United States Patent [19]

Cook et al.

[11] **4,074,352**

[45] **Feb. 14, 1978**

[54] **MODULAR BLOCK UNIT FOR INPUT-OUTPUT SUBSYSTEM**

[75] Inventors: **Darwen John Cook**, Alhambra; **Donald Alexander Millers, II**, San Clemente, both of Calif.

[73] Assignee: **Burroughs Corporation**, Detroit, Mich.

[21] Appl. No.: **728,456**

[22] Filed: **Sept. 30, 1976**

[51] **Int. Cl.**[2] **G06F 3/04**

[52] **U.S. Cl.** **364/200**

[58] **Field of Search** 364/200

[56] **References Cited**

U.S. PATENT DOCUMENTS

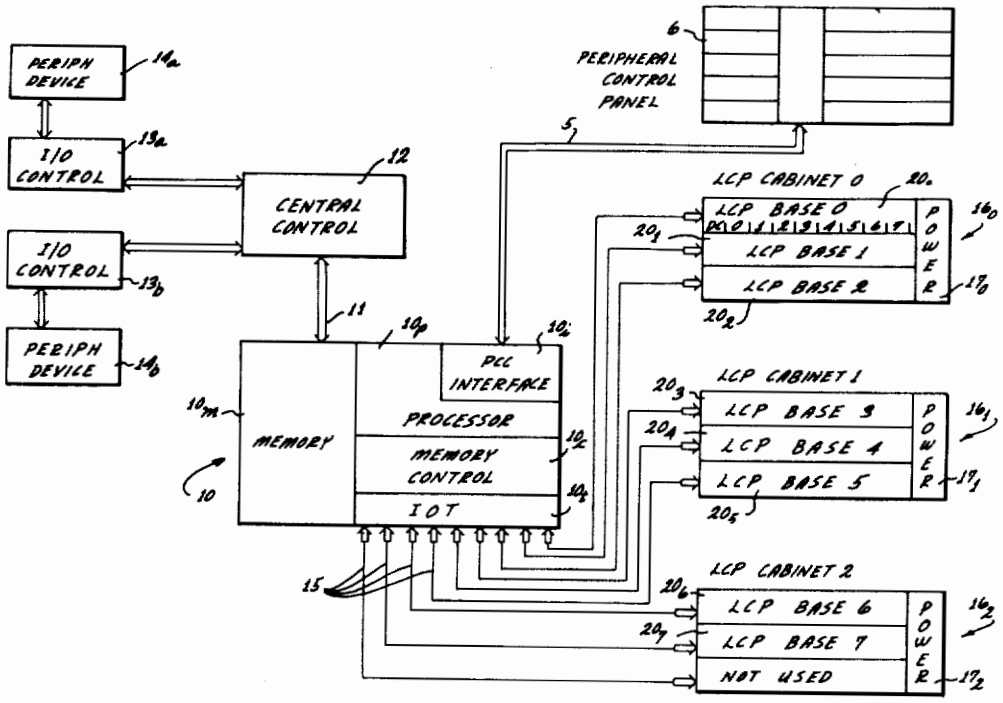
3,302,181	1/1967	Lee	340/172.5
3,510,843	5/1970	Bennett et al.	340/172.5
3,514,758	5/1970	Bennett et al.	340/172.5
3,526,878	9/1970	Bennett et al.	340/172.5
3,601,810	8/1971	Anderson et al.	340/172.5
3,704,453	11/1972	Blackwell et al.	340/172.5
3,976,977	8/1976	Porter et al.	340/172.5

Primary Examiner—Gareth D. Shaw
Assistant Examiner—C. T. Bartz
Attorney, Agent, or Firm—Alfred W. Kozak; Nathan Cass; Kevin R. Peterson

[57] **ABSTRACT**

A modular interface unit to serve as a building block in the framework of an Input-Output Subsystem of a digital data processing system. The unit, designated as a Base Module, manages a group of Intelligent I/O Interface Control units (Line Control Processors) and provides a Base Module distribution-control interface called a Distribution Card unit. A group of Line Control Processors have a common set of backplane connections to the distribution-control means and to a maintenance unit and termination control unit. The maintenance unit permits the exercising and diagnosis of any selected Line Control Processor on the Base Module. The termination control unit provides clocking and control functions for all of the Line Control Processors in the module and also common termination impedances for the interconnecting backplane lines. Means are provided for priority ranking of the various Line Control Processors within the Base Module unit. The Base Module unit provides a building block by which a central processing unit of a digital data processing system may be expanded to handle a large number of peripheral devices with simplicity and reliability.

11 Claims, 29 Drawing Figures



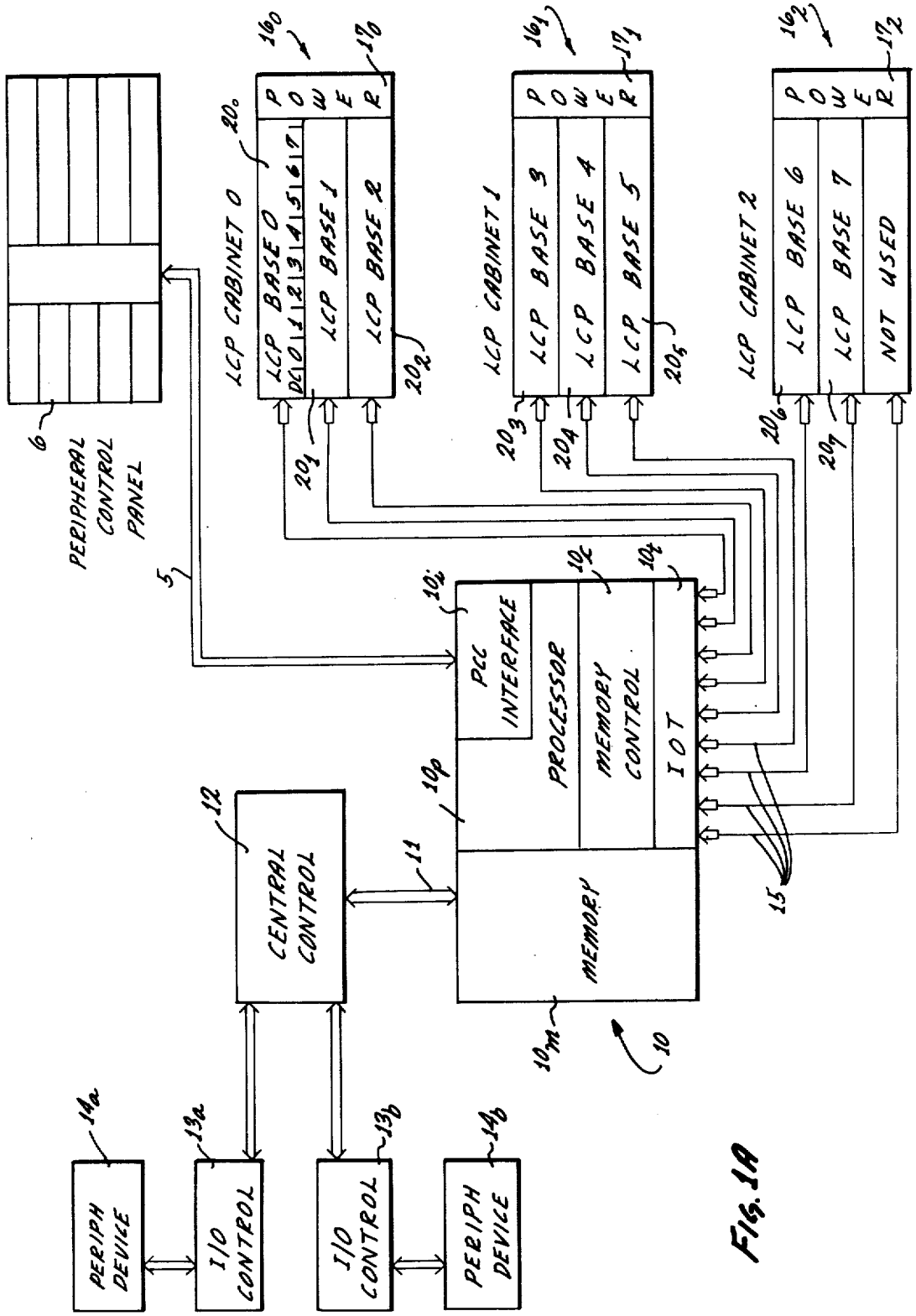

PERIPHERAL CONTROL PANEL
LCP CABINET 0
LCP BASE 0
LCP BASE 1
LCP BASE 2
LCP CABINET 1
LCP BASE 3
LCP BASE 4
LCP BASE 5
LCP CABINET 2
LCP BASE 6
LCP BASE 7
NOT USED
POWER
PCC INTERFACE
PROCESSOR
MEMORY CONTROL
IOT
MEMORY
CENTRAL CONTROL
PERIPH DEVICE
I/O CONTROL
I/O CONTROL
PERIPH DEVICE

FIG. 1A

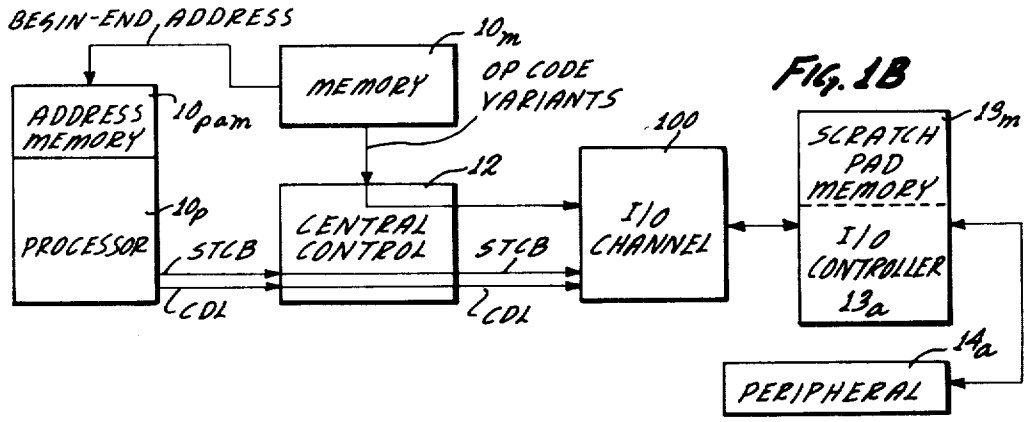


FIG. 1B

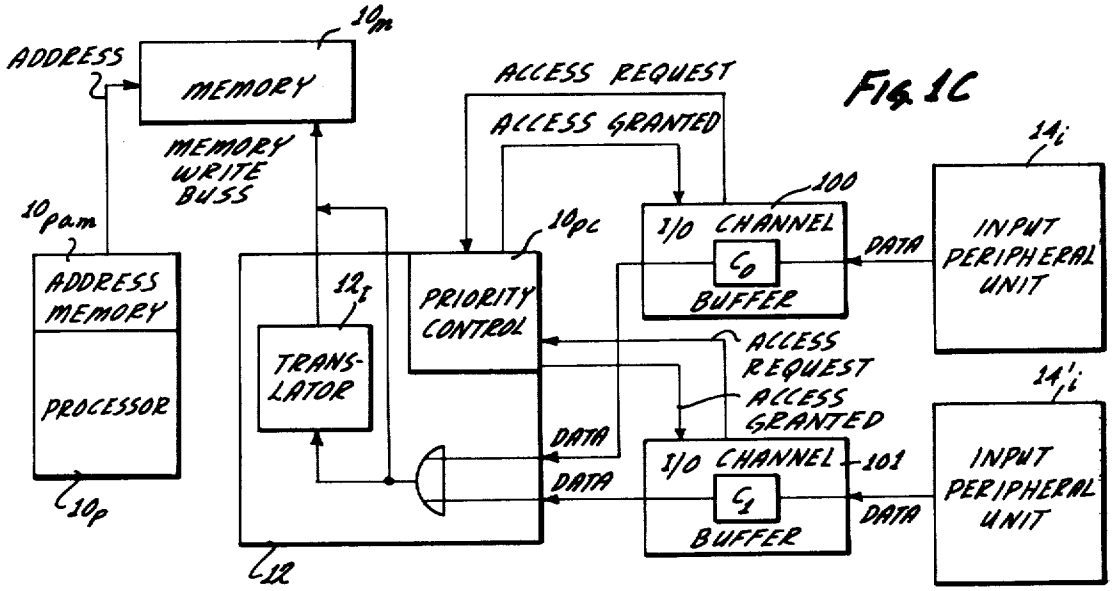


FIG. 1C

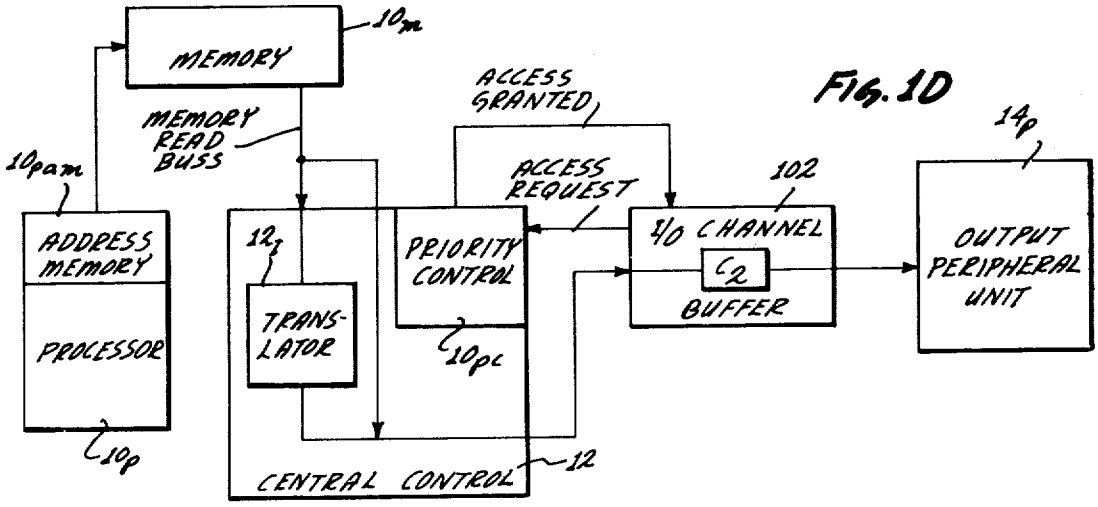


FIG. 1D

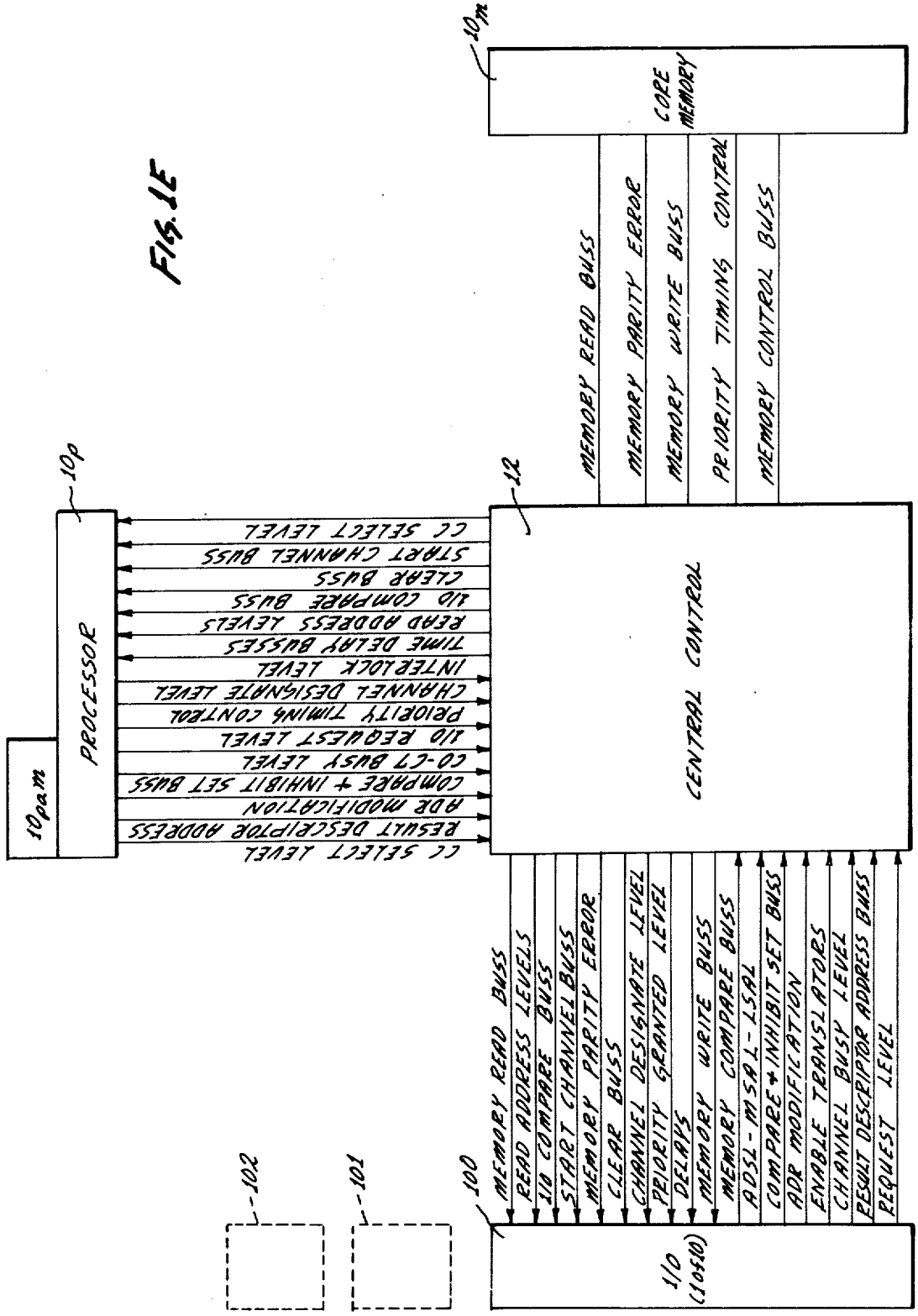


FIG. 1F
10m
CORE MEMORY
MEMORY READ BUSS
MEMORY PARITY ERROR
MEMORY WRITE BUSS
PERIORITY TIMING CONTROL
MEMORY CONTROL BUSS
10p
12
CENTRAL CONTROL
CC SELECT LEVEL
START CHANNEL BUSS
CLEAR BUSS
I/O COMPARE BUSS
READ ADDRESS LEVELS
INTERLOCK LEVEL
TIME DELAY BUSSES
CHANNEL DESIGNATE LEVEL
PRIORITY TIMING CONTROL
I/O REQUEST LEVEL
CO-CT BUSY LEVEL
COMPARE + INHIBIT SET BUSS
ADR MODIFICATION
RESULT DESCRIPTOR ADDRESS
CC SELECT LEVEL
10pam
PROCESSOR
MEMORY READ BUSS
READ ADDRESS LEVELS
I/O COMPARE BUSS
START CHANNEL BUSS
MEMORY PARITY ERROR
CLEAR BUSS
CHANNEL DESIGNATE LEVEL
PRIORITY GRANTED LEVEL
DELAYS
MEMORY WRITE BUSS
MEMORY COMPARE BUSS
ADSL-MSAL-LSAL
COMPARE+INHIBIT SET BUSS
ADR MODIFICATION
ENABLE TRANSLATORS
CHANNEL BUSY LEVEL
RESULT DESCRIPTOR ADDRESS BUSS
REQUEST LEVEL
100
I/O (10x10)
101
102

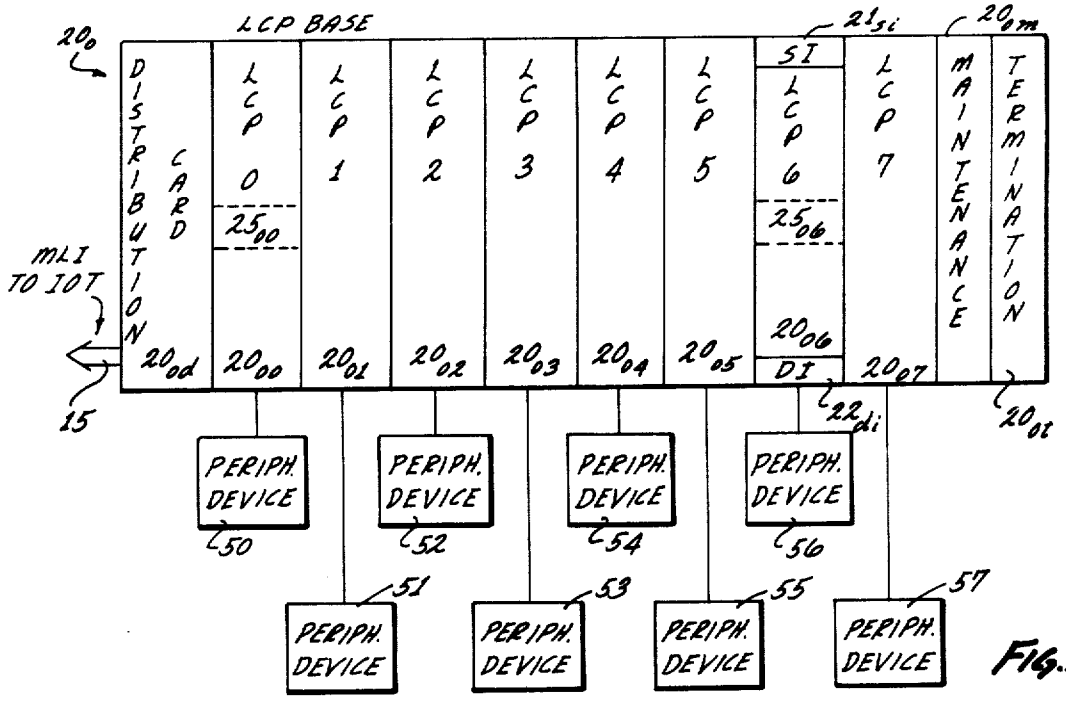

LCP BASE
DISTRIBUTION CARD
LCP 0
LCP 1
LCP 2
LCP 3
LCP 4
LCP 5
LCP 6
LCP 7
SI
DI
MAINTENANCE
TERMINATION
MLI TO IOT
PERIPH. DEVICE

FIG. 2

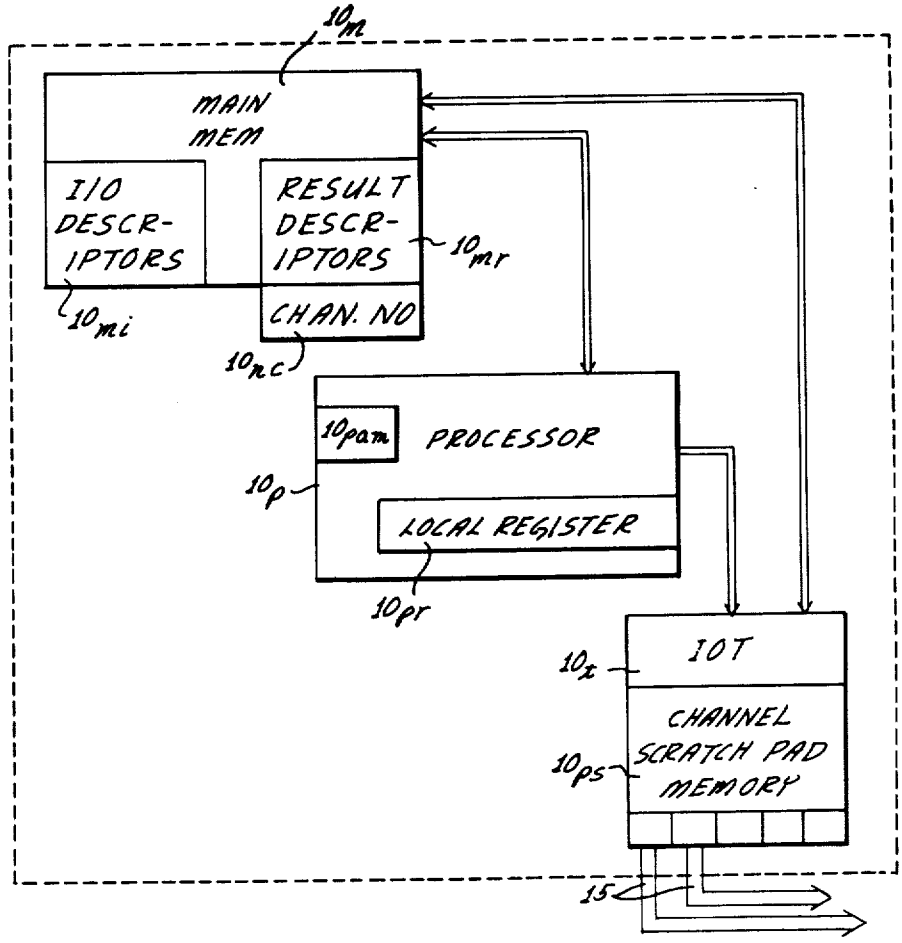

MAIN MEM
I/O DESCRIPTORS
RESULT DESCRIPTORS
CHAN. NO
PROCESSOR
LOCAL REGISTER
IOT
CHANNEL SCRATCH PAD MEMORY

FIG. 3

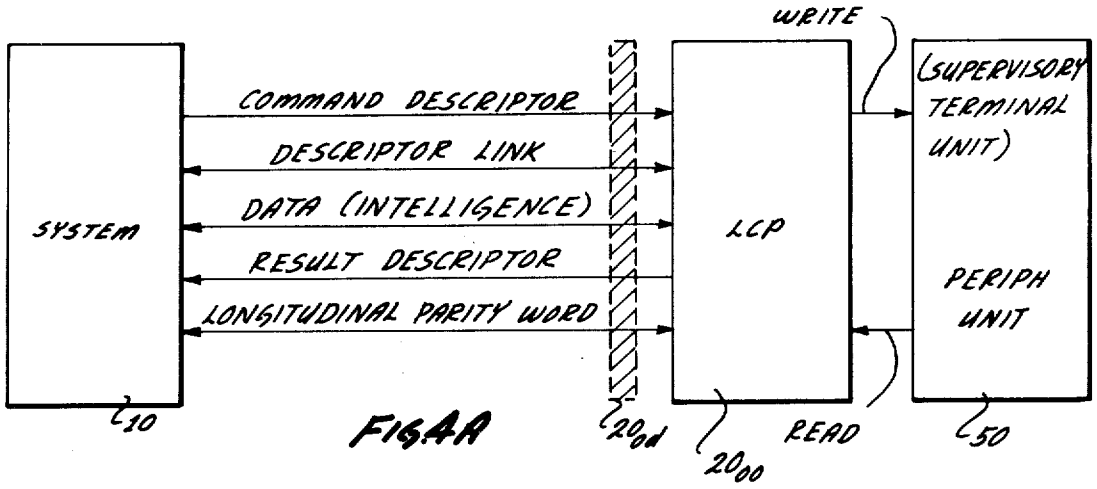


FIG.4A

		WRITE	READ	WRITE FLIP	TEST	TEST ENABLE	CONDITIONAL CANCEL	ECHO
OP DIGIT	A8	0	1	0	0	0	0	0
	A4	1	0	1	0	0	0	0
	A2	0	0	0	1	1	1	0
	A1	0	0	0	0	0	0	1
VARIANT DIGIT 1	B8	0	0	1	0	0	1	0
	B4	0	0	0	0	1	0	0
	B2	0	0	0	0	0	0	0
	B1	0	X	X	0	X	0	—
VARIANT DIGIT 2	C8	MUST BE EQUAL TO ZERO						
	C4							
	C2							
	C1							
VARIANT DIGIT 3	D8	MUST BE EQUAL TO ZERO						
	D4							
	D2							
	D1							

NOTES: IF BIT B1 OF VARIANT DIGIT 1 IS EQUAL TO 1, THE OCCURRENCE OF "TIMEOUT" WILL BE INHIBITED DURING A READ OPERATION. FOR AN ECHO OPERATION, BIT B1 OF VARIANT DIGIT 1 CAN BE EITHER 0 OR 1

FIG.4B

FIG. 4C

RESULT DESCRIPTOR

A8 RESERVED	B8 CONDITIONAL CANCEL OPERATION COMPLETED.	C8 TERMINATE SIGNAL RECEIVED DURING TRANSFER OF DATA FROM THE SYSTEM BEFORE AN ENDING CODE WAS RECEIVED BY THE LCP.	D8 RESERVED
A4 DESCRIPTOR ERROR.	B4 TIMEOUT OCCURRED DURING TRANSMISSION FROM TERMINAL UNIT	C4 TERMINAL UNIT BLOCK CHECK CHARACTER ERROR.	D4 RESERVED
A2 SYSTEM VERTICAL PARITY ERROR.	B2 TERMINAL UNIT BLOCK CHECK CHARACTER ERROR.	C2 ACCESS ERROR. LCP BUFFER NOT SERVICED IN TIME TO COMPLETE A DATA TRANSFER DURING A READ OR WRITE OPERATION.	D2 RESERVED
A1 SYSTEM LONGITUDINAL PARITY ERROR.	B1 LCP BUFFER PARITY ERROR DURING TRANSFER OF DATA FROM TERMINAL UNIT TO LCP.	C1 TERMINATE SIGNAL RECEIVED DURING ECHO OPERATION OR DURING TRANSFER OF DATA FROM LCP TO SYSTEM BEFORE AN ENDING CODE WAS RECEIVED BY THE LCP.	D1 RESERVED
A DIGIT	B DIGIT	C DIGIT	D DIGIT

1A DESCRIPTORS USED TO GENERATE LCP COMMAND MESSAGES CIM,

DESCRIPTION	OP	VARIANT	A ADDRESS	B ADDRESS	C ADDRESS
READ, FORWARD	40	SLLL	AAAAAA	BBBBBB	
READ, FORWARD	50	SLLL	AAAAAA	BBBBBB	CCCCCC
READ, BACKWARD	41	SLLL	AAAAAA	BBBBBB	
READ, BACKWARD	51	SLLL	AAAAAA	BBBBBB	CCCCCC
WRITE, FORWARD	42	SLLL	AAAAAA	BBBBBB	
WRITE, FORWARD	52	SLLL	AAAAAA	BBBBBB	CCCCCC
WRITE, BACKWARD	43	SLLL	AAAAAA	BBBBBB	
WRITE, BACKWARD	53	SLLL	AAAAAA	BBBBBB	CCCCCC
TEST	44	SLLL			
TEST	54	SLLL	AAAAAA	BBBBBB	CCCCCC
ECHO	48	SLLL	AAAAAA	BBBBBB	
ECHO	58	SLLL	AAAAAA	BBBBBB	CCCCCC
		↑↑↑↑ ABCD DIGITS			

NOTES:
S = VARIANTS USED BY THE IOT
8 = INHIBIT DATA TRANSFER TO MEMORY
4 = RESERVED
2 = ASCII TRANSLATION
1 = RESERVED
L = LCP VARIANTS WHICH ARE SENT UNMODIFIED TO LCP

1B DESCRIPTORS USED BY IOT

EXTENDED RD STORE	70 XX CC	AAAAAA	BBBBBB
CONDITIONAL CANCEL	71 XX CC	(CONVERTS TO TEST OP FOR LCP)	
UNCONDITIONAL CANCEL	72 XX CC	(CLEARS THE LCP)	

CC = CHANNEL DESIGNATION (00-77)

