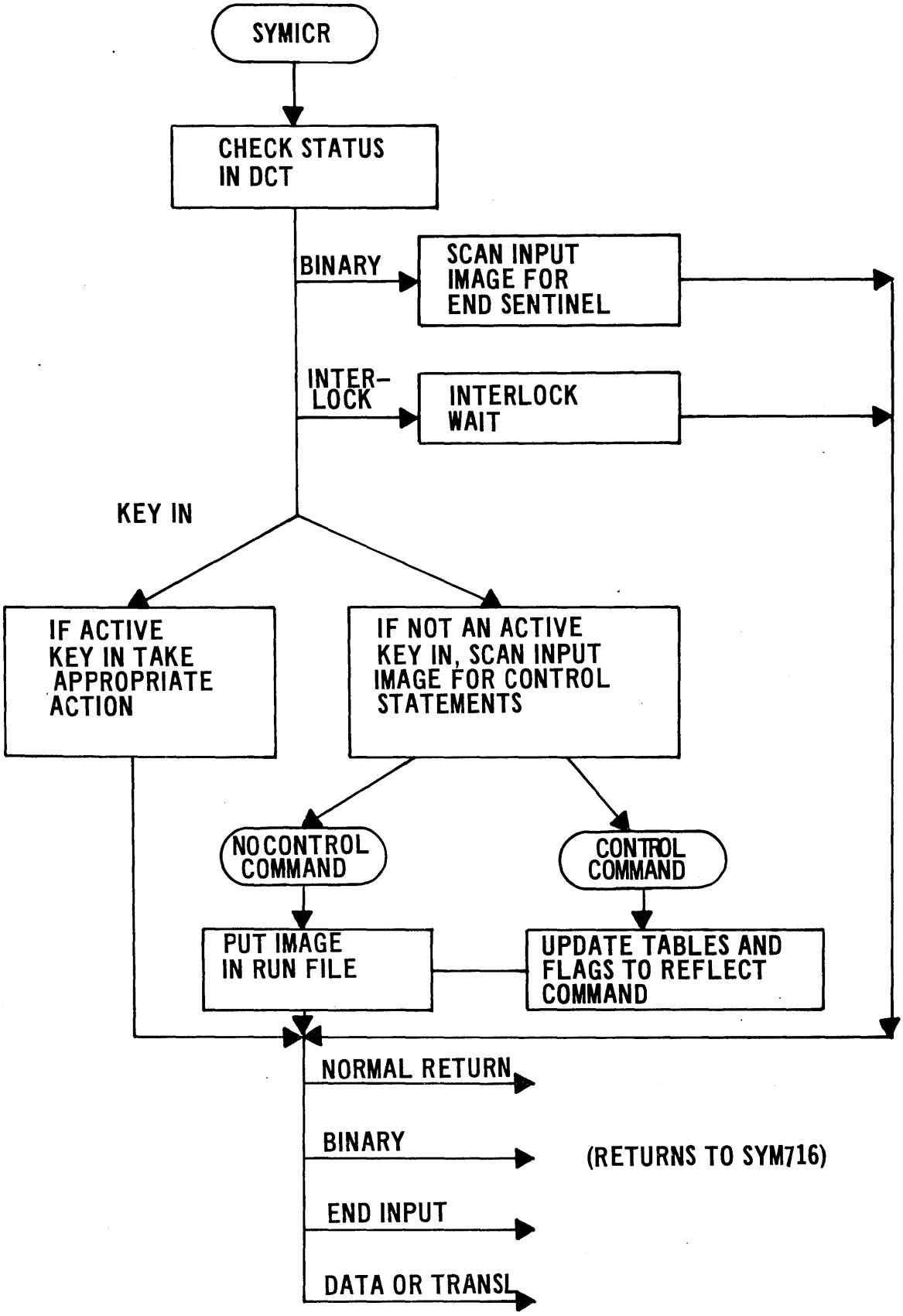


SCP ROUTINES

(Input Control Routine)

Called from the 716 card reader routine to analyize the input image. SYMICR takes the appropriate action after checking the status in the Device Control Table (DCT) and returns to the card reader routine.

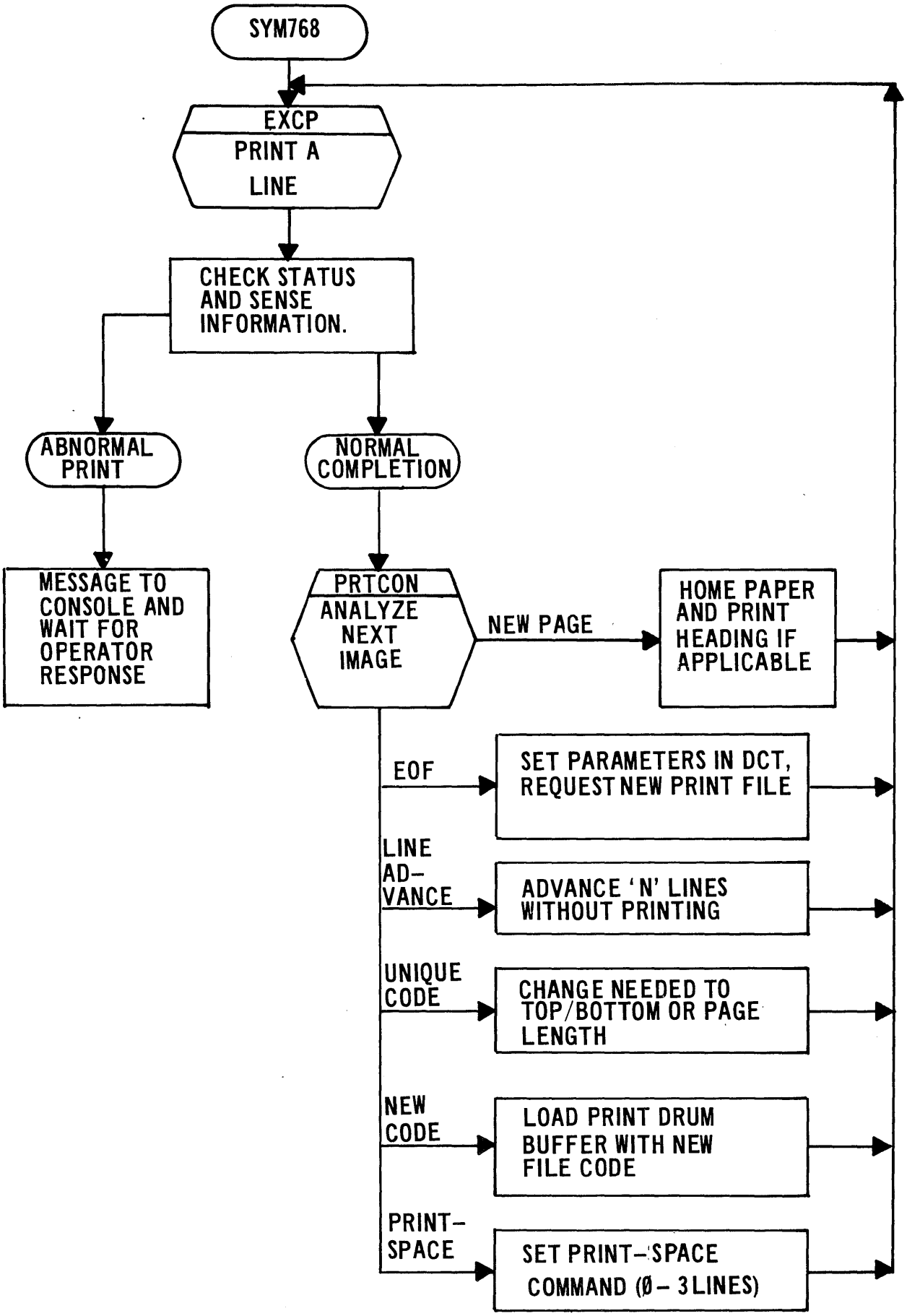
SCP ROUTINES



SCP ROUTINES

(768 Printer)

Issues the start I/O (SIO) via macro call 'EXCP (Execute Channel Program)' to print a line. Checks status, sense information and takes appropriate action.


SYM768
EXCP
PRINT A LINE
CHECK STATUS AND SENSE INFORMATION.
ABNORMAL PRINT
NORMAL COMPLETION
MESSAGE TO CONSOLE AND WAIT FOR OPERATOR RESPONSE
PRTCON
ANALYZE NEXT IMAGE
NEW PAGE
HOME PAPER AND PRINT HEADING IF APPLICABLE
EOF
SET PARAMETERS IN DCT, REQUEST NEW PRINT FILE
LINE AD-VANCE
ADVANCE 'N' LINES WITHOUT PRINTING
UNIQUE CODE
CHANGE NEEDED TO TOP/BOTTOM OR PAGE LENGTH
NEW CODE
LOAD PRINT DRUM BUFFER WITH NEW FILE CODE
PRINT-SPACE
SET PRINT-SPACE COMMAND (Ø – 3 LINES)

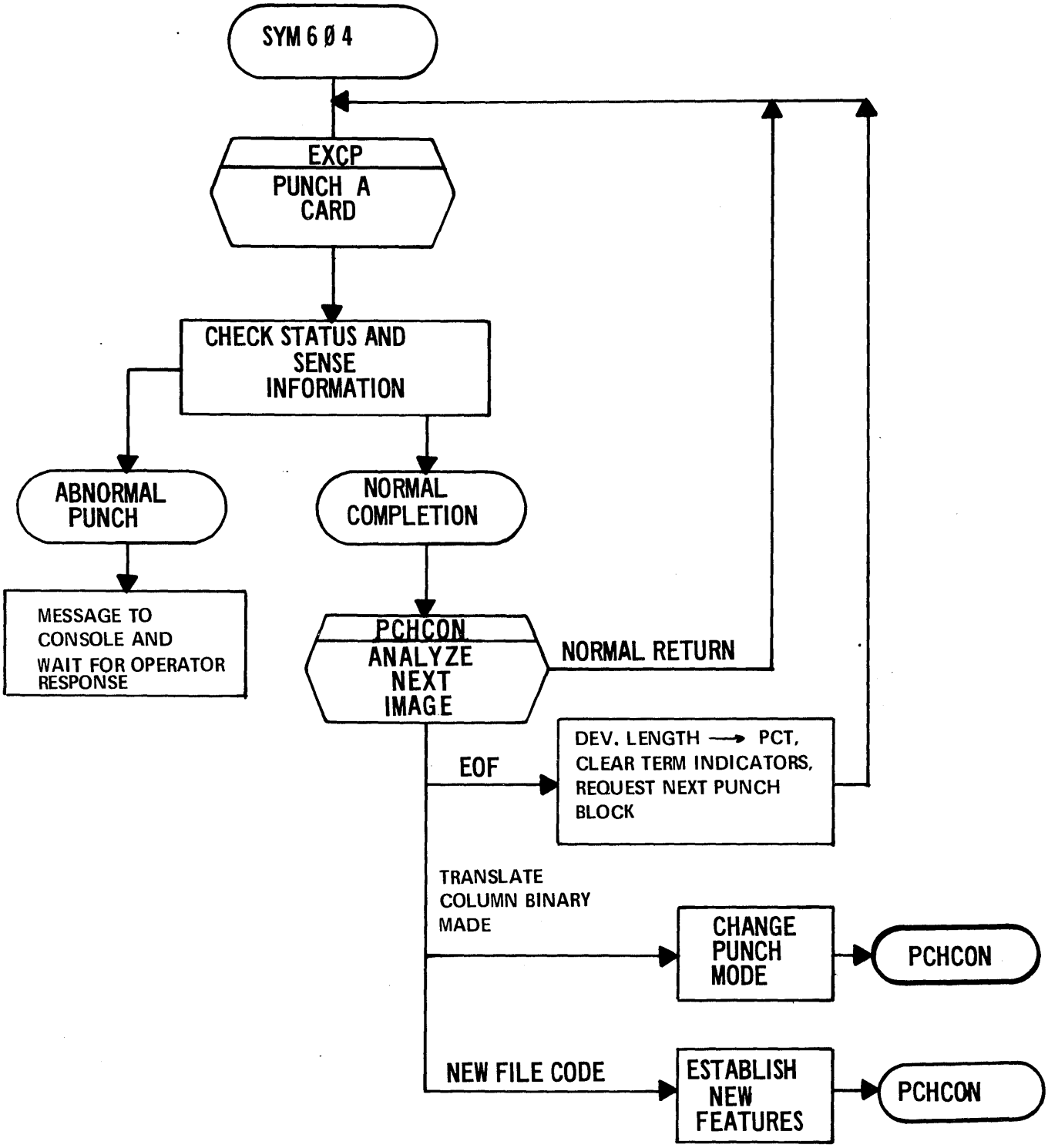
SCP ROUTINES

(604 Card Punch)

Issues the start I/O (SIO) via macro call "EXCP (Execute Channel Program)' to punch a card. Checks status, sense information and takes appropriate action.

SCP ROUTINES

PUNCH HANDLING - GENERAL FLOW



SCP - HOST INTERFACE

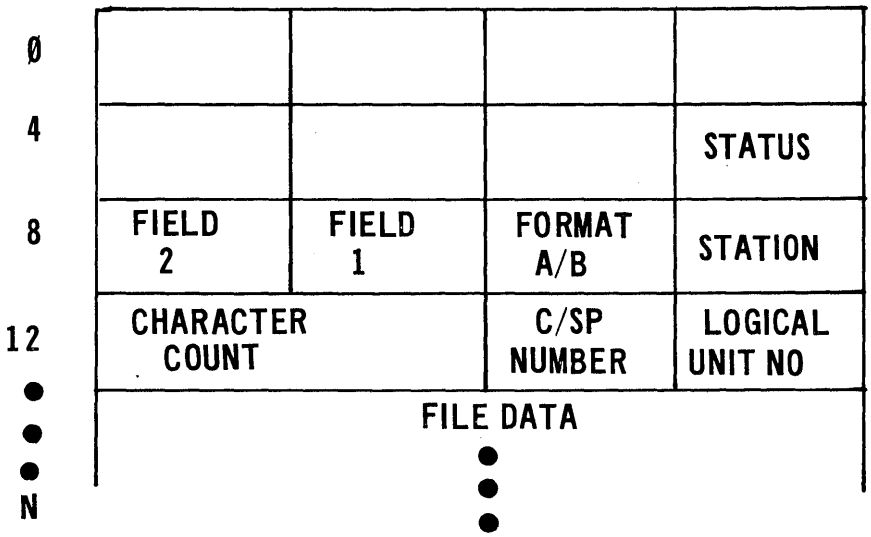
The SCP and the Host system communicate via symbiont information blocks which are classified as data blocks and control blocks. Data blocks are used for the transfer of data and control blocks are used for computer cross talk.

DATA BLOCKS - A run/output file is constructed in units of 224 Host words. The transmission of file data is in whole or partial units (N X 28 Host words, where N = 1,2, or 4). The transfer size is set at system generation time. Data blocks are transferred in ICA Format-A or Format-B.