FIG. 5A

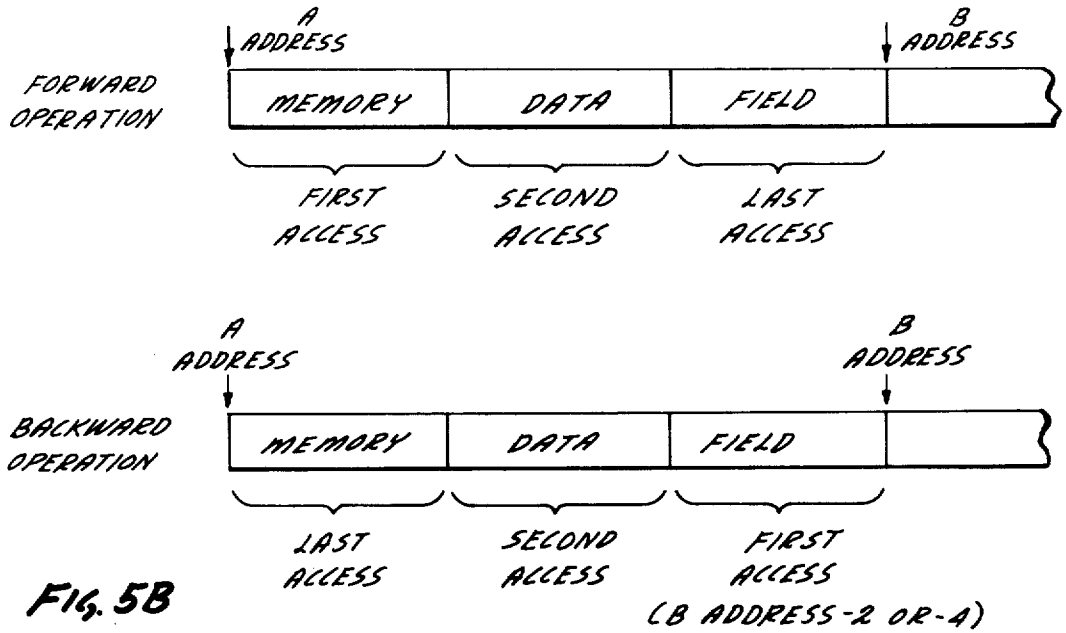


FIG. 5B

DATA FIELD BOUNDARIES

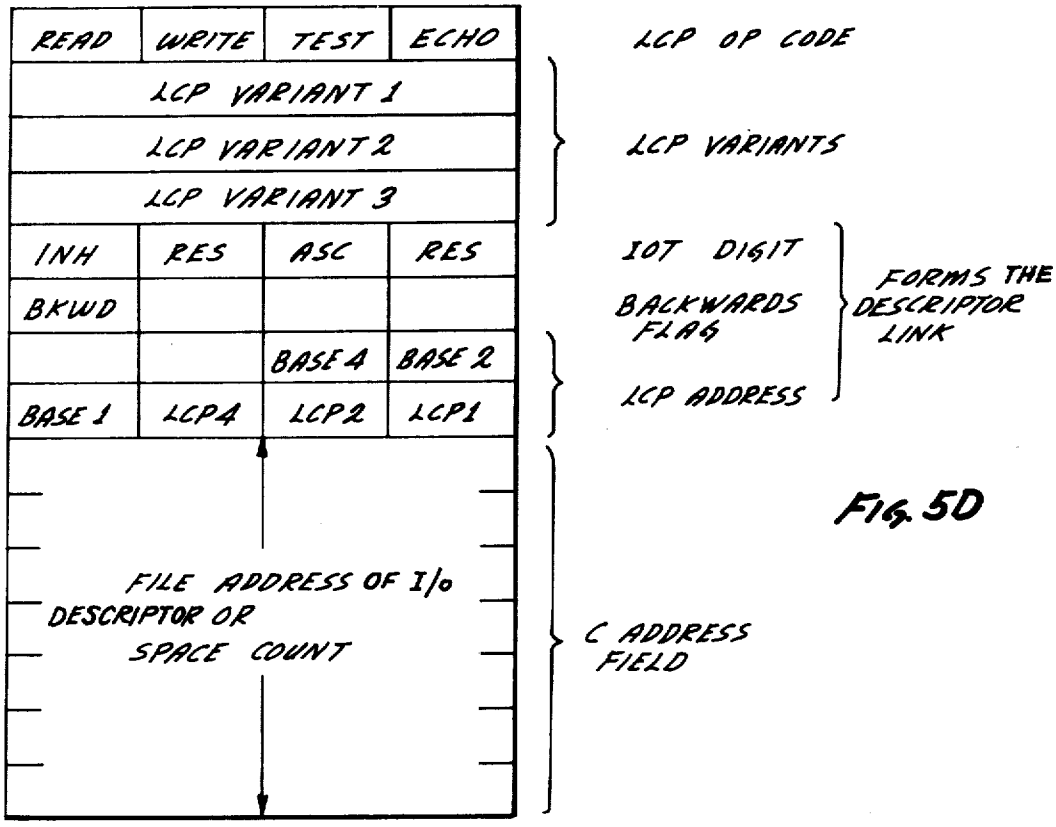


FIG. 5D

IOT DESCRIPTOR REGISTER INFORMATION

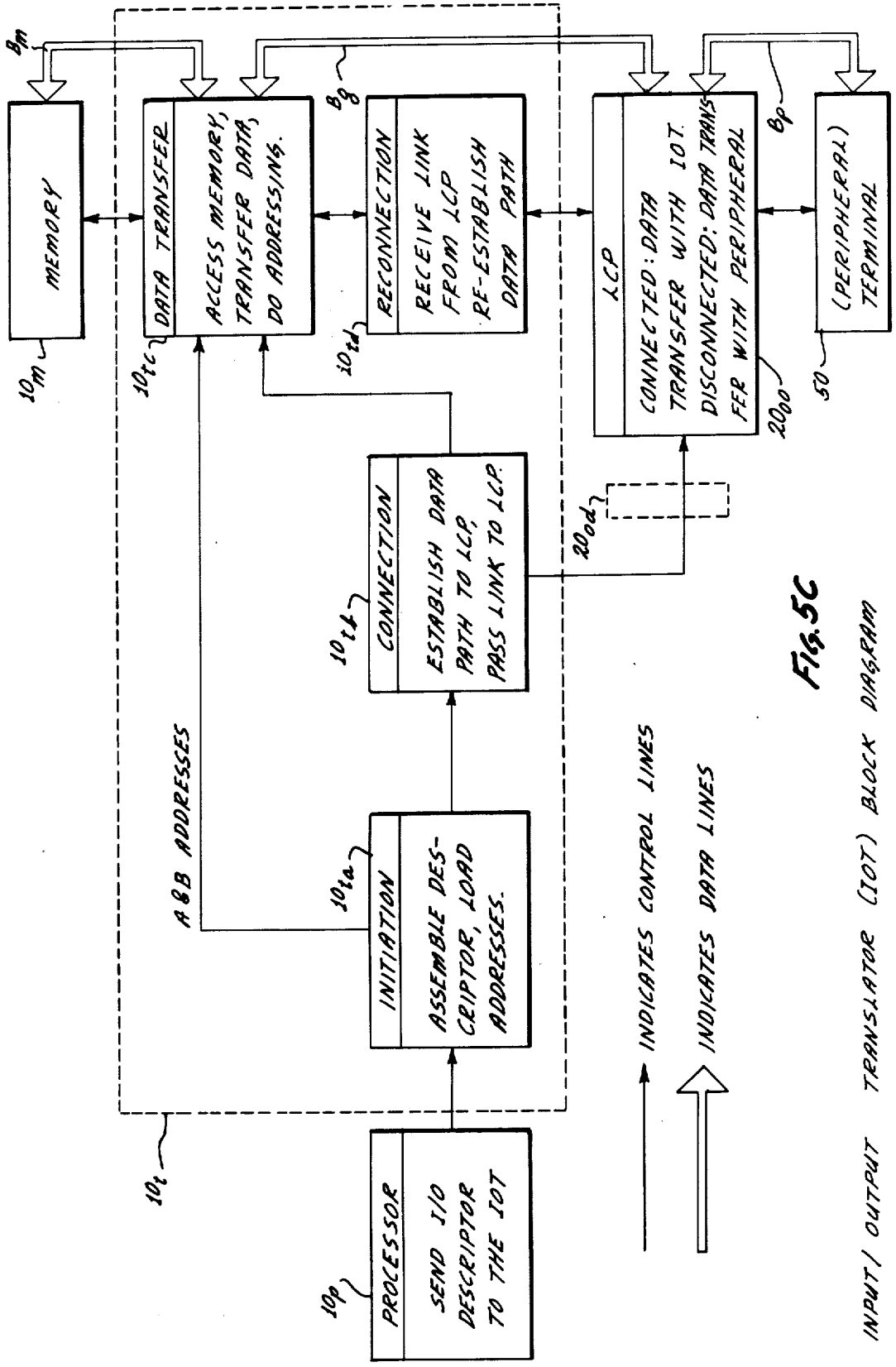
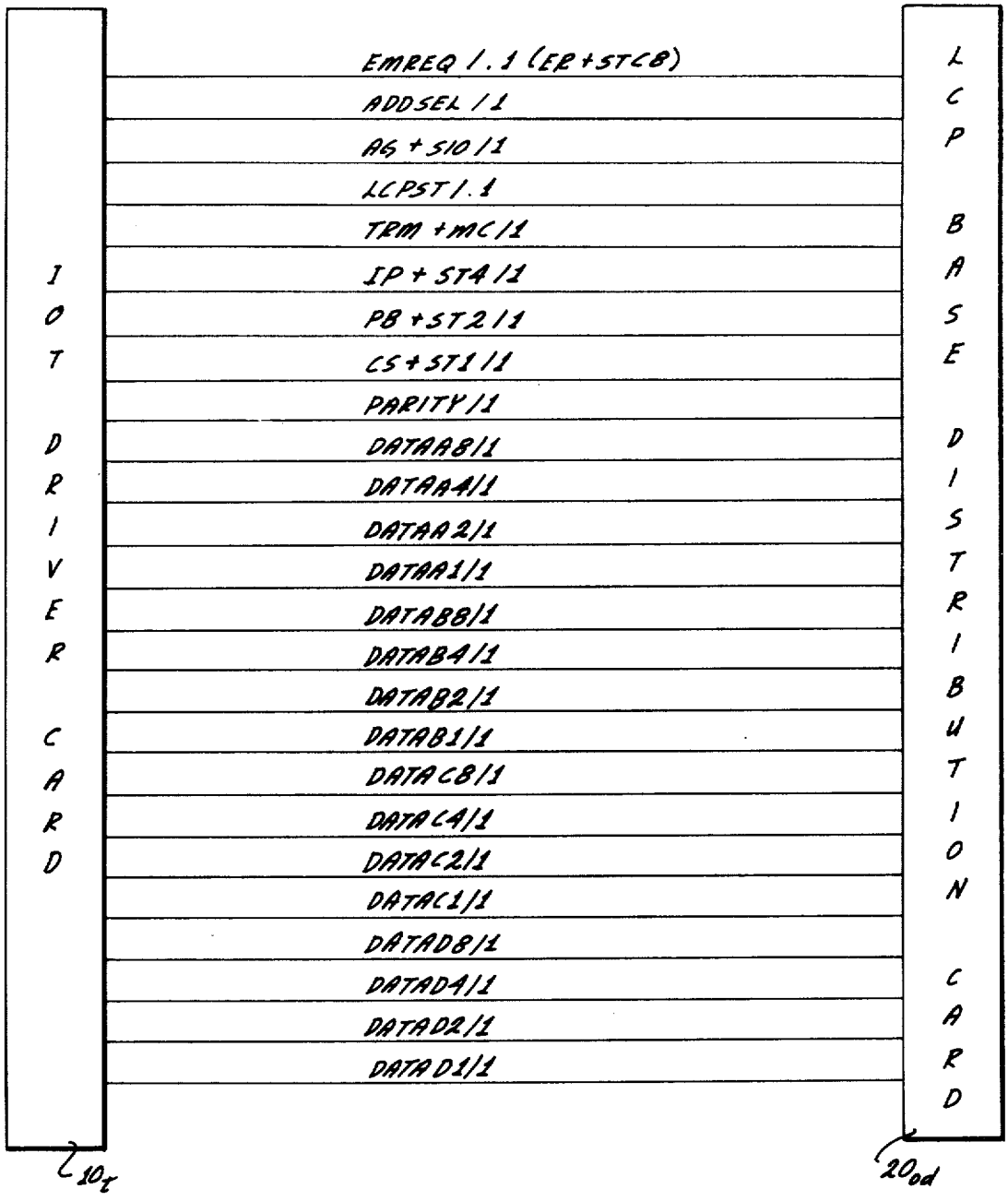


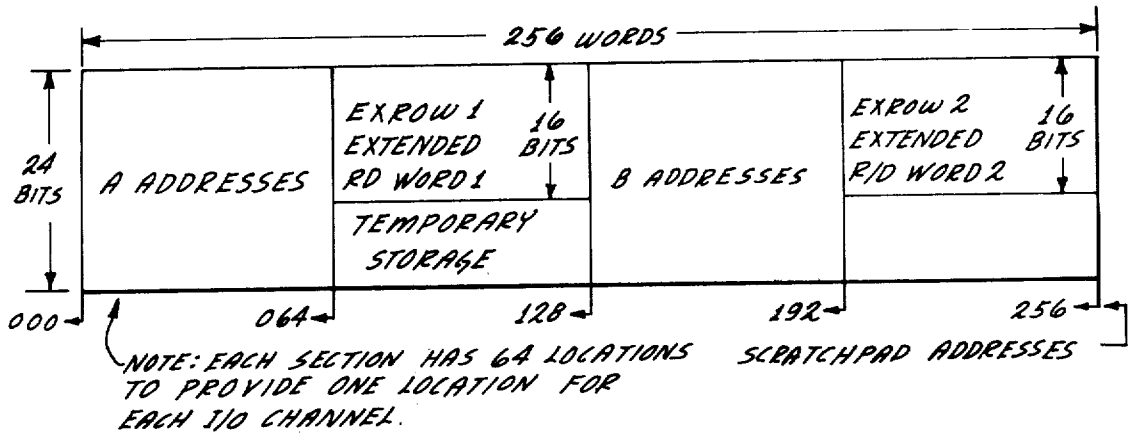
Fig. 5C

INPUT/OUTPUT TRANSLATOR (IOT) BLOCK DIAGRAM



MESSAGE LEVEL INTERFACE

FIG. 5E



ADDRESSING SCHEME

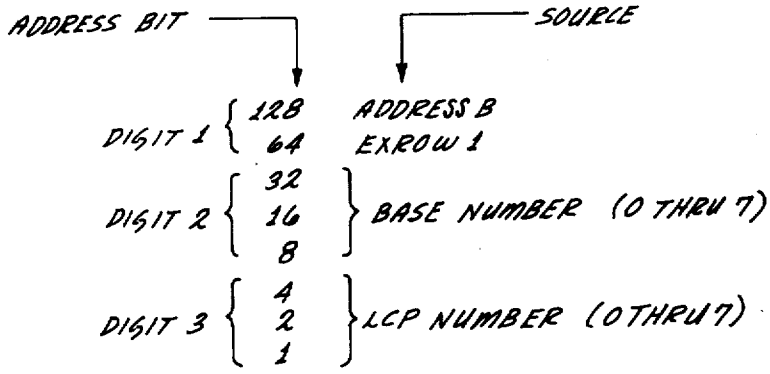


FIG. 5F
IOT SCRATCHPAD MEMORY 10 ps

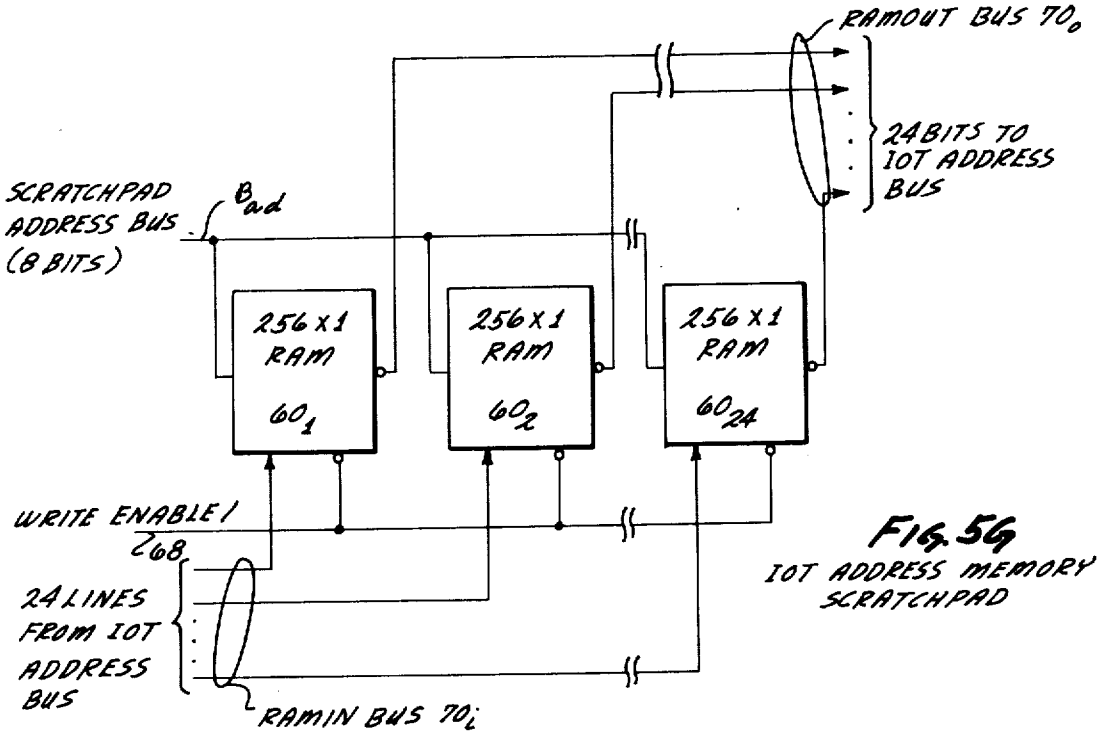


FIG. 5G
IOT ADDRESS MEMORY SCRATCHPAD

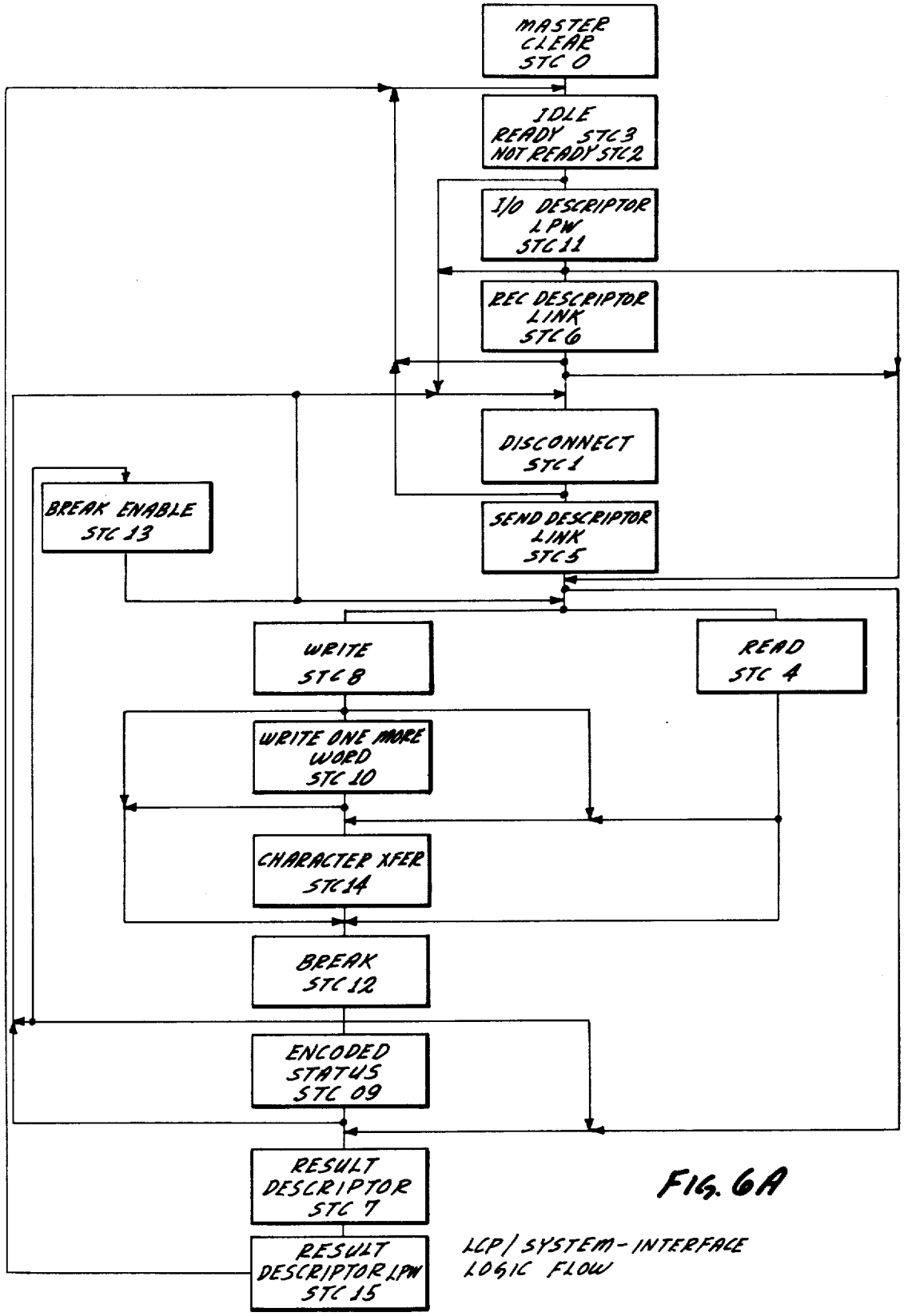


FIG. 6A

LCP/SYSTEM-INTERFACE LOGIC FLOW

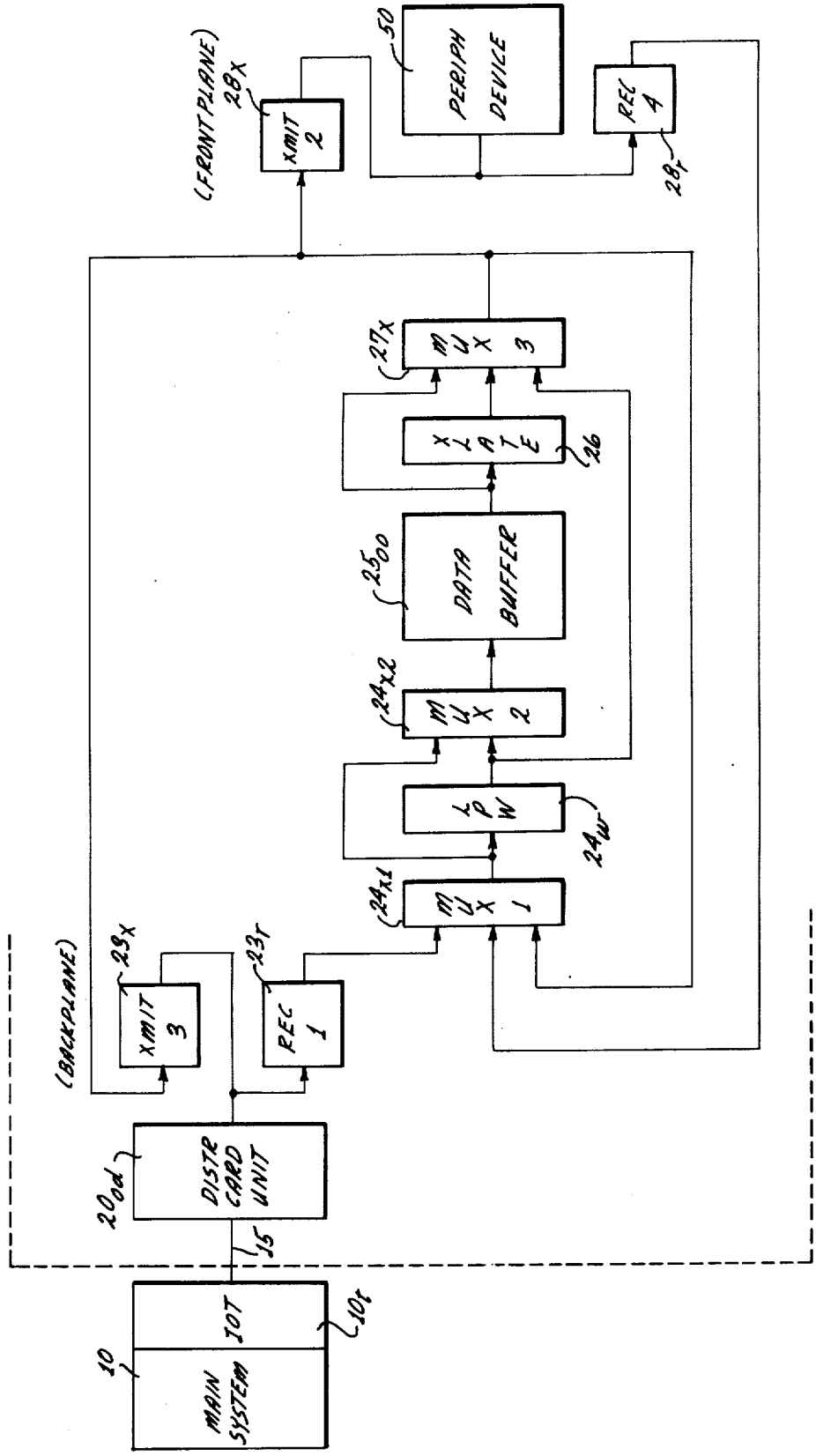


FIG. 6B

LCP BLOCK DIAGRAM

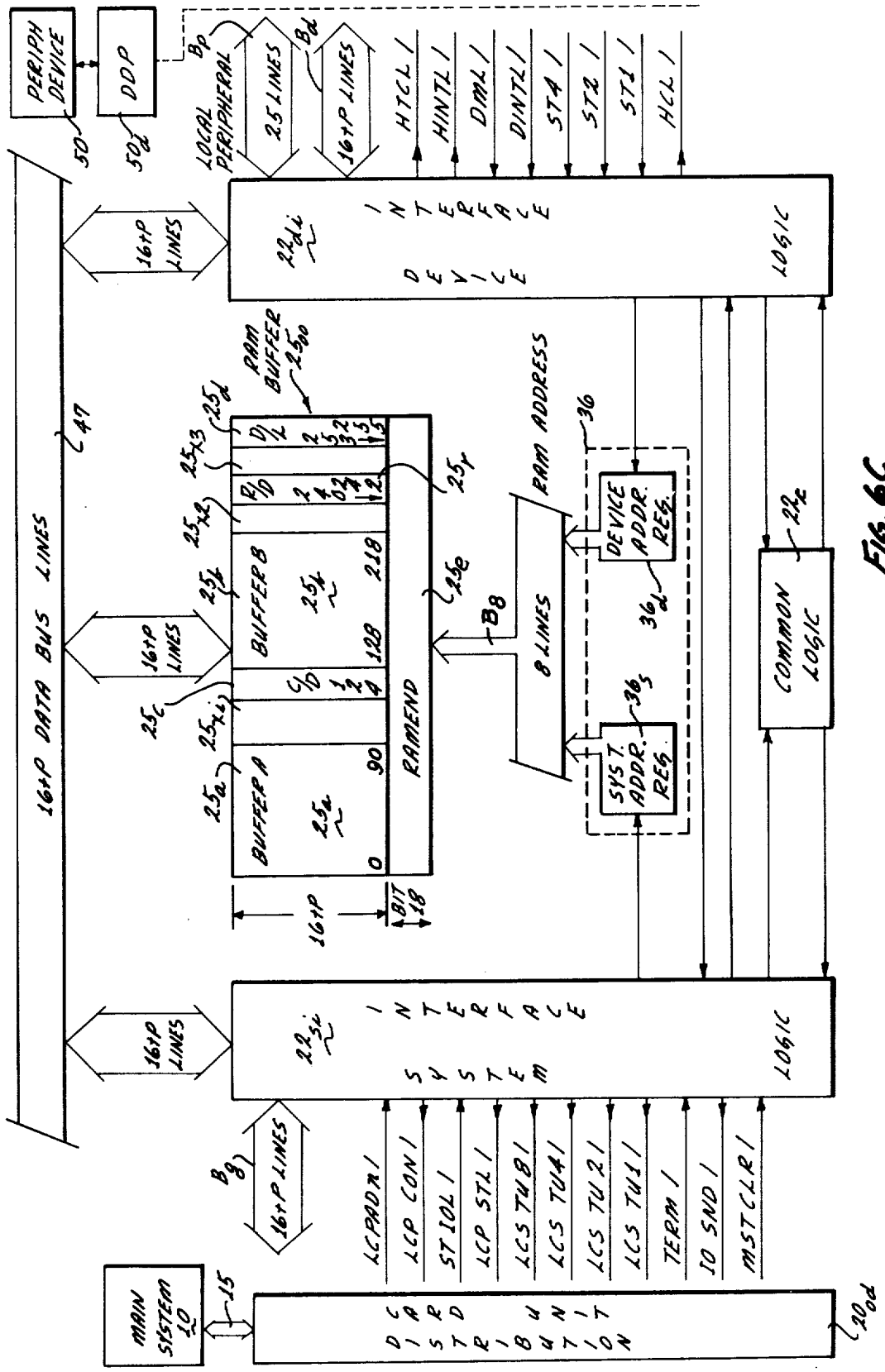

PERIPH DEVICE
DDP
LOCAL PERIPHERAL
25 LINES
16+P LINES
HTCL/
HINTL/
DML/
DINTL/
ST4/
ST2/
ST1/
HCL/
INTERFACE DEVICE
LOGIC
16+P DATA BUS LINES
RAM BUFFER
BUFFER B
BUFFER A
RAMEND
RAM ADDRESS
8 LINES
DEVICE ADDR. REG.
SYST ADDR. REG.
COMMON LOGIC
INTERFACE SYSTEM
LOGIC
LCPADR/
LCP CON/
STIOL/
LCP STL/
LCS TU8/
LCS TU4/
LCS TU2/
LCS TU1/
TERM/
IO SND/
MSTCLR/
MAIN SYSTEM IO
DISTRIBUTION

FIG. 6C

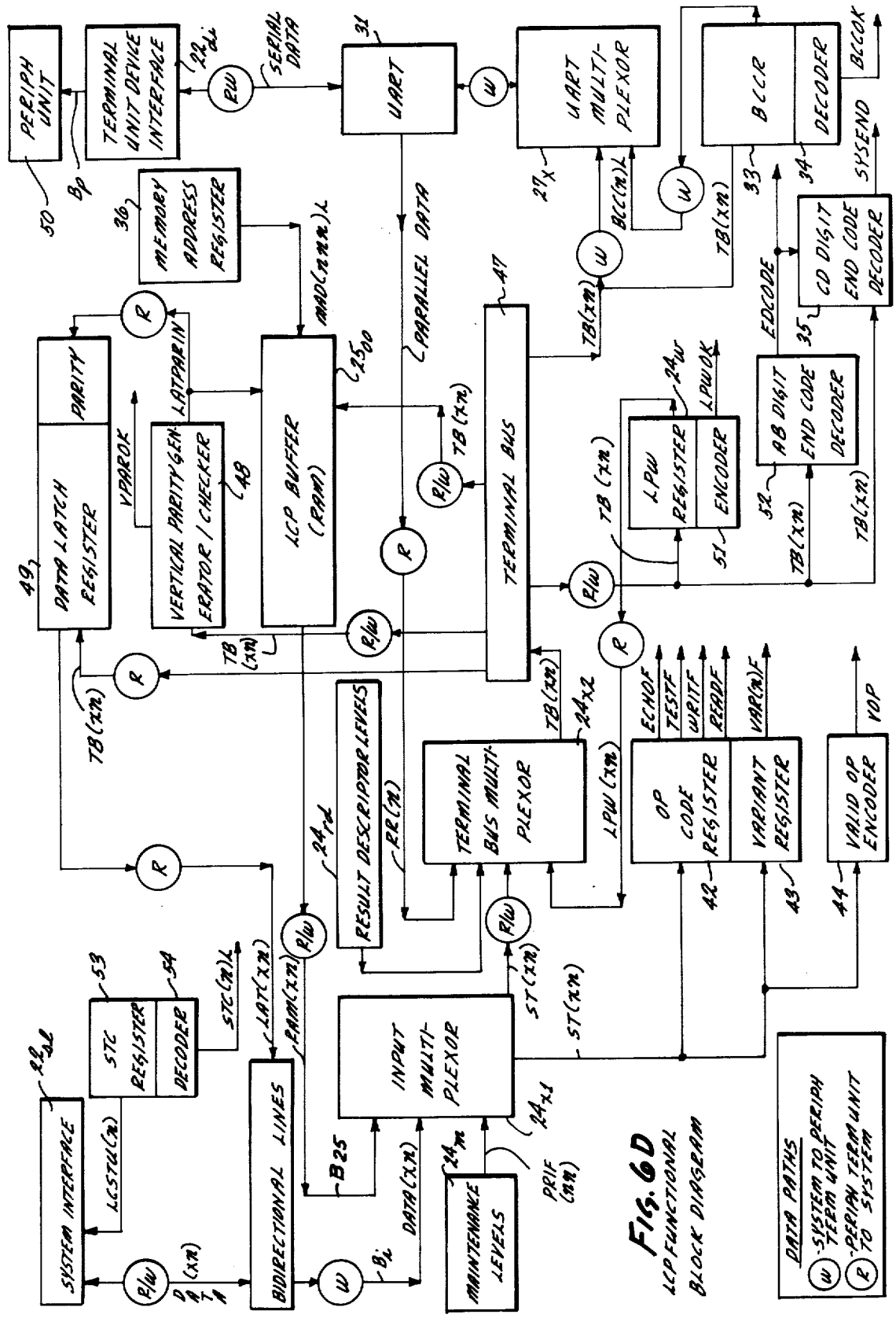


FIG. 6D
LCP FUNCTIONAL BLOCK DIAGRAM

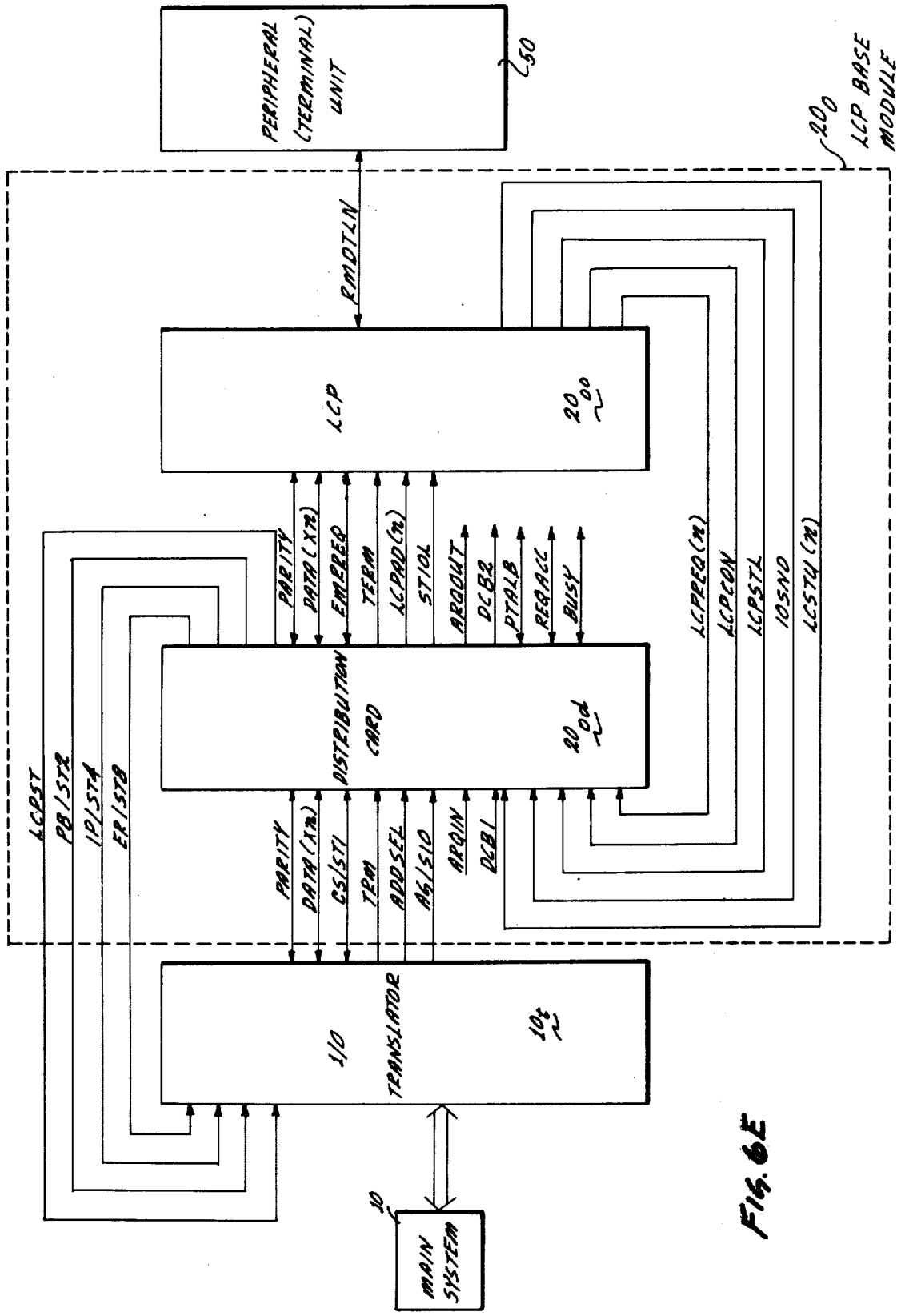
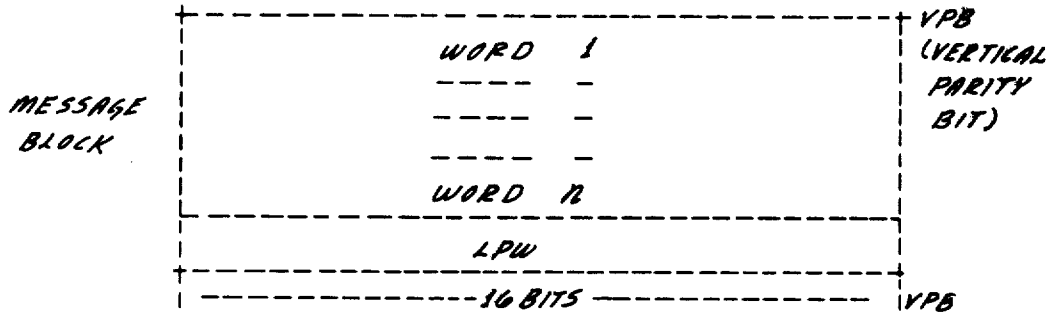

PERIPHERAL (TERMINAL) UNIT
50
20₀ LCP BASE MODULE
RMDTLN
LCP
20₀₀
PARITY
DATA (Xn)
EMERGEQ
TERM
LCPAD(n)
STIOL
ARQOUT
DCB2
PTALB
REARAC
BUSY
LCPREQ(n)
LCPCON
LCPSTL
IOSND
LCSTU(n)
DISTRIBUTION CARD
20₀₀
LCPST
PB/STR
IP/STA
ER/STB
PARITY
DATA (Xn)
CS/STI
TERM
ADDSEL
AG/SIO
ARQIN
DCB1
I/O TRANSLATOR
10₂
10
MAIN SYSTEM

FIG. 6E



n = OR LESS THAN 90 FOR DATA
n = OR LESS THAN 9 FOR R/D
n = 3 FOR C/D
n = 1 FOR C/M

A WORD IS COMPOSED OF FOUR (4) DIGITS A, B, C, AND D PLUS A PARITY BIT, OR A TOTAL OF 17 BITS. A DIGIT IS COMPOSED OF FOUR (4) BITS 8, 4, 2, AND 1.

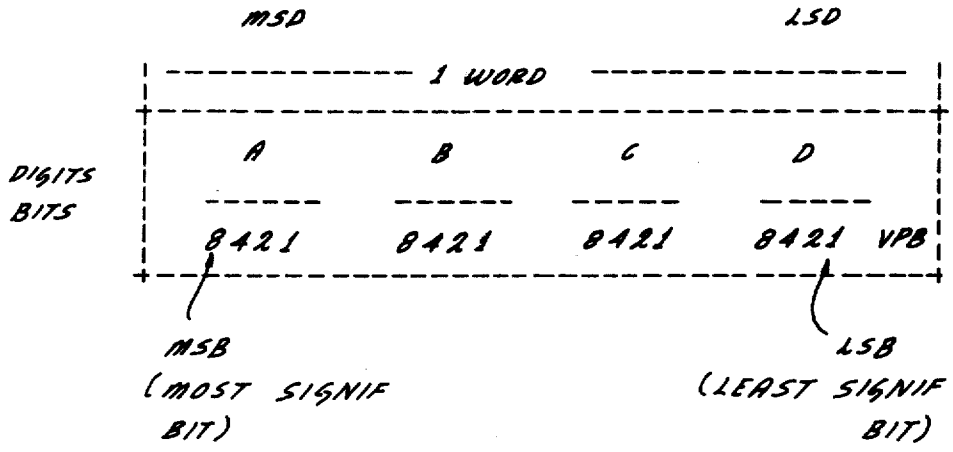


FIG. 6F

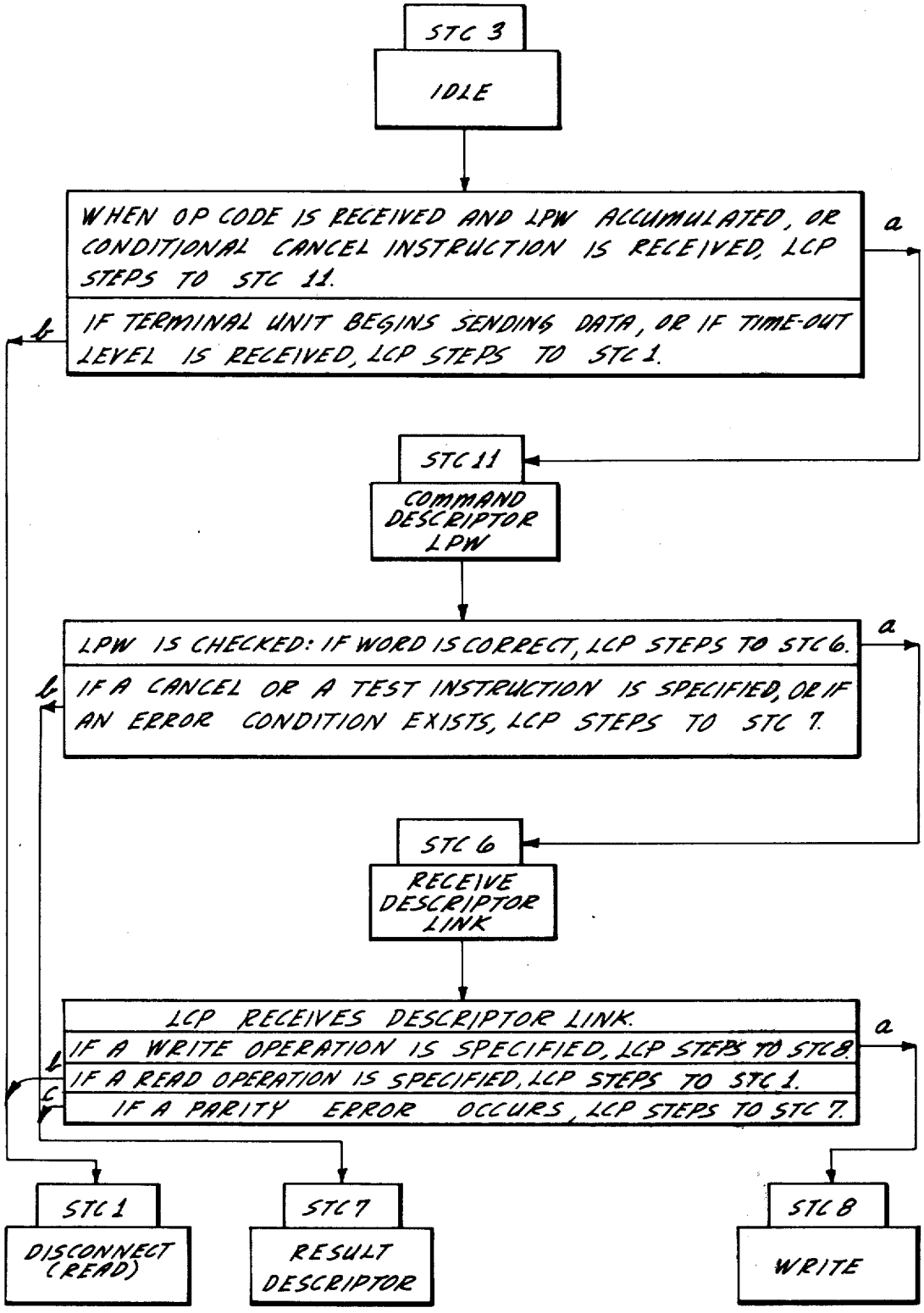


FIG. 7A SIMPLIFIED RECEIPT-OF-INSTRUCTIONS FLOW

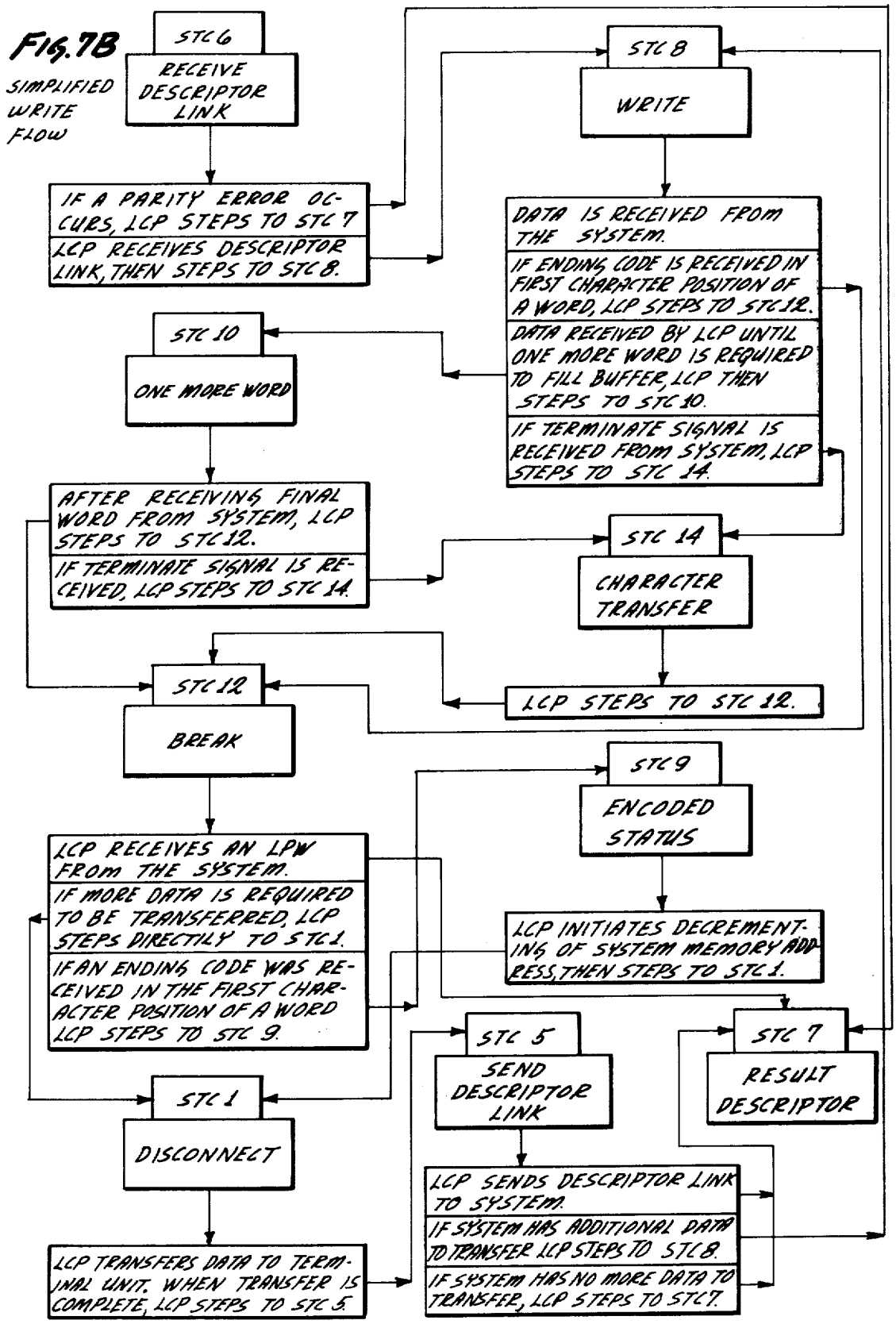


FIG. 7B
SIMPLIFIED WRITE FLOW
STC 6
RECEIVE DESCRIPTOR LINK
IF A PARITY ERROR OCCURS, LCP STEPS TO STC 7
LCP RECEIVES DESCRIPTOR LINK, THEN STEPS TO STC 8.
STC 8
WRITE
DATA IS RECEIVED FROM THE SYSTEM.
IF ENDING CODE IS RECEIVED IN FIRST CHARACTER POSITION OF A WORD, LCP STEPS TO STC 12.
DATA RECEIVED BY LCP UNTIL ONE MORE WORD IS REQUIRED TO FILL BUFFER, LCP THEN STEPS TO STC 10.
IF TERMINATE SIGNAL IS RECEIVED FROM SYSTEM, LCP STEPS TO STC 14.
STC 10
ONE MORE WORD
AFTER RECEIVING FINAL WORD FROM SYSTEM, LCP STEPS TO STC 12.
IF TERMINATE SIGNAL IS RECEIVED, LCP STEPS TO STC 14.
STC 14
CHARACTER TRANSFER
LCP STEPS TO STC 12.
STC 12
BREAK
LCP RECEIVES AN LPW FROM THE SYSTEM.
IF MORE DATA IS REQUIRED TO BE TRANSFERRED, LCP STEPS DIRECTLY TO STC 1.
IF AN ENDING CODE WAS RECEIVED IN THE FIRST CHARACTER POSITION OF A WORD LCP STEPS TO STC 9.
STC 9
ENCODED STATUS
LCP INITIATES DECREMENTING OF SYSTEM MEMORY ADDRESS, THEN STEPS TO STC 1.
STC 1
DISCONNECT
LCP TRANSFERS DATA TO TERMINAL UNIT. WHEN TRANSFER IS COMPLETE, LCP STEPS TO STC 5.
STC 5
SEND DESCRIPTOR LINK
LCP SENDS DESCRIPTOR LINK TO SYSTEM.
IF SYSTEM HAS ADDITIONAL DATA TO TRANSFER LCP STEPS TO STC 8.
IF SYSTEM HAS NO MORE DATA TO TRANSFER, LCP STEPS TO STC 7.
STC 7
RESULT DESCRIPTOR

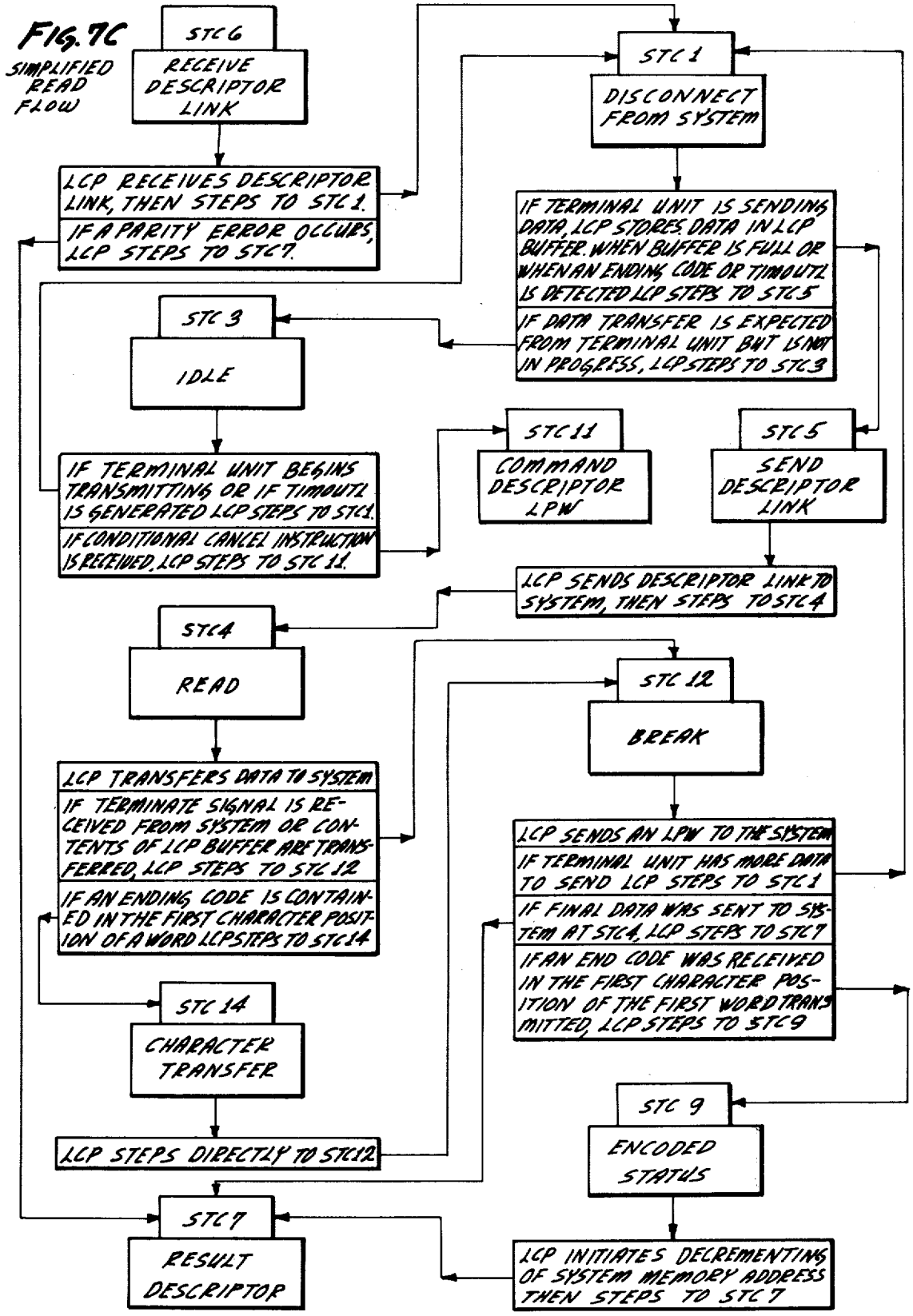


FIG. 7C
SIMPLIFIED READ FLOW
STC 6
RECEIVE DESCRIPTOR LINK
LCP RECEIVES DESCRIPTOR LINK, THEN STEPS TO STC 1.
IF A PARITY ERROR OCCURS, LCP STEPS TO STC 7.
STC 1
DISCONNECT FROM SYSTEM
IF TERMINAL UNIT IS SENDING DATA, LCP STORES DATA IN LCP BUFFER. WHEN BUFFER IS FULL OR WHEN AN ENDING CODE OR TIMOUT 2 IS DETECTED LCP STEPS TO STC 5
IF DATA TRANSFER IS EXPECTED FROM TERMINAL UNIT BUT IS NOT IN PROGRESS, LCP STEPS TO STC 3
STC 3
IDLE
IF TERMINAL UNIT BEGINS TRANSMITTING OR IF TIMOUT 2 IS GENERATED LCP STEPS TO STC 1.
IF CONDITIONAL CANCEL INSTRUCTION IS RECEIVED, LCP STEPS TO STC 11.
STC 11
COMMAND DESCRIPTOR LPW
STC 5
SEND DESCRIPTOR LINK
LCP SENDS DESCRIPTOR LINK TO SYSTEM, THEN STEPS TO STC 4
STC 4
READ
LCP TRANSFERS DATA TO SYSTEM
IF TERMINATE SIGNAL IS RECEIVED FROM SYSTEM OR CONTENTS OF LCP BUFFER ARE TRANSFERRED, LCP STEPS TO STC 12
IF AN ENDING CODE IS CONTAINED IN THE FIRST CHARACTER POSITION OF A WORD LCP STEPS TO STC 14
STC 12
BREAK
LCP SENDS AN LPW TO THE SYSTEM
IF TERMINAL UNIT HAS MORE DATA TO SEND LCP STEPS TO STC 1
IF FINAL DATA WAS SENT TO SYSTEM AT STC 4, LCP STEPS TO STC 7
IF AN END CODE WAS RECEIVED IN THE FIRST CHARACTER POSITION OF THE FIRST WORD TRANSMITTED, LCP STEPS TO STC 9
STC 14
CHARACTER TRANSFER
LCP STEPS DIRECTLY TO STC 12
STC 9
ENCODED STATUS
STC 7
RESULT DESCRIPTOR
LCP INITIATES DECREMENTING OF SYSTEM MEMORY ADDRESS THEN STEPS TO STC 7

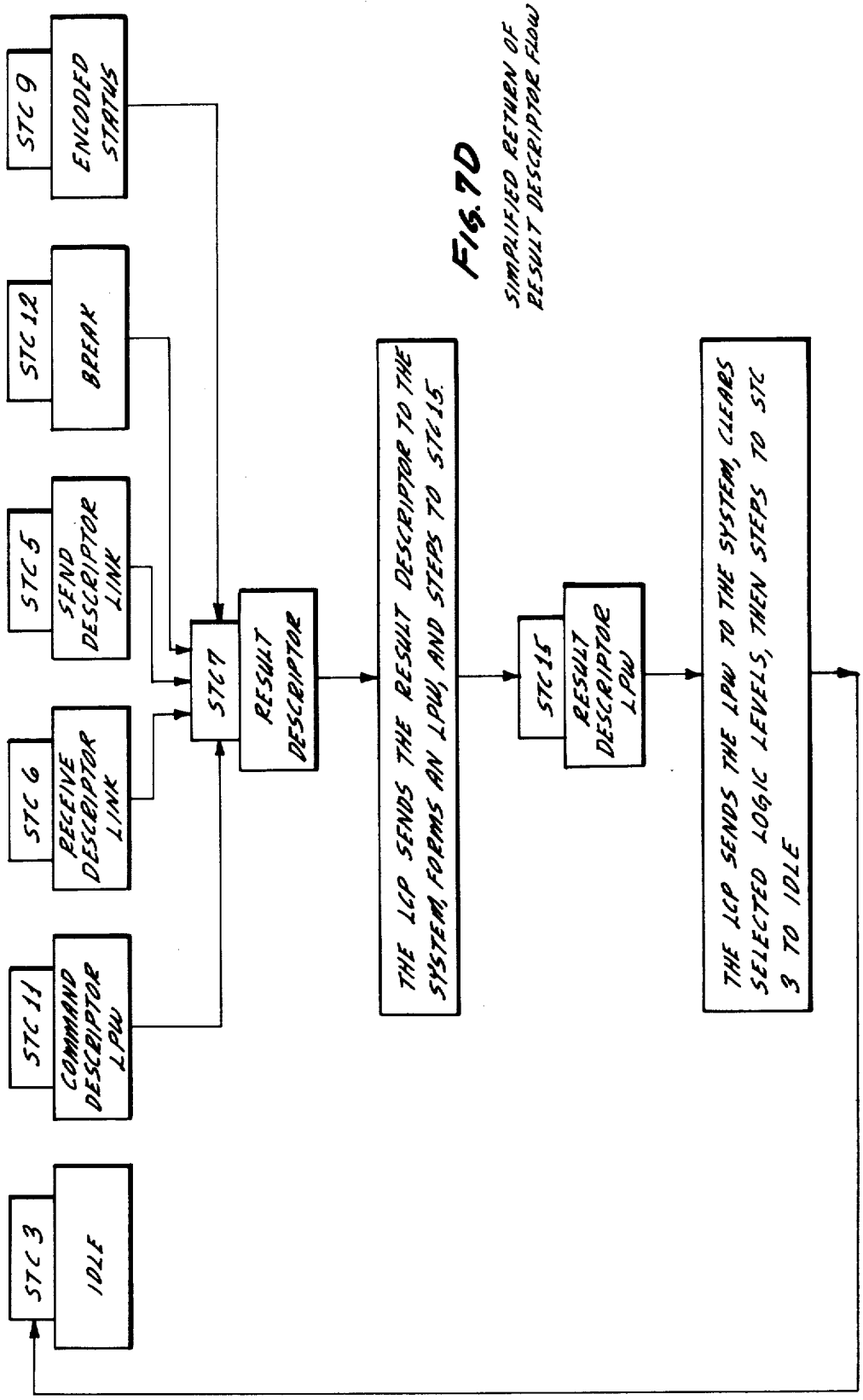


FIG. 7D
SIMPLIFIED RETURN OF RESULT DESCRIPTOR FLOW

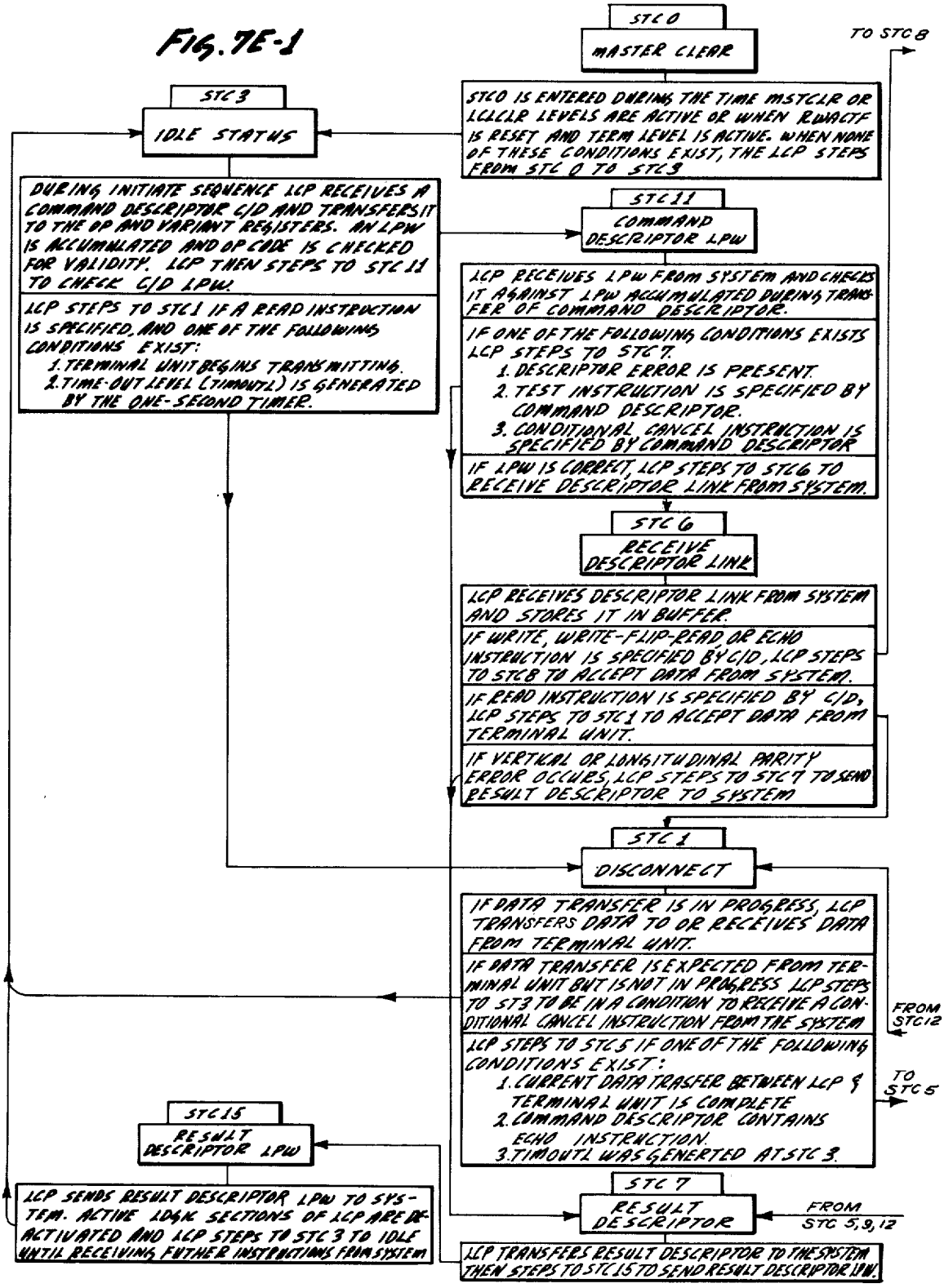


FIG. 7E-1
STC 0
MASTER CLEAR
TO STC 8
STC0 IS ENTERED DURING THE TIME MSTCLR OR LCLCLR LEVELS ARE ACTIVE OR WHEN RWACTF IS RESET AND TERM LEVEL IS ACTIVE. WHEN NONE OF THESE CONDITIONS EXIST, THE LCP STEPS FROM STC 0 TO STC 3
STC 3
IDLE STATUS
DURING INITIATE SEQUENCE LCP RECEIVES A COMMAND DESCRIPTOR C/D AND TRANSFERS IT TO THE OP AND VARIANT REGISTERS. AN LPW IS ACCUMULATED AND OP CODE IS CHECKED FOR VALIDITY. LCP THEN STEPS TO STC 11 TO CHECK C/D LPW.
LCP STEPS TO STC 1 IF A READ INSTRUCTION IS SPECIFIED, AND ONE OF THE FOLLOWING CONDITIONS EXIST:
1. TERMINAL UNIT BEGINS TRANSMITTING.
2. TIME-OUT LEVEL (TIMOUTL) IS GENERATED BY THE ONE-SECOND TIMER.
STC 11
COMMAND DESCRIPTOR LPW
LCP RECEIVES LPW FROM SYSTEM AND CHECKS IT AGAINST LPW ACCUMULATED DURING TRANSFER OF COMMAND DESCRIPTOR.
IF ONE OF THE FOLLOWING CONDITIONS EXISTS LCP STEPS TO STC 7.
1. DESCRIPTOR ERROR IS PRESENT.
2. TEST INSTRUCTION IS SPECIFIED BY COMMAND DESCRIPTOR.
3. CONDITIONAL CANCEL INSTRUCTION IS SPECIFIED BY COMMAND DESCRIPTOR
IF LPW IS CORRECT, LCP STEPS TO STC 6 TO RECEIVE DESCRIPTOR LINK FROM SYSTEM.
STC 6
RECEIVE DESCRIPTOR LINK
LCP RECEIVES DESCRIPTOR LINK FROM SYSTEM AND STORES IT IN BUFFER.
IF WRITE, WRITE-FLIP-READ OR ECHO INSTRUCTION IS SPECIFIED BY C/D, LCP STEPS TO STC 8 TO ACCEPT DATA FROM SYSTEM.
IF READ INSTRUCTION IS SPECIFIED BY C/D, LCP STEPS TO STC 1 TO ACCEPT DATA FROM TERMINAL UNIT.
IF VERTICAL OR LONGITUDINAL PARITY ERROR OCCURS, LCP STEPS TO STC 7 TO SEND RESULT DESCRIPTOR TO SYSTEM
STC 1
DISCONNECT
IF DATA TRANSFER IS IN PROGRESS, LCP TRANSFERS DATA TO OR RECEIVES DATA FROM TERMINAL UNIT.
IF DATA TRANSFER IS EXPECTED FROM TERMINAL UNIT BUT IS NOT IN PROGRESS LCP STEPS TO ST3 TO BE IN A CONDITION TO RECEIVE A CONDITIONAL CANCEL INSTRUCTION FROM THE SYSTEM
FROM STC12
LCP STEPS TO STC 5 IF ONE OF THE FOLLOWING CONDITIONS EXIST:
1. CURRENT DATA TRASFER BETWEEN LCP & TERMINAL UNIT IS COMPLETE
2. COMMAND DESCRIPTOR CONTAINS ECHO INSTRUCTION.
3. TIMOUTL WAS GENERTED AT STC 3.
TO STC 5
STC 15
RESULT DESCRIPTOR LPW
LCP SENDS RESULT DESCRIPTOR LPW TO SYSTEM. ACTIVE LOGIC SECTIONS OF LCP ARE DEACTIVATED AND LCP STEPS TO STC 3 TO IDLE UNTIL RECEIVING FUTHER INSTRUCTIONS FROM SYSTEM
STC 7
RESULT DESCRIPTOR
FROM STC 5, 9, 12
LCP TRANSFERS RESULT DESCRIPTOR TO THE SYSTEM THEN STEPS TO STC 15 TO SEND RESULT DESCRIPTOR LPW.

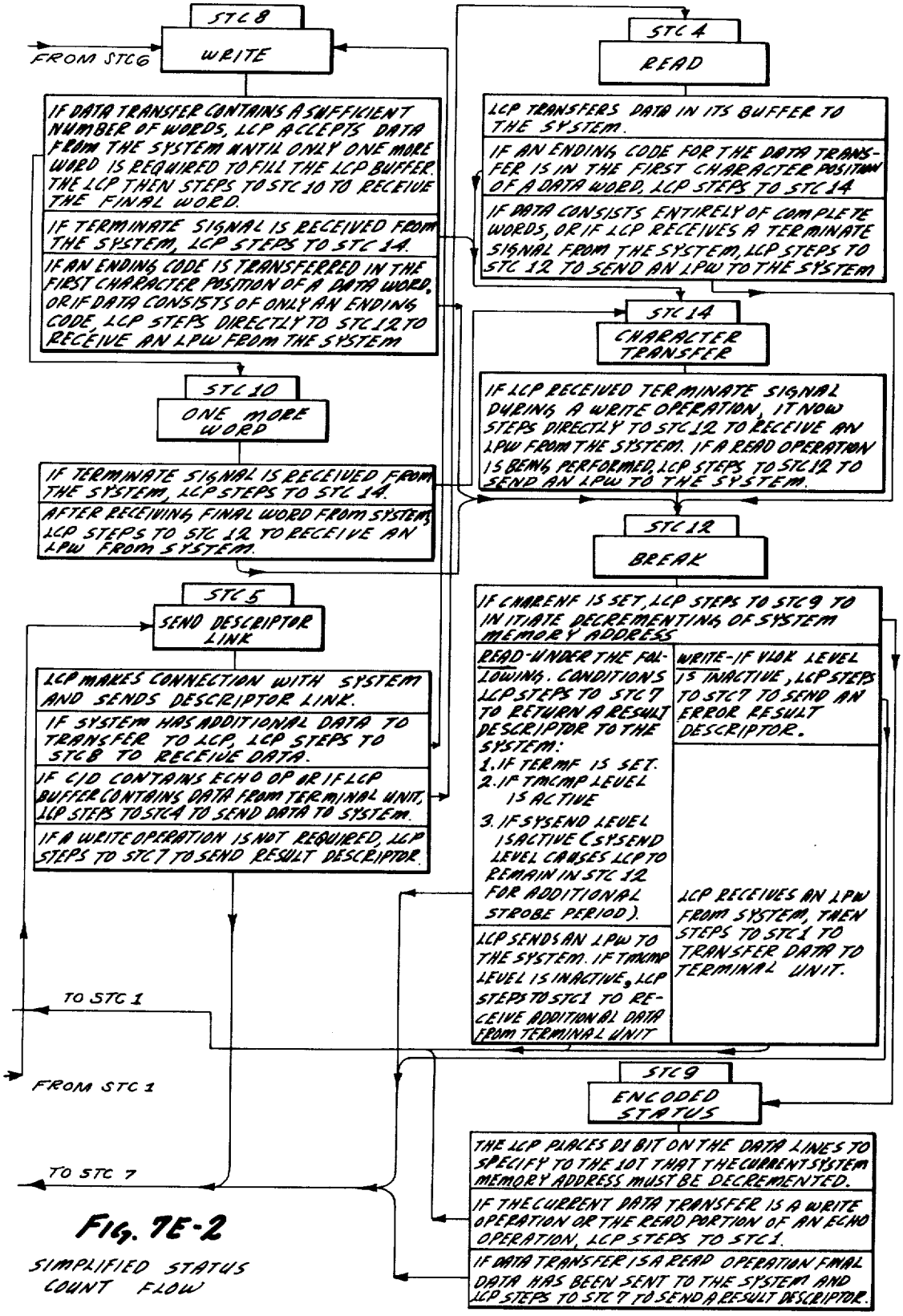


FIG. 7E-2

SIMPLIFIED STATUS COUNT FLOW

MODULAR BLOCK UNIT FOR INPUT-OUTPUT SUBSYSTEM

CROSS REFERENCE TO RELATED APPLICATIONS

The following commonly assigned, concurrently filed patent applications are related to the subject matter of this application:

Ser. No. 728,458, filed Sept. 30, 1976, for Intelligent Input-Output Interface Control Unit for Input-Output Subsystem.

Ser. No. 728,455, filed Sept. 30, 1976, for Interface System Providing Interfaces to Central Processing Unit and Modular Processor-Controllers for an Input-Output Subsystem.

Ser. No. 728,457, filed Sept. 30, 1976, for Input-Output Subsystem for Digital Data Processing System.

FIELD OF THE INVENTION

This invention relates to digital computing and/or data processing systems and is concerned with the means and methods of controlling the transfer of data between a variety of different peripheral devices and the main memory of a central processing unit or main system. Basically the system involves taking the load off of the processing unit and distributing it among a variety of Intelligent I/O Interface units which can work independently of the central processor in handling data transfer operations.

The invention involved herein describes an organized device, designated as a Base Module, which houses and supports a plurality of Intelligent I/O Interface units within an integrated modular structure. This sub-unit or Base Module comprises a basic building block which can be used as a plurality of modular units to build up a system around a central processing unit for handling a large number of peripheral devices independently of intervention by the central processor.

BACKGROUND OF THE INVENTION

The general configuration of a data processing system typically comprises a processor or processors, a main memory, and a plurality of various types of peripheral devices or terminals (sometimes called I/O units), which more specifically may be card readers, magnetic tape units, card punches, printers, disk files, supervisory terminals, and so on. The optimum systems generally involve the configuration wherein the peripheral devices are handled by independent interface control units so that the processor is free to access and process data contained in the main memory. In configurations having separate control means for the peripheral input-output devices, it is possible to have parallel or concurrent processing occurring at the same time that input-output (I/O) operations occur. These concurrent processing I/O operations occur within the same program which operates through one of the processors, and which also initiates all input-output operations. In addition the program must have some means of determining when the I/O operations are inactive or have been completed.