CONTROL BLOCKS - Transferred between the Host and SCP to coordinate run/output file processing. The control blocks are transferred in ICA Format-B.

SCP - HOST INTERFACE

DATA BLOCK



FIELD 2 – DATA TYPE

FIELD 1 – BLOCK TYPE (DATA)

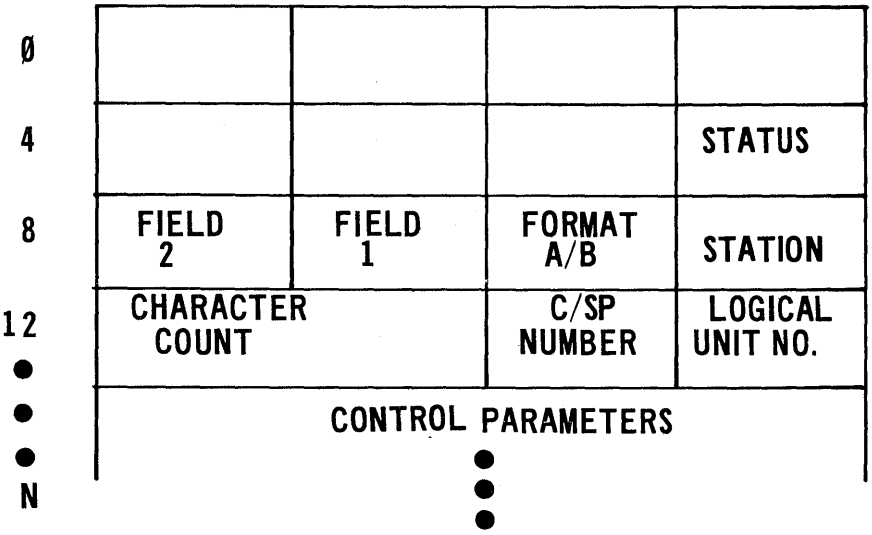
CHAR. COUNT – NO. OF BYTES

C/SP NO. – C/SP SUBSYT. I.D.

L.U. N.. – DEVICE SUBSYST. I.D.

FILE – DATA (F.D.OR ASCII)

CONTROL BLOCK



FIELD 1 – BLOCK TYPE (CONTROL)

PARAMETERS – (FIELDATA)

SECTION VIII

HOST HANDLER

SECTION VIII INDEX - HOST HANDLER

HOST HANDLER

Resident within C/SP and controls all commands between C/SP and host system. Accessed through a read or write command from the supervisor, system MCP, or user control programs (same basic format of all I/O commands). Through setting of options in the I/O command, special action may be initiated; output to host console, load program to C/SP from host storage, initiate program load, and read or write to the host mass storage device.

HOST HANDLER

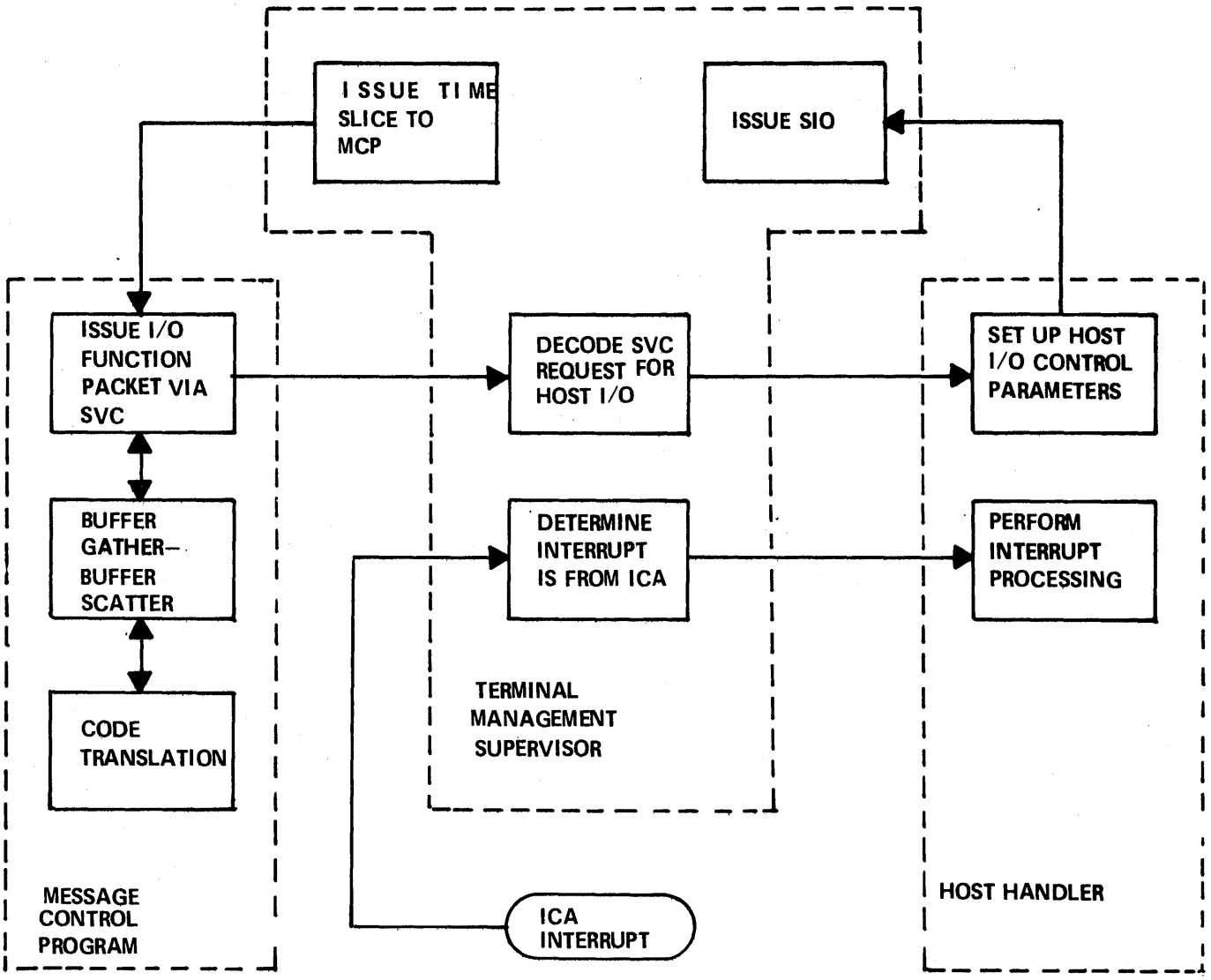
1. CONTROLS ALL COMMANDS BETWEEN C/SP AND HOST
   - A. COMMUNICATE WITH HOST CONSOLE
   - B. LOAD PROGRAM TO C/SP
   - C. TRANSFER DATA
   - D. READ/WRITE TO HOST MASS STORAGE
   - E. IDENTIFY FORMAT OF MESSAGE

HOST HANDLER

(I/O FLOW)

The Host Handler (H.H.) is entered as a result of the execution of a SVC instruction. The main function of the HH is to set up Host I/O control funct ons and queue all I/O buffers according to priority. The Host Handler maintains the I/O queues and performs the interrupt processing of ICA interrupts.

HOST HANDLER



HOST I/O CONTROL FLOW

HOST HANDLER

(TABLE, QUEUES AND SUPERVISOR CALLS)

Each one has an entry for each key.

HOTAPN - Output completion appendage address queue.
HINAPN - Input completion appendage address queue.
HOTSUC - Output start of queue chain.
HCTINQ - Control information input queue.
HRTINQ - Real time data input queue.
HSMINQ - Symbiont data input queue.
HIOSRQ - I/O service request queue.

There are four SVC instructions that call the Host Handler. Each passes an address to be validated.

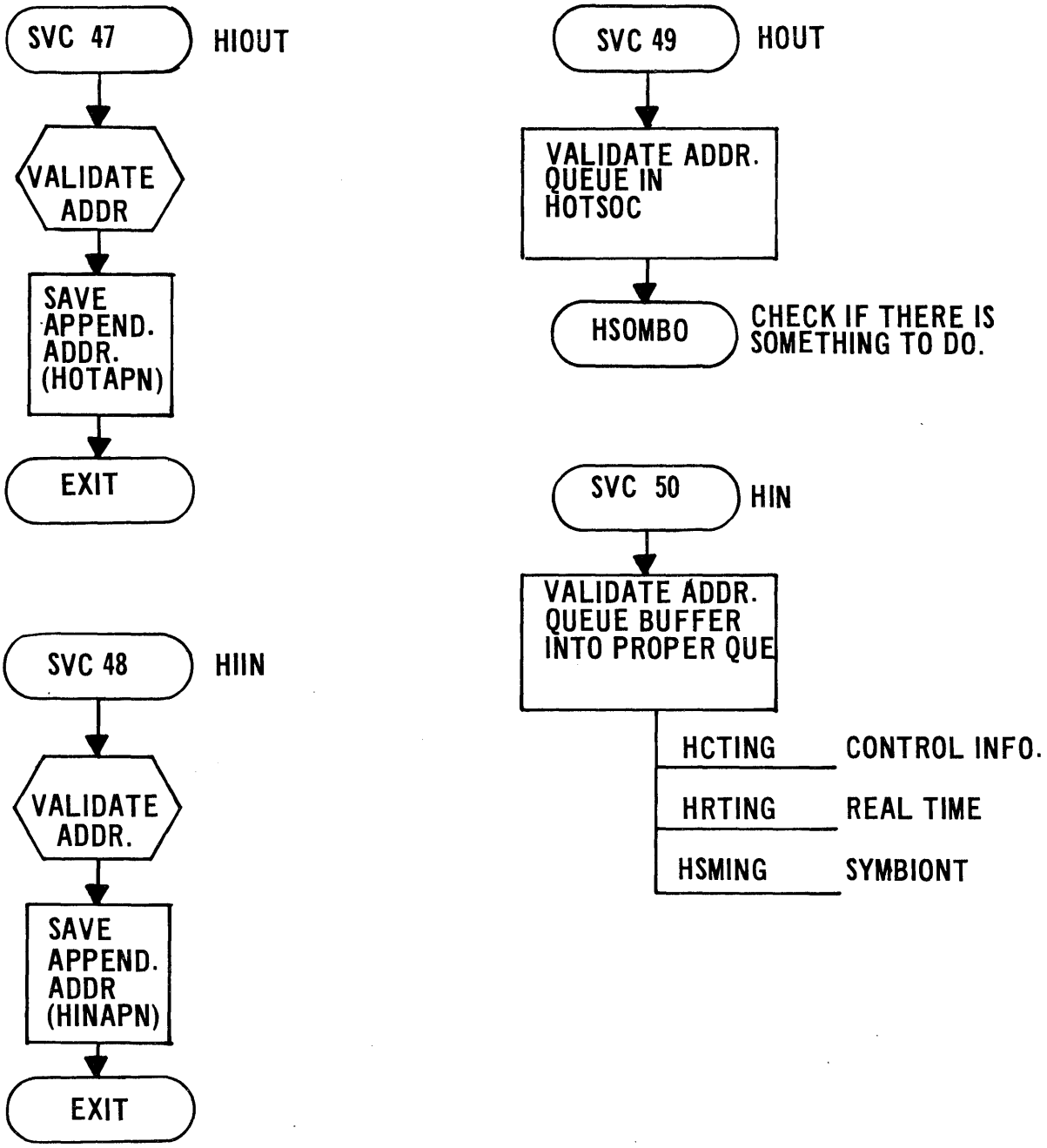
HIOUT (SVC 47) - Initialize output completion appendage address. Stores the address in HOTAPN.

HIIN (SVC 48) - Initialize input completion appendage address. Stores the address in HINAPN.

HOUT (SVC 49) - Send output to C/SP.

HIN (SVC 50) - Send input to host.

HOST HANDLER
(SUPERVISOR CALL INSTRUCTIONS)



HOST HANDLER

(TABLES)

The Host Handler maintains these tables by program key:

HOTSOC - Start of output chain
HRTINQ - First real time input buffer
HCTINQ - First control input buffer
HSMINQ - First non-real time buffer
HSDINQ - First buffer of SDF
HOTAPN - Output appendage address
HINAPN - Input appendage address
HIOSRQ - Address of I/O Service Requests
HSTCSP - PUB associated with Host logical unit number 1, etc.

HOST HANDLER (TABLES)