As an example, if a program calls for a file of data to be loaded into the main memory, it must be able to determine when that operation has been completed before it can go ahead to make use of the data. Thus, an input-output operation is initiated or started by the program, as by some type of "initiate instruction" which provides, typically, an address pointing to an "I/O descriptor" which is stored in the main memory. This descriptor identifies the peripheral device from which data is to be received and/or transmitted, it identifies the type of operation such as a "Read" or a "Write", and also identifies the field of main memory locations to be used in the input-output operation. Generally this I/O descriptor is transferred to a control means (I/O control means) to control the transfer of data between the peripheral terminal device and the main memory.

When the input-output operation is "complete", such as by the transfer of the data from the peripheral unit to the main memory to load the main memory, then there is a need for some type of a completion statement, which is typically referred to as a "Result Descriptor". Usually this is transferred from the I/O control means to some specific location in main memory known to the program being used. Typically, the Result Descriptor includes information identifying the particular peripheral terminal device and further includes information as to the result of or the status of that particular input-output operation — thus, to provide information as to whether the transfer was complete and correct, or whether any exception conditions occurred or whether any errors occurred or any other peculiar situations arose in regard to the transaction involving that particular peripheral terminal device.

Thus, when a program initiates an input-output operation, the program must have some means to determine when the input-output operation has been completed. A standard technique in this respect is for the program to have instructions to interrogate the Result Descriptors periodically, to determine when and if a particular input-output operation has been completed. However, it is much simpler if the input-output control means indicates when the transfer operation is finished. In accomplishing this, it is usually necessary to interrupt whatever operation the processor has underway, and force it to examine the Result Descriptors and to take appropriate action. This stopping or interruption of the processor's activities is generally designated as an "Interrupt".

Thus, when an interrupt occurs, the processor must stop the program it is working on, it must make a fixed notation of what point in the program execution it was interrupted and it must then store the contents of certain registers and control flip-flops so it can have information as to where it should return in the program after the completion of the interrupt cycle; and then the processor must transfer its attention and operation to the program designed to handle and service the Interrupt condition.

Certain systems such as the system described herein, have a program for servicing "Interrupt" conditions, which program is sometimes referred to as MCP or a master control program. This program must keep a record of current input-output operations and associate the particular Interrupt with the particular input-output operation that caused it. Then it must analyze the results of this Interrupt cycle to see if any unusual circumstances or exceptions occurred or if an error condition was reported, so that corrective and appropriate action may be taken. The Interrupt program must take the results of the input-output operation and make them available to the program that initiated the input-output operation and then further determine if other input-output operations are waiting to be initiated and, if so, to

ake action to initiate other needful input-output operations.

In many of the prior and present system configurations, many calls or request for memory access would ome in to get memory service, but because of the limed bandpass and time available for various peripheral nits, many I/O transfers would be incomplete and ause "access errors".

Also many of the prior art system configurations rovided only one or two communication paths or hannels to a multitude number of peripheral terminal nits so that I/O transfers of a particular peripheral erminal unit had to wait their turn in sharing access and se of a communications bus. This introduced congeson and delay into the system. It also made difficulties systems involving multi-programming since efforts re made to match a job having heavy input-output equirements with another job that is "processoround" and which has only limited input-output reuirements.

Many of the present day data processing systems ave a single communication path or a limited number f communication paths between the central processing nit and the peripheral units. Generally within the comunication path there is one or more "input-output ontrol" means. When an input-output path is requested y a processor, the path will only generally become vailable when: the peripheral unit is not initiating a ansfer operation; the peripheral unit is not busy in a ansfer or other operation with the input-output conol means; the peripheral unit or its input-output conol means is not busy with other operations.

The data-transfer rate of the input-output control eans is, of course, a limiting factor in the operation of e system since the often slow transfer rate of certain eripheral units (which are passed through the inpututput control means) will unnecessarily tie up the proessor and memory activity to the low speed of the eripheral terminal unit.

Thus, many data processing systems have come to be ovided with a plurality of input-output control means hich include buffers, to permit a particular peripheral group of peripherals to communicate with the main stem. When there are a plurality of input-output conol means (through which pass the communication annels to individual peripheral units or groups of such its) some prior art systems have used the method of erating the data transfer operation in a sequential shion so that the various input-output control means ke turns in serving the peripherals which are associed with them.

A difficulty arises here in that certain peripheral units d their associated input-output control means are sier than others, and certain of the channels involved tually need more communications-time than they are tting. A "channel" may be looked at as a communicaon path between the main system, through the inpututput control means, over to the peripheral unit. Thus, ere can occur situations where certain channels are hort changed" to the extent that a great number of ccess errors" will be developed. Access errors inolve the situation where the data bytes being transrred through the input-output control means do not mprise complete message units but consist only of n-usable fractions of message units. As a result of this, e central processing unit would not be getting or ansferring useful information and would have to beme fixated on continually requesting the same input-output operations over and over again. Thus, when the peripheral units are placed in a situation where they are unable to send or receive an entire message unit or record, then the likelihood of access errors occurs which leads to uncompleted cycles in regard to a particular channel and no successful completion of transfer of the required informational data.

It is desired that the maximum transfer of data occur through the mentioned plurality of input-output control means, and without such access errors which lead to incomplete cycles of data transfer (which are unusable, and the time period of which is wasted and of no use, thus tying up valuable processor time).

Thus, in such a system configuration, problems arise in regard to how much time should be allocated to each of the individual channels for data transfer operations and the further problem of which channels should be given priority status over the other channels.

Now, in data processing systems where multitudes of peripheral units are involved (many of which are at differently located installation sites) it is necessary to have groupings of input-output control means to handle the variety of peripheral units at each given site. Thus, the priority problems involve not only the priority to be given as to the competition among peripheral units at one local given site, but also involve the priority problems of priority allocation as between the different locational sites, each of which have their own input-output control means.

SUMMARY OF THE INVENTION

The present invention involves a digital data processing system for the control and handling of input-output operations (data transfers) as between a plurality of various types of peripheral units and a central Main System (Processor and Main Memory). Two types of I/O Subsystems are provided for communication to the central Main System.

One I/O Subsystem is a system wherein a type of intelligent interface control unit, designated as a "Line Control Processor" (LCP), is used, and wherein each LCP, while performing the same basic functions, is specifically oriented to control and handle data transfers to and from a specific type of peripheral terminal unit. For example, a basic LCP would be adapted for each specific instance to handle a card reader, a disk unit, a train printer, or other special type of peripheral unit. The LCP's are placed in groups, typically, of eight units, to form an LCP Base Module. Then each of the Base Modules are grouped in a set of three to form an LCP Cabinet Unit. A plurality of such LCP Cabinet Units may be used to constitute the first I/O Subsystem.

Another I/O Subsystem is provided for those types of peripheral terminal units for which no specific Line Control Processors (LCP) have been developed. This second I/O Subsystem is organized so that a Central Control unit is provided to control the pathing from the Central Processor and Main Memory to selected input-output channels which provide a data path to individual peripheral units. These individual channels will each have their own memory buffer and connect through the Central Control unit over to the Main System.

In the I/O Subsystem using the Line Control Processors, the Main System (of Processor and Main Memory) is also provided with a unit called an Input-Output Translator unit (IOT) which becomes part of the Main System and provides an interface between the Main System and another distribution-control interface desig-

nated as "Distribution Card Unit" which handles a Base Module, (a group of Line Control Processors) and which connects a selected individual Line Control Processor into the LCP I/O Subsystem.

The Line Control Processor's (LCP's) are organized in groups of eight called the LCP Base Module each of which has a single "Distribution Card Unit" which provides the interface between the Input-Output Translator, IOT, of the Main System and the eight LCP's of any given Base Module. Each Base Module also carries a Maintenance Card unit which can provide all maintenance and checking functions for the group of eight LCP's of the Base Module. Each Base Module is also provided with one common "Termination Card Unit" which provides common clocking functions for all the LCP's of the group and also provides proper terminations for the transmission lines which connect the various LCP's, the Distribution Card, and the Maintenance Card of that particular Base Module.

The IOT of the Main System works in a unique relationship with the Distribution Card Unit of the Base Module of the LCP's in the LCP I/O Subsystem, serving to setup data-transfers between the peripheral units and the Main Memory in a fashion that does not burden the Central Processor and which permits concurrent data-transfer operations to occur between any number of peripheral units and the Main Memory. This is facilitated by the use of a record-length buffer memory in each LCP. The data-transfer cycle is accomplished in using complete data message-blocks which thus prevent "access errors" from occurring.

The embodiment of the invention described herein provides a system which helps alleviate certain problems inherent in prior art systems. By providing a separate channel from the Main System to each peripheral unit, there is no need for data transfers (between a particular peripheral unit and the Main System) to have to wait for the use of a shared communication channel, since each individual peripheral unit is provided with its own channel, and thus each of the plurality of peripheral units can simultaneously and concurrently consummate input operations without any further requirements from the processor or without interference to processor operations. The input-output data-transfer control means in the Subsystem is provided by an individual "Line Control Processor" (LCP) for each peripheral unit. The "Line Control Processors" accept input-output commands from the Main Memory (via the I/O Translator unit) and they execute these commands independently of the main processor, so that input-output control operations are performed in parallel with and asynchronously with processing.

A memory control unit 10c, FIG. 1A, regulates the flow of data between the Main Memory, the Central Processor and the I/O Subsystem. It allows each of the system components to have access to Main Memory on a priority basis, giving the highest priority to the I/O Subsystem. Since the memory control operates independently of the Processor, the Processor is free to perform memory-independent functions at the same time that memory accesses are being granted to the I/O Subsystem.

The Line Control Processors are each provided with memory buffers which can store an entire message-block or record-length of data. Thus, data transfers between Main Memory and the Line Control Processor can take place at high speed and constitute a complete message-block in itself. Since a complete message-block of data is transferred in any given cycle, the problem of access errors is eliminated so that no further memory cycle time is required to complete "incomplete former data transfer cycles", which might occur absent the record-length buffer.

The Line Control Processors are functionally the same except that minor variations may occur so that they are adaptable to work with different types of peripheral terminal units, and, as such, the LCP's are "transparent" to the Main System.

In certain cases, there are peripheral units and data storage devices involved for which no specific Line Control Processors (LCP) have been developed. In this case, there is used another input-output control subsystem which can operate in parallel with the first I/O Subsystem and its Line Control Processors (LCP).

The main or central system of the described embodiment, which may include the Processor, the Main Memory, and the Memory Control, is furnished with a unit called an Input-Output Translator or IOT. The IOT is a special portion of the Processor which, upon receipt of an I/O descriptor from memory, works in conjunction with the LCP Base Module to establish connection to the particular LCP in the channel specified by the "Initiate I/O" instruction from the program. The IOT translates the I/O descriptor into a form (Command Descriptor) recognizable to the LCP (Line Control Processor), and when connection is established, passes the translated descriptor over to the LCP after which the data transmission may begin. During the time that data is being transferred between LCP and the Main System, the IOT, upon demand by the LCP, requests memory accesses, addresses memory, then modifies and compares the data addresses. The IOT controls the routing of data between the selected LCP and the Main System, and it performs translation (ASCII/EBCDIC) of the data if so required. ASCII/EBCDIC refers to American Standard Code for Information Interchange/Extended Binary Coded Decimal Interchange Code. Upon completion of an operation, the IOT accepts Result Descriptor (R/D) information from an LCP and then stores the Result Descriptor (R/D) in a predetermined location.

The Line Control Processor, (LCP), is a device which upon receipt of a Command Descriptor (C/D) from the Main System via the IOT establishes a communication path to a selected peripheral unit. Once this path is established, the LCP accepts data from or passes data to, the peripheral device. Since each LCP has a "data buffer" (typically 256 words), then data can be transferred to and from the peripheral device at the comparatively low speed rate of the peripheral device; then, when the buffer is full, the data can be transferred to the central Main System at the highest rate permitted by speed of the Main Memory. Thus, a unique interworking relationship exists between the IOT (Input-Output Translator) of the Main System and the LCP which is the interface control between the peripheral units and the Main System. Further, a unique working relationship exists between each LCP and the Distribution Card Unit of its Base Module, which interfaces a given LCP to the IOT of the Main System. The Distribution Unit not only provides for interconnection of the Main System to a selected LCP but also regulates priorities among LCP's for access to Main Memory.

The invention particularly claimed within the I/O Subsystem described herein, is a modular interface unit designated as Base Module and which permits grouping

of Line Control Processors via a local distribution-control interface designated as the Distribution Card Unit.

Some of the major objectives of the Line Control Processor I/O Subsystem may be summarized as follows:

To relieve the Central Processing Unit from getting involved in monitoring and controlling data transfers between the System's Main Memory and a large number of peripheral units.

To increase the rate of data transfer between a variety of different peripheral units all connected to the Main System having a Main Memory and Processor. This includes transfers from the Main Memory to any individual peripheral in this system and vice versa.

To provide an Intelligent I/O Interface control unit (Line Control Processor) which will relieve the Central Processor of many burdens and which will be responsive to the needs of various peripheral units for access to the Main Memory.

To provide an Intelligent Interface I/O control unit which can receive an I/O instruction from the Central Processing unit and then independently continue in regard to controlling, monitoring, and executing this instruction so as to accomplish data transfer between Main System Memory and any specifically desired peripheral unit. This is done asynchronously as the needs and the requirements arise. The interface unit (LCP) also handles the error-checking of all word and message block transmissions in addition to keeping the Main System informed of the status of any data-transfer cycle, as to its completeness, incompleteness, error-status. The Line Control Processor also monitors requests from a peripheral unit for access to Main Memory and informs the Main System of "busyness" of the peripheral unit or its unavailability.

To permit easy modular system expansion. The I/O Subsystem of the Central Processing unit servicing a plurality of terminal units is setup such that the interface units (Line Control Processor) are organized in Base Modules in groups of eight units. Each module has a Distribution Card Unit which interfaces the group of eight Line Control Processors to the Main System via the IOT of the Main System. The Distribution Unit can thus set priorities as between any one of the eight Line Control Processors in the Base Module. Further, when a plurality of Base Modules occur in the System then the Distribution Unit of each Base Module can be given a priority ranking (designated global priority) as between the priority rank granted to any given Base Module, within the full set of Base Modules. Thus, another object of the I/O Subsystem involved is to provide arrangements for setting up Global Priority (priority as between Base Modules in the System) and also Local Priority (Priority as to precedence status of each Line Control Processor in the group of eight Line Control Processors in the Base Module).

To eliminate "access errors" so that all the data required at any given time for the Main System (that is, a message length block of data) is always transmitted and error-checked in one complete cycle without interruption (except under emergency conditions).

To permit the rapid completion of a data-transfer operation as between the System's Main Memory and a given peripheral unit, without interruption or incomplete data-transfer, once a communication channel is established (except for certain emergencies).

To provide the Main System with the current status of any Line Control Processor at all times and the results (complete, incomplete or in error) of any given data-transfer cycle.

To provide modular building blocks for facilitating the expansion of the System by increasing the number of peripheral devices that can be included in the System in a simple economical fashion.

To provide an I/O Subsystem whereby the Central Processor is relieved of executing I/O data-transfer cycles and this work load is distributed throughout the system via I/O control units grouped in modular block units (Line Control Processors).

BRIEF DESCRIPTION OF THE DRAWINGS

The I/O Subsystem described herein and the operative components involved will be better understood with reference to the following drawings of which:

FIG. 1A is a schematic of a Central Data Processing System having two different types of I/O Subsystems; the two I/O Subsystems are designated as (a) the Central Control Subsystem (CC) with Input-Output Controllers (IOC) and (b) the Line Control Processor (LCP) Input-Output Subsystem;

FIGS. 1B, 1C, 1D, and 1E, are schematics which indicate various components of the Central Control type of I/O Subsystem;

FIG. 2 is a schematic drawing of a modular unit of the LCP I/O Subsystem known as the LCP Base Module showing its relationship to a variety of peripheral devices;

FIG. 3 is a schematic drawing of the central processing unit of the Main System of the Line Control Processor I/O Subsystem;

FIG. 4A is a simplified schematic showing the basic connective relationships between the Main System, the Line Control Processor and a peripheral unit within the Line Control Processor I/O Subsystem;

FIG. 4B is a chart indicating various codes for the various instructions executable by a Line Control Processor, LCP;

FIG. 4C is a chart showing how four informational digits (ABCD) are organized such that a Line Control Processor can inform the Main System of operational results via a "Result Descriptor";

FIG. 5A is a chart of digital information (Descriptors) used by the Input-Output Translator (IOT) to generate Command Messages (C/M);

FIG. 5B is a schematic showing the data field boundaries of the Descriptors in FIG. 5A;

FIG. 5C is a block diagram of the Input-Output Translator (IOT) in its relationship to the Main System (Processor and Memory) and to the Line Control Processor (LCP);

FIG. 5D is a chart showing the information array in the IOT Descriptor Register;

FIG. 5E shows the message level interface between the IOT and the Distribution Card unit of the LCP Basic Module;

FIG. 5F is a sketch of the IOT scratchpad memory;

FIG. 5G is a sketch illustrating the address memory scratchpad of the IOT (Input-Output Translator);

FIG. 6A is a logic flow diagram of the interface between the Main System and the Line Control Processor (LCP);

FIG. 6B is a generalized block diagram of a Line Control Processor;

FIG. 6C is another generalized block diagram of a Line Control Processor with detail in regard to its data buffer memory;

FIG. 6D is a detailed functional block diagram of the Line Control Processor;

FIG. 6E is a diagram showing the intercooperating logic and control signals between the Input-Output Translator (IOT) of the Main System and the Distribution Card unit for Line Control Processors within a Base Module;

FIG. 6F is a chart showing the arrangement of a message block and the composition of a digital word;

FIG. 7A is a logic flow diagram of a Line Control Processor which handles a peripheral unit and shows the "status counts" for "Receipt of Instructions";

FIG. 7B is a flow diagram showing how the Line Control Processor handles a "Write" operation;

FIG. 7C is a flow diagram showing how a Line Control Processor handles the "Read" operation;

FIG. 7D is a flow diagram showing how the Line Control Processor logically handles the Result Descriptor;

FIGS. 7E-1 and 7E-2 together form a logic diagram showing the overall logic flow of the Line Control Processor.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The digital system described herein consists of a Processor, a Memory, a series of Input-Output Controllers (IOC's) forming a first I/O Subsystem and a system of Line Control Processors (LCP's) that make up a second I/O Subsystem. The Line Control Processors basically handle input-output operations for specific peripherals with minimal interference to main processor operations. Further, no peripheral device is "hung up" waiting for memory access, since the LCP for that peripheral is always readily available to service its peripheral.

A substantial number of prior data processing systems utilize a hierarchical system of Main Memory in which a large capacity, slow bulk memory must transfer information to a small high-speed processor memory before that information can be used. The presently described system allows the Processor and the I/O Subsystem to directly access any area of memory, and since the memory size may go up to one-million bytes, far more information is available to the Processor without the imposition of additional I/O activity. This system may be provided with high-speed (250-nanosecond cycle time) bipolar memory together with an error correction system. Bipolar memory is not only fast, but is inherently more immune to the type of errors that cause program failures. If an error is detected, the error correction occurs during the normal memory cycle and there is no additional time required for a correction cycle. Various operating relationships between the processors main memory and other units of the present system may be found in a Burroughs Corporation publication entitled "Burroughs B 2800/B 3800/B 4800 series, MS-2 Reference Manual, Catalog 1090560, Copyright 1976".

Normally, I/O memory cycles account only for a small fraction of the total number of memory cycles available. However, during periods of high I/O activity, the probability of any two devices requesting the same memory cycle increases. When, due to simultaneous requests, a device fails to get access to memory within a system-allotted time period, then valuable time is lost while the operation is retried. Furthermore, during periods of low I/O activity, many memory cycles are unused.

The I/O activity problems are solved in the present system by distributing the I/O processing among a group of LCP's or Line Control Processors organized into Base Modules of eight LCP's each. In so doing, the Central Processor is only required to initiate the I/O activity and it takes no further role in the Input-Output (I/O) operation. The Central Processor initiates the I/O activity through a device called the Input-Output Translator (IOT).

The LCP, once initiated, can buffer large amounts of data and, in most cases, an entire message block. At some point in the operation, the LCP requests an access to memory and when the access is granted, LCP transfers the information from its "word buffer" to the Memory at the maximum rate of memory operation. Now, if the requested access to memory is not granted, LCP continues to fill its word data buffer while waiting for an opportunity to access Memory. Thus, the peripheral device is now protected against no-activity since its data transfers to the buffer of the LCP, which transfers it to the Main Memory without missing a memory access period.

The result of this method and system is that the peak loads imposed upon the Memory by the demands of I/O activity are eliminated; instead, the I/O Subsystem utilizes those memory cycles that would otherwise be missed. Since this method of I/O processing is more efficient, the system is more capable of a higher input-output (I/O) data transfer rate and can also support more I/O devices.

In the instant computer system wherein there are two categories of Input-Output Subsystems, that is, the second Subsystem of I/O controls and the first Subsystem of an Input-Output Translator working with a group of Line Control Processors, the control of the system is facilitated by the use of "descriptor" information which is passed among the various units.

A "Result Descriptor" is a report to the Main operating system that describes the manner in which an operation was completed or the reason why the operation could not be completed. The Result Descriptors for the Processor and for the I/O control systems are 16 bits (one word) long. The LCP Result Descriptors may be longer than one word, however, and each bit in the Result Descriptor represents the status of some condition that is to be reported to the Main operating System.

The LCP's (Line Control Processors) and the I/OC's (I/O Controllers) always write Result Descriptors upon completion of an operation; the Processor writes a Result Descriptor only if an error condition was encountered. Result Descriptors are written into predetermined locations in Memory; for the Processor, the location is address **80**, for example.

The Result Descriptors for the LCP's and the I/OC's are written into locations beginning at the address specified by the equation (CH×20) plus 200, where CH is the channel number of the initiated device. The IOT Result Descriptor is written into address **260**. After the Result Descriptor has been written, an interrupt is generated.

LCP Result Descriptors, R/D

Upon the completion of its assigned operation, the LCP stores a Result Descriptor, which describes to the Processor the manner in which the operation was completed. An LCP Result Descriptor may consist of one, two, or three 16-bit words. The first Result Descriptor, R/D, is stored in Memory at the location specified by

he equation (CH × 20) plus 108, where CH is the channel number of an LCP. If more than one word of Result Descriptor information is to be written (extended Result Descriptor), the additional words are stored in the address memory of the IOT. As shown in the table I below, the first LCP Result Descriptor word is preceded by a 1-word link and the channel (IOT) Result Descriptor. Typically, the link is used by the Operating System as an address to the next Result Descriptor to be examined. Table II shows the basic word format for a "data" word having four digits, A, B, C, D, where each digit has 4 bits and each character has 8 bits. Symbols are used to designate parts of each digit, as A8, A4, A2, A1, etc.

TABLE I : Result Descriptor

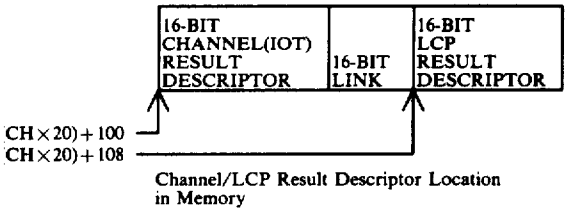


TABLE IV

Descriptor Link also see FIG. 5D	
Data Bit	Designation
A8	Inhibit Access to system memory.

specify the various options that a specific input-output operation can incorporate.

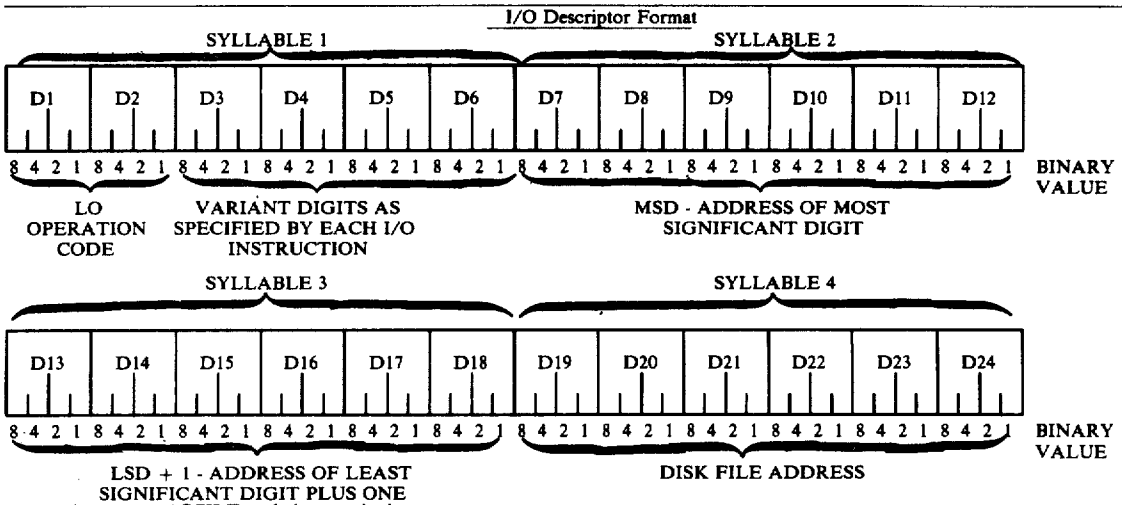
Syllable 2 contains the address of the most significant digit (MSD) of the Main Memory section which is used in this particular input/output operation as a memory buffer area. This buffer area is referred to as the beginning address. Syllable 3 contains the address of the least significant digit plus 1 (LSD+1) of the input/output core memory buffer area which is referred to as the "ending address". The most significant address and the least significant address plus 1 represent the maximum memory boundary limits of a record being transmitted. The length of the record may or may not utilize the entire area within this limit. But an attempt to exceed this limit causes termination of data transmission to that area.

Syllable 4 is used only for disk file descriptors and contains the disk address.

The length of the record may or may not utilize the entire area within the beginning address and ending address limits. As stated, an attempt to exceed this limit causes termination of data transmission to that area. For example, punch cards may be read into an area greater than 80 characters, that is, with a MSD and an LSD+1 at 80 characters apart, or they may be read into an area less than 80 characters; for example, the record area defined in a particular object program reflects 40 characters in a card reader record. Data within columns 1 through 40 of the punch card are stored in the record area of core memory allocated by MSD and LSD+1.

TABLE III

I/O Descriptor Format



A2	ASCII Translation required.
C2	Base Modulue Address: 4 bit.
C1	Base Module Address: 2 bit.
D8	Base Modulue Address: 1 bit.
D4	LCP address: 4 bit.
D2	LCP address: 2 bit.
D1	LCP address: 1 bit.

The table III below indicates the format for the I/O Descriptor which is normally stored in Main Memory and then accessed in order to regulate a particular type of Input/Output operation. As will be seen there are four syllables, wherein each syllable is composed of 6 digits. These digits are numbered D1–D6, D7–D12, D13–D18, D19–D24, to indicate the relative positions of each digit. In syllable 1, the digits D1 and D2 always specify the type of input/output operation to be performed and are generally called the "OP-code". Digits D3–D6 are referred to as "variant digits" in that they

System Description: (General)

An I/O Subsystem provided as part of a digital system environment to supply means of communication between a central data processing system and a variety of peripheral devices which are attached to and work within the system. The peripheral devices which work with the overall digital system herein vary from mass storage devices, such as disks or disk packs, to system control devices such as the operator's supervisory terminal, or to a variety of other peripheral devices such as printers, card readers, card punches, magnetic tape storage devices, and so on.

The I/O Subsystem described herein can be divided into two major subsystem categories, based on the

method by which the various peripheral devices are controlled. The first category uses a method which employs I/O Controllers (IOCs) working in conjunction with the Processor and a Central Control to handle I/O activity. The second category uses an Input-Output Translator (IOT) in the central processing unit which works with individual units called Line Control Processors (LCP's). The units known as Line Control Processors are the devices which establish a communication path from the System (Main Memory and Processor) to a specific peripheral device. Once the communication path is established, the LCP can accept data from, or pass data to, the specific peripheral device, for later transmission to the Main System. Since each LCP has a built-in data buffer, then data can be transferred to and from the given peripheral device at the comparatively low speed rate of the device; however, when the data buffer of the LCP is connected to transmit to the Main System Memory and Processor, the data can be transferred to the Main System at the highest rate allowed by the Memory of the Central System.

The first category of I/O Subsystems which use IOC's as an interface from a peripheral to the Main Memory and Processor has a Central Control (CC) unit which links the I/O channel and IOC with the Central Processor and Memory. These Input-Output Controllers accept instructions from the Processor and they return data information involving the result of what happened regarding that particular instruction. This result information is placed in a specified location in the Main Memory.

In the second category of I/O Subsystem is the system wherein the Processor and Main Memory communicate, via an Input-Output Translator (IOT), to a group of LCP Base Modules, each Module of which constitutes a unit supporting a group of 8 Line Control Processors (LCP's). Thus, an instruction from the Processor is translated by the IOT into a specialized set of commands which is acceptable to individual LCP's. After an LCP accepts instructions from the IOT, it will then report back certain "result information" which is stored in a specified location in the Main Memory.

Thus, in this second I/O Subsystem, all communications between the main system Processor and Memory over to a specified peripheral device are controlled by an LCP which is uniquely suited to that particular peripheral device.

When a Line Control Processor, LCP, (or an Input-Output Control means having a Central Control) is installed, it is assigned a unique number called its "channel number". For I/O Controls this number corresponds to a word of scratchpad memory located in the Processor. For Line Control Processors (LCP's) this "channel number" corresponds to a word of scratchpad memory in the Input-Output Translator (IOT).

To accomplish an input-output operation in the system, an I/O request is initiated by an Initiate I/O Instruction which tells the Processor where to find the appropriate I/O Descriptor in the Main Memory and also which channel number it is intended for. The I/O Descriptor contains the OP code and also the variants for the kind of I/O operation selected, and the beginning (A) and ending (B) Main Memory address of the memory area involved.

The Processor accesses this I/O Descriptor and then sends the OP code and its variants to the selected IOC (first Subsystem) or to the IOT (second Subsystem). The IOC or the IOT verifies the OP code and signifies acceptance or rejection of the request.

In the first Subsystem the Processor then loads the beginning (A) and the ending (B) addresses into a local register and informs the IOC that the addresses are available. These particular addresses are transferred by the IOC into the scratchpad memory location for that designated I/O channel.

In the second Subsystem the IOT accesses the A and the B addresses directly from the memory address lines leading to the Processor's "local register 10_{pr}", FIG. 3, at the time of transfer from Main Memory and thus the IOT loads its own local scratchpad memory 10_{ps}.

The access to Main Memory is shared by the IOT, the Central Control and the Processor. The highest priority is shared by the IOT and the Central Control. The timing may be so arranged that each Central Control is guaranteed and limited to every fourth memory cycle (at, for example, 8 MHz). The IOT is guaranteed the remaining cycles. When the Central Control is not requesting memory, then the IOT can take all the memory cycles. The Processor takes all memory cycles available on a lowest priority basis.

Thus, I/O communications in the system require that the Processor execute an Initiate I/O Instruction (which may be designated, for example, as OP = 94). This Initiate Instruction specifies the channel number of the requested device and also the location of the I/O Descriptor in Main Memory. The I/O Descriptor specifies the action to be taken by the peripheral device and specifies the boundaries in Memory of the data field. The Descriptors, and the manner in which they are executed, vary, depending on the method by which the peripheral device is controlled.

If an Initiate I/O Instruction is executed for a channel containing an I/O Control (first I/O Subsystem), then the Processor sends the Descriptor OP code, variants and a C address (if used) to the I/O Control. The A (beginning) and B (ending) addresses of the Descriptor are stored in the Processor's I/O channel address memory. The I/O Control verifies that the OP code is valid, then signals the peripheral device that a data transfer is to begin.

As was previously discussed, the embodiment of the present digital system involves a duality of Input/Output Subsystems. The second of these involve a Central System with Input/Output Translator (IOT), a Base Module having a plurality of Line Control Processors (LCP) and a plurality of peripheral units, the first I/O Subsystem involves, as seen in FIG. 1A, a Central Control unit **12** which interfaces with a plurality of I/O controls 13_a and 13_b which interface with a plurality of peripheral devices 14_a and 14_b etc.

The following discussion will involve the first I/O Subsystem involving IOC's with Central Controllers, CC.

The FIG. 1B shows the system of connecting the I/O channels with the Processor, 10_p, and the Main Memory 10_m through the Central Control **12**. Logic levels are generated in each I/O channel **100**, **101** (FIG. 1C) and combined by Central Control **12** before being sent to the Processor 10_p and the Main Memory 10_m. Other logic levels are generated by the Processor, and within the Memory, and distributed by Central Control **12** to each I/O control such as 13_a, FIGS. 1A and 1B. There are also logic levels which pass through the Central Control **12** with the Central Control performing as the connecting block between the Processor 10_p and I/O

channels. Priority logic, 10_{pc} of FIG. 1C, determines which of the I/O channels will be allowed access to the Main Memory 10_m, should more than one channel need access at the same time.

As seen in FIG. 1C, there is included, as part of Central Control 12, a plug in translator which is capable of translating BCL (Burroughs Common Language) data to or from EBCDIC (Extended Binary Coded Decimal Interchange Code) as it goes to or comes from the Core Memory 10_m. The I/O Control units, 13_a, 13_b, FIG. 1A, request Central Control 12 to use the translator, 12_t, FIG. 1C, or to bypass it. The translation takes place as data is transferred between the I/O Control unit, such as 13_a, and the Main Memory 10_m. Additional time is not required for I/O operation even though translation is necessary. The translator logic translates incoming Burroughs Common Language (BCL) data into EBCDIC (Extended Binary Code Decimal Interchange Code) data or the outgoing EBCDIC data into Burroughs Common Language (BCL). Those EBCDIC codes which are not assigned a BCL code, will cause to be generated a code for a BCL symbol "?".

The Central Control 12 functions as an interface between an I/O channel and the Main Memory 10_m during system operation, as seen in FIGS. 1B and 1C. It determines the priority of memory accesses, should more than one channel need access, and it translates data coming to the I/O channel, as 100, from Memory 10_m or from the I/O channel to Memory. The Central Control correlates various functions of the channels. The sequence of events is initiated by the Processor 10_p when an I/O channel is needed.

When the program being performed has need of a peripheral unit such as 14_a or 14_b of FIG. 1A, the Processor 10_p executes the "Initiate I/O Instruction". This instruction reads an I/O Descriptor from Memory 10_m and then sends the necessary information to the I/O channel, 100, through Central Control 12. This information contains the type of operation (OP code) and the variant information. The remaining portion of the I/O Descriptor including the beginning (A) and ending (B) addresses, is stored in Address Memory, 10_{pam}, FIG. 1C, of the Processor 10_p. The channel is selected by the channel designate level (CDL) as seen in FIG. 1B, which line comes from the Processor 10_p.

Once all the information is available, the I/O channel, 100, is released by the start channel bus (STCB), FIG. 1B, to operate independently. When the I/O channel has been released, it operates as another processor and shares the Main Memory 10_m with the main Processor 10_p or other channels (FIG. 1C).

If the operation being performed involves an "input type" peripheral unit 14_i such as a card reader, the data is received by the I/O channel 100 seen in FIG. 1C, and the data is stored in a buffer C_0 within the I/O channel 100. The I/O channel then requests access to Main Memory 10_m via Central Control 12. This request is processed by the priority logic 10_{pc} which controls other requests at the same time. Once access to Memory has been granted to the channel, the information is transferred to Memory 10_m. The information may or may not be translated depending upon the I/O Descriptor. The information is then written into the Main Memory 10_m at the location specified by the beginning (A) and ending (B) addresses in the Address Memory, 10_{pam}.

If it is desired, at some point, for data or information to be transferred out to a peripheral terminal unit, this is called an "output" operation, FIG. 1D. If an "output" operation is being performed, a similar sequence of events occurs as before, except that data goes from the Main Memory 10_m to an I/O channel such as 102 of FIG. 1D. Then when a peripheral unit as, for example, a printer 14_p needs data, the memory access request is made to the Central Control 12 via the I/O channel, 102. When priority is granted to the channel, the data is read from Main Memory 10_m from the address specified by the beginning and ending addresses located in the Address Memory 10_{pam}; this data is then transferred to the I/O channel buffer C_2 through the translator 12_t (or bypassed around the translator depending upon the I/O Descriptor). As seen in FIG. 1D, the data is then transferred to the peripheral unit, such as 14_p.

As seen in FIG. 1E the Central Control 12 provides an interface to/from the I/O channels, the Processor 10_p, and the Core Memory 10_m. Control information from the Processor 10_p is sent to the Central Control 12, where it is distributed to each I/O channel as 100, 101, etc. The Central Control 12 handles all of the Core Memory requests made by the I/O Control units in this first I/O Subsystem. Data from each I/O channel, which is to be written into Core Memory 10_m is placed on the Memory Write Bus by the Central Control 12, and data which is to be read from the Core Memory 10_m is placed on the Core Memory Read Bus and distributed to each I/O channel.

When a request is made by an I/O channel unit, the Central Control 12 will obtain the Core Memory address from the Address Memory location reserved for that specific I/O channel. This address is used to access Main Memory 10_m and the memory cycle is then initiated. The memory cycle could be either a "Read" or "Write" depending on the specific I/O operation.

When the Processor 10_p requests a memory access, the memory address involved is obtained from the Address Memory 10_{pam} located in the Processor 10_p. This address is used to access Main Memory 10_m, and the memory cycle (either a read or a write) is initiated.

Since only a single memory access can be made at a given moment, multiple memory requests must be handled individually, and this handling is accomplished automatically via Priority Control 10_{pc} (FIG. 1C, 1D) by Central Control 12, as previously discussed. Each Central Control 12 contains "priority logic" 10_{pc} which is established or changed by a field engineering adjustment. As I/O channels are added to the Central Control 12, they are also added to the priority network. The Processor 10_p, in this case, has a lower priority than a Central Control 12. The highest priority request is granted first, and as soon as it is completed, the next highest request is automatically granted. This process is repeated until all of the multiple requests are handled. The requests are alternately granted to each Central Control unit (when multiple Central Controls are used) depending on which control was granted the last request. If a Central Control does not want the access, then it is granted to the Processor 10_p.

During the course of a data transfer operation within the first category Subsystem, the I/OC (Input/Output Controller) may perform several functions depending on the OP code, the variants, and the type of peripheral device. Typically, the I/O Controls have the ability to buffer only one byte or at most one word. Thus, when the data buffer of a control is loaded, the I/O Controller or I/O Channel Unit 100, 101, 102, must request a memory access; therefore, the rate at which data is transferred to or transferred from the System is controlled

primarily by the speed rate at which the peripheral device can read or write. The second I/O Subsystem using Base Modules with Line Control Processors does not have this speed limitation.

When the I/O Controller requests a memory access, it is, in effect, asking the Processor to perform a series of operations; these operations include: (a) the transfer of the data field address from the processor's I/O channel address memory to the local address register; (b) the initiation of a memory cycle; (c) and the restoration of the data field address to the address memory of the channel. The I/O Controller also indicates to the Processor the amount by which the address must be incremented so as to point at the next data field location. Upon completion of the operation, the I/O Controller builds a Result Descriptor (R/D) indicative of how the operation was effectuated, then the I/OC stores the Result Descriptor in a reserved memory location, after which it sets the Processor Interrupt flip-flop.

In the second category of controlling I/O activity, use is made of an Input/Output Translator (IOT) interface unit which is located in the central processor unit **10**. The IOT interfaces with a group of Line Control Processors (LCP) which are installed in LCP Base Modules. Up to eight LCP's may be housed in an LCP Base Module. The Base Module for the LCP's holds up to as much as eight LCP's. The LCP is an intelligent interface unit which establishes a buffered data transfer path between the peripheral device involved and the main system of Processor and Memory. This communication path is established by the LCP upon receipt of a Command Descriptor (C/D) from the IOT which has translated an original I/O Descriptor into a specialized Command Descriptor for the LCP.

Since each LCP has a large "data buffer" of, typically, 256 words, then data can be transferred to and from a specific peripheral device at the comparatively low rate of the device; however, when the data buffer is full, data can be transferred to the Main System at the highest rate allowed by the memory speed of the Main Memory, which is at a fast rate.

The LCP Base Module, which houses up to eight LCP's, operates in conjunction with the IOT to establish connection to and to initiate operation of a particular LCP. The LCP Base Module also supplies the timing signals, the maintenance logic, the power supply and cooling which is supportive of each group of individual LCP's.

The IOT is that portion of the central processing unit which, upon receipt of an I/O Descriptor, works in conjunction with the LCP Base Module to establish connection to a particular LCP in the channel specified by the Initiate I/O Instruction. The IOT translates the I/O Descriptor to a form (Command/Descriptor) recognizable to the LCP, and, when connection is established, passes the translated descriptor to the LCP, after which data transmission may begin. During the time that the data is being transferred between the LCP and the Main System, then the IOT, upon demand from the LCP, requests memory accesses, addresses memory, then modifies and compares the data addresses. Further, the IOT controls the routing of data between the selected LCP and the Main System, and it performs translations (ASCIII/EBCDIC) of the data if so required. Upon completion of an operation, the IOT accepts R/D (Result Descriptor) information from the LCP, and then stores the Result Descriptor in a predetermined location.

The LCP system configuration allows up to 68 I/O channels. In the I/O Control Subsystem there may be two CC's (Central Controls) with eight I/O Controllers each for a total of only 16 channels.

In the LCP subsystem however, there may exist up to eight LCP Base Modules per single IOT. Each Base Module may service and carry up to eight LCP's. Thus, one IOT may serve as many as 64 LCP's. A Multiplex Adapter may be used to provide the effect of "two" IOTs connected to common LCP Base Modules. This configuration may be used to improve I/O band pass to the Main Memory.

The entire I/O System has channel addresses which must be unique in themselves. Access to Main Memory is shared by the IOT, the Central Control and also the Processor.

In FIG. **1A** there is seen an overall system diagram showing the dual categories of I/O Subsystems. The first I/O Subsystem is made of Central Control **12** which supports I/O Controls $\mathbf{13}_a$ and $\mathbf{13}_b$ which connect respectively to peripheral devices $\mathbf{14}_a$ and $\mathbf{14}_b$. This first I/O Subsystem (using Central Control) is connected to the Main System **10** by means of interconnecting bus **11**.

The Main System **10** is shown comprising a Main Memory $\mathbf{10}_m$, the Central Processor $\mathbf{10}_p$, the Memory Control $\mathbf{10}_c$, and the Input-Output Translator $\mathbf{10}_t$. A PCC (Peripheral Control Cabinet) interface $\mathbf{10}_i$ connects via bus **5** to a Peripheral Control Cabinet **6** which houses the Central Control and the I/O Control units of the first I/O Subsystem.

The Input-Output Translator $\mathbf{10}_t$ of the Main System, FIG. **1A**, forms a second I/O Subsystem through the use of cabinets shown as LCP cabinet numbers **0, 1, 2**, designated as $\mathbf{16}_0$, $\mathbf{16}_1$, $\mathbf{16}_2$. Each of the LCP cabinets supports three LCP Base Modules, **0–8**; for example, base cabinet $\mathbf{16}_0$ carries Base Modules $\mathbf{20}_0$, $\mathbf{20}_1$, $\mathbf{20}_2$; while LCP cabinet $\mathbf{16}_1$ supports LCP Base Modules $\mathbf{20}_3$, $\mathbf{20}_4$, and $\mathbf{20}_5$; likewise, LCP cabinet $\mathbf{16}_2$ supports LCP Base Module $\mathbf{20}_6$ and $\mathbf{20}_7$. Each of the individual LCP Base Modules is connected to the IOT $\mathbf{10}_t$ by means of message level interface cables (MLI) **15**, each of which is made up of 25 lines.

Referring to FIG. **2**, a typical LCP Base Module $\mathbf{20}_0$ is shown in greater detail. The Base Module $\mathbf{20}_0$ is composed of eight Line Control Processors (LCP's) $\mathbf{20}_{00}$ through $\mathbf{20}_{07}$, in addition to a common Distribution Card $\mathbf{20}_{od}$, a common Maintenance Card $\mathbf{20}_{om}$ and a common Termination Card $\mathbf{20}_{ot}$. The Distribution Card $\mathbf{20}_{od}$ connects to one set of the message level interface cables **15** which connect to the IOT $\mathbf{10}_t$ (also see FIG. **5E**).

Each individual Line Control Processor is seen connected by output lines to a particular peripheral device, wherein, as seen in FIG. **2** the LCP's $\mathbf{20}_{00}$ through $\mathbf{20}_{07}$ respectively connect to peripheral devices **50, 51, 52, 53, 54, 55, 56, 57**.

While each LCP of the Base Module may be slightly different in certain aspects in order to accommodate the idiosyncrasies of each particular peripheral device which the LCP handles, each LCP is of basically the same design and functional capability. With reference to FIG. **2**, a typical example of each LCP is seen in the LCP $\mathbf{20}_{06}$ which is seen having a System Interface $\mathbf{21}_{si}$, a Device Interface $\mathbf{22}_{di}$ and having a Word Buffer $\mathbf{25}_{06}$ which is typically capable of holding 256 words.

Referring to FIG. **3** there is seen a more detailed block diagram of the Main System as it relates to the I/O LCP Subsystems. The Main System **10** has a Main

Memory 10_m in which there is a reserve portion 10_{mi} for I/O Descriptors and another reserve section 10_{mr} for Result Descriptors. In addition the Main Memory 10_m has another reserve portion 10_{nc} for storage of channel numbers. The I/O Descriptors, Result Descriptors, and Channel Numbers are information used by the System for control and for recognition of the status of operations. These will be described in detail hereinafter.

The Processor 10_p has a local register 10_{pr} which is useful for storing information for the IOT. The Input-Output Translator 10_t holds a channel scratchpad memory 10_{ps}.

The local register 10_{pr} of the Processor 10_p is used for storing the beginning (A) and the ending (B) addresses of the appropriate I/O Descriptor. (In the case of the first I/O Subsystem using a Central Control, FIG. 1A, the I/O C causes these addresses to be transferred into a temporary storage location called channel scratchpad memory or channel address memory.) In the case of the second Subsystem using the IOT, the IOT accesses the A and B addresses directly from the memory address lines leading to the local register 10_{pr} of the Processor. The channel scratchpad memory 10_{ps} for all 64 LCP's is contained in the IOT. The channel scratchpad memories will also contain the required channel numbers.

With reference to FIG. 4A and the transfer of information as between the Main System 10 and a typical LCP 20_{00}, a brief look at these information words and their functions will indicate the nature of the operating relationship.

Command Descriptor (FIG. 4A)

The Command Descriptor (C/D) is a modified form of the I/O Descriptor. The I/O Descriptor is the information residing in Main Memory 10_m, FIG. 1 and specifically in 10_{mi} of FIG. 3), which provides data and information as to the type of Input-Output operation to be accomplished. The modification of the I/O Descriptor is accomplished by the IOT 10_t (Input-Output Translator, FIG. 1) which receives the I/O Descriptor from the System Memory 10_m, retains a portion of the instruction, and then transmits the applicable portion to the LCP 20_{00} as a Command Descriptor.

The Command Descriptor is a 17-bit word, A, B, C, D, (FIG. 4B) consisting of an OP code digit (A), variant digits 1 (B), 2 (C), and 3 (D), and a parity bit. However, the LCP 20_{00} makes use of only the OP code digit and variant digit 1 for instructional purposes. Variant digits 2 and 3 are always equal to 0. The OP code digit (A) defines the basic operation to be performed by the LCP 20_{00}, and the variant digit 1 (B) specifies modifications of the basic operation. No memory address information is sent to the LCP; the System Memory address functions are accomplished by the IOT 10_t. FIG. 4B contains the Command Descriptor codes for all operations that can be performed by the LCP. These operations include: Write, Read, Write Flip Read, Test, Test Enable, Conditional Cancel, and Echo. These operations will be later described hereinafter.

Descriptor Link (FIG. 4A)

The Descriptor Link (D/L) consists of two 16-bit information words accompanied by a longitudinal parity word (LPW). The Descriptor Link is exchanged between the IOT 10_t (FIG. 1) and a LCP, as LCP 20_{00} at specific times during communication between the two units. The content of the Descriptor Link is shown in the following table. The data bits which are not listed are reserved for future use.

TABLE IV

Descriptor Link Also see FIG. 5D	
Data Bit	Designation
A8	Inhibit Access to system memory.
A2	ASCII Translation required.
C2	Base Module Address: 4 bit.
C1	Base Module Address: 2 bit.
D8	Base Module Address: 1 bit.
D4	LCP address: 4 bit.
D2	LCP address: 2 bit.
D1	LCP address: 1 bit.

Data (Intelligence) (FIG. 4A)

These are the bidirectional communication lines for transfer of data from the System 10 over to the LCP such as 20_{00} for eventual transfer to a peripheral unit such as 50; or otherwise for transfer of data from the peripheral unit 50 over to the LCP 20_{00} and thence to the System 10 for storage in Memory 10_m. In FIGS. 1 and 3, these channels would be the message level interface (MLI) 15. Data transmission between the System 10 and the LCP 20_{00} is in the form of words (Table II) except for certain transmissions which are limited to a single character or for transmissions ending in an odd number of characters. Each "data word" is composed of two 7-bit ASCII characters and a single parity bit. Data bits A8 and C8 are not used, (Table II).

It should be noted in regard to the Command Descriptor, that after receipt of a Command Descriptor, but prior to execution of an operation, the LCP 20_{00} receives the Descriptor Link from the IOT 10_t and stores it in the LCP buffer 25_{00} (FIG. 2). When the LCP 20_{00} disconnects from the System 10, then reconnects for further communication, the Descriptor Link is returned to the IOT 10_t to identify the LCP and the operation in progress.

Result Descriptor (FIG. 4A)

A Result Descriptor is generated by the LCP 20_{00} and forwarded to the System 10, after the instruction contained in a Command Descriptor (C/D) is executed, or when an error occurs during receipt of a Command Descriptor or a Descriptor Link. The Result Descriptor is sent to the System 10 by the LCP, in a 16-bit word format, with a parity bit. FIG. 4C shows the 16-bit format for a Result Descriptor, wherein digits A, B, C, D will each have 4-bits.

Longitudinal Parity Word (FIG. 4A)

The Longitudinal Parity Word (LPW) is a 16-bit word representing the longitudinal parity of each transmission between the System 10 and the LCP 20_{00}. An LPW is accumulated in both the IOT 10_t and the LCP 20_{00} during a transfer of information between the two units. An LPW register is provided in the LCP 20_{00} wherein accumulation of the LPW by the LCP 20_{00} consists of applying each word being transferred to the input of the LPW register and performing a binary add operation without carry (exclusive OR function). Then at the end of a data transfer, the exclusive OR function is again performed between LPW's of the sending and the receiving unit. If no errors have occurred, both LPW's will be identical, and the resultant value in the LPW register will be "all 0's".

Input-Output Translator (IOT) (FIG. 5C)

The IOT 10_t translates the system I/O Descriptors into the appropriate operational messages relevant to each LCP. In return the result messages from the LCP in the form of Result Descriptors are not translated by the IOT, but are stored directly into Memory 10_m at 10_m, as transmitted by the LCPs. The IOT performs all the information transfers between the LCP's and the Main Memory 10_m necessary to support the input-output capability of the second I/O LCP Subsystem.

The I/O Descriptors, which are sent to the IOT from Memory 10_m, are shown in FIG. 5A. Section 1A of this figure shows the descriptors used by the IOT to generate command messages C/M for the LCP. These can also be referred to as Command Descriptors C/D. Section 1B indicates descriptors used by the IOT. Operations 40 through 58 are translated into LCP OP codes and sent to the LCP's in "message" format. The "L" digits in the variant field carry information used in the variant digits (B, C, and D) of the descriptor information sent to the LCP's. The S-digit is used by the IOT as shown by the note of section 1A of FIG. 5A.

Each operation shown in FIG. 5A has two OP codes; the difference is in the number of addresses used by the LCP. The first digit of the OP code designates the number of addresses required. For example, a value of 4 designates two-address operation (except "test" which has none); a value of 5 for the first digit of the OP code designates three address operation. The second digit of the OP code is mapped into the actual OP codes sent to the LCP's as the "A" digit.

FIG. 5B shows the data field boundaries of operations going in the forward direction and in the backward direction. (Forward = System to LCP).

FIG. 5A also shows the four types of standard operational messages used for controlling the LCP's: these are

1. Read
2. Write
3. Test
4. Echo

The specific descriptor information is obtained in the form of variants which accompany these OP codes. "Read" and "Write" require system memory access. All operations which do not transfer data are considered "Test". Thus, a "Test" is defined as an operation which results in the IOT receiving result information only. "Echo" is a confidence test operation which causes the LCP to accept a buffer load of information from the System 10 and then return it to the System 10 for checkout.

All communications between the Main System 10 and the LCP is over a standard message level interface 15 (MLI). This communication between the IOT and the various LCP's is accomplished by a standard flow discipline which is common to all LCP's.

In FIG. 5C the IOT 10_t receives I/O Descriptors from the Processor 10_p. The IOT then connects via Distribution Unit 20_{od} to the requested LCP channel and sends the translated descriptor information (Command Descriptor C/D) in a message format which indicates the LCP's task. The IOT then becomes LCP "status driven". This means that the IOT responds to the various LCP states (including memory requirements) as indicated via the control lines between the LCP and the IOT FIG. 4A. The IOT manages the transfer of information between Main Memory and the LCP's. The LCP's memory requirements drive the IOT for all data transfers except that of initiation.

Either the IOT or the LCP can initiate a connection to Main Memory 10_m. The IOT initiates a Main Memory connection to an LCP (and its associated peripheral) by performing an algorithm which is called a "Poll Test". On the other hand, the LCP initiates a connection to IOT and Main Memory by an algorithm called a "Poll Request". Once the LCP is connected, it indicates its status via the control lines of FIG. 4A. An LCP which is initiating a "Poll Request" must compete with the other LCP's in the system; a connection to Main Memory 10_m is granted on a priority basis which will be described hereinafter. During an operation, the IOT 10_t may disconnect from one LCP in order to service another LCP.

The message transmissions between the IOT and the LCP involve data and control messages which are transmitted 16-bits at a time along with a vertical odd parity bit. Following the last message, a 16-bit longitudinal odd parity word (LPW) is transmitted accompanied by a vertical odd parity bit. Parity is checked by both the IOT and the LCP. If a parity error is detected by the LCP, then the LCP reports this in its result information transmission (Result Descriptor) and halts the operation. If the IOT detects a parity error, it is inserted in the LCP Result Descriptor.

The Input-Output Translator 10_t (IOT) consists of four major functional sections, each concerned with one particular aspect of input-output operation. These functional sections are shown in FIG. 5C. Further, the operating relationships between the IOT and the Main System (Processor and Main Memory) and also the LCP and the peripheral device, are also shown.

Referring to FIG. 5C, the Input-Output Translator 10_t communicates with the Processor 10 and the Main Memory 10_m. The IOT 10_t also communicates with a selected LCP as Line Control Processor 20_{00} and the peripheral device 50. A series of control lines in FIG. 5C are shown from the Processor 10 to the Initiation Module 10_{ta}, the Connection Module 10_{tb}, the Data Transfer Module 10_{tc} and the Reconnection Module 10_{td}.

Initiation Module

The Initiation Module 10_{ta} accepts the descriptor information, including the addresses, from the Processor 10, and then translates the descriptor OP code and assembles the information into a form usable by the LCP 20_{00}. The A and the B addresses of the descriptor are stored in the IOT scratchpad memory 10_{ps}, FIG. 3, which has locations reserved for each designated channel; the rest of the descriptor information is assembled in a register (as shown in FIG. 5D) for subsequent transmission to the LCP 20_{00}. Once the information is assembled in this "descriptor information register" and the addresses are stored, then the contents of the first register are shifted to a second identical register. In this manner, the first register can be cleared and the Initiation Module 10_{ta} is thereby freed to accept a second descriptor.

The information contained in the descriptor register of FIG. 5D consists of a number of items:

(a) LCP OP CODE : these are four mutually exclusive bits, which are translated by the IOT from the I/O Descriptor OP code; they indicate to the LCP the type of operation that is to be commenced.

(b) LCP Variants : these are three digits which are used to pass supplementary information to the LCP concerning the operation that is to be commenced.

(c) IOT Digit : this digit specifies if data transfers are to be inhibited and whether or not data is to be translated.

(d) Backwards Flag : when on, this flag bit indicates that a reverse operation is to occur.

(e) LCP Address : this is decoded from the "BF" (channel number) of the processor Initiate I/O instruction; this field contains three bits which specify one of the eight LCP Base Modules, and the other three bits which are used in combination to select a particular LCP in the designated Base Module.

(f) C Address : this is a six-digit C-Address field (file address) of the I/O Descriptor.

The combination of the IOT digit, the backwards flag, and the LCP address constitute the Descriptor Link (D/L) which is used by the LCP to re-establish connection to the System following a previous disconnection. When the Processor signals the IOT that the entire I/O Descriptor has been sent, the IOT disconnects from the Processor, and the Initiation Module 10_{ta} passes control to the Connection Module 10_{tb}.

Connection Module

The Connection Module 10_{tb} of FIG. 5C has the purpose of establishing a communication path between a designated LCP, such as LCP 20_{00}, and the Input-Output Translator 10_t. The Connection Module 10_{tb} decodes the channel number which appears in the Processor Initiate Instruction, and, with the decoded value, selects a communication path to the LCP Base Module such as 20_0, FIG. 1A, in which the desired LCP is located. The Connection Module 10_{tb} then sends the LCP address to the selected LCP Base Module, and then signals the Base Module, such as 20_0, to begin a "Poll Test".

Poll Test

The "Poll Test" is an algorithm used by the LCP Base Module to establish connection between the Base Module and a particular LCP; the Poll Test algorithm is a connection which is initiated by the IOT (as contrasted with an algorithm called "poll request" which is a connection initiated by the LCP). Once the connection between the LCP Base Module and the specific LCP is established, the Base Module, such as 20_0 of FIGS. 1A and 2, becomes transparent to data transfers between the LCP and the IOT. The "Poll Test" algorithm also checks for priority, transmission errors, and busy conditions, any one of which, if detected, could abort the connection attempt.

If the connection attempt is successful, the specific LCP remains connected to the IOT 10_t until the connection is terminated by the IOT. The LCP Base Module takes no further role in the communications between the chosen LCP and the IOT.

In the course of the attempted connection, certain conditions may be detected which will stop or abort the connection attempt, with the result that the existing condition is reported in the IOT Result/Descriptor. The following are the types of conditions detected and reported:

(a) The channel addressed does not contain an LCP or the LCP in the channel is off line.

(b) the LCP in the particular channel addressed is "busy", (that is, the LCP status is not 2 or 3; the use of "status counts" will be described hereinafter).

(c) The port is busy, that is, some other LCP in that Base Module is presently connected to the System **10**.

(d) the LCP address has in it a parity error.

When the IOT and Base Module Distribution Control means uses the "Poll Test" for connection to a particular LCP, then if the Poll Test results in connection to that LCP, the IOT 10_t will transmit the Descriptor Link (D/L), the LCP OP code and variants, and the C address to the LCP selected. After receiving this information, the LCP signals the IOT 10_t that it is either going to disconnect, or that it is now prepared to begin to transfer data. Typically, a "Write" operation (data from Main Memory 10_m to the peripheral device, such as peripheral **50**) causes the LCP selected to request a "data transfer"; on the other hand a "Read" operation typically results in a disconnection.

If a data transfer is requested, the Connection Module 10_{tb} passes control over to the Data Transfer Module 10_{tc}. If the LCP 20_{00} disconnected, then communication between the LCP 20_{00} and the IOT 10_t is terminated until the LCP requests a re-establishment of communication via the Reconnection Module 10_{td}.

Data Transfer Module

In FIG. 5C the Data Transfer Module 10_{tc} is used by the IOT 10_t to control and to direct the flow of data between a connected LCP 20_{00} and the Main Memory 10_m. The LCP may be in a connected state as a direct result of the actions of the Connection Module 10_{tb}, or as a result of the actions of the Reconnection Module 10_{td}; in either case the operation of the Data Transfer Module 10_{tc}, is the same. When control is passed over to the Data Transfer Module 10_{tc}, the A and B addresses of the descriptor are retrieved from IOT scratchpad memory 10_{ps} of FIG. 3, where they had been stored by either the Initiation Module 10_{ta}, or by the Data Transfer Module 10_{tc} of FIG. 5C, at the end of a prior data transfer operation. A memory access request is made and the A address is transferred from the IOT 10_t over to the Processor memory address register 10_{pam} in the Main System **10**, FIG. 3.

Assuming that a "Write" operation is in progress, in FIG. 5C, the data from the memory location specified by the A address is bussed via B_m to the IOT Data Transfer Module 10_{tc}. Once in the module, the data is translated (if specified by the descriptor), and used to generate longitudinal parity, and then is gated via bus B_g of the selected LCP such as LCP 20_{00}, accompanied by a strobe pulse. When the LCP 20_{00} receives the data, it acknowledges the reception by returning a strobe pulse back to the IOT 10_t.

While the data transfer from Memory 10_m over to the LCP 20_{00} is occurring, the IOT 10_t increments the A address and compares it to the B address. As long as the A address is less than the B address, the reception of the acknowledged strobe pulse from the LCP 20_{00} will cause another memory access to be requested and will allow the data transfer sequence to continue.

When the LCP buffer, such as 25_{00}, FIG. 2, is filled with data from the Memory 10_m, the LCP signals the IOT 10_t that it is going to disconnect; the IOT 10_t then restores the incremented A address to the IOT scratchpad memory 10_{ps}, FIG. 3, after which it terminates the

connection between the IOT and the LCP. The LCP, such as LCP 20_{00}, then begins data transmission via B_p with its peripheral device **50**; the IOT 10_t is now free to establish connection to another LCP.

Upon transferring the contents of its data buffer 25_{00} to the peripheral device **50**, the LCP 20_{00} requests a re-establishment of the data path to Main Memory 10_m. This re-establishment is handled by the LCP Base Module 20_0 and the IOT Reconnection Module 10_{td}.

In order to increase the overall rate of input-output (I/O) activity, the IOT 10_t may contain, as an option, an IOT Multiplexor. This multiplexor would enable the IOT to service an LCP during those memory cycles which would otherwise be lost while the IOT was busy with some non-memory function.

Reconnection Module

An LCP, such as 20_{00}, after having been connected to the IOT 10_t and receiving the Command Descriptor (C/D) and the Descriptor Link (D/L), then the LCP 20_{00} may disconnect from the system in order to communicate with its associated peripheral device, such as device **50**. Now, if that LCP subsequently requires access to Memory 10_m, it sends a request to the Base Module 20_0. An algorithm called the "Poll Request" is the method by which the LCP Base Module (in response to the request of the LCP) attempts to connect the LCP back to the IOT 10_t. The Base Module Distribution Card contains hard wired logic to accomplish this. The purpose of the Reconnection Module 10_{td} is to acknowledge the "Poll Request" and to re-establish a data path over to the IOT 10_t.

The Reconnection Module 10_{td}, during the reconnection attempt, and working with the Base Module, as 20_0, resolves any priority conflicts that may arise between various requesting LCP's. When priority is resolved, the Reconnection Module establishes the data path from the requesting LCP over to the Main Memory 10_m.

Once the data path is re-established, the LCP returns the Descriptor Link over to the IOT 10_t. (The Descriptor Link was originally passed to the LCP 20_{00} during the original connection sequence). The Base Module 20_0 takes no further role in the LCP-IOT communication. Following the transfer of the Descriptor Link, the Reconnection Module 10_{td} passes control to the Data Transfer Module 10_{tc}.

The IOT 10_t must have the ability to accept, store and to modify data field addresses in order to transfer data to and from the correct memory locations. Because Main Memory 10_m may include up to two-million digits (addresses 0 to 1,999,999), and because the various input-output devices may address the Memory 10_m directly, then the I/O descriptor data field addresses must be seven digits long. An I/O descriptor data field address must be either MOD 2 or MOD 4 (modulus is abbreviated to MOD); no odd addresses are permitted. Because odd addresses are not allowed, the least significant bit of the least significant digit is not required. Furthermore, since the most significant digit can be only a "1" or a "0", only one bit is required for the digit position. With these facts, it is possible to construct a seven digit address using 24-bits. The format for the I/O descriptor data field address is shown in the table V below.

TABLE V

	G	F	E	D	C	B	A	Digit Position
Bit Value 8	☒							
Bit Value 4	☒							
Bit Value 2	☒							
Bit Value 1							☒	

I/O Descriptor Data Field Address

Note: ☒ indicates bit not used; must be zero

In the address, the digit G may be a one or a zero, digits B through F may be any decimal value (0 through 9), and digit A may be any even decimal value (0 through 8).

As was indicated in FIG. 3, the IOT 10_t has a scratchpad memory 10_{ps}. This is shown in greater detail in FIG. 5F. The IOT contains 256 words of scratchpad memory, each word of which is 24-bits long. As seen in FIG. 5F, the scratchpad memory is divided into five major areas. The areas marked A and B are used to store the begin (A) and the end (B) addresses of the memory data field; both of these addresses are 24-bits long. The areas marked EXRDW 1 and EXRDW 2 are used to store extended result descriptors wherein each of these words are 16-bits long. The area marked "temporary storage" is used to store flags indicative of errors detected during IOT operation. When the Result Descriptor is assembled, the information from the temporary storage area is added to any existing Result Descriptor information. Each of the five major areas is subdivided into 64 individual locations, one for each channel.

The scratchpad locations are addressed by a combination of eight bits which represent the Base Module number and the LCP number, the end address flag (ADDRESB), and the extended result descriptor flag (EXRDW 1). The six least significant bits of the scratchpad address (Base Module number and LCP number) are derived from the BF portion of the Processor's Initiate Instruction (BFA = base number, BFB = LCP number). The EXRDW 1 signal is generated by the IOT 10_t whenever access is required to either the extended Result Descriptor word, or to the temporary storage area. ADDRESB is generated by the IOT whenever access is required to a B address or to the second extended Result Descriptor area.

The memory elements of the scratchpad 10_{ps} consist of 24 RAMs (256 × 1), organized in a 64 × 4 × 24 array (64 channels, 4 words per channel, 24-bits per word). As seen in FIG. 5G, the eight-bit address bus, B_{ad}, goes to all RAMs, 60_0, 60_1 . . . 60_{24}, in the array, as does the Write Enable line **68**. Each RAM has one data input line and one data output line; these individual data lines are combined to make up the data input (RAMIN) 70_i and the data output (RAMOUT) 70_o busses respectively.

When the scratchpad address is applied to the array, and the "Write Enable" is made active, the data on the IOT address bus is written into the RAMs. In order to read from the scratchpad, the desired location must be specified with the scratchpad address and the "read enable" must be made active. The requested data is then transferred from the scratchpad to the IOT address bus.

Address Store

During the execution of an Initiate I/O Instruction, the Processor 10_p assembles the beginning (A) and the ending (B) addresses of the data field. The Processor then transfers the complete A address from the Proces-

sor register 10_p, to the IOT address bus. At the proper point of the IOT initiation sequence, the IOT generates the appropriate signals, then gates the Base Module and the LCP address bits to the scratchpad 10_{ps}. Now, with the channel's scratchpad location addressed and with the "Write Enable" active, the A address can be written into the scratchpad. Subsequently the Processor 10_p places the end (B) address on to the IOT address bus and again the IOT generates the proper control signals along with the Base Module and LCP address. This time, however, the IOT also generates ADDRESB, thus causing the address on the bus to be written into the B address area of the scratchpad (FIG. 5F). The beginning and ending addresses of the data field have now been stored in the channel's address memory scratchpad 10_{ps}. When the data transfer operation begins, these scratchpad locations will be accessed by the Data Transfer Module 10_{tc} (FIG. 5C).

Message Level Interface

As was previously described in reference to FIG. 2, the LCP Base Module 20_0 is typical of the other Base Modules in that each individual Base Module contains a Distribution Card 20_{od} which services up to eight LCP's. In addition, each LCP Base Module has a Maintenance Card such as 20_{om} and a Termination Card 20_{ot}.

The Distribution Card for each LCP Base Module provides an interface between the LCP Base Module and the Input-Output Translator 10_t of the Main System 10. As seen in FIG. 2, the message level interface 15 provides a channel to the IOT 10_t from each LCP Base Module by means of 25 lines. These lines are shown in FIG. 5E. The functions of each of these individually identified lines are listed in table VI herein below:

TABLE VI

(Refer to FIG. 5E also)

Signal Name	Description
ADDSEL	Address Select. This signal, when active, indicates that the IOT is connected to, or is attempting to connect to, a specific LCP. Once the connection is made, the LCP remains connected until the IOT drops ADDSEL.
Ag+SIO	Access Granted or Strobe I/O. If an LCP is not connected, this signal indicates that the LCPs request for reconnection has been granted, and initiates the "Poll Request" algorithm. If the LCP is connected, this signal is the IOT's acknowledgement for information received, or strobe for information transmitted.
TRM+MC	Terminate or Master Clear. If no LCP's are connected, this signal will cause all on-line LCP's to clear. If an LCP is connected, this signal will terminate the connected LCP.
LCPST	LCP Strobe. If an LCP is connected, this signal is the LCP's acknowledgement for information received, or the strobe for information transmitted. This signal is also used by the Distribution Card as an acknowledgement during Poll Test and Poll Request.
ER+ST8	Emergency Request or LCP Status 8. When activated by an unconnected LCP, this signal indicates that the LCP requires immediate access to the IOT. If activated by a connected LCP, this signal indicates that bit 8 of the LCP status is set.
IP+ST4	Interrupt Request, Poll Test Parity Error, or LCP Status 4. When activated by an unconnected LCP, this signal indicates that The LCP requires access to memory, i.e., the LCP is requesting a reconnection. If activated during a system-initiated connection sequence (Poll Test), this signal indicates that a parity error was detected during the Poll Test. If activated by a connected LCP, IP+ST4 indicates that bit 4 of the LCP status is set.
PB+ST2	Port Busy or LCP Status 2. When detected during a Poll Test, this signal indicates that the LCP Base is "busy". If activated by a connected LCP, PB+ST2 indicates that bit 2 of the LCP status is set.
CS+ST1	Channel Select or LCP Status 1. When activated by the IOT and transmitted to an LCP Base, this signal indicates "channel select", and that a connection or reconnection attempt has been initiated. If activated by a connected LCP, CS+ST1 indicates that bit 1 of the LCP status is set.
PARITY	Parity. This bidirectional line carries the proper (odd) parity for the information on the 16 data lines.
DATAxn	Data Lines (x=A, B, C, or D; n=1, 2, 4, or 8). In the unconnected state, these 16 bidirectional lines are used for addressing and priority resolution in connection or reconnection attempts. In the connected state, these lines are used for the transfer of data between the IOT and the LCP.