HOTSOC

ADDR.	FIRST	OUT – BUF.	KEY Ø
ADDR.	FIRST	OUT – BUF.	KEY 1
		⋮	

HRTINQ

ADDR.	FIRST	R. T.	IN – BUF.	KEY Ø
ADDR.	FIRST	R. T.	IN – BUF.	KEY 1
			⋮	

HCTINQ

ADDR.	FIRST	CONTROL	IN – BUF.	KEY Ø
ADDR.	FIRST	CONTROL	IN – BUF	KEY 1
			⋮	

HSMINQ

ADDR.	FIRST	NON – R.T.	IN – BUF.	KEY Ø
ADDR.	FIRST	NON – R.T.	IN – BUF	KEY 1
			⋮	

HSDINQ

ADDR.	FIRST	BUF.	IN	CHAIN	OF	SDF	KEY Ø
ADDR.	FIRST	BUF.	IN	CHAIN	OF	SDF	KEY 1
				⋮			

HOTAPN

OUTPUT	APPENDAGE	ADDR.	KEY Ø
PUTPUT	APPENDAGE	ADDR.	KEY 1
		⋮	

HINAPN

INPUT	APPENDAGE	ADDR.	KEY Ø
INPUT	APPENDAGE	ADDR.	KEY 1
		⋮	

HIOSRQ

ADDR.	I/O SERVICE REQUEST	KEY 1
	⋮	

HSTCSP

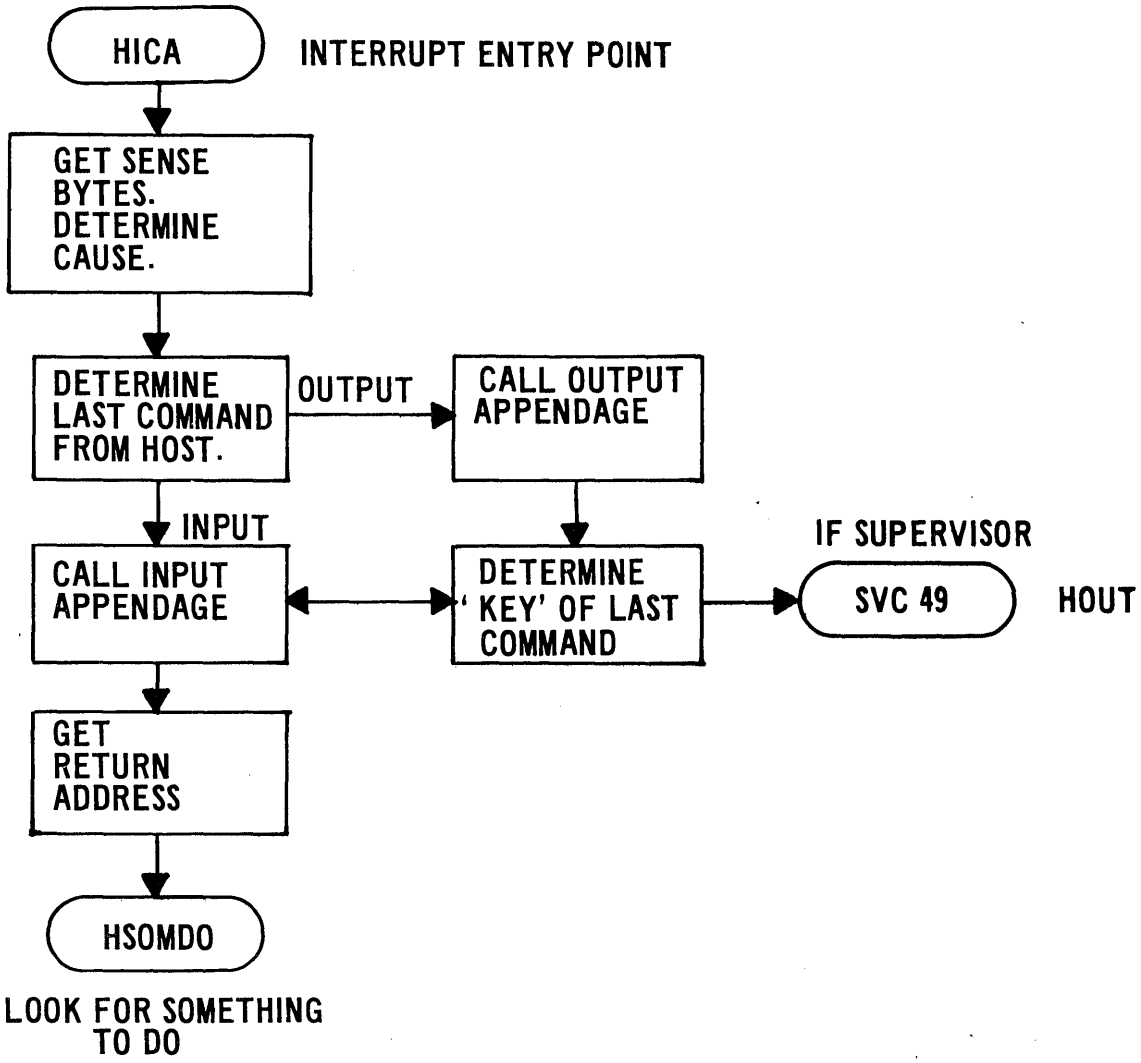
ADDR.	PUB	ASSOCIATED	HOST	L.U.N. 1
ADDR.	PUB	ASSOCIATED	HOST	L.U.N. 2
		⋮		

HOST HANDLER

(INTERRUPTS)

The ICA interrupt handling routine is called HICA. HICA makes use of the interrupt status halfword for the I/OB class of interrupt. An SIO instruction must be used to obtain the sense bytes. The TIO function is used to determine the condition of the ICA and also clears the 'device end' of the sense information.

HOST HANDLER (INTERRUPTS)



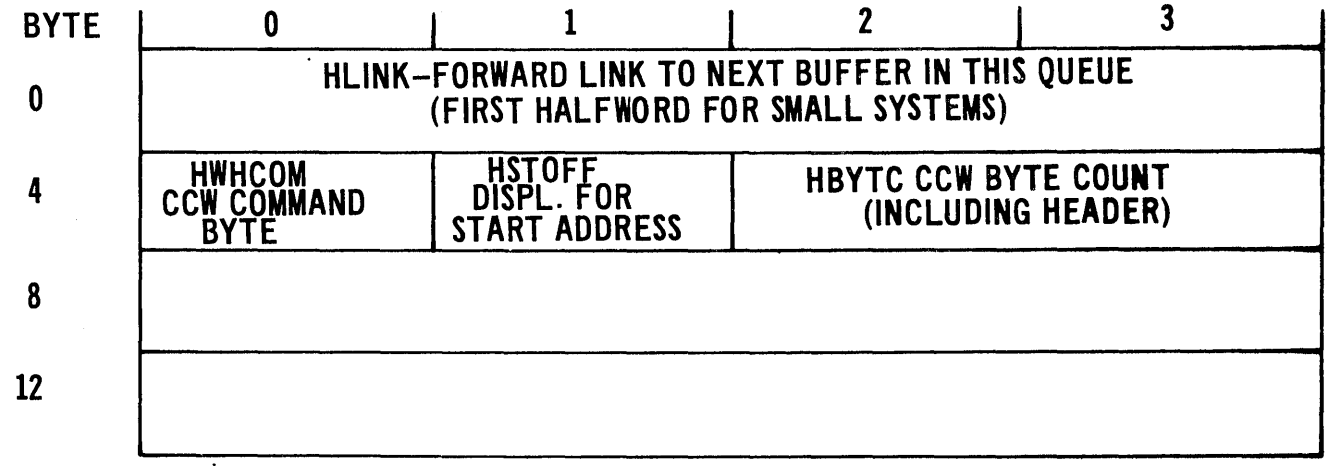
HOST HANDLER

(Common Buffers)

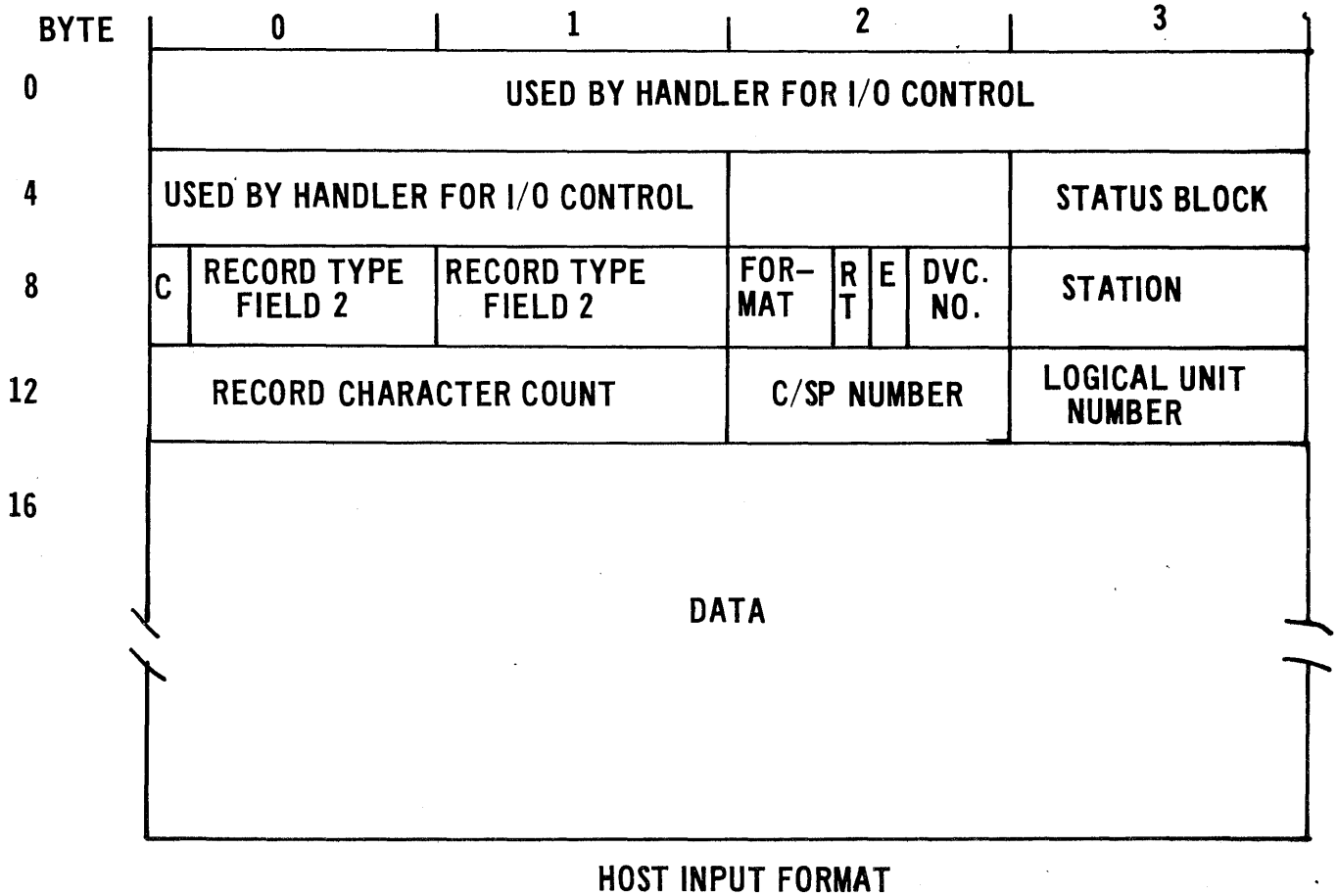
A common buffer format is used within the C/SP to pass information to/from the Host Handler (HH). The first two words are used by the Host Handler as a means of linking to more buffers in this particular chain, byte count of the buffer, etc. The next two words indicate such things as to the format (whether format A,B, or C), real time buffer, etc. These first four words are the header for the Host buffers.

HOST HANDLER

(COMMON BUFFERS)



FIELDS USED BY H.H.



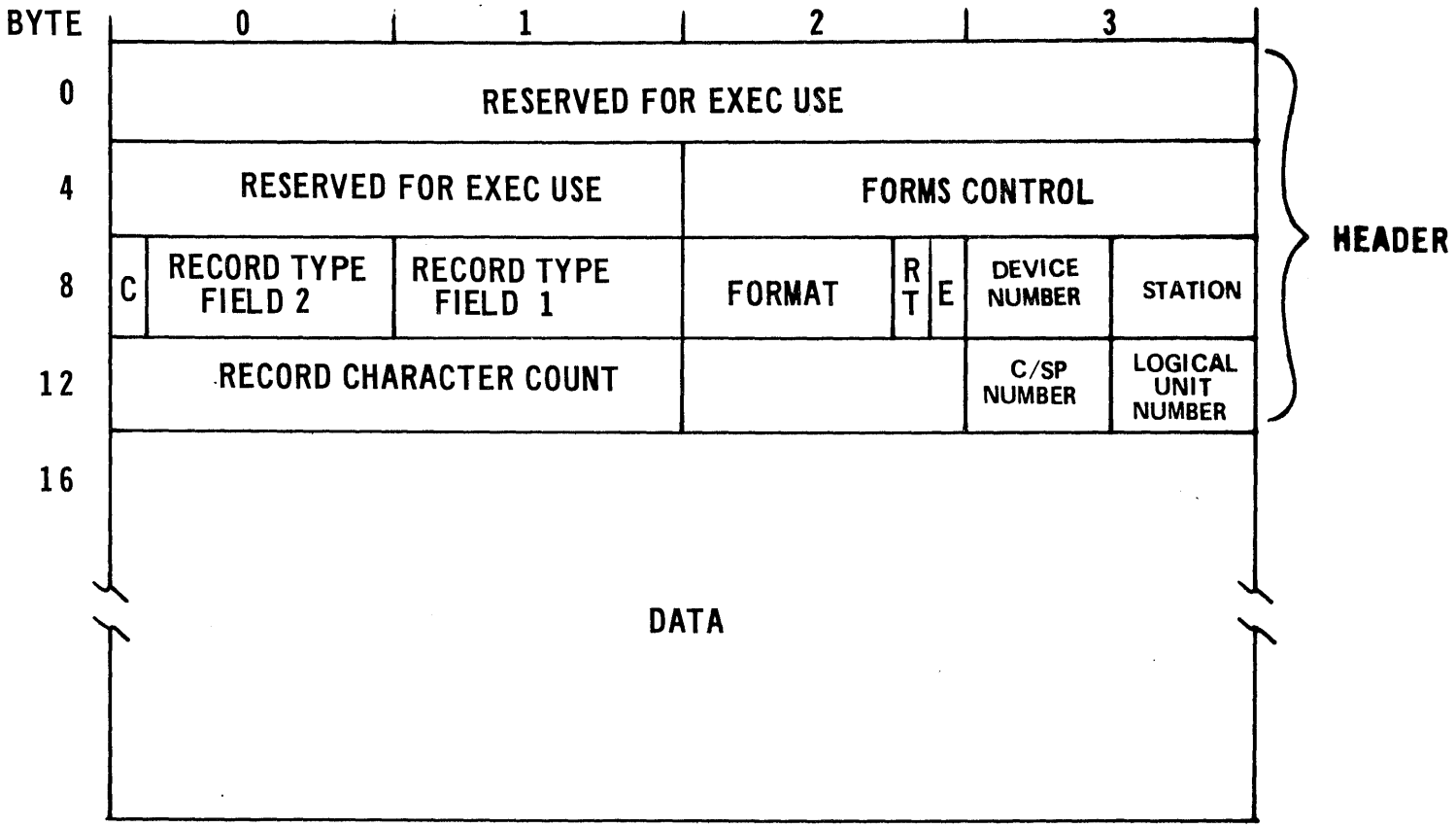
HOST INPUT FORMAT

HOST HANDLER

(Common Buffers)

Reference the Technical Document for the Host Handler to determine the detail of each field of the header for the Host buffers.

HOST HANDLER (COMMON BUFFERS)



HOST OUTPUT COMPLETION FORMAT

HOST HANDLER

(Transmission Buffers)

The Host Handler contains routines to reformat the common buffers to transmission buffers for transmission to the Host computer according to Format A, B, or C.

The transmission buffers received from the Host computer are also reformatted to common buffers.