The message level interface 15 (MLI) which consists of 25 signal lines connecting the Distribution Card as 20_{od}, of a particular LCP Base Module as 20_0, to the IOT 10_t provides assurance that the signal discipline presented to the IOT is a standard one regardless of the variations of logic and operation found in the different types of LCP's. It will be noted that some of the MLI signal lines 15 shown in FIG. 5E are bidirectional, and are assigned multiple functions, depending on the source of the signal and the state (connected or disconnected) of the LCP.

The Distribution Card 20_{od} for a given LCP Base Module is used to provide a part of the Message Level Interface between the IOT and the individual LCP's within the Base Module. The Distribution Card also works on conjunction with the IOT Connection Module 10_{tb} to establish a data path to a specified LCP (Poll Test), and, upon request by an LCP, works with the IOT Reconnection Module 10_{td} to establish a path from that particular LCP to the IOT (Poll Request).

LCP Status Counts

During the time a particular LCP is connected, it follows a standard communication procedure with the IOT. Although the sequence of events followed in the communication procedure may not be identical for all LCP's, the events occurring in any one point in the sequence will be identical. The steps in the sequence, which are numbered 0 through 15, are called "Status Counts" and are transmitted to the IOT. The IOT examines the "Status Counts" each time it receives a strobe pulse from the LCP and, based upon that status count, takes appropriate action. More detail in the sequence and use of status counts will be provided hereinafter. FIG. 6A is a diagram showing the various status counts and the logic flow which they involve. Detailed explanation of this logic and the status counts involved will be provided hereinafter.

LCP Base Module Backplane

A local common backplane is provided in each of the LCP Base Modules 20_0, 20_1, 20_2, etc. Each backplane connects to all the eight LCP's in the Base Module. The backplane is constructed so that all signal lines are bussed the length of the backplane, thus making each line available to all LCPs in that Base Module. From the individual position of a single LCP, these backplane

lines fall into two general types: (a) those going to the Distribution Card and on to the IOT; and (b) those going to the Maintenance and Termination Cards. With the exception of the various clock and voltage lines, those lines going to the Maintenance Card, (such as for example, 20_{om} of FIG. 2) are used for local or off-line maintenance functions.

Of those lines which go to the Distribution Card, and on to the IOT, some, such as the data an the parity lines, must be gated to individual LCPs. This gating is enabled only when the LCP is in the "connected" state; when the LCP disconnects, the gating is disabled. The LCP is in a "connected" state when the LCP can transfer data between the IOT and itself. The "disconnected" state of an LCP is where the LCP is disconnected from the IOT, but is now able to transfer data between itself and its peripheral unit.

In addition to the gated lines, there are some lines which are dedicated to each individual LCP, for example, the line which goes from the Distribution Card to only one LCP. Those lines, which require no gating, are used for signals such as the LCP request for reconnection or the LCP address lines.

During the time an LCP is connected to the IOT, that LCP has the exclusive access to the Base Module Backplane. It is during this "connected" time that the IOT-LCP data transfer occurs. Upon cessation of the data transfers, the LCP disconnects from both the IOT and the Base Module Backplane, thus freeing them for use by other LCP's in the system. Once disconnected, the LCP is free to communicate, via the frontplane, with its associated peripheral device, such as device **50**. When a disconnected LCP requires that the connection to the IOT be re-established, that LCP sends a request signal, via one of its dedicated backplane lines, to the Distribution Card, such as 20_{od}. Reception of the LCP request causes the Distribution Card to begin the "Poll Request" algorithm and to initiate the IOT Reconnection Module, 10_{td}, FIG. 5C.

Line Control Processor

An LCP, Line Control Processor, is a device which is used as an interface unit between a specific peripheral device and the Main System. The LCPs are made in a variety of types, each designed to operate with a specific type of peripheral device. Since peripheral devices are different in their operational characteristics, the LCP is devised to handle, control and be particularly adaptable to its own specific peripheral device. However, there are certain general characteristics of the LCP interface unit which establish a common characteristic for all LCPs. Basically, the common characteristics of each LCP involve: the ability to transform serial data to parallel data or to transform parallel data to serial data; to transform format from character to word-format, or to transform from word to character-format; to recognize and take appropriate action in response to certain standard control characters or signals.

A generalized block diagram of a Line Control Processor is shown in FIG. 6B, which also indicates the relationship to Distribution Card Unit 20_{od} and IOT 10_t. If the LCP is assumed to be in the "connected" state, and that a "write" operation has been initiated, then data from the IOT 10_t enters the LCP through the backplane receivers 23_r. Then the Multiplexor 24_{x1} is used to select the "data source" for the operation, which in this case is the IOT 10_t.

The output of Multiplexor 24_{x1} is bussed to both the LPW (longitudinal partiy word) circuitry 24_w and also to the Multiplexor 24_{x2}, which gates the data from Multiplexor 24_{x1} into the data buffer 25_{00}. The LCP continues to receive data from the IOT 10_t until the data buffer 25_{00} is filled.

In the period that the LCP is receiving data, the LPW circuitry 24_w is generating the LPW sum; then at the end of the transmission, the IOT 10_t sends a longitudinal parity word (LPW) which, if there were no errors in the transmission, clears the LPW circuitry 24_w. If the circuitry 24_w does not clear, then an error is indicated.

When the data buffer 25_{00} is filled, the LCP disconnects from the Main System (IOT) by disabling its backplane transmitter drivers 23_x and backplane receivers 23_r; the LCP then establishes a data path to the peripheral device, such as **50**, by enabling its frontplane transmitter drivers 28_x and frontplane receivers 28_r. Once this path is established, the LCP uses Multiplexor 27_x to select data (translated or untranslated) from the data buffer 25_{00} to be transmitted to the peripheral device **50**. The transmission continues until the data buffer 25_{00} is empty, at which time the LCP requests a "reconnection" (to the IOT), either to store a Result Descriptor or to request more data.

If a "read" operation is in progress and the LCP is disconnected from the Main System (IOT), data from the peripheral device **50** enters the LCP via the frontplane receiver 28_r. The output of the receiver 28_r is bussed to Multiplexor 24_{x1}, which now selects the peripheral device **50** (through frontplane receiver 28_r) as the "data source". The output of Multiplexor 24_{x1} bypasses the LPW circuitry 24_w and goes on to Multiplexor 24_{x2}, which selects Multiplexor 24_{x1} as the input to the data buffer 25_{00}. When the data buffer 25_{00} is filled, the frontplane receivers 28_r and the frontplane drivers 28_x are disabled, then the LCP reconnects to the IOT 10_t, and the backplane receivers 23_r and backplane drivers 23_x are enabled.

The LCP now begins transmission (to the Main System **10**) of the data from the data buffer 25_{00}, through the Multiplexor 27_x and driver 23_x, over to the IOT 10_t. During this transmission, the output of Multiplexor 27_x also goes through the Multiplexor 24_{X1} over to the LPW circuit 24_w. When the data buffer 25_{00} becomes emptied, the LCP sends a signal to the IOT 10_t indicating that the longitudinal parity word, LPW, is coming, after which it then gates the final LPW sum through Multiplexor 27_x and driver 23_x over to the IOT 10_t.

After the transmission of the longitudinal parity word (LPW), the LCP may either disconnect from the Main System (IOT) in order to receive additional data from the peripheral device **50**, or, if there is no further data, the LCP may store a Result Descriptor and go on to an "idle" state.

In the above described operations, the informational data could have been transferred between the LCP and the peripheral device in the form of bits, characters, or words, depending on the type of peripheral device involved. The method of data transmission is typically controlled by the type of peripheral device used.

Typically, the informational data is transferred between the LCP and the IOT 10_t as "words", with some instances of character transfers, as for example, the first or the last character of a transmission. These data transfers between the IOT 10_t and the LCP of FIG. 6B are controlled by the exchange of strobe pulses, and the

recognition by the IOT 10_i of the LCP "status counts", to be described hereinafter.

As previously introduced in connection with FIG. 6A, the status count of an LCP provides standardized information which is transmitted to the IOT 10_i and which permits the IOT to take the next appropriate action based on the status count information.

During the time an LCP is "connected" to the Main System, if follows a standard communication procedure with the IOT 10_i. Even though the sequence of events followed in the communication procedures may not be identical for all LCPs, the particular events which occur at any one point in the sequence of communication procedure are all similar. The steps in the communication sequence, numbered 0 through 15, are called "status counts" and designated "STC". These status counts are transmitted to the IOT 10_i which examines the status count (STC) each time it receives a strobe pulse from the LCP, and, based upon that status count, the IOT can take appropriate action.

Referring to FIG. 6A and the following table, it will be seen that each status count has a particular function and further, depending on the type of LCP and Descriptor involved, the status count will have different exits. The following table VII briefly describes the various LCP status counts:

TABLE VII

Status Count	Description
STC=0	Master Clear
STC=1	Disconnect. The LCP is communicating with it's peripheral device.
STC=2	Not Ready. The LCP is idle. The peripheral device is not ready. The LCP can receive descriptor information from the System.
STC=3	Ready. The LCP is idle. The peripheral device is ready. The LCP can receive descriptor information from the System.
STC=4	Read. The LCP transmits data from its buffer to the System.
STC=5	Send Descriptor Link. The LCP sends the Descriptor Link to the IOT in order to re-establish connection.
STC=6	Receive Descriptor Link. The LCP receives the Descriptor Link from the IOT during the IOT "connection" sequence.
STC=7	Result Descriptor. The LCP transmits its Result Descriptor to the IOT.
STC=8	Write. The LCP receives data from the System.
STC=9	Encoded Status. One character transmitted; LCP sets D1 bit (FIG. 4C Result Descriptor) as a flag to the IOT. The IOT decrements the address by 2.
STC=10	Write One More Word. The LCP data buffer can hold only one more word.
STC=11	I/O Descriptor LPW. The LCP receives and checks the LPW for the I/O Descriptor received in STC=2 or STC=3. The I/O Descriptor after being translated by the IOT, then becomes known as the Command Descriptor.
STC=12	Break. There is no more data to be transferred. The LPW is transmitted and checked.
STC=13	Break Enable. Data transfer has been halted; the LCP is requesting a return to STC=8 (Write) or to STC=4 (Read).
STC=14	Character Transfer. The last transmission consisted of a character instead of a word.
STC=15	Result Descriptor LPW. The LCP sends the LPW for the Result Descriptor to the IOT.

Referring to FIG. 5C, the Processor 10_p starts the chain of input-output operations by the execution of an initiate I/O Instruction. In this situation, the Processor passes certain information, including the channel number of the desired LCP over to the IOT Initiation Module 10_{ia} of FIG. 5C. The channel number is decoded to determine the Base Module number and the address of the LCP, which are then passed over to the Connection Module 10_{ib}. The Connection Module then selects the proper LCP Base Module and sends a signal (channel select) to the appropriate Distribution Card, as 20_{od}, for that Base Module, as 20_0, requesting that a connection attempt be made. The above described operation is called a "Poll Test" and is a means for the Main System to seek connection to an LCP; it is, further, a method by which the Distribution Card 20_{od}, in response to the connection request, also attempts to connect to a specific LCP.

Following the transmission of a "Channel Select", the IOT 10_i sends the address of the desired LCP to the Distribution Card in the selected Base Module. At the same time, the IOT sends "Address Select" to all Base Modules in the system. The Distribution Card that receives both the Address Select and Channel Select begins a "Poll Test" and responds to the IOT with an "LCP Strobe"; the Distribution Cards that received the Address Select only, consider it as a "busy" signal, and they are inhibited from communication with the IOT. When the IOT 10_i receives the LCP Strobe, it drops the Channel Select.

When the Distribution Card receives an "Address Select" and "Channel Select", a signal is generated which enables the LCP address to be placed into an LCP address register in the Distribution Card. The BCD (Binary Coded Decimal) output of the LCP address register is decoded to enable one of eight lines. Each line represents one LCP in the Base Module. When an LCP detects that its address line is active, then that LCP responds to the Distribution Card with the signal LCPCON meaning "LCP connected". When this connected signal is received in the Distribution Card, a connect flip-flop (CONF) is set. Then depending on the state of the I/O send line (IOSND/FIG. 6C) from the connected LCP, this will cause an activation of control lines for either receiving data or sending data as between the LCP and the IOT (FIG. 6C).

If a Distribution Card detects the absence of Channel Select, it responds to the IOT with the LCP's status, accompanied by a strobe. The LCP is now connected to the IOT and remains connected until the IOT drops Address Select; the Distribution Card takes no further part in the IOT-LCP communications.

The above events show the steps leading to a successful "connection" attempt; however, the connection attempt could have failed due to one of the following causes:

(a) there was no LCP at the location addressed or the LCP at the address location was off-line;
(b) the LCP was busy, that is the LCP status count was not 0 or 2 or 3;
(c) the port was busy, that is, a second Distribution Card in the Base Module was busy;
(d) a parity error was detected in the address.

The detection of any of these errors would cause the connection attempt to be aborted and a Result Descriptor indicative of the type of failure to be written and sent to the Main System in 10_{mr} of Memory 10_m (FIG. 3).

In subsequent discussions, reference may occasionally be made to specific flip-flops and signal levels which are not specifically shown within the block diagrams. Since the design and use of such elements are well known, it is considered to be redundant and overcomplex to show all such elements.

Poll Request

An LCP, after having been connected to the IOT 10_t and receiving the Command Descriptor and the Descriptor Link, may "disconnect" from the Main System 10 in order to communicate with its associated peripheral device, such as 50. If that LCP subsequently requires access to Memory 10_m, it sends a request (LCPRQ) over to the Distribution Card. The "Poll Request" is the method by which the Distribution Card, in response to the LCP's request, attempts to reconnect the LCP to the IOT. A number of events occur during a "Poll Request" operation.

If several LCPs within the Base Module 20_0 simultaneously request access, the Distribution Card 20_{od} determines which one of them is to gain access by checking their priority levels; thus, the requesting LCP which has the highest priority level (this priority selected at installation time) is given access to the Distribution Card. This priority level is called "Base Priority" as it involves which LCP has what level of priority as among the eight LCPs residing in that particular Base Module.

Once the "Base Priority" is resolved, the Distribution Card assigns a "Global Priority" (which has also been assigned and selected at installation time) to the requesting LCP. The "Global Priority" establishes the priority rank between different Base Modules in the overall system rather than just the priority rank of LCPs in one single Base Module. The Distribution Card 20_{od} contains a series of pins or socket-type connections which are connected to each individual LCP. These pin-socket connections can be jumpered (by a field engineer) to a priority encoder which assigns an internal base priority number from zero (low) to seven (highest) to each LCP. Thus, if several LCP's in the same Base Module request connection concurrently, then the Distribution Card control means will put through the LCP with the highest priority.

Another set of pin-sockets on the Distribution Card are connected to each LCP. These are "jumpered" or "strapped" by a field engineer so that each LCP is given a "global" or external priority number to permit the Input/Output Translator interface of the Main System to select amongst LCP's which reside in different Base Modules of the system. Thus, when the "global" priority number is received by the IOT, and there are concurrent requests from other LCP's in other Base Modules, the IOT will select the LCP with the highest global priority number, but this occurs only after internal base priority has been resolved by the Distribution Card.

Those Distribution Cards receiving requests from their associated LCPs, each send an "Interrupt Signal" (IP+ST4) over to the IOT 10_t. (See message level interface FIG. 5E and Table VI). When the IOT 10_t detects the signal IP+ST4, it begins the "reconnection" sequence and sends a signal (Access Granted) to all the Base Modules in the system. The "Access Granted" signal causes those Distribution Cards that sent the IP+ST4 to the IOT 10_t to begin their individual "Poll Request" algorithms.

In response to the "Access Granted" signal, the requesting Distribution Cards send their individual Global Priorities over to the IOT 10_t. The IOT compares the Global Priorities of the requesting Distribution Cards (that is, sends the Channel Select signal over to the requesting Distribution Card which has the highest Global Priority one clock-time later) and the IOT sends an Address Select signal to all Distribution Cards in the system. The Distribution Card that receives both the "Channel Select" and the "Address Select" responds to the IOT with the LCP Strobe, then sets its LCP Address flip-flop, thus driving the specific address line of the requesting LCP. When the LCP detects that its own address line is active, it then responds to the Distribution Card with the LCP connected signal (LCPCON).

Upon receipt of the LCP Strobe, the IOT 10_t drops "Access Granted" signal and the "Channel Select" signal; and when the Distribution Card detects the absence of the "Access Granted" and the "Channel Select" and detects the presence of LCPCON, it then assumes a connection to be completed and responds to the IOT with an LCP Strobe, accompanied by the LCP Status Count and the Descriptor Link.

The Poll Request is now complete; the Distribution Card takes no further part in the LCP-IOT communication. The LCP and the IOT continue with the reconnection sequence until the LCP is connected, after which control is passed to the IOT Data Transfer Module 10_{tc}, FIG. 5C. The LCP remains connected until the time when the IOT drops its "Address Select" signal.

Error Checks

Each transmission between the IOT and a particular LCP is checked for errors. The error checking methods used are

(a) vertical parity checking on each word transmitted, and
(b) longitudinal parity checking on each block transmitted.

(a) Vertical Parity

In "Read" operations, the LCP sends information to the IOT 10_t on 16 message level interface (MLI) data lines, (FIG. 5E) accompanied by the parity bit on the MLI parity line, FIG. 5E. The data and parity lines go to a parity generator-checker on an IOT base driver card. In "Read" operations, the parity generator checker is used to count the number of 1-bits on the MLI data and parity lines. If the total number of 1-bits (including the parity bit) is odd, then parity is correct and a signal term (PAROK, FIG. 6D) from parity-generator 48 is generated. If the total number of one bits is even, then the PAROK signal is not generated; the absence of PAROK at the time that data is received, causes the IOT to set a vertical parity error flip-flop (VPERRF).

Similarly, in "Write" operations the 16 data lines from the Main System 10 are bussed to a parity generator-checker on the IOT base driver card. The data on the 16 lines is examined and if an even number of 1-bits is detected, the term PARGEN is generated. This PARGEN signal is then used to force a "1" bit onto the message level interface parity line to accompany the data to the LCP. On the LCP Base Distribution Card, the state of the parity bit controls the parity generator-checker circuit. The parity generator-checker circuit examines the states of the 16 data lines and generates PAROK if the total number of 1-bits, including parity, is odd.

(b) Longitudinal Parity Checking

Longitudinal parity checking is an error detection method in which a check word generated by a sending unit is compared to a check word generated in the same

ıanner by a receiving unit. These check words are enerated by treating each word in the transmission as 16-bit number, then performing an exclusive OR operion (binary addition without carry) of each word in ıe transmission. At the end of the transmission, the ending or transmitting device sends the check word it as assembled over to the receiving device. If there ave been no errors in the transmission, the addition of ıe check word from the transmitting device to the heck word in the receiving device results in a sum of 0". Thus, if the sum is not "0", a longitudinal parity rror flip-flop is flagged (LPERRF).

As was discussed in connection with FIG. 6B, the .CP was provided with LPW circuitry 24_w. Likewise, ıere is longitudinal parity checking circuitry in the OT 10_t. This circuitry connects in a parallel path to the ata bus shown as the lower 16 lines of FIG. 5E.

The Line Control Processor (LCP), such as element 0_{00}, may be better understood with reference to FIG. C which represents a basic block diagram of the major lements involved in addition to some specific details ith regard to the RAM buffer such as 25_{00} of the LCP, 0_{00}.

The LCP buffer 25_{00} is a random access memory RAM) which is functionally 256 bits (0–255) wide and 8-bits deep. It can thus hold 256 words of 18-bits each. n one typical embodiment, the buffer 25_{00} may have a ection designated buffer A, 25_a, having provision for **90** ongitudinal words of **18** bits each; another section desgnated 25_{x1}; a Command Descriptor C/D section desigated 25_c; a buffer area B, 25_b which may typically be **90** ords long, (i.e., from address **128** over to address **218**); nother buffer area designated 25_{x2}; a Result Descriptor /D area 25_r; another area designated 25_{x3}; and a Deriptor Link D/L area designated 25_d.

The RAM buffer 25_{00} is addressed by a memory adress register **36** having a system address register secon **36**, and a device address register section 36_d, which ommunicate to the buffer 25_0 via an eight-bit address us, B_8. The RAM buffer 25_{00} is functionally composed ı the vertical direction (FIG. 6C) of 16-bits plus a arity bit, plus an eighteenth bit called an "end flag bit", he end flag bits residing in a storage section designated s 25_e.

A "data bus" **47** provides a data input and output hannel for the buffer 25_{00} to communicate to the Main ystem **10** through the system interface logic 21_{si}; and or the buffer 25_{00} to communicate to its peripheral unit ia a device interface 22_{di}. The system interface logic 1_{si}, the device interface logic 22_{di}, and the common ogic 22_c schematically represents blocks which refer to ıore specific elements which are described in connecion with FIG. 6D.

Referring to FIG. 6F, there is shown a "message lock" of the type used in the LCP buffer 25_{00} of FIG. C.

As mentioned with the discussion of FIG. 6C in reard to the RAM buffer 25_{00}, this is typically a message lock of "*n*" words, which block provides **90** words (or = 90) for data storage; and also there may be proided three words for Result Descriptors R/D; there ıay be provided three word locations for Command Descriptors C/D; and there may be one word location or Command Message C/M.

FIG. 6F also shows the basic word format, in that a vord is composed of four digits which are: A, B, C, and D plus a parity bit marked VPB (vertical parity bit), vhich normally makes a total of 17-bits per word.

As seen in the drawing of FIG. 6F, the four digits A, B, C and D are each made up of four bits designated as the "8" bit, the "4" bit, the "2" bit, and the "1" bit.

In FIG. 6C, the buffer 25_{00} is also provided with an 18th bit or "end flag" bit which is placed in the location designated 25_e of FIG. 6C.

The central or Main System **10** communicates with the peripheral terminal unit via the LCP. The LCP provides the means for transferring control information and data from the Main System **10** to the peripheral terminal units, such as **50**, and vice-versa. The LCP looks at the Command Descriptor C/D received from the Main System **10** and sets itself up to perform the operation required if it is sensitive to that particular command. It also transfers the same Command Descriptor C/D unmodified to the peripheral terminal unit. The peripheral terminal unit acts upon the Command Descriptor C/D and returns Result Descriptors R/D to the Main System **10** via the LCP. The message block and the word formats have been shown in FIG. 6F. Typical Command Descriptors C/D and Result Descriptors R/D will be shown subsequently hereinafter.

The LCP accepts the Command Descriptor C/D transmitted by the Main System **10**. The C/D contains a digit of the OP code, 3 digits of variants, and 6 digits of C address. The Command Descriptor C/D is received by the LCP via 4 digits per transmission for a total of 3 words (4 digits per word). The two least significant digits contain all zeros. With each word there is a vertical parity bit (VPB) and the entire C/D is followed by a longitudinal parity word (LPW). Should a parity error be detected on transmission of the C/D, the LCP will branch to a Result Descriptor R/D mode and report a descriptor error to the Main System **10**.

The random access memory buffer 25_{00} (RAM of the LCP) buffers the entire Command Descriptor, the vertical parity bit and the longitudinal parity word within the LCP, Line Control Processor.

The LCP examines the first word of the Command Descriptor C/D and determines whether it is an ECHO OP, HOST LOAD OP, or READ NO timeout OP. If it is one of these, it sets the appropriate flag.

Descriptor Link (D/L)

Following the receipt of the Command Descriptor C/D, the Line Control Processor LCP proceeds to accept the Descriptor Link D/L. This is a two word transmission followed by a longitudinal parity word LPW. Should there be an error, the LCP branches to the Result Descriptor R/D mode, and reports a descriptor error to the System **10**.

The random access memory RAM of the buffer (such as 25_{00}) acts as the buffer for the entire Descriptor Link D/L, the vertical parity bit (VPB) and the longitudinal parity word LPW.

Disconnect Mode

Following the receipt of the Descriptor Link D/L, the LCP goes to the "disconnect mode".

Reconnect Mode

If it is an ECHO OP, the Line Control Processor LCP proceeds to "reconnect mode" and starts operating on the ECHO OP which involves the receiving of two buffers of data (each 180 bytes, or 90 words of 16-bits) and the transmitting of the same data back to the System Memory 10_m.

If it is other than an ECHO OP, the LCP examines the readiness of the peripheral terminal unit. Should the peripheral device be in the "not-ready" state, the LCP branches to the Result Descriptor R/D mode and reports this to the System **10**.

If the peripheral device is "ready" the LCP starts communicating the Command Descriptor C/D to the peripheral device, while at the same time branches to the "idle" state to make itself available for a possible "Conditional Cancel OP". The Line Control Processor LCP stops in this "idle" state until one of two things happen:

1. The peripheral device sets up the Line Control Processor LCP to a "data transfer" state.
2. The System **10** communicates a "Conditional Cancel OP" or an Unconditional Cancel.

If it is number (2) above, the Line Control Processor LCP accepts one word from the System **10** followed by the longitudinal parity word LPW, and the LCP determines if it is a valid Conditional Cancel OP. In any case the LCP communicates this to the peripheral device. If the situation involves number (1) above, the LCP branches back to the "disconnect" state, where data transfer between the LCP and its peripheral can occur.

After transmission of the Command Descriptor C/D to the peripheral, the LCP is driven by the peripheral device "state", which defines the operation mode and the memory requirements. Data is transferred in "message blocks" together with a longitudinal parity word (LPW) of 16-bits following each block and with a parity bit on every word (except in a disk pack controller situation, the message block would consist of a segment). If the Line Control Processor LCP detects an error on data received from the peripheral device or from the Main System **10**, it reports this information to the peripheral device and then branches to the Result Descriptor R/D mode and reports it to the Main System **10**.

In the "Read" mode, the data transfer between the Line Control Processor LCP and the peripheral device is dependent on the requirements of the peripheral device. On the other hand, data transfer between the LCP and Main Memory $\mathbf{10}_m$ is dependent upon the memory access rate of the Main System **10**. Since the peripheral device may operate in a "stream" mode, and the LCP must compete with other LCP's for access to memory, the LCP alternates between its two buffer areas to accommodate the transfer rate of the peripheral device.

Table VIII below indicates certain types of Command Descriptors C/D which are used and acted on by the LCP. All other C/D's are transparent to the LCP and pass through to the peripheral device:

TABLE VIII

Command Descriptors	
The LCP is transparent to all Command Descriptors except for the following as determined by testing the first word of the C/D:	
1. ECHO OP	(bit A1 is true)
2. HOST LOAD	(A4 and B8 are true)
3. READ NO T/O (timeout)	(A8 and B8 are true)
4. CONDITIONAL CANCEL OP	(A2 and B8 are true).
5. UNCONDITIONAL CANCEL	
OP code digits of the C/D are defined as follows:	
Read (A8) -	Any operation where data is transmitted from the LCP buffer to the Main System. (1000)
Write (A4) -	Any operation where data is transferred from Main System Memory to LCP buffer. (0100)
Test (A2) -	Any operation where no data transfer takes place between LCP and System Memory but results in a R/D storage in System Memory. (0010)

TABLE VIII-continued

Echo (A1) -	Operation that results in receiving a message block from System Memory and then transmitting the same block back to System Memory. (0001).

Normally Result Descriptors R/D are generated by the peripheral unit and accepted by the LCP in one, two or three words. When the LCP generates a R/D, only one word is sent to the Main System **10**. Table IX shows the conditions for the LCP to generate a Result Descriptor:

TABLE IX

Result Descriptors	
Bits	Condition
A8	Not Ready
A4	Descriptor Error
A2	System Vertical Parity Error
A1	System LPW Error
B8	Time-Out
B4	Remote Device Vertical Parity Error
B2	Remote Device LPW Error
B1	(blank)

Referring to FIG. **6C** with respect to the lines between the device interface $\mathbf{22}_{di}$ and the peripheral unit, the peripheral device unit may be provided with a port interface which may be designated as a DDP or device dependent port interface, $\mathbf{50}_d$, which is tailored to the requirements of each specific type of peripheral device.

The LCP communicates to the peripheral via the DDP in an asynchronous mode. The "Write" operation is defined as a transfer where the LCP is writing into the peripheral device unit. The "Read" operation is defined as a transfer where the LCP is reading from the peripheral device unit.

Referring to FIG. **6C** the line marked HTCL/ may be designated as the Host Transfer Control Level, and when the LCP "Writes" into the peripheral device unit, this signal is the asynchronous level, which signifies the presence of data on the data lines. This level is deactivated by the peripheral unit sending DML/ (peripheral message level) or by sending DINTL/ (peripheral device interrupt level) to the LCP.

When the LCP is "Reading" data on a Result Descriptor R/D from the peripheral unit, this HTCL/ signal is the asynchronous acknowledgement that the data on the data lines has been received by the Line Control Processor LCP. Upon receipt of this level, the peripheral device unit must deactivate DML/ or DINTL/. When the peripheral unit causes the de-activation of DML/ or DINTL/, then the LCP de-activates HTCL/ (the Host Transfer Control Level).

When the peripheral device unit drives the LCP to the Command Message C/M mode, the Host Transfer Control Level HTCL/ is sent to the peripheral device unit when the LCP's buffers are empty and no system terminate has been detected. The HTCL/ must be answered by the peripheral device unit with a DINTL/ and a change-of-state.

The line in FIG. **6C** marked HINTL/ is designated as the Host Interrupt Level and is used by the LCP to indicate to the peripheral unit that the LCP wishes to interrupt the operation. The response to this level by the peripheral device must be DINTL/ and a change-of-state, to which the LCP responds by de-activating its Host Transfer Control Level, HINTL/. Following the detection of the trailing edge of HINTL/, the LCP will

respond to the new mode of operation described by the state line shown on FIG. 6C as ST-4/, ST-2/, ST-1/.

When an interrupt from the System 10 is activated in the "Write" mode, the Host Interrupt Level HINTL/ signifies that the last word of data has been transmitted and the LPW is on the data line of bus 47. The peripheral unit needs to respond to the interrupt with a DINTL/ and a change-of-state.

In the "Read" mode when the LCP detects the "Read Terminate Command", the LCP will activate the Host Interrupt Level HINTL/. In the Command Message C/M mode, the LCP will activate the Host Interrupt Level HINTL/ if a "Read Terminate" has been detected.

The line of FIG. 6C designated HCL/ refers to "Host Clear" which indicates to the peripheral unit that the LCP is being cleared by the Main System 10, or that a parity error has occurred during a read.

A combination of the Host Transfer Control Level and the Host Interrupt Level (HTCL/ - HINTL/) indicates to the peripheral unit the presence of a Host Load Command Descriptor C/D. The peripheral unit responds by activating the line marked DINTL/ (peripheral interrupt level) and the Status Count ST = 2; the CP acknowledges by de-activating both levels of HTCL/ - HINTL/. Following the trailing edge of DINTL/, the LCP transfers data in the "Write mode".

In FIG. 6C a bidirectional data bus B_d is provided having 16 data lines and a parity line between the LCP and the peripheral unit. When controlled by the LCP, these lines are active as long as the Host Transfer Control Level HTCL/ is active. When control is held by the peripheral unit, these lines are active as long as the peripheral device message level DML/ is active. The direction of transfer is determined by the status of the peripheral unit. The line designated DML/ refers to the peripheral device message level and is a unidirectional line. When the LCP is reading data or a Result Descriptor R/D from the peripheral unit to the LCP, the peripheral device message level DML/ is used as a transit signal to indicate the presence of stable data on the data lines. When the peripheral device receives a Command Descriptor C/D or data from the LCP, this signal, DML/, is used as an acknowledge level for data.

The peripheral device (via its port interface) uses the DINTL/ (peripheral interrupt level) to request the LCP to change its mode of operation. This is done by activating DINTL/ and presenting the proper state on the state lines, ST-4/, ST-2/, ST-1/. The state lines must be stable during the time that DINTL/ is active.

change-of-state to occur for either the above. When the LCP is writing into the peripheral unit, the peripheral device interrupt level DINTL/ is based on the leading edge of HTCL/ or HINTL/. DINTL/ is de-activated by the trailing edge of these signals (HTCL/ - HINTL/).

In the "Read Mode"

The peripheral interrupt level DINTL/ is a no-data transfer "strobe" used exclusively to change states, DINTL/ is acknowledged by the Host Transfer Control Level HTCL/ in the Read Mode. When the LCP is reading from the peripheral device unit, the peripheral device activates DINTL/ instead of the peripheral message level DML/, and de-activates DINTL/ when the peripheral unit detects the leading edge of the Host Transfer Control Level HTCL/.

In the Host Load Mode

This mode involves the transfer or loading of data from the peripheral device, as 50, FIG. 6C, into the LCP (Host) for the "Read Mode" and vice versa for the "Write Mode".

The peripheral device interrupt level DINTL/ is the acknowledge level to HTCL/ - HINTL/ as the Host Load Command. The peripheral device activates DINTL/ and changes to State 2 (Table X). The LCP acknowledges this by de-activating both HTCL/ - HINTL/, and if in the "Write Mode", starts writing into the peripheral device memory. To interrupt this mode the peripheral device unit 50 activates DINTL/ in the same manner as in a regular "Write Mode".

The "State"Lines

In FIG. 6C, these unidirectional lines ST-4/, ST-2/, ST-1, indicate to the LCP the state of the peripheral device, and from this, the LCP determines what kind of operation mode is required. For example, in a typical embodiment, there may be eight states, 0–7, as seen in Table X, for the peripheral device which might be used to indicate the following conditions: peripheral device not on-line; Read operations; Write operations; Result Descriptor; Command Message (C/M); reset LCP timer (RT); ready or writing Command Descriptor (C/D); last word of a block or the Result Descriptor and longitudinal parity word (R/D-LPW) is next to be transmitted.

A typical coding system for the state lines from a typical peripheral device unit is shown herein below in Table X:

TABLE X

	State Lines					
ST=	ST-4/	ST-2/	ST-1/	Condition	Data Line Direction	
0	0	0	0	Peripheral Device Not On Line		
1	0	0	1	Read	From	Peripheral Device
2	0	1	0	Write	To	Peripheral Device
3	0	1	1	R/D	From	Peripheral Device
4	1	0	0	Command Message	No Data Xfer	
5	1	0	1	Reset HD Timer (RT)	No Data Xfer	
6	1	1	0	Ready or Writing C/D	To	Peripheral Device
7	1	1	1	Last Word of a Block or R/D-LPW is next	To or From	Peripheral Device

In the "Write Mode"

DINTL/ is the acknowledge level to the Host Transfer Control Level HTCL/ and the LPW data word; or else it is the response to HTCL/ or HINTL/ in the Command Message C/M mode. DINTL/ will cause a

The interface discipline between the LCP and the peripheral device unit via the peripheral device unit port interface (DDP 50_d, FIG. 6C) may again be looked at in terms of a "Reading Mode" and a "Writing Mode".

Reading Mode

With the Line Control Processor LCP reading from the peripheral device unit (State 1 + 7), the peripheral device unit (as **50**, FIG. **6C**) places a word on the data lines and activates the peripheral device message level DML/. The LCP acknowledges this by activating the Host Transfer Control Level (HTCL/). The peripheral device unit now de-activates DML/, and then the LCP de-activates the HTCL/. This process continues in State = 1 until:

1. The LCP activates the Host Interrupt Level (HINTL/). The peripheral unit acknowledges by de-activating the peripheral message level (DML/), if active and activates the peripheral device interrupt level (DINTL/) with a change-of-state. This indicates to the peripheral device that the LCP has a "Command Message" C/M to send to the peripheral device.