HOST HANDLER
(TRANSMISSION BUFFERS)

BYTE	0				1	2	3
0	NOT TRANSMITTED						
4	NOT TRANSMITTED						TRANSMITTED IN FORMAT C ONLY BUT NOT USED
8	FF	R	E	FORMS CONTROL	RECORD TYPE FIELD 2	RECORD TYPE FIELD 1	DEVICE NUMBER
12	STATION				CHARACTER OF WORD (SYMBIONTS) COUNT	C/SP NUMBER	LOGICAL UNIT NUMBER

FORMAT A AND C

BYTE	0							1		2			3	
0	NOT TRANSMITTED													
4	XX	NOT USED		XX	NN	FF	R	N	XX	FORMS CONTROL			XX	FORMS CONTROL
8	XX	RECORD TYPE FIELD 2		XX	RECORD TYPE FIELD 1				XX	DEVICE NUMBER	STATION		XX	STATION
12	XX	CHARACTER OR WORD COUNT		XX	CHARACTER OR WORD COUNT				XX	C/SP NO.	LOG. UNIT NO.		XX	LOGICAL UNIT NUMBER

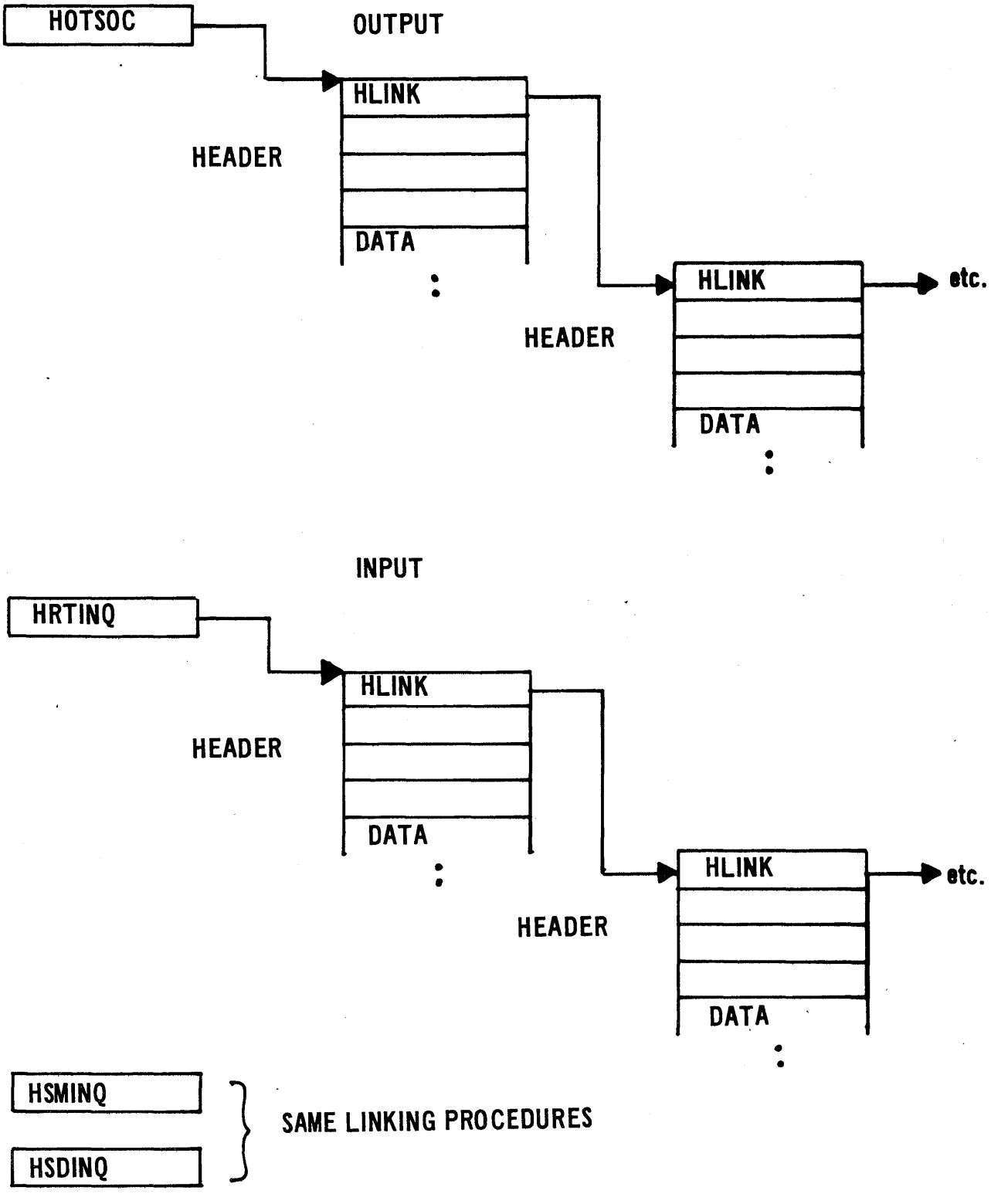
FORMAT B

HOST HANDLER

(Buffer Linking)

The first word of each buffer header is used for linking all the buffers in this particular chain.

HOST HANDLER

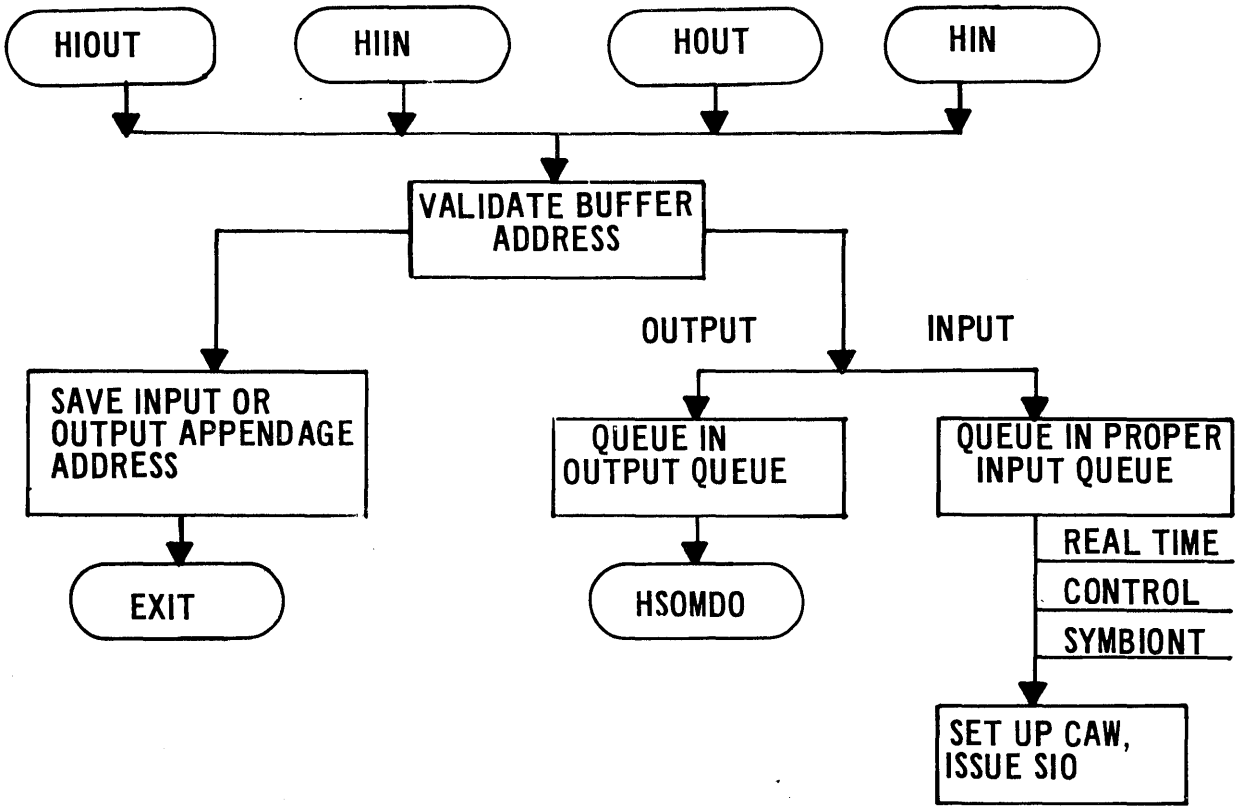


HOST HANDLER

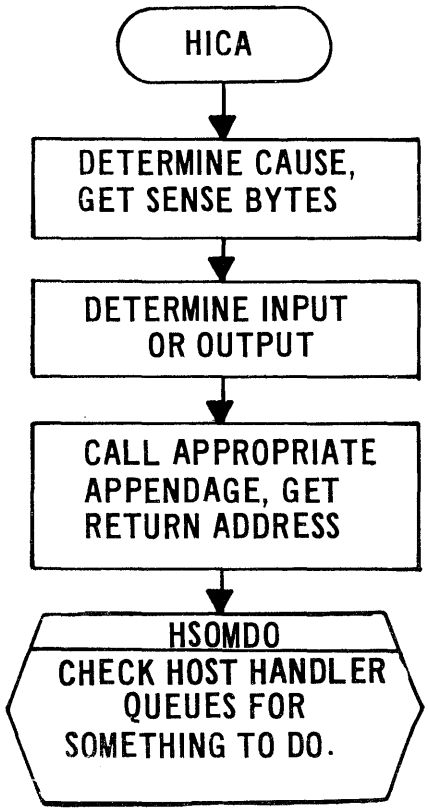
(General Flow)

There are two entries to the Host Handler, function (SVC interrupt) and hardware interrupt. The Host Handler maintains its own queues, issues the SIO instruction, and processes the interrupts.

HOST HANDLER
FUNCTION ENTRY (SVC)



HARDWARE INTERRUPT ENTRY



SECTION IX

C/SP - 1100 INTERFACE

SECTION IX INDEX - C/SP 1100 INTERFACE

C/SP - 1100 INTERFACE

The 1100 Interface to the C/SP is designed to provide the user an interface with a variety of remote devices to utilize the advantages of the C/SP. A complete re-education of the user is not necessary as the interface causes as little impact on the Host executive as possible.

Modification and Additions:

1. Host Elements - modification of certain elements is needed to facilitate the Host - C/SP interface.

2. Symbionts - additional symbionts are needed to handle remote run submission from devices on the C/SP and interface to the 1100 EXEC.

3. Symbiont and Real-time code removal - code which is not used by the C/SP interface when symbionts and a CTMC are not in the system.

4. C/SP Handler - needed to communicate with the C/SP through the Intercomputer Adapter (ICA).

5. Host Handler - within the C/SP and communicates with the Host (1100 computer) through the ICA.

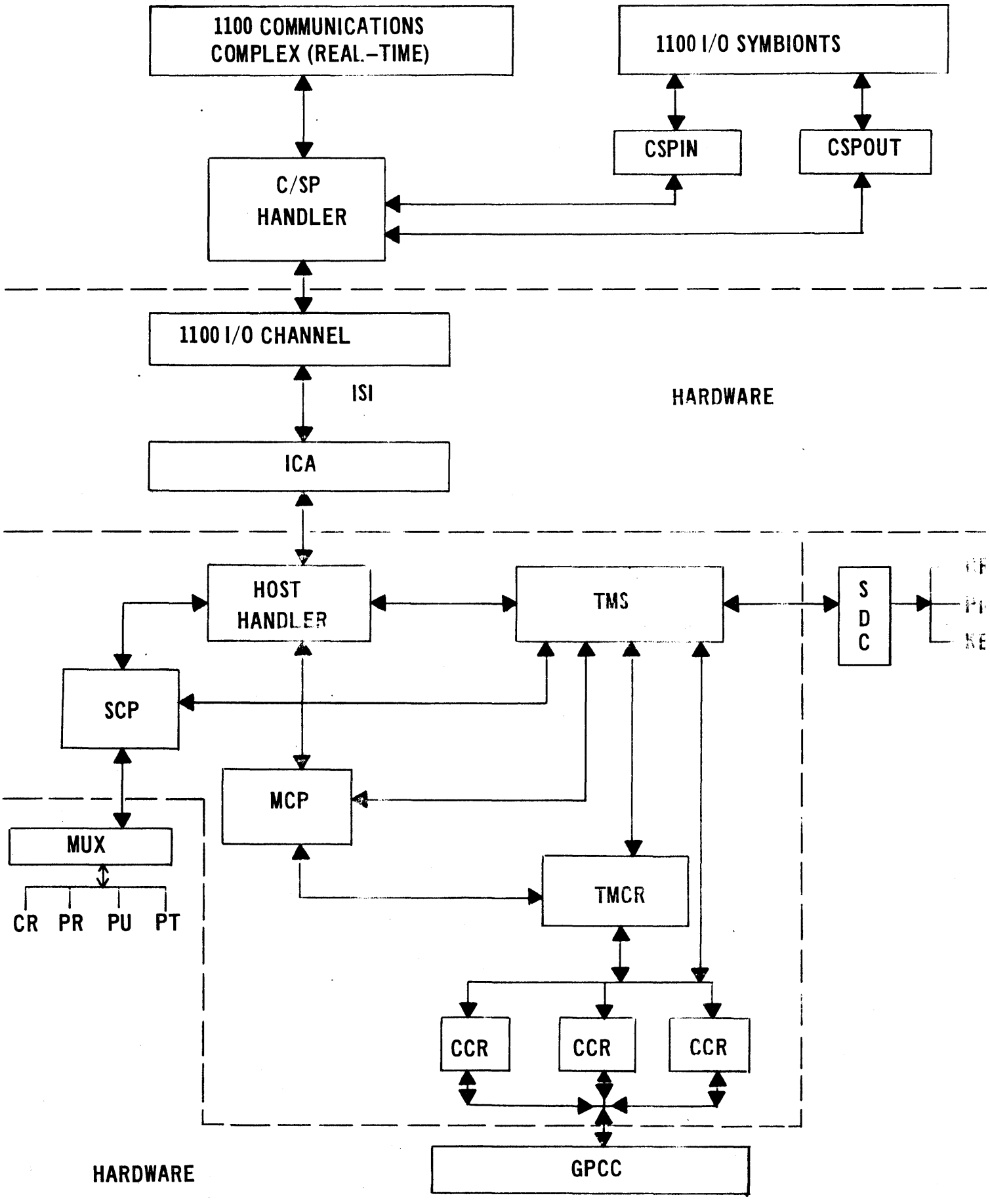
C/SP - 1100 INTERFACE

1. HOST ELEMENT MODIFICATION
2. SYMBIONTS
3. SYMBIONT AND REAL-TIME CODE REMOVAL
4. C/SP HANDLER
5. HOST HANDLER

1100 - C/SP BLOCK OVERVIEW

The 1100 software, to support a C/SP, consists of a C/SP Handler and C/SP Symbionts. A reduction of the 1100 software results from the communications/device handlers now residing in the C/SP. The C/SP Handler consists of a number of small routines to interface with the 1100 communications real-time, demand, and batch symbionts by providing a common format for the 1100 - C/SP interface. The C/SP symbionts are designed to interface with existing 1100 I/O symbionts presently in the 1100 EXEC.

1100 - C/SP BLOCK OVERVIEW



1100 - C/SP ELEMENTS (Additional)

CDISP - C/SP QCIO Dispatcher. Processes the QCIO buffers supplied by IH and COME for dispatching.

COINT - Initialize C/SP for Real-time, in conjunction with COMS.

CQCELL - Queue cell management for expool, cell and main header block.

CSPCOM - Console communications to handle unsolicited keyins.

CSPESI - Real-time ESI processing called from COME.

CSPIH - Process al C/SP interrupts.

CSPIN - Resident part of input symbiont.

CSPINR - Non-resident part of input symbiont.

CSPIO - Queue and process I/O requests.

CSPMSS - Mass Storage symbiont to handle program load to C/SP.

CSPOUT - Handle all output to C/SP except real-time.

CSPREC - Recovery, termination and clean up.

CSPSON - Handle sign-on, sign-off.

CSPSSS - SCP support for; EOF, I/O errors for SCP output.

CSPUTL - Utility routines as panic dump, error logging.

1100 - C/SP ELEMENTS (ADDITIONAL)

CDISP	C/SP DISPATCHER
COMINT	C/SP INITIALIZATION (REAL TIME)
CQCELL	C/SP QUEUE CELL MANAGEMENT
CSPCOM	C/SP CONSOLE COMMUNICATIONS
CSPESI	C/SP REAL-TIME ESI PROCESSING
CSPIH	C/SP INTERRUPT HANDLING
CSPIN	C/SP INPUT SYMBIONT (RESIDENT)
CSPINR	C/SP INPUT SYMBIONT (NON-RESIDENT)
CSPIO	C/SP I/O REQUEST PROCESSING
CSPMSS	C/SP MASS STORAGE SYMBIONT
CSPOUT	C/SP OUTPUT SYMBIONT
CSPREC	C/SP RECOVERY
CSPSON	C/SP SIGN-ON
CSPSSS	C/SP SCP SUPPORT
CSPUTL	C/SP UTILITY ROUTINES

SYSTEM TABLES

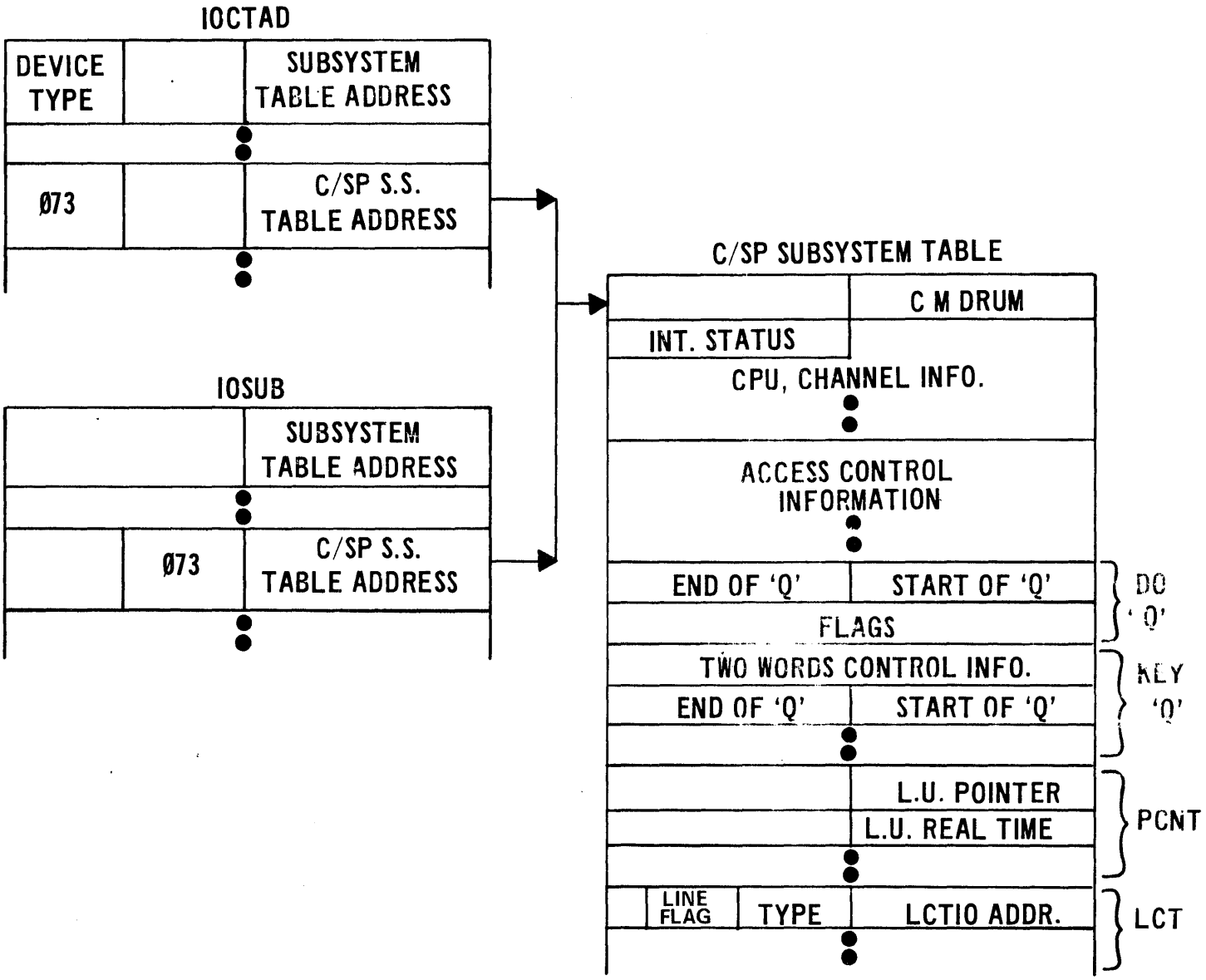
System tables are generated at system generation time describing to the software the information needed to control the environment. The tables described are contained in the 1100 EXEC element IOD.

IOCTAD - Reflects the type of equipment for the particular processor and channel on that processor. The length of the table is equal to 16 times the number of processors (24 times the number of processors for the 1110). IOCTAD is used to obtain the address of the appropriate tables and information when an interrupt occurs on a C/SP channel.

IOSUB - Reflects the subsystems by number. Used by 1100 EXEC to access the information contained in the subsystem table.

SUBSYSTEM TABLE - Holds the various lists and chains that control the environment of the subsystem. In this case, the C/SP. The subsystem table is located through IOCTAD or IOSUB.

SYSTEM TABLES



C/SP USER INTERFACE
(REAL - TIME)

Line Terminal Table (LTT) - must be constructed by the user in the user-program data bank. The LTT enables the user to control each line terminal group. The LTT is the same as the table used with the CTMC operations and is divided into three parts: Output, Input, and DIAL.

CHNAME - Up to twelve character file name used to identify the line terminal group.

CHOSTA - Output - Status

CHXUSE - Used by 1100 EXEC

CHOUSE - Output usage

CHOACT - Output completion activity address

CHOSEN - Output character count

CHOPOL - Output buffer or pool - I.D.

CHOQND - End of output queue of buffers filled

CHOQST - Start of output queue of buffers filled

CHOPAR - Partial buffer character count

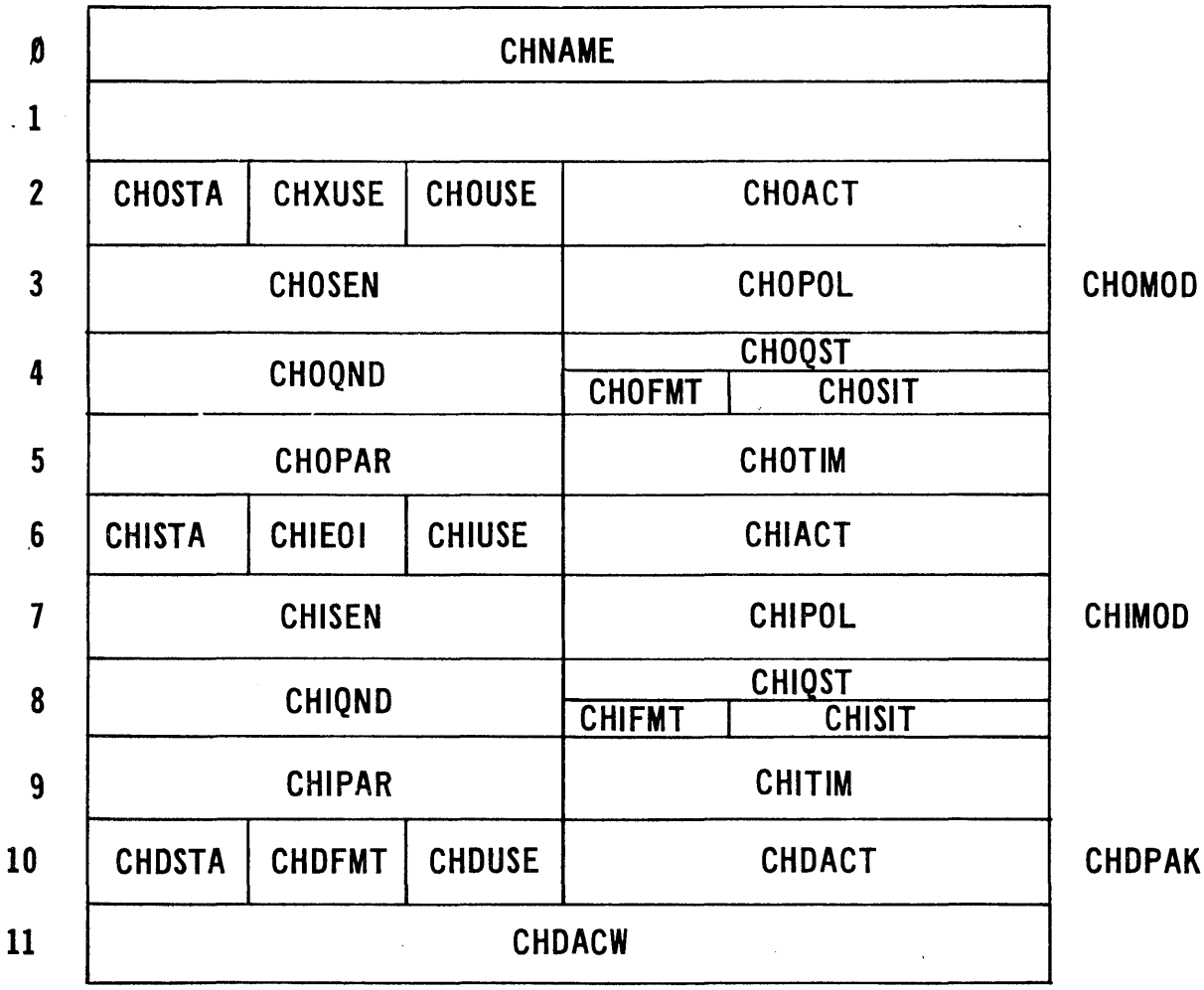
CHOTIM - Buffer completion count

Words 6 - 9 are used for input and are similar to the definitions for output.

Words 10 - 11 are used for DIAL.

Reference UP-7917 (C/SP Operating System - 1100 Series Supplement) for a more detailed explanation.

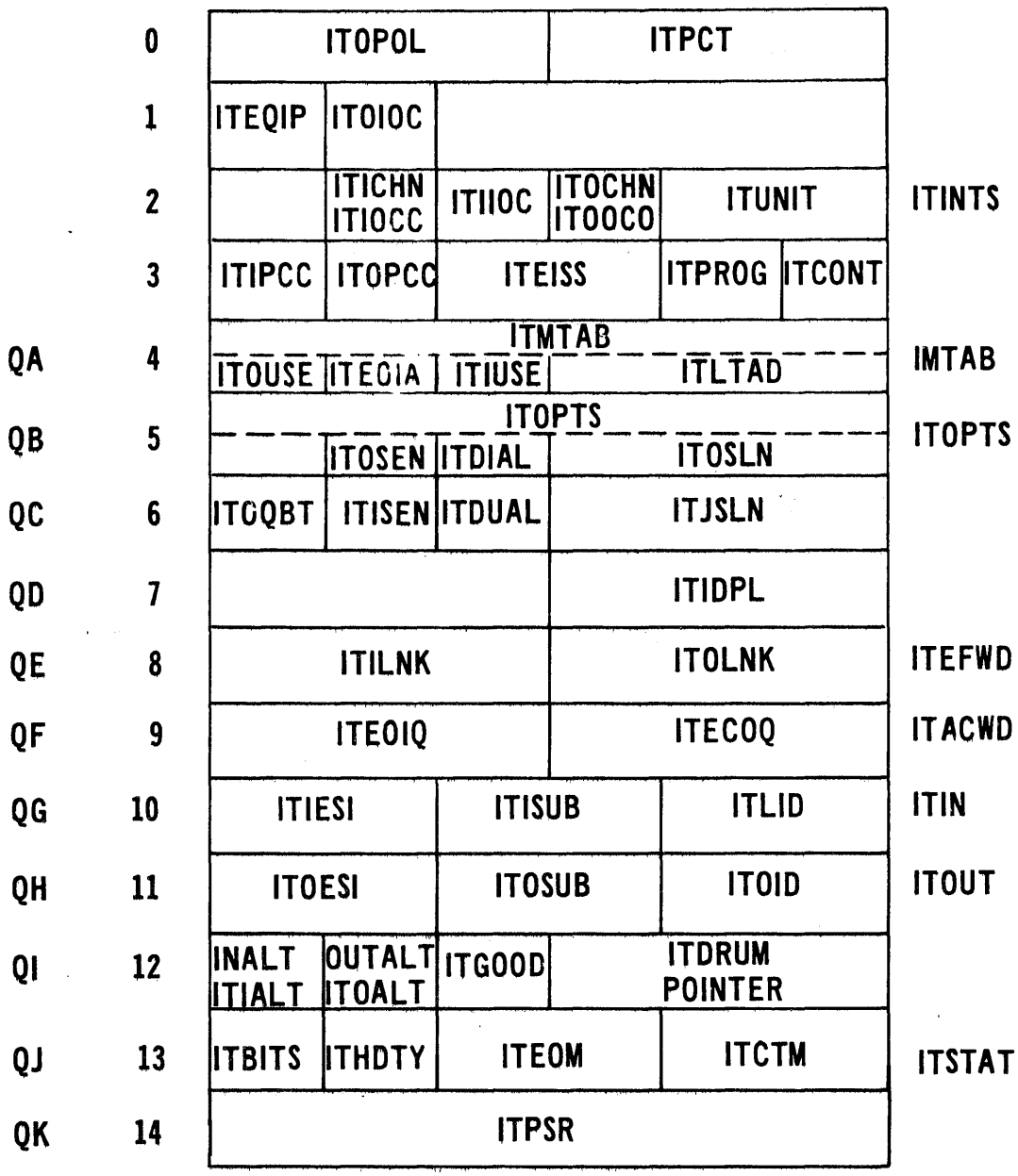
LINE TERMINAL TABLE (LTT)



C/SP USER INTERFACE
(REAL - TIME)

Internal Control Table (IT) - The IT table is formed by the host executive assignment routine upon submission of the user's request for a particular line assignment. It is built in the user's PCT and serves as the table to contain executive information necessary to control the usage of the assigned line. For a detailed explanation of the fields reference the Univac Technical Document for the C/SP - 1100 Interface.