2. The peripheral device activates the peripheral device interrupt level DINTL/ instead of the peripheral message level DML/, with the proper change in the State Lines. The LCP acknowledges by activating the Host Transfer Control Level (HTCL/), and, following the de-activation of DINTL/, it de-activates the Host Transfer Control Level HTCL/ and goes on to the proper State. DINTL/ does not transfer data on the data lines.

3. When the peripheral device detects it is transmitting the Last Word of a block, the peripheral device changes to Status ST = 7 with the leading edge of DML/. The LCP answers the peripheral device with a Host Transfer Control Level (HTCL/) and expects the next transfer to be the longitudinal parity word LPW. The LPW is transmitted with the peripheral message level DML/ and answered with a Host Transfer Control Level (HTCL/).

4. If the LCP detects a vertical or longitudinal parity error, the LCP will not acknowledge the peripheral message level DML/ from the peripheral device. Instead the LCP will generate a Host Clear Level (HCL/).

In the Writing Mode

If the LCP is writing data into the peripheral device (State = 2 + 7), the following actions take place:

The LCP places a word on the data lines and activates the Host Transfer Control Level (HTCL/). The peripheral device acknowledges by activating the peripheral device message level (DML/). The LCP now de-activates the Host Transfer Control Level (HTCL/), and then the peripheral device de-activates the peripheral message level (DML/).

This process continues (Table X) in Status ST = 2 until:

1. The peripheral device changes state to ST = 7, then activates the peripheral message level DML/ which flags the LCP that the Last Word of that block has been received. The next word in the data lines must be a longitudinal parity word LPW when the Host Transfer Control Level HTCL/ becomes active again. Then the peripheral device activates the peripheral interrupt level DINTL/ instead of the peripheral device message level DML/, accompanied by a change in the State Lines.

2. At ST = 2 or ST = 7, the LCP activates the Host Interrupt Level HINTL/ instead of the Host Transfer Control Level HTCL/. In this mode, HINTL/ signifies an interrupt and that a longitudinal parity word LPW is on the data lines. The peripheral device acknowledges by activating the peripheral interrupt level DINTL/ and a change-of-state. The LCP de-activates the Host Interrupt Level HINTL/ and goes to the proper mode after DINTL/ is de-activated.

In another mode called the "Result Descriptor R/D Mode", the LCP reads a Result Descriptor R/D from the peripheral device (State = 3 + 7). When in the R/D Mode, the LCP is reading the Result Descriptor on the data lines from the peripheral device. The Result Descriptor R/D can be from 1 to 3 words long plus a longitudinal parity word LPW. The first and second words of the 3-word Result Descriptor R/D are read in Status ST = 3. The last word of the Result Descriptor R/D is read in Status ST = 7. The peripheral device message level DML/ signifies there is stable data on the data lines. Each Result Descriptor R/D word transferred is then acknowledges with a Host Transfer Control Level HTCL/. If a 1-word Result Descriptor R/D is received by the LCP, then data transfer occurs after going from Status ST = 3 to Status St = 7 together with a peripheral device message level DML/ which signifies a 1-word Result Descriptor R/D. The next word on the data lines is the R/D longitudinal parity word LPW which is strobed by the peripheral device message level DML/. After the LCP finishes reading a complete Result Descriptor R/D together with its appropriate longitudinal parity word LPW, the peripheral device returns to Status ST = 6. It can now accept a Command Descriptor C/D.

Command Message C/M Mode

This involves the situation in which the LCP in writing a Command Message into the peripheral device (State ST = 4). When the LCP is in the "Read" mode and is directed to the Command Message C/M mode (DINTL/ + ST = 4), the LCP continues to send data to the Main System **10** until:

1. The "read-system terminate" is detected which results in activating the Host Interrupt Level HINTL/ or:

2. Data buffer areas A and B (of buffer $\mathbf{25}_{00}$, FIG. **6C**) are empty and the "read-system terminate" is not detected. This causes the LCP to activate the Host Transfer Control Level HTCL/, indicating that the main System **10** expects more data.

The Reset Timer (R/T) Mode

This occurs when the peripheral device resets the LCP timer (State ST = 5). A change-of-state to ST = 5 resets the LCP timer. This change-of-state occurs without a strobe. The peripheral device unit must remain in ST = 5 for at least 500 nanoseconds.

The Send Command Descriptor (C/D) Mode

In this case the LCP is writing a Command Descriptor C/D into the peripheral device (State = 6). In this send Command Descriptor Mode C/D, the LCP writes 3-words followed by a longitudinal parity word LPW. The Host Transfer Control Level HTCL/ that accompanies the C/D and LPW is acknowledged by the peripheral device interrupt level DINTL/ and a change to the proper state.

The Last Word of Block Mode

This is the State = 7 (of Table X) and during a "Read" operation with ST = 7, the LCP is reading the last word of a block of data (or else a Result Descriptor

:/D) from the peripheral device. The next word will e an LPW. During a "Write" operation with St = 7, ıe LCP is writing the last word of a block into the eripheral device. The next word will be a longitudinal arity word LPW.

Conditional Cancel

After the LCP writes the Command Descriptor C/D ıto the peripheral device unit and before the peripheral evice changes from Status ST = 6 with the peripheral ıterrupt level DINTL/, the Main System **10** can termiate the operation (OP) by issuing a "Conditional Canel". In this case, the LCP de-activates the Host Trans:r Control Level HTCL/ and then activates the Host ıterrupt Level HINTL/ as long as the Status ST = 6 nd DINTL/ is not active.

Unconditional Cancel

The Main System **10** can generate an "Unconditional 'ancel". This causes the LCP to generate the Host 'lear Level HCL/ to the peripheral device. No acnowledgement is required from the peripheral device.

The LCP (Line Control Processor) Subsystem consts of a number of individual LCPs which communiate to the Main System **10** through the IOT 10_i. While ıch of the several LCPs have basically the same design ıd provide the same basic system functions, there are ariations of a minor nature as between the various 'pes of LCPs, since each LCP is tailored to meet the perational requirements of the particular peripheral :rminal unit that it services.

The discussion following herein will involve an oper:ional description of one preferred embodiment of a articular LCP which is provided for a peripheral terıinal unit known as the "Supervisory Terminal".

The necessary functional elements of the LCP in.ude registers, counters, encoders, decoders, busses, ıgic elements, etc. In addition there is a large scale ıtegrated (LSI) receiver/transmitter for implementing ɔmmunication between the LCP and its peripheral :rminal unit. Within the LCP, there are functionally vo divisions that are used for communication between ıe LCP and the Main System **10**. They are designated ; the "read module" and "write module". These modles exist "functionally", but they are not separate comɔnents, since many of the logic levels of which they :e composed are shared by both modules. The "read ıodule" is used to transfer data from the LCP over to ıe Main System **10**, and is active when the transmit ip-flop (XMITF) in the LCP is set. The "write modle" is used to transfer data from the Main System **10** ver to the LCP, and is active when the receive flipop (RECVF) is set.

Functionally, the components of the LCP are coniined in three major sections: (A) Terminal Control; 3) Data Flow; and (C) System Logic Section. In order ı understand the means by which the LCP communiıtes with the Main System **10** and with the associated :ripheral terminal unit, such as **50**, the functional char:teristics of the following components will be disıssed:

A. Peripheral Terminal Control Section
1. Universal Asynchronous receiver/transmitter (UART).
2. UART Multiplexor.
3. Block check character register (BCCR).
4. Block check character decoder.
5. End code decoder.
6. Memory address register.

B. Data Flow Section
1. Input Multiplexor
2. OP code register.
3. Variant register.
4. Valid OP encoder.
5. LCP buffer (RAM).
6. Terminal bus multiplexor.
7. Terminal bus.
8. Vertical parity generator/checker.
9. Data Latch register.
10. Longitudinal parity word (LPW) register.
11. LPW encoder.
12. End code decoder.

C. System Logic Section
1. Status Count (STC) register.
2. STC decoder.

The above mentioned functional components will be understood with reference to FIGS. **6B**, **6C**, **6D**, **6E** and **6F**, with particular references to FIG. **6D**. Examples of types of interconnections between peripheral devices and I/O interface units may be found in U.S. Pat. Nos. such as 3,510,843; 3,514,785; 3,526,878. Examples of the circuitry involved in communication between remote units and corresponding buffer registers in a typical fashion can be found by reference to U.S. Pat. No. 3,390,379.

With reference to FIG. **6D** and the Peripheral Terminal Control (Section A) previously mentioned, the universal asynchronous receiver transmitter (UART) **31** is used as the interface between the asynchronous serial data channel of the terminal unit device interface 22_{di} and the parallel data transmission channel of the LCP. The transmitter section of the UART **31** converts a parallel data character and the control levels into serial information containing a start bit, data, a parity bit, and a stop bit. The receiver section of the UART **31** converts serial information, containing a start bit, data, a parity bit, and a stop bit, into a parallel data character. The UART **31** generates a parity bit for information transferred to the terminal unit device interface 22_{di}, and it also checks the vertical parity of information received from the device interface terminal unit 22_{di}.

The UART **31** has provisions for selecting various character lengths, odd or even parity generation/checking, and a choice of one or two stop bits. For use with a particular LCP, the UART **31** has options selected to provide the following characteristics:

(a) a character containing seven data bits;
(b) generation/checking of even vertical parity;
(c) one stop bit.

The UART Multiplexor 27_x accepts an 8-bit character from either the AB (first two) digits of the terminal bus **47** or from the block check character register (BCCR) **33**. The selected input is sent to the parallel data input bus of the UART **31**. The UART Multiplexor 27_x is used only for the transfer of data or for a block check character from the LCP over to the terminal device interface 22_{di}.

The Block Check Character Register (BCCR) **33** is a register which consists of either separate flip-flops operated in the "toggle" mode, with inputs connected to the AB digits of the terminal bus **47**. While the LCP is transferring data to the terminal device interface 22_{di}, the BCCR **33** accumulates a block check character (BCC) to be sent to the device interface terminal unit 22_{di}. When the LCP is receiving data from the device interface terminal unit 22_{di}, the BCCR **33** also accumu-

lates a "block check character" to be checked against yet another "block check character" (BCC) sent from the device interface terminal unit 22_{di}. The block check character accumulation is started upon the receipt of the first character following a STX (start of text) or a SOH (start of heading) character, and continues until a ETX (end of text) character is received. Only messages and control sequences containing a STX or a SOH character will cause a block check character (BCC) to be accumulated.

The accumulation of the BCC consists of applying each character being transferred to the input of the BCCR 33 and performing a binary addition without carry (Exclusive OR function). Prior to each operation in which a BCC will be accumulated in the BCCR 33, the register is cleared. At the end of a data transfer, the exclusive OR function is again performed between BCC's of the sending and receiving units. If no errors have occurred, both BCC's will be identical and the resultant value is the BCCR 33 will be "all zeros".

The block check character decoder 34 receives the output of the BCCR 33. At the end of a transmission from the peripheral terminal unit 50, a BCC is received and checked against the contents of the BCCR 33. If the two BCC's are identical, then the output of the BCCR is equal to "all zeros" and the decoder 34 generates the BCCOK level (Block check character OK) which is used in the BCC error logic.

The memory address register 26 is an eight bit register which develops addresses for a 256 word LCP buffer 25_{00}. The register 36 is controlled so as to provide selective or sequential addressing of the buffer, as required by the data transfer operation which is to be performed.

The Termination Card 20_{ot} (of FIG. 2) provides a 1-second timer which is enabled for operation only during a "read" operation when the LCP is conditioned to receive data from the peripheral unit, such as 50. When enabling inputs are active, the timer allows the peripheral terminal unit a one-second period in which to begin a transmission or continue an interrupted transmission over to the LCP. If the 1-second period elapses without a transmission from the peripheral terminal unit, a time-out flip-flop (TIMOUTF) is set, generating time-out level (TIMOUTL), and the LCP then initiates an end to the read operation by setting an end flip-flop (ENDF). However, this timer can be programmatically inhibited from operating by placing the proper code in the variant-1 digit of the Command Descriptor (FIG. 4B).

With reference to FIGS. 6B and 6D and the prior discussion regarding the Data Flow section of the LCP, (Section B), the input multiplexor 24_{x1} provides the selection of a 17-bit word from three sources: the data input lines B_i, the output lines B_{25} from RAM data buffer 25_{00} or the peripheral device interface levels 24_m which are generated on the Maintenance Card (such as 20_{om}, FIG. 2) from the outputs of push-button switches on the maintenance panel. The selected levels received by input multiplexor 24_{x1} are transferred to the OP code Register 42 and variant register 43, the terminal bus multiplexor 24_{x2} or the valid OP encoder 44, as required by the operation to be performed.

The OP code Register 42 receives the digital OP code of the Command Descriptor C/D, and in conjunction with the output of the variant register 43, specifies the operation to be performed by the LCP. The variant register 43 receives the variant digits contained in the Command Descriptor C/D and, in conjunction with the output of the OP code register 42, specifies further details of the operation to be performed by the LCP.

The Valid OP Encoder 44 is a network which receives Command Descriptor C/D information at its input; then, if the OP code digits and the variant digits 1, 2, and 3 coincide with values representing valid operations for the LCP, this encoder develops the valid OP (VOP) level, which enables the Command Descriptor C/D to be loaded into the OP code register 42 and the variant register 43.

The LCP RAM buffer 25_{00} is made of a network of 18 RAM devices, each one of which has a capacity of 256 information bits. Reference to FIG. 6C will show more detail of the RAM buffer 25_{00}. The buffer network can store 18-bits in each of its 256 address locations; 16 are data bits, one bit is a parity bit, and one bit is an end-flag bit (25_e of FIG. 6C) to identify a word location containing an ending code.

Referring again to FIG. 6D, the terminal bus multiplexor network 24_{x2} provides selection of a 17-bit word from four sources: the input multiplexor 24_{x1}; the UART 31 parallel data output line; the LPW 24_w register output; and the Result Descriptor levels 24_{rd}. The output of the terminal bus multiplexor network 24_{x2} goes to the terminal bus 47. Appropriate voltage levels are provided to those LCP components, (such as the data latch register 49, vertical parity generator/checker 48, buffer 25_{00}, LPW register 24_w, decoder 52 and end code decoder 35 etc.) which have inputs received from the terminal bus 47.

The terminal bus 47 connects the output of the terminal bus multiplexor network 24_{x2} over to the following components: the data latch register 49, the LCP RAM buffer 25_{00}, the LPW register 24_w, the vertical parity generator/checker 48, the BCC register 33, the end code decoders 52 and 35, and the UART multiplexor 27_x.

The vertical parity generator/checker 48 generates odd parity for every word transferred by the LCP over to the Main System 10. The generator/checker 48 also checks for odd parity of every word transferred from the Main System over to the LCP. Each word to be transferred from the particular LCP over to the Main System 10 is first placed in the 17-bit register called the data latch register 49. The data latch register 49 then transfers the word over to the Main System 10. The use of the data latch register increases the rate of data transfer by allowing quicker access to data stored in the LCP RAM buffer 25_{00}.

The longitudinal parity word (LPW) register 24_w is made of 16 separate flip-flops operated in the "toggle" mode. It receives its inputs from the terminal bus 47. When the main System 10 sends a Command Descriptor C/D, a Descriptor Link D/L, or data, over to the LCP, the LPW register 24_w accumulates a LPW (longitudinal parity word) to be checked against an LPW from the System 10. When the LCP sends data or a Result Descriptor R/D over to the System 10, the LPW register 24_w also accumulates an LPW to be sent to the System 10. Accumulation of the LPW consists of applying each word being sent or received to the input of the LPW register 24_w and performing a binary addition without carry (exclusive OR function).

The LPW register 24_w is initialized to "all ones" prior to each operation in which an LPW will be accumulated in the LPW register. At the end of a data transfer from the Main System 10, the exclusive OR function is

performed between the accumulated LPW and an LPW from the System **10**. If no errors have occurred, both LPW's will be identical and the resultant value in the LPW register 24_w will be "all zeros".

In FIG. **6D** the end code decoders **52** and **35** are used to determine the receipt of an ending code character. Decoder **52** handles the AB digits and decoder **35** handles the CD digits. The AB digits end-code decoder **52** is used to identify an ending code in the first character position of a word from the Main System. This decoder is also used to identify an ending code in any character sent from the terminal unit device interface 22_{di}. If decoder **52** receives such an ending code, it causes the level EDCODE and the level SYSEND to be generated. The CD digit decoder **35** is used to identify an ending code in the last character position of a word from the System. Receipt of such an ending code by decoder **35** will cause the voltage level SYSEND to be generated.

The above discussion involved the second section B of the LCP. Now the third section C, the System Logic Section of the LCP, will be discussed with reference to FIG. **6D**.

The Status Count Register **53** (STC) is a four-bit register. This register develops Status Count levels (STCnL) for use in the LCP and levels designated LCSTUn (LCP Status Levels) for transmission to the Main System **10**. In conjunction with providing floating logic levels, the STC register **53** also controls the sequencing of operations for the LCP. Each Status Count developed by the STC register **53** specifies a different phase of operation in the execution of a Command Descriptor C/D, as was previously outlined in connection with FIG. **6A**. The decoder **54** is a binary coded decimal (BCD) to decimal decoder which changes the BCD values of the STC register **53** to decimal values required by the LCP system.

Reference to FIG. **6E** will be instructive in reviewing the system interrelationships between the major LCP elements involved in regard to the logic and control signals operating between these elements. FIG. **6E** shows the major logic and control lines between the IOT (Input-Output Translator) 10_p, the Distribution Card 20_{od} (for the Base Module 20_0), the particular Line Control Processor LCP 20_{00} and the peripheral terminal unit **50**.

First referring to the lowermost group of control lines, the LCP 20_{00} and its Distribution Card 20_{od}, the designation LCPREQ (*n*) is a group of eight "request" lines where the letter "*n*" represents the numbers **0–7** for each specific LCP in the Base Module 20_0. Each of these signals is driven by one particular LCP over to the Distribution Card 20_{od}. This signal is used by a particular LCP to "request" a connection to the System **10** and causes the Distribution Card 20_{od} to set up a "Poll Request".

The next designation LCPCON is the designation for "LCP connected". This line is driven by the connected LCP (**0–7**) to the Distribution Card 20_{od}. This signal is driven by the LCP when it detects its own particular LCP address and it is not in an "off-line" condition. The signal is a response to the LCP address and signifies to the Distribution Card 20_{od} the presence of the LCP addressed.

The designation LCPSTL signifies "LCP Strobe Level". This line is driven by the "connected" LCP over to the Distribution Card. It is the particular LCP's designation of "send" or "acknowledge", depending on the data direction involved.

The IOSND designates I/O send. This line is driven by the "connected" LCP to the Distribution Card 20_{od}. This lines defines the direction of the bidirectional data lines marked DATA (xn). When this line is active low, the data lines will be driven by the Distribution Card 20_{od} to the Main System **10** via IOT 10_p.

The LCSTU (*n*) designates the status of the particular LCP where "*n*" may designate either of LCP's **0–7**. This line is driven by the particularly-connected LCP to the Distribution Card 20_{od} and reveals the "status" of the LCP as shown in FIG. **6A**.

Referring to FIG. **6E**, a number of connections are provided as between the LCP, such as 20_{oo} and the Distribution Card 20_{od}. The DATA (xn) represents the "message level interface" (as previously shown in FIG. **5E** of which the lower 16 lines are the data lines for the digits ABCD). The next higher line is the PARITY line which carries the parity bits. These 17 lines constitute the message level interface and are of a bidirectional nature, that is to say, transmission may occur in either direction along these lines depending on the logic control lines used to determine the direction of transmission.

The designation EMRREQ in FIG. **6E** signifies the "emergency request" line. This line is driven by one or more LCPs to the Distribution Cards. The LCP may drive the emergency request line at any time. The emergency request signifies that an LCP needs system access quickly to avoid a data transfer failure. Only LCP's whose lack of system access will necessitate operator intervention or difficult error recovery, will drive the emergency request in conjunction with their LCP request. Those LCPs which are not emergency requesting, will disable their LCP request with this line. A Distribution Card detecting an emergency request, will cause a Global Priority of "seven" to be transmitted to the Main System **10** during a "Poll Request".

The designation TERM in FIG. **6E** designates a "terminate" voltage level. This is generated on a Distribution Card and is sent to the LCP to terminate or end an operation.

The designation LCPAD "*n*" in FIG. **6E** designates the LCP address (where "*n*" can be **0–7**) to designate the individual LPCs. One of these eight signal lines is driven by the Distribution Card to each particular LCP. The receiver in the LCP will be jumpered to the proper line. This signal is functionally a connection line to the LCP. An LCP receiving its LCP address is "connected" to the Main System **10** through the Distribution Card.

The STIOL in FIG. **6E** signifies the "Strobe I/O Level". This line is driven by the connected Distribution Card. It represents the System's "send" or "acknowledge" depending on the data direction.

The ARQOUT line of FIG. **6E** is the output end of the Distribution Card which has an input designated ARQIN. These represent "access request in" and "access request out". These signals are driven and received by Distribution Cards only and consist of short lines between adjacent Distribution Cards. They are used during "Poll Test" to resolve Distribution Card priority. The lines DCB **1** and the DCB **2** represent Distribution Card "busy" levels. These are generated on each active Distribution Card in a Base Module to resolve Distribution Card priority in the module during a "Poll Request" sequence.

The PTALB line designates "Poll Test active level". This is a bidirectional signal level between Distribution Cards in the same Base Module. A Distribution Card performing a "Poll Test" operation sends this level to the other Distribution Cards, thus inhibiting them from conducting a "Poll Test" or a "Poll Request" sequence. Each Base Module may service not only one "main system" **10** via its Distribution Card (20_{od}, FIG. **2**) but may be provided with multiple Distribution Cards to cooperate with and service other host "main systems". Each Distribution Card in a Base Module can service a different host system and each host system would follow the same basic organization shown in FIG. **3**.

The REQACC line designates "Request Access". This line is driven by and received by Distribution Cards only. The line is used to signify an interrupt request as being "active" by the Distribution Cards.

The BUSY line of FIG. **6E** designates a Base Module "busy" level. This is a bidirectional signal level developed on a Distribution Card when that card has made a "connection" with the Main System **10**. The level is sent to other Distribution Cards on the same Base Module to indicate that the LCP backplane is in use.

Now further in reference to FIG. **6E**, the relationships between the IOT 10_t and the Distribution Card 20_{od} will be discussed. At the upper left of FIG. **6E**, the LCPST designates the LPC Strobe Pulse. This is generated on a Distribution Card from the LCP strobe level and is sent on to the Main System via the IOT 10_t.

The PB/ST **2** designates "Port Busy" or the LCP status **2** line. This line resides on the message level interface as shown in FIG. **5E**. In the "unconnected" state, this line indicates a Port Busy condition during a "Poll Test" algorithm. In the "connected state", this line carries bit **2** of the LCP's status to the System **10**.

The IP/ST **4** designates an Interrupt Request or a Poll Test parity error, or an LCP status **4** line. In the unconnected state, this line is used to carry an "Interrupt Request" from the LCP or else to indicate an address parity error during a "Poll Test" connection attempt. An Interrupt Request indicates that an LCP is requesting access to Memory. In the "connected" state, this line carries bit **4** of the LCP's status to the Main System.

The ER/ST **8** designates "emergency request" or the LCP status **8** line. In the "unconnected" state, this line represents an emergency request from the LCP. "Emergency request" designates that an LCP needs immediate access to the Main System. In the "connected" state, this line carries bit **8** of the LCPs status to the Main System. Once connected, the LCP indicates its System Memory requirements by its status. The LCP status is gated continuously and may only be considered valid by the System at LCP "Send/Acknowledge" time.

Further in FIG. **6E** to the connections between the IOT 10_t and the Distribution Card 20_{od}, the connections designated PARITY and DATA "*xn*" refer to the message level interface lines previously discussed. The CS/ST **1** designates "Channel Select" or LCP status **1** line. In the "unconnected" state, this lines carries "Channel Select" from the System **10** to the Distribution Card. "Channel Select" is used in conjunction with "address select" in both connection algorithms. However, in the "connected" state, this line carries bit **1** of the LCP's status to the Main System **10**. This line is a bidirectional line. The receiver on the Distribution Card will be any standard TTL device. The driver on the Distribution Card will be a tri-state driver such as a 8097/8098 (National Semiconductor Corp.) or equivalent which will be active only in the connected state.

The TRM designates the "terminate" level. This is sent from the Main System **10** to a Distribution Card when a data transfer operation is to be terminated.

The ADDSEL line of FIG. **6E** designates "address select". This signal line indicates that the Main System is connected or is attempting connection to a specific LCP. This line is used in conjunction with "Channel Select" for both connection algorithms to achieve connection. Once a connection to the LCP is achieved, the System and LCP remain connected until the signal line is inactivated by the System. When the line is active, the System can be considered "busy".

Again referring to FIG. **6E**, the AG/SIO designates "access granted" or "Strobe I/O". When the interface is in an "unconnected state", this line carries and "access granted" signal. "Access granted" is used to acknowledge an Interrupt Request for connection and to begin a "Poll Request" algorithm. With the interface in a "connected" state, this line carries a "Strobe I/O" signal. This signal is the System's Send/Acknowledge line in transferring information between the System **10** and the LCP Base Module. The actual signal is a 100 nanosecond minimum pulse sent from the System and latched by the Distribution Card. The Distribution Card will generally clip the first 50 nanoseconds from the signal to allow for cable settling time.

In regard to FIG. **6E**, the control signals as between the LCP 20_{00} and the peripheral terminal unit **50** indicate a line designated RMDTLN. This designates Remote Data Line Level. This is a bidirectional signal level which permits the transfer of serial data between the LCP and the peripheral terminal unit in one direction or the other direction as determined by the level.

Discussed here and below are the operational sequences of the LCP. The logic terms are referred to as either being active or inactive in order to avoid any ambiguity that might result from using the terms True and False.

Receipt of Instructions by Line Control Processor

Previously in FIG. **6A**, the logic flow involving the Status Counts (STC) between the LCP 20_{00} and the Main System **10** was discussed. Now referring to FIG. **7A** there will be seen in greater detail a simplified flow diagram illustrating the receipt of instructions by the LCP. This flow chart shows the basic actions of the LCP during receipt of instructions and also shows those actions which can occur due to modification of the original instructions, the receipt of a time-out level, and the occurrence of error conditions.

Prior to receiving any of the seven possible instructions from the Main System **10**, the LCP is normally in an "idle" state at Status Count **3**. However, the LCP can also be in a STC **3** during a "Read" operation, awaiting either a conditional cancel instruction from the Main System **10**, or a data transmission from the peripheral terminal unit, such as **50**.

The following will describe the actions of the LCP during receipt of instructions from the System **10** and during preparation for the instruction execution. These actions are itemized as (a), (b), and (c).

(a) System-LCP connection: with the LCP in STC **3**, the System makes a connection with the LCP through a "Poll Test" sequence, and the LCP receives its unique address level (LCPAD) (*n*), as was illustrated in FIG. **6E**. The receipt of LCPAD (*n*) causes the LCP to send the LCP connection level (LCPCON, FIG. **6E**) to the

ssociated Distribution Card 20_{od} and generates CPADL (LCP Address Level), which "enables" porons of the LCP system logic section. The address level CPAD (*n*) also enables the LCP backplane network y generating a gate system level (GATSYS). Then a robe (STIOL) is received from the Distribution Card 0_{od}, FIG. 6E) causing STIOF (Synchronous Strobe lip-Flop) to be set. The setting of STIOF activates the esired module of the LCP by setting RECVF (Receive lip-Flop), enables setting of the LPW register (24_w, IG. 6D) to logic "1's", and sets selected flip-flops to a eginning state. The Command Descriptor C/D is reeived in the LCP and is loaded into the OP code regisr 42 and variant registers 43 (FIG. 6D). Receipt of the /D results in an LPW being placed into the LPW gister 24_w. The C/D is checked for validity and the alid OP flip-flop (VOPF) is set. The LCP then steps om STC 3 over to STC 11 (FIG. 7A) to receive an PW from the System 10.

(b) Receipt of LPW by the LCP: In FIG. 7A at STC l, a longitudinal parity word (LPW) is received from e System 10 and is checked against the contents of the PW register 24_w to validate longitudinal parity of the /D transfer. Vertical parity is also checked, then a ertical level OK (VLOK) and vertical parity OK level /PAROK) is set. The LCP buffer address is preset to 53 in the memory address register MADR 36 (FIG. D), and setting of the LPW register 24_w to logic "1's" again enabled; then the LCP steps to STC 6 to receive e Descriptor Link D/L from the System 10.

(c) Receipt of Descriptor Link and Descriptor Link PW: at STC 6, the LCP receives the two words of the escriptor Link D/L from the System 10 and an LPW accumulated in the LPW register 24_w. An LPW is en received from the System 10 and is checked gainst the contents of the LPW register 24_w. The Deriptor Link D/L and the LPW are stored in buffer ddress locations specified by the memory address regter MADR 36 as addresses 253, 254 and 255 (FIG. C). From STC 6, the LCP branches to STC 8 for a Write" operation, or to STC 1 for a "Read" operation, r to STC 7 if a Descriptor Link error occurred.

There are alternate flow path situations such as: (a) hen a "conditional cancel" instruction is received om the System 10, or (b) a data transmission is reeived from the peripheral terminal unit, such as 50, or c) a time-out level is generated, or (d) a receipt of test istructions. To further amplify these alternate flow ath situations per FIG. 7A:

(a) Receipt of Conditional Cancel Instruction: at STC , if a conditional cancel instruction is received from the ystem 10, while the LCP is awaiting a transmission om the peripheral terminal unit 50, a cancel flip-flop CANCF) is set and the LCP steps to STC 11 to receive Command Descriptor longitudinal parity word, PW. From STC 11, the LCP steps to STC 7 and sends Result Descriptor to the System 10, indicating that the ancel operation is completed.

(b) Receipt of Transmission from the Peripheral Terinal Unit: at STC 3 during a "Read" operation, if the rminal busy flip-flop (TRMBSYF) is set, indicating at terminal unit has started transmitting, the LCP eps to STC 1 to receive data from the peripheral terinal unit. The LCP continues to receive data and ompletes the remainder of the Read operation in acordance with instructions contained in the Command escriptor C/D.

(c) Receipt of Time-Out Level: during a "Read" operation, with the LCP in STC 3 awaiting a transmission from the peripheral terminal unit (and if the 1-second timer is not inhibited) then if there is a 1-second delay in receiving the transmission, the time-out level (TIMOUTL) is generated. With TIMOUTL active, the end flip-flop (ENDF) is set, the terminal complete (TMCNP) level is generated, and the LCP steps to STC 1. At STC 1 a request for reconnection to the System is initiated and the LCP steps to STC 5, FIG. 7B. At STC 5 with ENDF set, the Read operation is terminated and the LCP steps to STC 7 to send a Result Descriptor, R/D, to the System 10. A time-out level can also be received with the LCP at STC 1.

(d) Receipt of Test Instructions: at STC 11, FIG. 7A, if TESTF (Test Flip-Flop) is set indicating that a test instruction was received, the LCP completes the test operation by stepping to STC 7 and sending a Result Descriptor R/D to the System 10. Error Conditions: the occurrence of two types of error conditions (ea) and (eb) during the receipt of instructions will be acted upon by the LCP, as follows: (ea) Command Descriptor parity error: In FIG. 7A, at STC 11, if the VLOK (Validity Level OK) level is not active, or if VOPF (Valid Operation Flip-Flop) is not set, the LCP steps to STC 7 to send a Result Descriptor R/D containing a descriptor error to the System; (eb) Descriptor Link parity error: at STC 6, if the VLOK level is not active, the LCP steps to STC 7 to send a Result Descriptor R/D containing a Descriptor Link error to the System 10.

Write Operation

Referring to FIG. 7B, there is seen a sequential logic diagram which is simplified to show the steps involved in the "Write" operation. Let us assume that one buffer load of data will be transferred from the System 10 to the peripheral terminal unit 50, followed by a partial buffer of data containing an ending code character in the last character position (CD digits) of a word.

The following steps (a through i) describe actions of the LCP, such as 20_{00}, during transfer of data from the System 10 over to the LCP, and fron the LCP to the peripheral terminal unit, such as 50.

(a) Receipt of data from system: at STC 6, if a "Write" operation is specified by the Command Descriptor C/D, the LCP enables the setting of the LPW register 24_w to logic "1's", then steps to STC 8 to receive data from the System 10. An IOSF (I/O Send Flip-Flop) is used and put in a reset state at this time to enable the bidirectional data lines for transfer of data from the System 10 over to the LCP. There are provided multiplexor control levels SLAIN (Select A Input Multiplexor) and SLBIN (Select B Input Multiplexor). These are both inactive, connecting the data lines to the input multiplexor network 24_{x1} of FIGS. 6B and 6D. There are other multiplexor control levels SLARAM (Select A Level Terminal Bus Multiplexor) and SLBRAM (Terminal Bus Multiplexor Select B Level). These also are both inactive, connecting the input multiplexor network 24_{x1} to the input of the Terminal Bus Multiplexor Network 24_{x2}.

At STC 8, the Receive Flip-Flop (RECVF) is set, activating the write module of the LCP. The setting of RECVF causes the write enable level (WESVS) for the LCP buffer (WESYS) to be active. Thus, data is transferred from system Main Memory 10_m over to the LCP buffer 25_{oo}, one word at a time, by way of the terminal bus 47 of the LCP. An asynchronous Strobe (STIOL)

from the associated Distribution Card 20_{od} (FIG. 6E) accompanies the transfer of each word, and as each word is received by the LCP, the LCP sends a strobe level (LCPSTL) to the System 10 to "acknowledge" receipt of the word. As each word is placed on the terminal bus 47, then, in addition to being sent to the buffer 25_{00}, it is also applied to the input of the vertical parity generator/checker 48, the LPW register 24_w and the end code decoders 52 and 35. Vertical parity is checked and a longitudinal parity word is accumulated in the LPW register 24_w. Transfer of words continues until the next to last data word address 251 is attained in the Memory Address Register 36. The LCP then steps to STC 10 of FIG. 7B to receive one final word from the System. At STC 10, the LCP receives the final word to fill the buffer, and then steps to STC 12 to receive an LPW from the System 10.

(b) Receipt of LPW and disconnect from the System 10: at STC 12, the LCP receives an LPW from the System 10 and checks it against the LPW accumulated in the LPW register 24_w during the data transfer. The LCP then enables setting of the LPW register 24_w to logic "1's" and steps to its STC 1, disconnecting from the System 10 in order to transfer data to the peripheral terminal unit, such as 50. Terminal bus multiplexor control levels SLARAM and SLBRAM (Select A and Select B of 24_{x2}) are both inactive thus to connect the output of the buffer 25_{00} with the input to the terminal bus multiplexor network 24_{x2}. The input multiplexor 24_{x2} has control levels SLAIN (Input Multiplexor Select A Level) and SLBIN (Input Multiplexor Select B Level) which will be controlled during the data transfer by the state of the even flip-flop (EVNF) in order to access a character alternately from the AB digits and the CD digits of a word in the buffer 25_{oo}.