INTERNAL TABLE (IT)



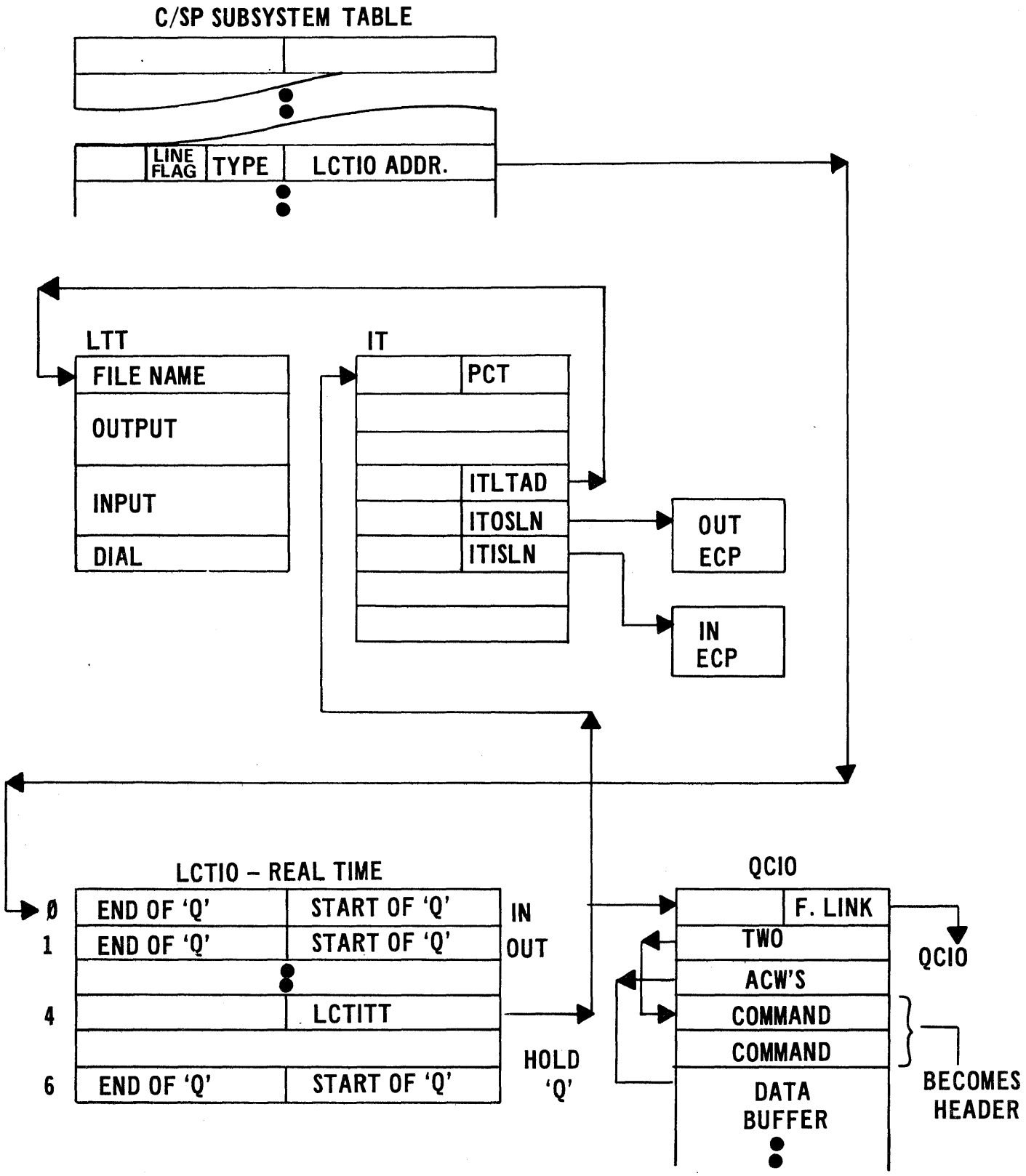
SYSTEM TABLES
(LINE CONTROL - REAL TIME)

The Line Control Table (LCT) portion of the subsystem table consists of one word entries for each line on the C/SP. Each word contains the basic status of the line and a pointer to the I/O control table (LCTIO).

The LCTIO is an extension of the IT table as the IT table did not have room for C/SP real-time handling information. Pointers to the beginning and end of input request queues, output request queues for the line. The 'Hold Queue' is used to insure that the C/SP does not receive multiple records for the same terminal.

The input/output queue cells (QCIO) are used to hold the information necessary to perform an I/O request and determine the appropriate action upon completion of the request.

SYSTEM TABLES
(LINE CONTROL - REAL TIME)



C/SP ER FUNCTIONS

All real-time functions on a line terminal group are performed through an LT table. The functions are identical in form and operation as those in the present 1100 Executive.

ER CMS$ - Initialize

ER CMT$ - Termination

ER CMD$ - Dial

ER CMI$ - Input

ER CMO$ - Output

ER CMH$ - Hang Up

ER Route$ - Route to New LTG

ER CPOOL$ - Establish Buffer Pool

ER CGET$ - Remove Buffers from Pool

ER CREL$ - Release one or all Pools

ER CADD$ - Return Buffers to Pool

ER CJOIN$ - Expand Pool

Each of these references are made with register A0 loaded with the address of the LT table defining the line terminal group.

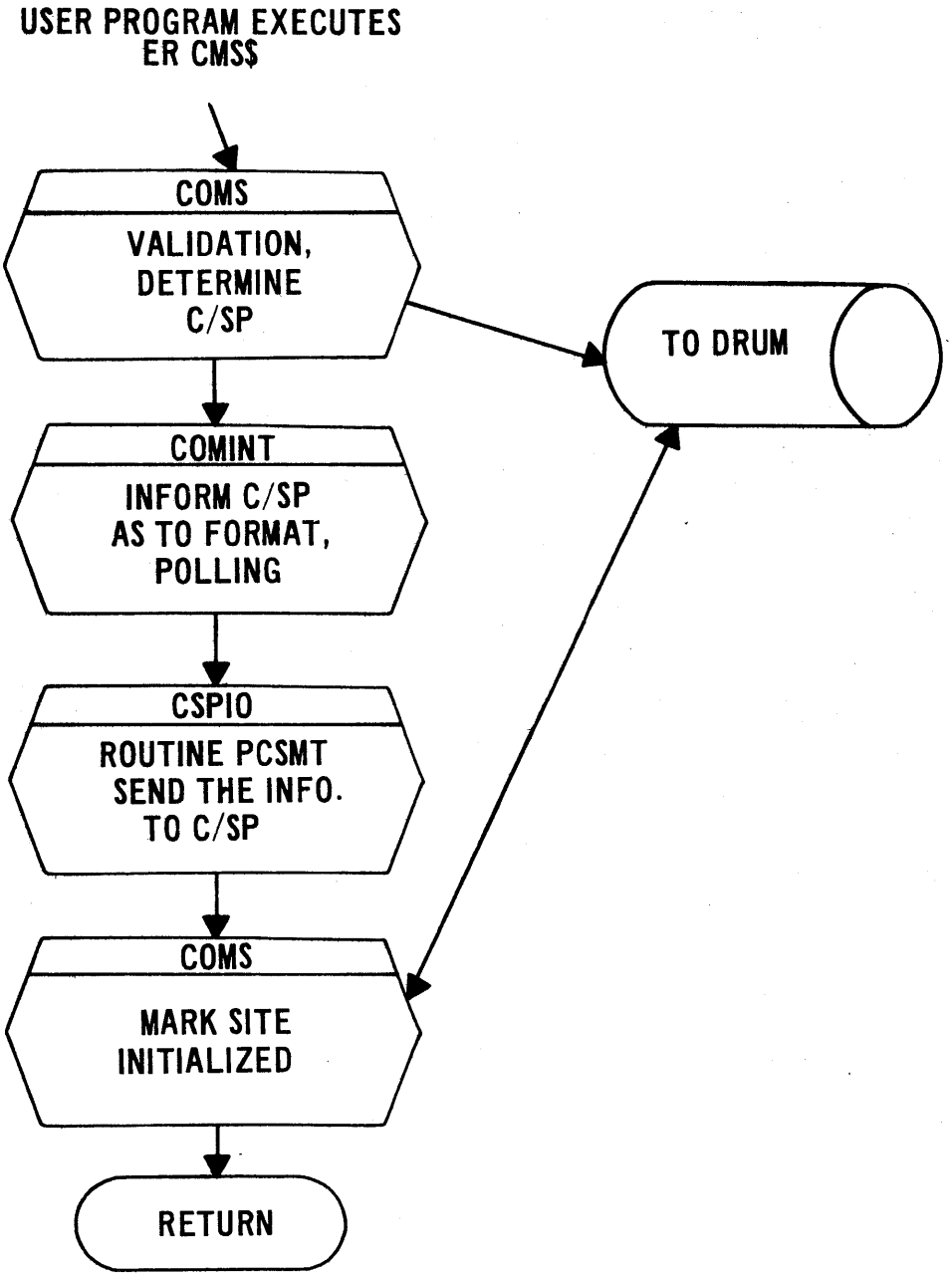
CISP ER FUNCTIONS

```
L       A0,(LT COUNT-1, LT ADDRESS)
ER      CMS$

L,u     A0, LT ADDRESS
ER      CMD$

L,u     A0, LT ADDRESS
ER      CMI$

L,u     A0, LT ADDRESS
ER      CMO$

L,u     A0, LT ADDRESS
ER      CMH$

L       A0,(LT COUNT-1, LT ADDRESS)
ER      CMT$

L       A0,(MODE, LT ADDRESS)
L,u     A1, POINTER TO NEW LTG
ER      ROUTE$

L,u     A0, PKT ADDRESS
ER      CPOOL$

L       A0, (NBR. BUFFER TO REMOVE, ADDRESS OF BUFFER)
ER      CGET$

L       A0, POOL-I.D.
ER      CREL$

L,u     A0, PKT ADDRESS
ER      CADD$

L,u     A0, PKT ADDRESS
ER      CJOIN$
```

LINE INITIALIZATION
(REAL - TIME)

Initialization is to set up the Internal Table (IT) with the general parameters required for I/O operations and inform the C/SP that the line is being initialized for real-time I/O. The 1100 Initialization routine (COMS) validates that the user has assigned the line and that the line is not currently initialized. COMS checks the LTT to determine whether input or output is to be initialized. When COMS determines the initialization is for the C/SP, control is given to COMINT (C/SP initialization routine) which builds an initialization buffer. Information such as ICA format, station information, the size of the input record desired on the 1100, if input is being initialized, is contained in the buffer. The buffer is output to the C/SP via CSPIO and control returns to COMS to mark the site initialized.

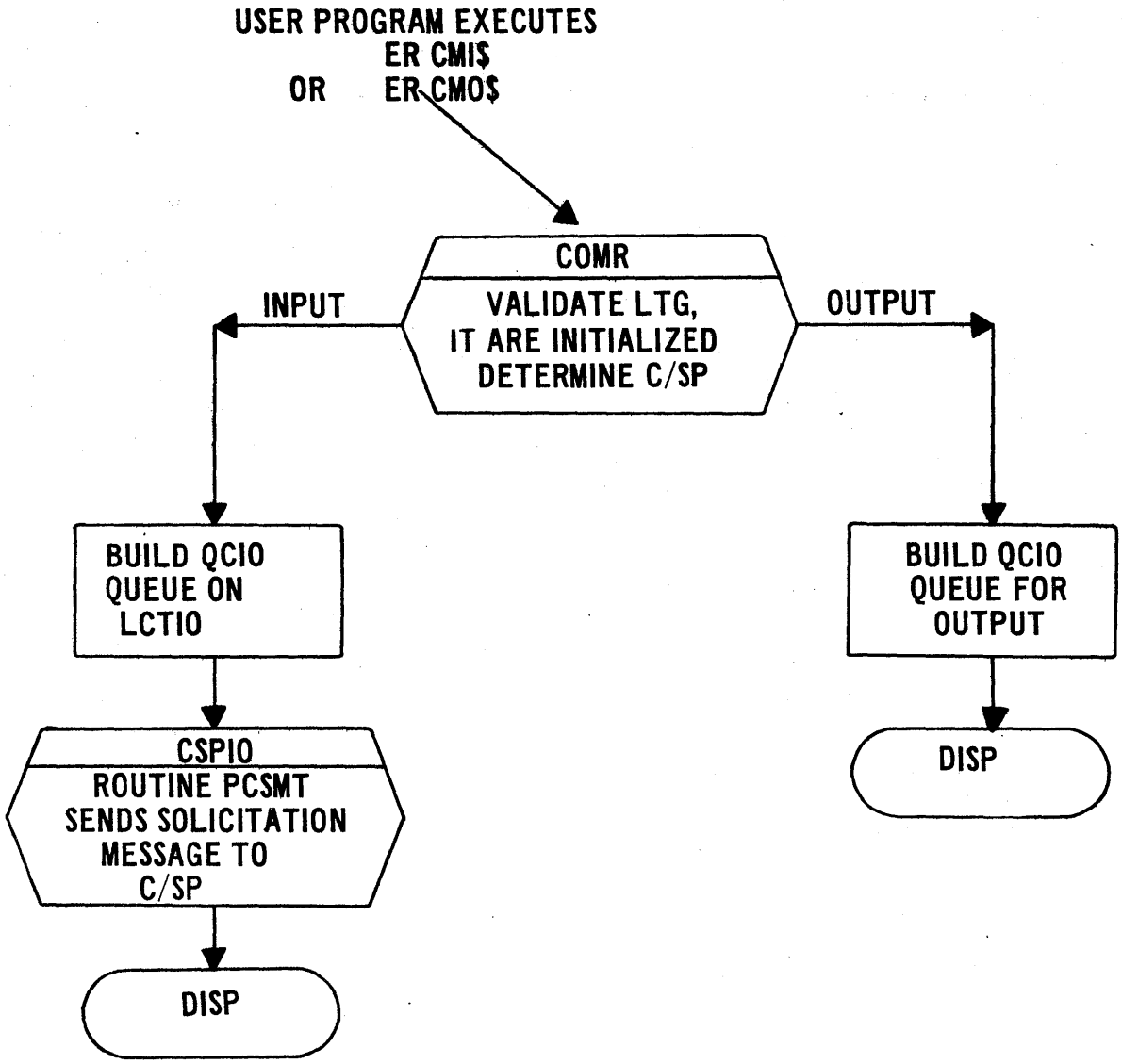
LINE INITIALIZATION
(REAL - TIME)



INPUT/OUTPUT
(REAL-TIME)

The 1100 element COMR is the interface between the real-time user and the C/SP for all input/output operations. COMR validates that the line has been assigned and the IT has been initialized for this LTT. If the request is for input, an access control word is built for the specified buffer. A QCIO (queue cell) for either single or pool mode is built and queued on the LCTIO table to wait for input from the C/SP. A solicitation message is sent to the C/SP to cause the C/SP to begin soliciting input for the given line. If the request is for output, an access control word, a QCIO is built and queued for output to the C/SP.

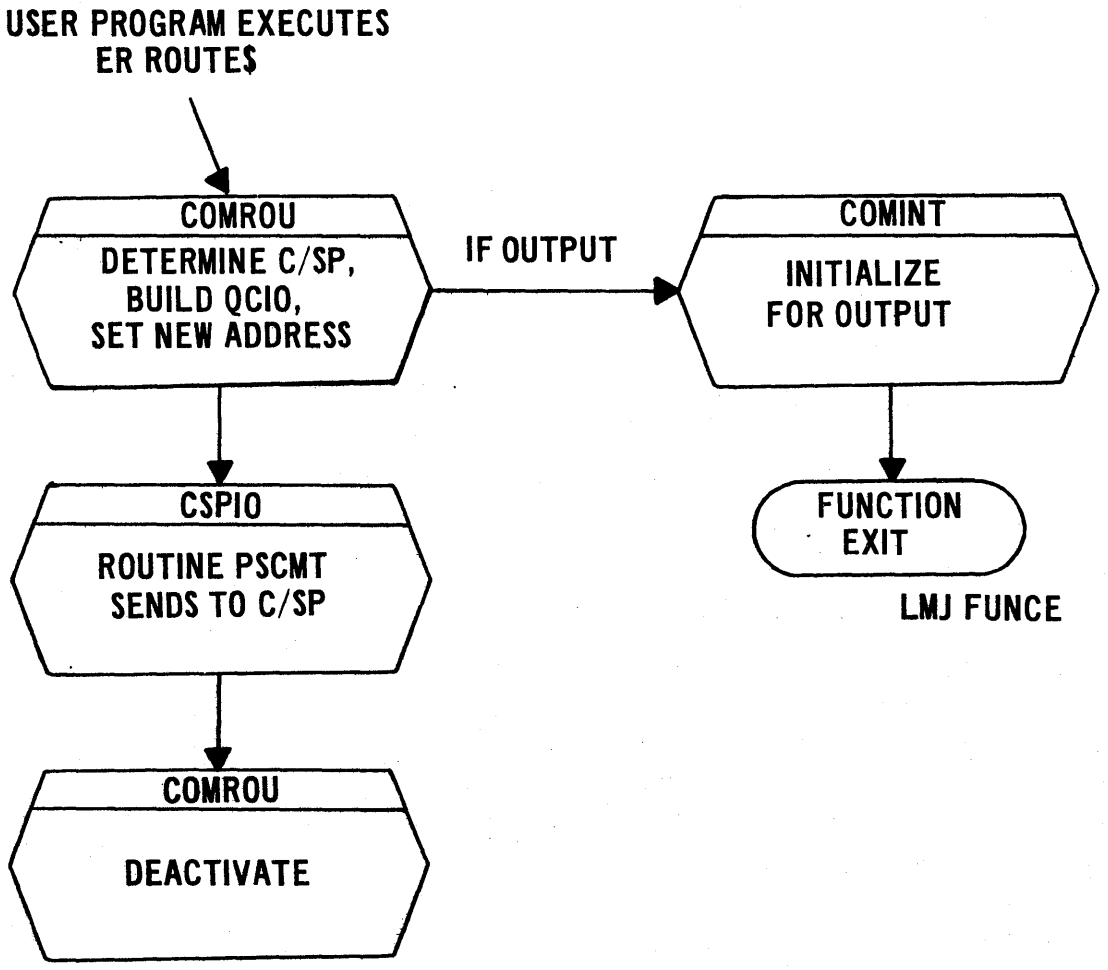
INPUT/OUTPUT
(REAL - TIME)



ROUTE TO NEW LTG
(REAL - TIME)

The 1100 element responsible for routing the input or output paths from one line terminal group (LTG) to another is COMROU. Validates that the line is assigned and initialized, then determines the path to be terminated and the site table for that path is read. Termination of the particular CLT is done and a message sent to the C/SP describing the path being terminated. The new path is determined and the site is read, marked assigned and the new line information is transferred to the IT. Relinking of the LCTIO takes place and the assign message is sent to the C/SP. When assigned the QCIO is constructed and queued for I/O.

ROUTE TO NEW LTG
(REAL - TIME)

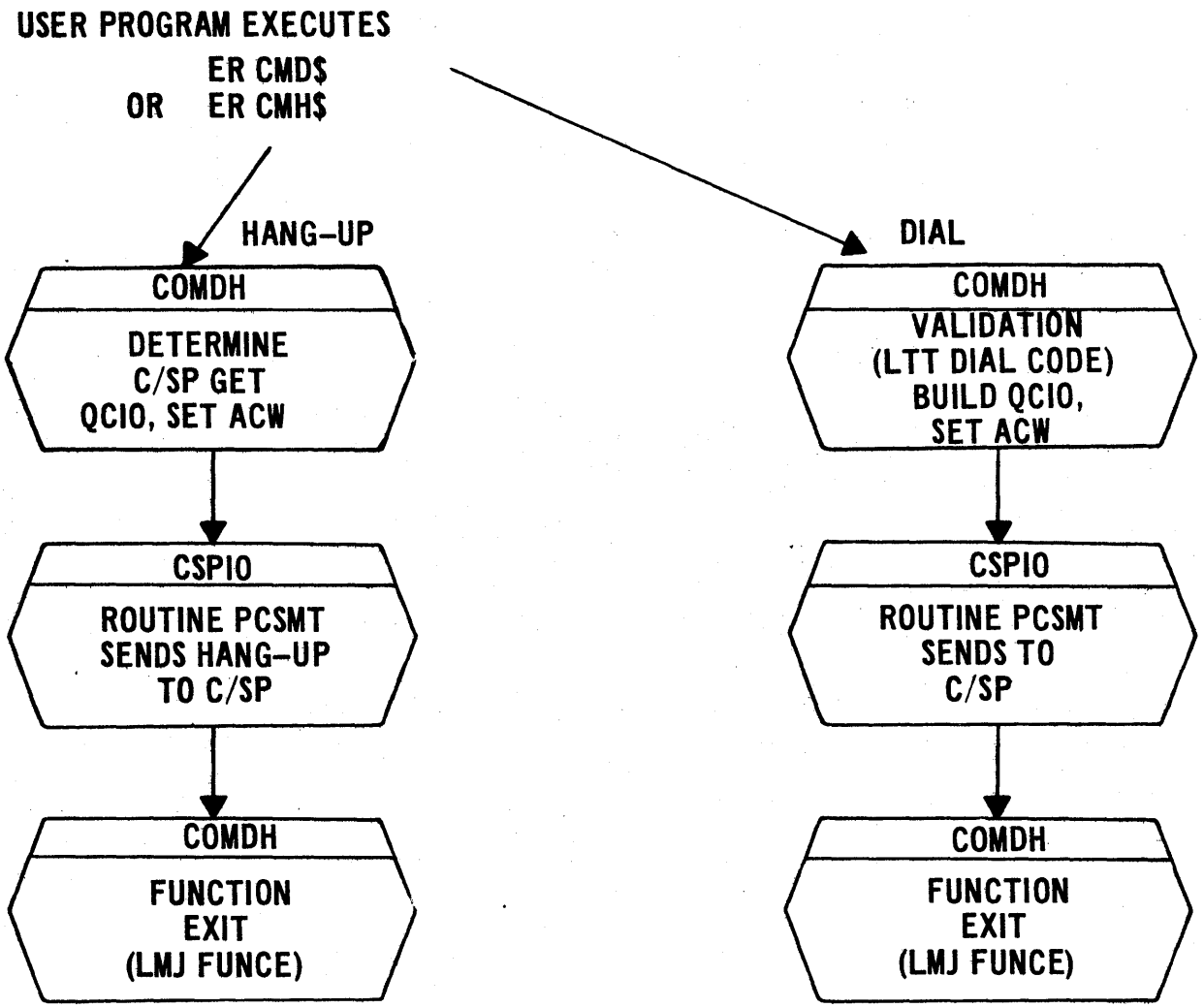


DIAL/HANG-UP
(REAL-TIME)

The 1100 element COMDH is responsible for the dial and hang-up functions. Validation of the users LTT to insure the LTG is assigned and the IT is initialized, occurs for either dial or hang-up. In the case of the dial function, the ACW, ESI control packet containing the dial digits for auto-dial are constructed. A QCIO is built and queued for output to the C/SP. If manual dial, a message is sent to the host console along with the digits to dial.

Hang-up function determines from the site table if the line is auto-dial, a QCIO is built and queued for output to the C/SP. For manual, a hang-up line message is sent to the host console.

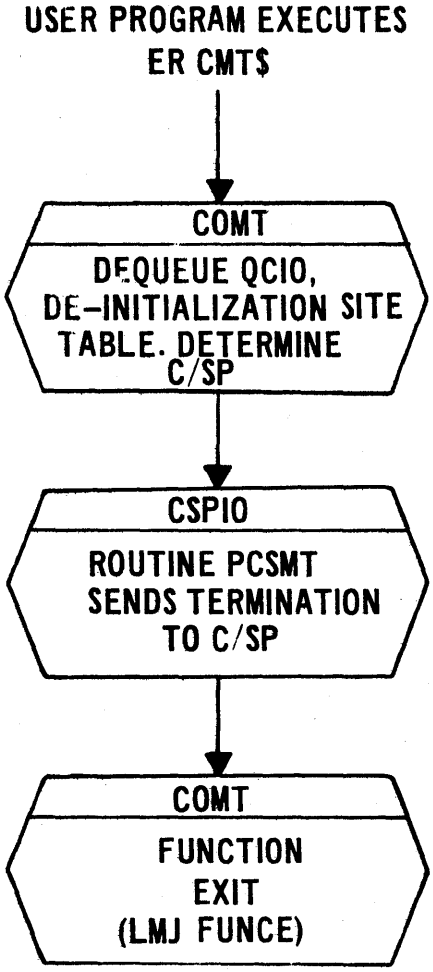
DIAL/HANG-UP
(REAL - TIME)



TERMINATION
(REAL - TIME)

Communications termination element (COMT) validates from the user LTT that the line is assigned and initialized. Then COMT sets termination flags in the LCTIO, dequeues the QCIO's, and marks the site table as de-inialized. A QCIO is built, queued for output to the C/SP to terminate the line.

TERMINATION
(REAL - TIME)

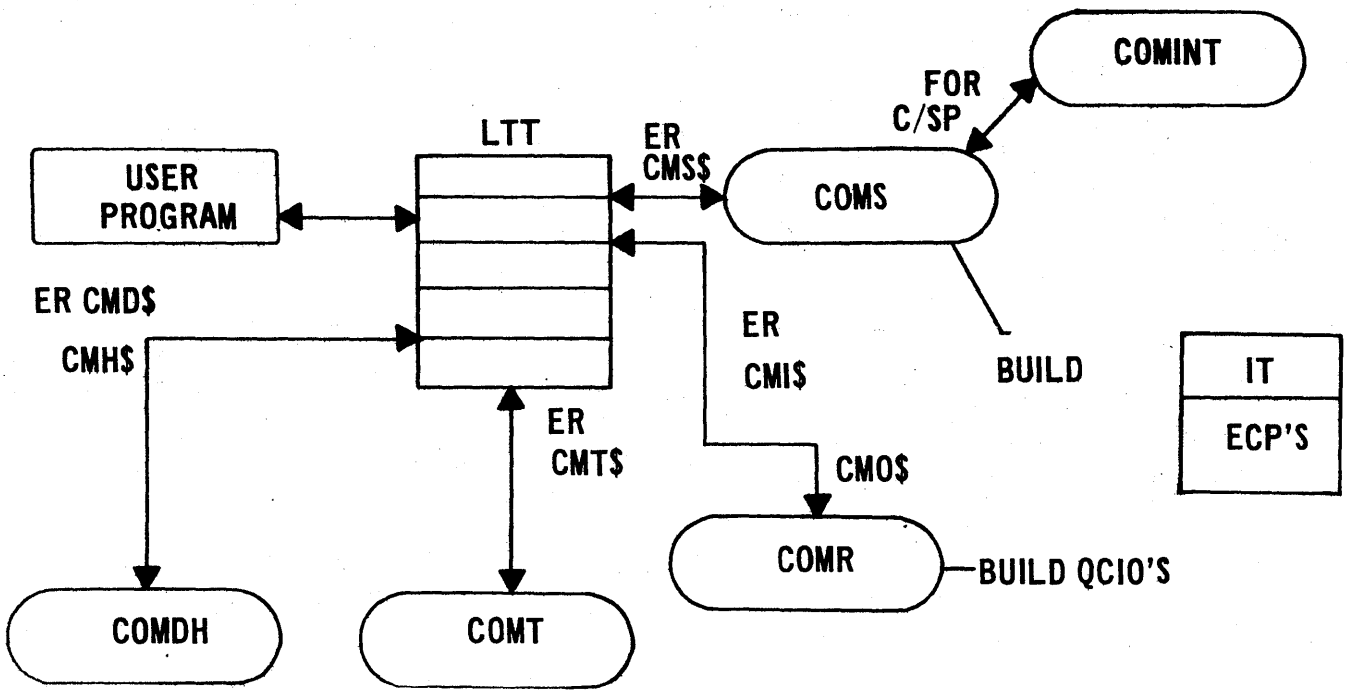


1100-C/SP REAL-TIME ELEMENTS

(OVERVIEW)

The user program enters the real-time mode via the execution of an ER RT$. The programmer must already have set up a line terminal table (LTT) in the programs data bank. The execution of the ER's to CMS$, CMI$, CMO$, CMD$, CMH$, CMT$ cause control to be given to the 1100 communications routines as indicated by the viewgraph on the opposite page. The 1100 communications routines perform the same functions as in previous systems. These routines determine the I/O is with the C/SP and control is passed to the C/SP routines for queueing the I/O, determining the ICA format, and any other information necessary for the C/SP. Each of the 1100 communication routines interface with CSPIO (the C/SP routine to handle queueing and processing of all I/O request for the C/SP). The CSPIO routine PCSMT is responsible for sending all output to the C/SP.

1100 - C/SP REAL TIME ELEMENTS



NOTE: THESE ELEMENTS USE THE ROUTINE CSPIO TO TRANSMIT TO THE C/SP. THE ELEMENTS ALSO USE THE QUEUEING ROUTINES FOR THE C/SP.

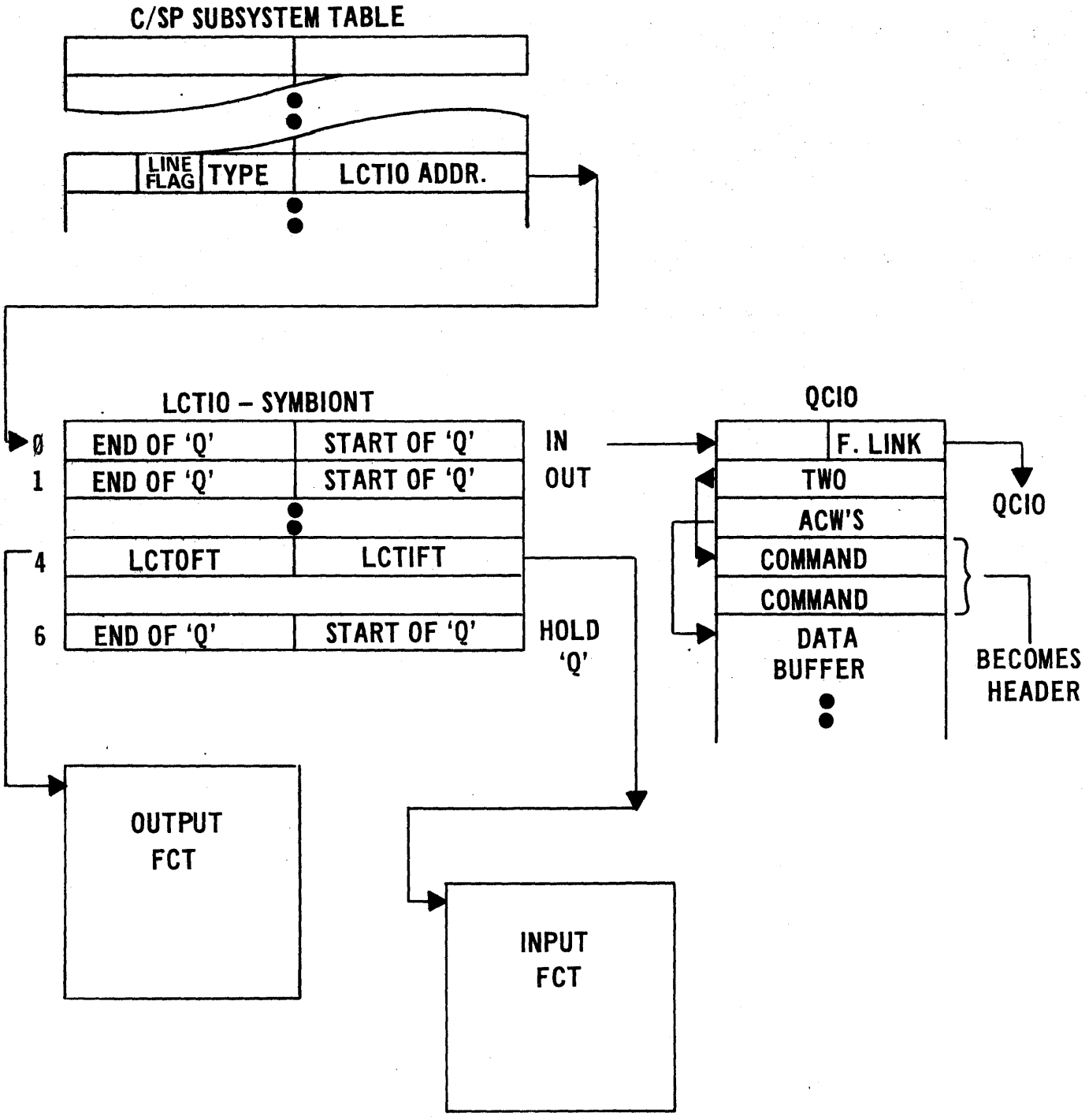
SYSTEM TABLES
(LINE CONTROL - SYMBIONT)

The line control table (LCT) portion of the subsystem table consists of one word entries for each line on the C/SP. Each word contains the basic status of the line and a pointer to the I/O control table (LCTIO).

The LCTIO is an extension of the IT table. Contains pointers to the beginning and end of input request queues, and output request queues for the line. The 'Hold Queue' is used to insure that the C/SP does not receive multiple records for the same terminal.

The input/output queue cells (QCIO) are used to hold the information necessary to perform an I/O request and determine the appropriate action upon completion of the request.

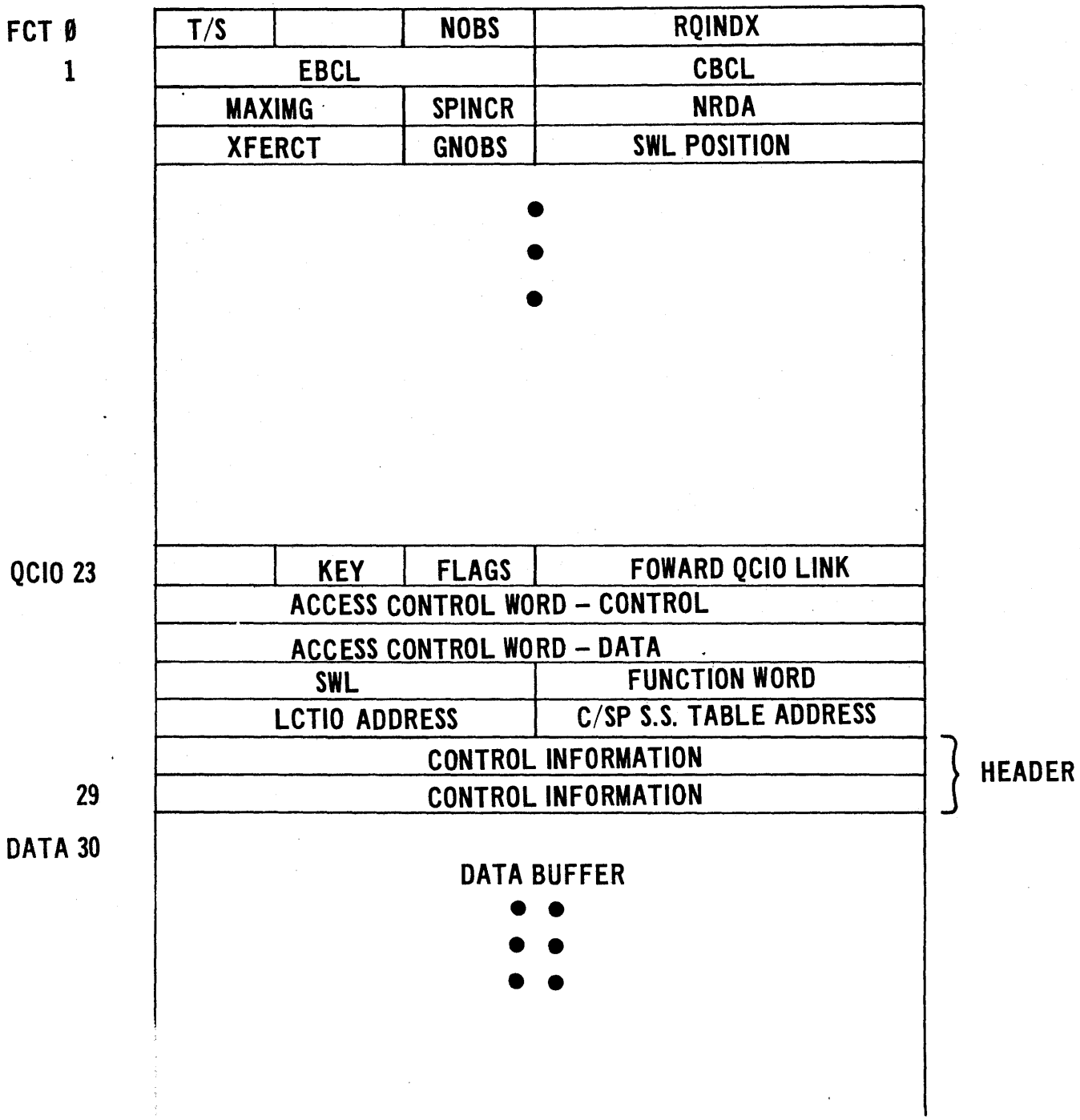
SYSTEM TABLES
(LINE CONTROL - SYMBIONT)



SYMBIONT FILE CONTROL TABLE

File Control Tables (FCT) are built for the card readers, printers, punches and contain the information needed by the software to control the device. A detailed description of each of the fields is contained in the C/SP-1100 Interface Technical Document. The QCIO (Queue Cells for I/O) are an extension of the FCT as indicated by the viewgraph on the opposite page, followed by the data buffer. Other QCIO's and buffers are chained in a forward link.

SYMBIONT FILE CONTROL TABLE

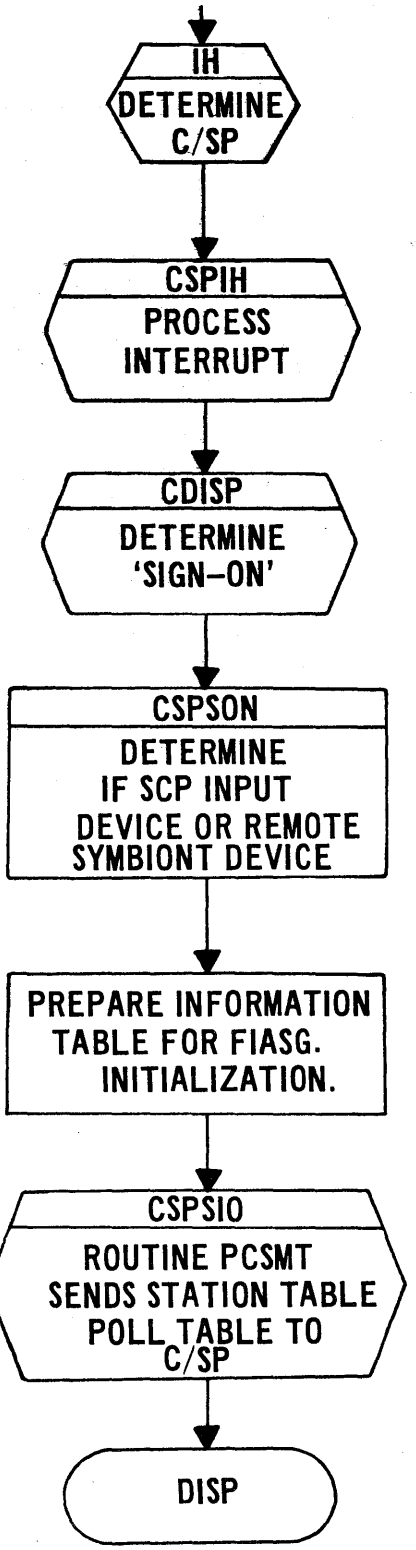


LINE INITIALIZATION (SYMBIONT)

A 'sign on' function is sent from the C/SP for symbiont line initialization. When it is determined that the interrupt is from the C/SP, the C/SP dispatcher (CDISP) passes control to the C/SP sign-on routine (CSPON). CSPON determines if this is an SCP input device or remote symbiont device and prepares an infor table for assignment (FIASG). The site table is marked as initialized, the station table, poll table, and other necessary information is sent to the C/SP via CSPIO.

LINE INITIALIZATION (SYMBIONT)

('SIGN-ON' FROM C/SP)



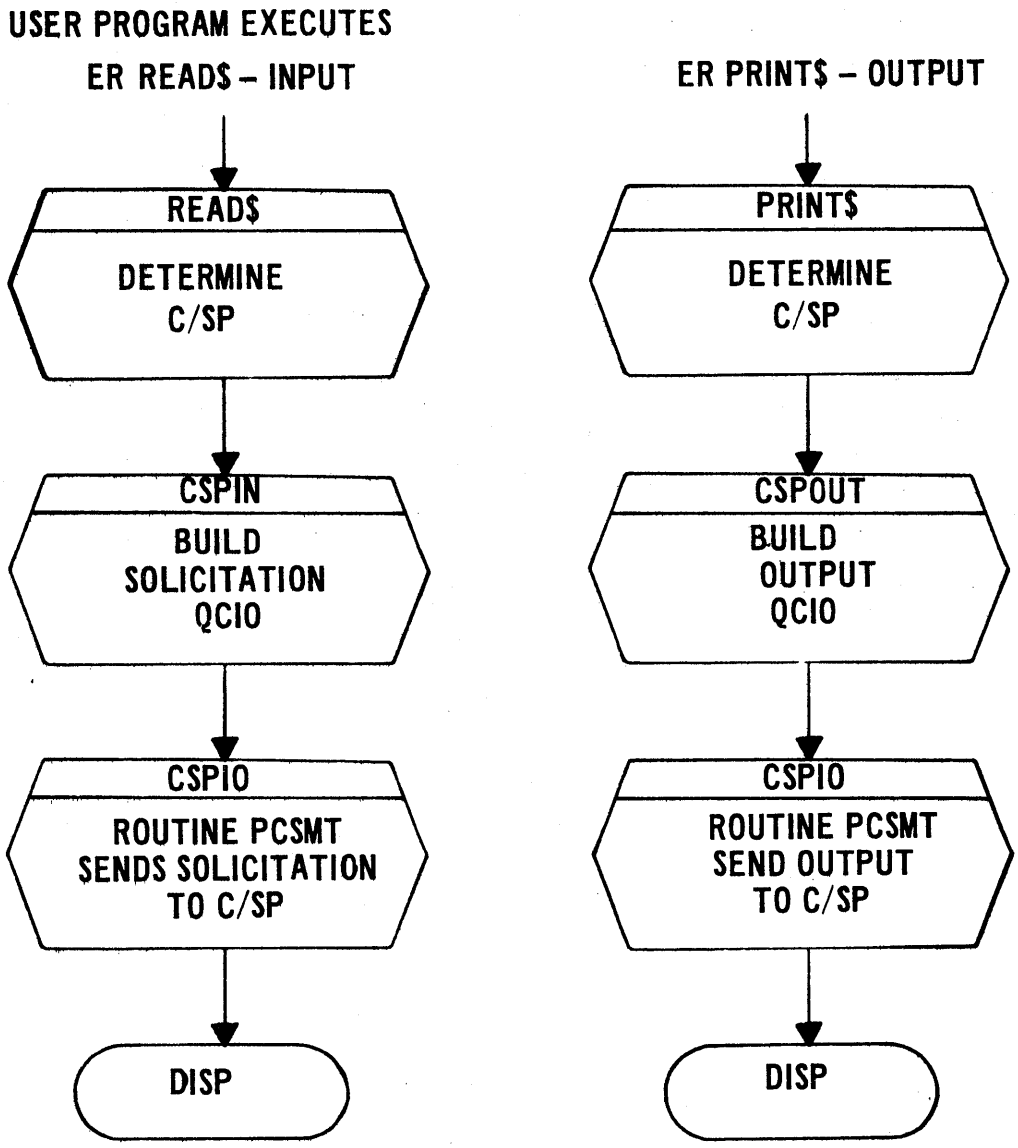
INPUT/OUTPUT

(SYMBIONT-DEMAND)

The symbiont READ$ is given control when the program executes an ER READ$. READ$ determines the input request is for the C/SP and passes control to the C/SP input symbiont (CSPIN). CSPIN builds a solicitation QCIO and the input solicitation is sent to the C/SP via the routine PCSMT in CSPIO. The requestor is deactivated until the dispatcher receives an input completion.

The symbiont PRINT$ is given control when the program executes an ER PRINT$. PRINT$ determines the output request is for the C/SP and passes control to the C/SP output symbiont (CSPOUT). CSPOUT builds the output QCIO and sends the output to the C/SP via the routine PCSMT in CSPIO. Control is given to the requestor. When the SNR acknowledge is received from the C/SP the symbionts are activated and look for more output. If there is no more output the symbionts are deactivated.

INPUT/OUTPUT
(SYMBIONT - DEMAND)



INPUT/OUTPUT
(SYMBIONT-BATCH)

The users program executes an ER READ$. READ$ is given control and reads an image from a mass storage READ$ file which was constructed to accept continuous input from the C/SP. Control is returned to the requestor.

The symbiont PRINT$ is given control when the user program executes an ER PRINT$. PRINT$ writes the output images to a mass storage PRINT$ file constructed for the user program. Control is given back to the requesting program. When the user program has terminated, the C/SP output symbiont (CSPOUT) builds the output QCIO and sends the output to the C/SP via the routine, PCSMT in CSPIO. When the last record has been sent to the C/SP, the site assignment is freed via FACILITIES.

INPUT/OUTPUT
(SYMBIONT - BATCH)