(c) Transfer of Data to Peripheral Terminal Unit: with further reference to FIG. 7B, at STC 1, the receive flip-flop (RECVF) is reset, thus enabling the receive module of the LCP. The terminal start level (TERST) is generated to prepare the LCP for operation with the peripheral terminal unit. The TERST level enables the setting of master clear UART flip-flop (MCUARTF) in order to clear the UART 31 (FIG. 6D). The setting of a terminal active flip-flop (TRMACTF), a send flip-flop (SENDF), and the terminal busy flip-flop (TRMBSYF) are also enabled, activating terminal control logic for a Write operation and specifying that the peripheral terminal unit is in a "busy" state. The Memory Address Register 36 (FIG. 6D) is set to MADR 0 to access the first word in the buffer 25_{00}. In the UART 31, the "transmitter holding register empty" (THRE) level is active, and the setting of the UART empty flip-flop (UARTETF) is enabled to provide a strobe level to the UART multiplexor 27_x.

The UART 31 accepts one character at a time from the LCP buffer 25_{00}. The even flip-flop (EVNF) is used in conjunction with the Memory Address Register 36 to control accessing of characters. When loaded with a character, the UART 31 transfers the character serially over to the peripheral terminal unit, such as 50. As each character from the buffer 25_{oo} is placed on the terminal bus 47, it is also applied to the input of the block check character register (BCCR) 33, which (after a STX/SOH, "start of test/start of heading" character has been received) begins to accumulate a block check character during the data transfer. The UART 31 continues to accept characters from the buffer 25_{00}, then transferring them to the peripheral terminal unit 50, until memory address level MADR 252 is attained in the Memory Address Register 36, indicating that the last word in the buffer has been accessed.

(d) Request for Reconnection to System 10: the memory address level MADR 252 causes the buffer transfer flip-flop (BFXFRF) to be set, indicating that the buffer 25_{00} needs service, and the LCP initiates a request for reconnection to the System by enabling the setting of the LCP request flip-flop LCPRQF. The setting of IOSF (I/O Send Flip-Flop which indicates the direction of data flow on the message level interface) is enabled to condition the data lines for transfer of data to the System 10, and the setting of MADR 253 level is enabled to allow access to the Descriptor Link D/L (FIG. 6C). The LCP then steps to STC 5 of FIG. 7B to send the Descriptor Link D/L to the System 10. There are floating logic levels which generate LCPADL (LCP Address Level) when the LCP address levels (0–7), LCPADn, is received from the associated Distribution Card during the reconnection sequence, and the LCP generates a level called gate system (GATSYS) to enable the backplane network. The level LCP connected (LCPCON) is sent to the Distribution Card 20_{od} to indicate that the LCP is connected.

(e) Transfer of Descriptor Link and the Descriptor Link LPW: in FIG. 7B, at STC 5, the transmit flip-flop (XMITF) is set, activating the "Read" module of the LCP. The LCP transfers the Descriptor Link D/L and the LPW (previously received at STC 6) back to the System 10. The LCP enables the setting of the LPW register 24_w to logic "1's" and if the Main System has more data to send, the LCP steps again to the STC 8 to receive additional data from the System 10.

(f) Receipt of Additional Data and Ending Code from System 10: at STC 8, the actions of the LCP while receiving the "Second" buffer load of data from the System 10 are the same as those performed during receipt of the first buffer load, up to the point that an "ending code" is recognized by the terminal bus 47. When an "ending code" in the last character position (CD digits) of a word is placed on the terminal bus 47, then a system end level (SYSEND) is generated. SYSEND level causes the data input for the end-flag 25_e of FIG. 6C (RAM 18 L) to be active and the end-flag bit (ENDFG) and the ending code character are both stored in the current buffer address. The LCP then steps to STC 12 to receive an LPW from the System 10.

(g) Receipt of LPW and Disconnect from System 10: at STC 12 of FIG. 7B, the LCP receives the longitudinal parity work LPW and checks it against the LPW accumulated in the LPW register 24_w. The LCP then steps to STC 1, disconnecting from the System 10, to transfer the remaining data and the ending code to the peripheral terminal unit.

(h) Transfer of Data and Ending Code to Peripheral Terminal Unit: at STC 1, the actions in transferring the remaining data to the peripheral terminal unit are the same as those performed during transfer of the first buffer load, up to the point that an "ending code" is recognized on the terminal bus 47. When an ending code is placed on a terminal bus 47 from the output of the buffer 25_{00}, the ending code is transferred and the end flip-flop (ENDF) is set. The accumulated block check character in the BCCR 33 (if a BCC is being generated) is then transferred to the peripheral terminal unit such as 50. SENDF (send flip-flop) and TRECF (terminal receive flip-flop) are both in a reset state, causing terminal complete (TMCMP) level to be active.

The terminal complete level causes the LCP to initiate a request for connection to the System **10**.

(i) Request for Reconnection to Terminate Write Operation: the LCP requests a reconnection to the System by enabling the setting of LCPRQF (LCP Request Flip-Flop). In conjunction with the reconnection, the LCP steps to STC **5** of FIG. **7B**, sends the Descriptor Link D/L to the System **10** and then steps to STC **7** to send a Result Descriptor R/D to the System **10**.

The above discussion completes the explanation of the general flow path for a "Write" operation in which more than one buffer load of data was transferred, and in which the operation was concluded by receipt of an "ending code". This describes the normal situation. However, there could be alternate flow paths and possible error conditions which might occur as follows, in reference to FIG. **7B**. The following items (a) through (c) describe the actions of the LCP when "modifications" to the original Write instructions are made by the System **10** or the LCP.

(a) Request for Emergency Access to System **10**: during transfer of data from the LCP to the peripheral terminal unit **50**, when the LCP buffer $\mathbf{25}_{00}$ is completely empty, a flip-flop BFXFRF is set. This is the buffer transfer flip-flop which is located on the Terminal Card; this flip-flop is set when the LCP buffer is filled with data from the terminal unit, or when emptied of data during transfer of data from the LCP to the peripheral terminal unit. When BFXFRF is set, this enables the setting of LCPRQF (LCP Request Flip-Flop, which, when set, indicates that the LCP requires access to the Main System Memory $\mathbf{10}_m$). The setting of the LCPRQF initiates a request for reconnection to the System **10** to either send data to the Main System or to obtain more data if the buffer is empty. If a reconnection is not completed prior to the time the transmitter-holding register of the UART **31** is ready to accept another character, the LCP causes the emergency request level (EMRREQ) to be generated. The EMRREQ level is sent to the associated Distribution Card $\mathbf{20}_{od}$ to initiate an emergency request for reconnection to the system.

(b) Receipt of Ending Code (AB digits): if an ending code is identified in the first character position (AB digits) of a word from the System **10**, then EDCODE (end code level) is generated. EDCODE is generated on the terminal control card when an end code character is in the A and B digits of the terminal bus **47**. Also generated is SYSEND (System End Code Level). When active, the SYSEND level indicates that an end code character is on the terminal bus **47**. At STC **8**, the EDCODE level enables the setting of a character end flip-flop (CHARENF), and the SYSEND level generates the **18**th bit Write end-flag level, RAM **18**L. The "Write" end flag level is generated on the terminal control card from the EDCODE level; this is the data input level for the end-flag RAM of the LCP buffer $\mathbf{25}_{00}$. The ending code and the ENDFG (end-flag level is generated on the data flow card from RAM **18** L; when active, this level identifies the address of an end code in the LCP buffer) are stored in the current buffer address of the LCP, and the LCP steps over to STC **12** (FIG. **7B**) to receive a longitudinal parity word LPW. AT STC **12**, the LCP receives an LPW from the System **10** and checks it against the accumulated LPW in the LPW register $\mathbf{24}_w$. The LCP then steps to STC **9** to initiate decrementing of the System Memory Address. (The address must be decremented by two digits to accurately reflect the address of the ending code in System Memory). From STC **9**, the LCP steps over to STC **1** to transfer data and the ending code to the peripheral terminal unit **50**. At STC **1**, recognition of the ending code on the terminal bus **47** causes the LCP to perform the same actions described during the previous "Write" operation at STC **1** when data, ending code, and block check characters are transferred to the peripheral terminal unit **50**, after which the LCP disconnects from terminal unit **50** and reconnects to the System **10** and terminates the "Write" operation.

(c) Receipt of Terminate Signal from System **10**: a terminate signal (TERM level, FIGS. **6C**, **6E**) is sent from the System **10** to the LCP whenever System Memory space designated for LCP operation is to be exceeded. During a "Write" operation, the TERM level can be received at STC **8**, STC **10**, or STC **12**, FIG. **7B**. The actions of the LCP upon receipt of the TERM level (Terminate Level) depend upon the Status Count in which the LCP is operating, and upon whether or not the receipt of TERM level is preceded by a receipt of an "ending code" from the System as follows:

(1) Receipt of Terminate Signal Before Ending Code: if the TERM level is received at STC **8** or STC **10**, the LCP steps over to STC **14**. AT STC **14**, regardless of whether TERM level remains active or is now inactive, the LCP steps over to STC **12**, receives and checks a longitudinal parity word LPW, then steps over to STC **7** to send a Result Descriptor R/D to the System **10**. If an ending code is received in the CD digits (last character) of a word at STC **8** or STC **10**, and the TERM level is also received, the LCP steps to STC **14**. At the STC **14**, if the TERM level is still active, the ending code was not placed in the LCP buffer $\mathbf{25}_{00}$. The LCP then steps to STC **12**, receives and checks an LPW, then steps over to STC **7** to send a Result Descriptor R/D to the System **10**.

(2) Receipt of Terminate Signal After Ending Code: if an ending code is received in the CD digits of a word at STC **8** or STC **10**, the LCP steps to STC **12** to receive LPW. At STC **12**, if the TERM level is now received, the ending code is transferred to the LCP buffer $\mathbf{25}_{00}$ and the LCP steps over to STC **1** to transfer remaining data and the ending code to the peripheral terminal unit **50**. At STC **1**, recognition of the ending code on the terminal bus **47** causes ENDF to be set. (End flip flop: when set, this flip-flop indicates that the terminal control section of the LCP has ended its operation). The setting of ENDF indicates that there is not more data to be transferred; after the data, ending code, and block check character are transferred to the peripheral terminal unit **50**, the LCP disconnects from terminal **50**, reconnects to the System **10**, to terminate the "Write" operation.

As illustrated hereinunder, at STC **1**, the recognition of the ending code on the terminal bus **47** causes the ENDF (end flip-flop) to be set. The setting of ENDF indicates that there is no more data to be transferred; after the data, ending code and block check character are transferred to the peripheral terminal unit **50**, the LCP connects to the System **10** to terminate the "Write" operation.

If an ending code is received in the AB digits of a word at STC **8** or STC **10**, and the TERM level is also received, the LCP steps to STC **14**. At STC **14**, if TERM level is inactive, the whole word containing the ending code in the AB digit was transferred to the LCP buffer $\mathbf{25}_{00}$. A correction of System Memory Address is

necessary. The LCP steps to STC **12**, receives and checks the LPW, then steps to STC **9** to initiate decrementing of the System Memory Address. The LCP then steps to STC **1** to transfer data and ending code to the peripheral terminal unit **50**.

If the TERM level was still active at STC **14**, then only the ending code character was transferred to the LCP buffer 25_{00} and no correction of System Memory Address is required. The LCP steps to STC **12**, receives and checks the LPW, then steps directly to STC **1** to transfer data and the ending code over to the peripheral terminal unit **50**.

Error Conditions

During a "Write" operation the following error conditions (a,b,c,d) will be acted upon by the LCP:

(a) Access Error: after transmitting EMRREQ level to the associated Distribution Card, if the LCP does not receive a reconnection to the System **10** prior to the time the UART **31** is completely empty, the LCP enables the setting of the access error flip-flop (ACCERF). The setting of ACCERF enables setting of the end flip-flop (ENDF), and the LCP initiates a request for reconnection to the System **10** to terminate the "Write" operation and to send an error Result Descriptor R/D to the System **10**.

(b) System Vertical Parity Error: during transfer of data from the System **10** to the LCP, if the vertical parity is not O.K. and the VPAROK level is not active after each check of vertical parity, then the vertical parity error flip-flop (VPERF) is set to indicate the existence of a vertical parity error. The absence of VPAROK level also prevents the vertical longitudinal OK level (VLOK) from being generated, and at STC **12**, the LCP steps over to STC **7** to send an error Result Descriptor R/D to the System **10**.

(c) Longitudinal Parity Error (FIG. 7B): when the longitudinal parity word is checked after a data transfer from the System **10** to the LCP, if longitudinal parity OK level (LPOK) is not active, the longitudinal parity error flip-flop (LPERF) is set to indicate existence of a longitudinal parity error. The absence of LPWOK level (the LPW OK level: is generated on the data flow card from the terminal bus **47** levels; when active, it indicates to the System Logic Section of the LCP that the LPW is correct) prevents VLOK level from being generated, and at STC **12**, the LCP steps over to STC **7** to send an error Result Descriptor R/D to the System **10**.

(d) Terminal Vertical Parity Error: during transfer of data from the LCP buffer 25_{00} to the UART **31**, if the vertical parity OK (VPAROK) level does not remain active for each character transferred, the terminal vertical parity error flip-flip (TVPERF) is set to indicate existence of a vertical parity error. When the LCP reconnects to the System **10** and terminates the "Write" operation, the Result Descriptor R/D sent to the System **10** at STC **7** will indicate the parity error.

Read Operation

Referring to FIG. 7C, there is seen a simplified logic chart showing the "Read" operation. A "Read" operation is generally accomplished in conjunction with some form of "Write" operation. As an example, assuming that a "Write" operation has been completed and the peripheral terminal unit **50** has responded with an acknowledge character (ACK), indicating that the peripheral terminal unit **50** is now capable of sending information. Again, assuming there will be no delay in receipt of data from the peripheral terminal unit **50**, and that one buffer load of data will be received followed by a partial buffer of data containing an ending code. It is also assumed that the ending code will be received in such a way that it will be placed in the last character position (CD digits) of a word in the LCP buffer 25_{00} (FIG. 6C).

General Flow Path

The following paragraphs (a) through (l) describe the actions of the LCP during transfer of data from the peripheral terminal unit **50** to the LCP, and also from the LCP over to the System **10**.

(a) Disconnect from Main System **10**: referring to FIG. 7C, at STC **6**, when a "Read" instruction is specified in the Command Descriptor C/D, from the System, the READF (read flip-flop: located on the data flow card; the logic state of the read flip-flop is controlled by output levels from the op code register; the set state of READF indicates that a "Read" operation is being performed by the System) is set. The LCP enables setting of the LPW register 24_w to logic "1's", then steps to STC **1**, disconnecting from the System **10** to receive data from the peripheral terminal unit **50**. The terminal bus multiplexor 24_{x2} (FIG. 6D) select A level (SLARAM) is active, and SLBRAM, SLAIN and SLBIN levels are inactive to provide a path for data from the UART **31** over to the LCP buffer 25_{00}.

(b) Receipt and Storage of Data from Terminal Unit: referring to FIG. 7C, at STC **1**, with READF set, the terminal start (TERST) level is active. This TERST level causes the UART **31** to be master cleared and enables the setting of TERMACTF (terminal active flip-flop; located on terminal control card; the logic state of this flip-flop is controlled by TERST, TRECF and SENDF; the set state of TRMACTF indicates that the terminal control section of the LCP has been activated for a "Read" or a "Write" operation) to activate terminal control logic. READF also enables the setting of the terminal receive flip-flop (TRECF) to allow receipt of data from the peripheral terminal unit **50**. The buffer 25_{00} has its address preset to MADR location **255** and if the even flip-flop (EVNF) is not already set, its setting is enabled to initiate control of buffer addressing. Data characters are transferred serially from the peripheral terminal unit **50** to the UART **31** in the LCP, and the UART checks each character for even vertical parity.

(b-1) Receipt of First Character and Generation of Vertical Parity: with the terminal receive flip-flop (TRECF) set, and the data store flip-flop (DATASTF), in a reset state, receipt of the first character causes the data received level (DR) to be active. The DR level enables setting of the reset UART flip-flop (RSUARTF) and also the terminal busy flip-flop (TRMBSYF). The even flip-flop, EVNF, is set, causing the buffer address to be incremented to MADR location **0**. The setting of the data store flip-flop, DATASTF, and the resetting of EVNF are then enabled, in preparation for storing the first character in the buffer. With RSUARTF set, the SLARAM level is generated which places the first character on the AB digits and also on the CD digits of the terminal bus **47**, forming a complete word. A parity bit is not included with this word. The contents of the terminal bus **47** are applied to the vertical parity generator/checker **48** of FIG. 6D. Parity for the word on the terminal bus **47** is generated and a flip-flop, used to designate odd vertical parity is set or

set, as applicable to indicate parity, until receipt of a cond character from the peripheral terminal unit **50**.

(b-2) Storage of First Character in Buffer: with the ata store flip-flop DATASTF set, the reset state of VNF causes the buffer write enable A (ERWA) level be active. The System Write Enable (WESYS) level also active, and these two levels provide the Write nable input for the AB and CD digits of the buffer etwork. The first character is then stored both in the B and the CD digit locations of MADR location **0** of lemory Address Register **36**. Transfer of the first char-ter from the UART **31** to the buffer 25_{00} causes the set UART flip-flop (RSUARTF) to be reset. The data ceive level (DR) is then made inactive, followed by e resetting of DATASTF (Data Store Flip-Flop). his combination of logic prepares the UART **31** to cept the second character from the peripheral termi-l unit **50**.

(b-3) Receipt and Storage of Second Character: when e second character is received by the UART **31**, the ata receive level (DR) is again made active and SUARTF is set. This logic in combination with the set state of the even flip-flop EVNF inhibits the buffer ldress from being incremented. The setting of the data ore flip-flop DATASTF and the even flip-flop EVNF e then enabled in preparation for storing the second aracter in the buffer. The terminal bus multiplexor lect A level, SLARAM, is still active and the charac-r is placed on both the AB and the CD digits of the rminal bus **47**. The contents of the terminal bus **47** are ain applied to the vertical parity generator/checker . Parity is generated for the word on the terminal bus and is compared with the parity generated during ceipt of the first character. From the results of the mparison, a single parity bit is generated for the first d second characters.

With the data store flip-flop DATASTF and the even p-flop EVNF set, the ERWB level (Write Enable vel for CD digits of LCP buffer) is generated and the cond character is stored in the last character position D digits) of buffer 25_{00} at address location MADR **0**, erwriting the character previously placed there. The aracter on the AB digits of the terminal bus **47** is not ored in the buffer 25_{00} because the ERWA level is not tive (ERWA is the Write Enable level for the AB gits of the LCP buffer). A parity bit from the vertical rity generator/checker **48** is added to the complete ord now contained in the Memory Address Register MADR **0**.

(b-4) Receipt of Additional Characters and Start of lock Check Character (BCC) Accumulation: addi-onal characters are accepted by the LCP. With the ceipt of each character, the logic state of the even ip-flop EVNF is complemented to control increment-g of the Memory Address Register **36**, so as to place ata into the buffer 25_{00} in word format. With the re-ipt of the "start of heading/start of text" character OH/STX) from the peripheral terminal unit **50**, the ock check character register **33** of FIG. **6D** is enabled d each character following the SOH/STX character applied to the BCCR **33** to accumulate a block check aracter BCC for the message being received. Accu-ulation of a BCC will continue through receipt of the rst buffer load of data and through receipt of succeed-g buffer loads of data until the ending code (ETX aracter) is received. The actions that occur when an ding code is received will be described subsequently ereinafter.

(c) Buffer Filled: when the LCP buffer 25_{00} is completely filled with data, the even flip-flop EVNF and the Memory Address MADR **252** level are set, enabling the setting of the buffer transfer flip-flop (BFXFRF). The setting of BFXFRF indicates that the LCP buffer 25_{00} needs service, and the LCP initiates a request for a reconnection to the System **10**.

(d) Request for Reconnection to System **10**: after disconnection, STC **1**, the LCP initiates a request for a reconnection to the System by enabling the setting of the LCP request flip-flop LCPRQF. The setting of the I/O send flip-flop (IOSF) is enabled also, to condition the data lines for transfer of data to the System **10**, and the setting of the Memory Address MADR **253** (FIG. **6C**) is enabled to allow access to the Descriptor Link D/L. The LCP then steps to STC **5** to send the Descriptor Link D/L and the LPW to the System **10**.

The term MADR refers to Memory Address levels. These are generated on the Terminal Control Card from outputs of the Memory Address Register **36**. These levels represent address locations, shown in Table XI, in the LCP buffer 25_{00} (FIG. **6C**) which are reserved for the following:

TABLE XI

Location	Description
251	Next-to-last data word
252	Last data word
253	Descriptor link information word
254	Descriptor link information word
255	Descriptor link LPW

When one of the eight LCP address levels, LCPADn, is received from the associated Distribution Card 20_{od} during the reconnection sequence, then the LCP address level, LCPADL, is active. The LCPADL address level is generated on the Terminal Control Card when the applicable LCPAD*n* level is active. the LCPAD*n* level also generates the gate system level, GATSYS, to enable the LCP backplane network. The LCP connected (LCPCON) level is sent to the Distribution Card 20_{od} to indicate that the LCP is reconnected. The SLAIN level is active and the SLBIN, SLARAM, and the SLBRAM levels are inactive in order to allow the Descriptor Link D/L to be transferred to the Latch Register **49** (FIG. **6D**).

(e) Transfer of Descriptor Link D/L and the Descriptor Link LPW: in FIG. **7C**, at STC **5**, the transmit flip-flop (XMITF) is set. The transmit flip-flop is located on the System Logic card and the set state indicates that the LCP is transferring data to the System **10**, thus, activating the "Read" module of the LCP. LCP transfers the Descriptor Link D/L and the longitudinal parity word LPW (previously received at STC **6**) back to the System **10**. The LCP then enables setting of the LPW register 24_w to logic "1's", and steps to STC **4** to transfer data to the System **10**.

(f) Transfer of Data to System **10**: at STC **4** of FIG. **7C**, the transmit flip-flop XMITF and the I/O send flip-flop, IOSF, are still in the "set" state from the operation at STC **5**. The asynchronous strobe flip-flop (ASYNCF) is set to enable asynchronous transfer of data to the System **10**. Data is transferred from the LCP buffer 25_{00}, by way of the data latch register **49** (FIG. **6D**) to the System **10** (via the system interface 22_{si} of FIG. **6C**). Transfer is accomplished one word (plus a parity bit) at a time. The LCP strobe level LCPSTL accompanies the transfer of each word, and as each word is received by the System **10**, the System sends a

strobe pulse to acknowledge receipt of a word. Each word placed on the terminal bus **47** of FIG. **6D** for transfer to the System **10** is applied simultaneously to the latch register **49** and the LPW register 24_w. The LPW register 24_w accumulates the longitudinal parity word LPW during the data transfer. When the last data word address of the LCP buffer 25_{00} (MADR **252**) is attained, the synchronous flip-flop (SF, which is located on the Terminal Control Card and is set when the LCP is also transferring data to the peripheral terminal unit) is set, resulting in the development of the synchronous level, SFL, and then the LCP steps over to STC **12** to send an LPW to the System **10**.

(g) Transmission of Longitudinal Parity Word to System **10**: in FIG. **7C** at STC **12**, the LPW accumulated in the LPW register 24_w during operation at STC **4**, is sent to the System **10**. The LCP then enables setting of the LPW register 24_w to logic "1's" and steps to STC **1** to receive additional data from the peripheral terminal unit **50** (via the terminal unit device interface 22_{di} of FIG. **6C**). After this, the LCP steps to STC **5** to send a Descriptor Link to the Main System **10**.

(h) Receipt of Additional Data and Ending Code from Peripheral Terminal Unit: upon the second entry to STC **1**, a terminal active flip-flop (TRMACTF) and a terminal receive flip-flop (TRECF) are both in a set state from the previous operation at STC **1**. The terminal receive flip-flop TRECF is located on the terminal control card and this flip-flop is set when the LCP is receiving data from the peripheral terminal unit; the terminal active flip-flop, TRMACTF, is also located on the terminal control card and, in its set state, indicates that the terminal control section of the LCP has been activated for a "Read" or "Write" operation. The LCP buffer address is again set to MADR **255** in preparation for receipt of data from the peripheral terminal unit **50**. At STC **1**, the actions of the LCP while receiving the second buffer load of data from the peripheral terminal unit **50** are the same as those performed during the receipt of the first buffer load, up to the point that an ending code is received on the terminal bus **47**.

Assuming that prior to receipt of the end code, at STC **1**, that the following two conditions exist: (1) EVNF is reset, indicating that the next character to be received will be placed in the last character position (CD digits) of a word; and (2) both RSUARTF (Reset UART Flip-Flop) and the data store flip-flop (DATASTF) are reset. When the ending code character is received, RSUARTF is set, providing the necessary logic level to generate the Write Enable (ERW **18**) level for the ending code RAM. Receipt of an ending code is recognized by the LCP when the character is on the terminal bus **47**. Recognition of the ending code causes the end code level, EDCODE, to be generated, which develops the data input level (RAM **18** L) for the ending code RAM; the end-flag bit (ENDFG) is then stored in the present buffer address of the buffer 25_{00}. The setting of EVNF and DATASTF is then enabled, which conditions the LCP to store the ending code in the buffer 25_{00}. With EVNF set, the ERWB (Write Enable level for CD digits) level is active and the character is stored in the last character position of the same word address in which the end-flag level, ENDFG, is stored.

(i) Check of BCC and Request for Reconnection to System **10**: with DATASTF set, the EDCODE level enables the setting of the end flip-flop (ENDF). The LCP now receives a block check character (BCC) from the peripheral terminal unit **50** and checks it against the accumulated BCC in the block check character register **33**. The setting of the end flip-flop ENDF causes the terminal receive flip-flop TRECF to be reset, and the terminal complete level (TMCMP) to be active, terminating the actions of the terminal control section of the LCP. The LCP then initiates a request for a reconnection to the System and steps from STC **1** to STC **5** to send the Descriptor Link D/L to the System **10**.

(j) Transfer of Descriptor Link D/L and the Descriptor Link LPW: as in the preceding reconnection to the System, at STC **5** the LCP sends the Descriptor Link D/L and the LPW to the System, and then steps to STC **4** (Read) to transfer data to the System **10**.

(k) Transfer of Data to System **10**: at STC **4**, the actions of the LCP are the same as described before at STC **4**, until the word containing the ending code character is placed on the transfer bus for transfer to the System **10**. Recognition of the ending code causes the System end level (SYSEND) to be developed, and the LCP steps to STC **12** to send an LPW to the System **10**.

(l) Transmission of LPW and Result Descriptor R/D to System **10**: the LCP sends the LPW accumulated in the LPW register 24_w to the System **10**. After the LPW is sent, since the terminate complete level (TMCMP) is now active, indicating that there is no more data to be transferred, the LCP steps to STC **7** to send a Result Descriptor R/D to the System **10**.

At STC **7**, the LCP sends a Result Descriptor R/D to the System **10**, then steps to STC **15** (FIG. **7D**), and sends an LPW, then returns to idle at STC **3** to await another instruction from the System **10**.

The above discussion has involved the general flow path for a "Read" operation in which more than one buffer load of data was transferred from a peripheral to the Main System, and in which the operation was concluded by receipt of an ending code.

However, during a "Read" operation, other situations may occur to cause alternate logic flow paths and the handling of possible error conditions. The following sections (a) through (d) indicate the actions of the LCP when modifications to the original "Read" instructions are made either by the System **10** or by the LCP:

(a) Receipt of Time-Out Level: referring now to FIG. **7E**, which is made of two sheets, **7E-1** and **7E-2**; at STC **1**, with operation of the one-second timer not inhibited, and data being received by the LCP from the peripheral terminal unit **50**; if the sending of the data is interrupted for a period of one second, the time-out level (TIMOUTL) is generated. With TIMOUTL active, the end flip-flop (ENDF) is set, and the terminal complete level (TMCMP) is generated. A request for reconnection to the System **10** is initiated and the LCP steps over to STC **5**. At STC **5**, with the end flip-flop (ENDF) set, the Read operation is terminated and the LCP steps over to STC **7** to send a Result Descriptor R/D to the System **10**. A time-out level can also be received with the LCP at STC **3** as can be seen in FIG. **7E** at STC **3** "idle status".

(b) Transmission Still Expected from Peripheral Terminal Unit: in FIG. **7E**, at STC **1**, with the LCP conditioned to receive data from the peripheral terminal unit **50**, then if data is not being received, the LCP steps immediately to STC **3** in order to be in a condition to receive a conditional cancel instruction from the System **10**. The LCP will return from STC **3** over to STC **1** if a data transmission begins (c) Request for Emergency Reconnection: during transfer of data from the

peripheral terminal unit **50** to the LCP, when the buffer 25_{00} is completely filled, a buffer transfer flip-flop (BFXFRF) is set, initiating a request for a reconnection to the System **10** to store data. (The buffer transfer flip-flop (BFXFRF) is set when the LCP buffer 25_{00} is filled with data from the peripheral terminal unit **50**, or when emptied during transfer of data from the LCP to the peripheral terminal unit). If a reconnection is not completed prior to the time the UART **31** receives another character, the emergency request level (EMRREQ) is generated. The EMRREQ level is sent to the associated Distribution Card 20_{od} to initiate an emergency request for a reconnection to the System **10**.

(d) Receipt of Ending Code (AB digits): the actions of the LCP relating to receipt of an ending code, which will be placed on the AB digits (first character) of a word, are more varied than those involved with receipt of an ending code to be placed on the CD digits of a word. This condition exists because a transmission from the peripheral terminal unit may consist of data followed by an ending code, or it may consist merely of an ending code by itself. Additionally, decrementing of the System Memory Address may or may not be required when storing the ending code, in order to reflect the accurate location of the ending code in System Memory 10_m. Thus, the following actions of the LCP for these various conditions are discussed below in paragraphs d1 and d2:

(d1) Receipt of Ending Code Following Data: if the ending code follows a series of data characters and is received on the terminal bus **47** when the even flip-flop (EVNF) is set, the character, when stored, will be placed in the AB digit position of a word in the LCP buffer 25_{00}. When the character is received, the end code level (EDCODE) is generated, causing RAM **18** L (Write end-flag level) to be active, and the end-flag level (ENDFG) is stored in the presently current buffer address. (The end code level (EDCODE) is generated on the terminal control card when an end code character is in the A and B digits of the terminal bus **47**. The end-flag level, ENDFG, is generated on the data flow card from RAM **18** L, and when active, this level identifies the address of an end code in the LCP buffer 25_{00}. The write end-flag level (RAM **18** L) is the data input level for the end-flag RAM of the LCP buffer 25_{00}). The set state of the even flip-flop (EVNF) then causes the buffer address to be incremented to the next word address. The setting of the data store flip-flop (DATASTF) and the complementing of the even flip-flop (EVNF) are then enabled. With EVNF reset, the write enable A (ERWA) level is generated and the ending code is stored in the AB digits of the buffer address following the one in which the end-flag level (ENDFG) was stored. The LCP then initiates a request for reconnection to the Main System to transfer data and the ending code to the System **10**.

During transfer of final data from the LCP buffer 25_{00} to the System **10** at STC **4**, an ending code in the AB digits of a word will be recognized when ENDFG (end-flag level) level is active and the system end code level (SYSEND) is inactive. This logic combination indicates that the next word to be transferred contains an ending code in the AB digits. In FIG. 7E, the LCP steps to STC **14** to accomplish transfer of a single character. At STC **14** the setting of a word transfer control flip-flop (WTCF) is enabled. unconditionally. The setting of the character transfer flip-flop (CTSF) is enabled to specify that the character transfer state was entered. The ending code is stored in System Memory 10_m, and the LCP steps first to STC **12** to send a longitudinal parity word LPW to the System **10**, then steps over to STC **7** to send a Result Descriptor R/D to the System **10**.

(d2) Receipt of Ending Code Only: as per FIG. 7E, at STC **1**, if the transmission from the peripheral terminal unit **50** consists of a single character (end code), it will be received on the terminal bus **47** with the even flip-flop EVNF in a set state, and will be placed on the AB digit position of a word in the LCP buffer 25_{00}. The character is stored and the LCP initiates a request for reconnection to the System to transfer the character, as seen in the 3rd block of FIG. 7E at STC **5**. This steps over to STC **4**, and with the end code level (EDCODE) active, the setting of the character end flip-flop (CHARENF) is enabled. The character is transferred (STC **14**) to the System **10** and the LCP steps over to STC **12** to send a longitudinal parity word (LPW) to the System **10**. At STC **12**, the set state of CHARENF (the character end flip-flop) causes the LCP to step directly to STC **9** to initiate decrementing of the System Memory Address 10_m. Then the LCP steps to STC **7** in order to send a Result Descriptor R/D to the System **10**.

(e) Receipt of Terminate Signal from System: a terminate signal (TERM level) is sent from the System to the LCP during a Read operation whenever available system memory space designated for the LCP operation is to be exceeded. During a Read operation, the TERM level may be received (FIG. 7E) at STC **4**, STC **14**, or STC **12**. The actions of the LCP upon receipt of the TERM level depend upon the status count in which the LCP is operating when the TERM level is received, and upon whether or not the receipt of the TERM level is preceded by the receipt of an ending code character from the peripheral terminal unit **50**. Under these conditions, the actions of the LCP are discussed in the following paragraphs e1 and e2:

(e1) Receipt of Terminate Signal Before Ending Code is Received: if the LCP receives the TERM (terminate signal) level from the System before it has sufficient time to receive and store an ending code, the LCP then acts as follows:

e1 (a) The receipt of the TERM level while the LCP is transferring data to the System at STC **4**, causes the terminate flip-flop (TERMF) to be set, and the LCP steps over to STC **12**. A longitudinal parity word LPW is sent to the System **10** and the set state of the terminate level (TERMF) causes the LCP to terminate the Read operation and step over to STC **7** to send a Result Descriptor R/D to the System **10**.

e1 (b) In FIG. 7E, the LCP steps from STC **4** over to STC **12** after transferring a word containing an ending code in the CD digits over to the System **10**. If the TERM level is now received at STC **12**, the setting of the word transfer control flip-flop (WTCF) is enabled, and the LCP remains in STC **12** for an additional strobe time. If during the second strobe time, the TERM level is still active, this indicates that the ending code was not transferred. The setting of TERMF (terminate flip-flop) is enabled and the LCP steps over to STC **7** to send a Result Descriptor R/D to the System **10**.

e1 (c) The LCP steps from STC **4** over to STC **12** when the last word in the buffer 25_{00} has been transferred. If the TERM level is now received at STC **12**, the LCP remains in STC **12** for additional strobe time. The word transfer control flip-flop (WTCF) is set and regardless of the logic state of the TERM level during

the second strobe time, the LCP terminates the Read operation and steps over to STC 7 to send a Result Descriptor R/D to the System 10.

e1 (d) The LCP will be in STC 14 if the last data word transferred at STC 4 is to be followed by an ending code in the AB digits of the next word. If the terminate (TERM) level is now received at STC 14, the ending code is not stored and the LCP steps to STC 12, it sends an LPW to the System, and then steps to STC 7 to send a Result Descriptor R/D to the System 10.

(e2) Receipt of Terminate Signal After Ending Code is Received: if the LCP receives the terminate level (TERM) from the System 10 after an ending code has been received from the peripheral terminal unit 50, then the LCP acts as shown in the following paragraphs e2 (a), e2 (b), e2 (c):

e2 (a) In FIG. 7E, the LCP steps from STC 4 over to STC 12 after transferring a word containing an ending code in the CD digits to the System 10. If the TERM level is now received at STC 12, the setting of the word transfer control flip-flop (WTCF) is enabled and the LCP remains in STC 12 for an additional strobe time. If during the second strobe time, the TERM level is no longer active, this indicates that the ending code was transferred. The LCP then steps over to STC 7 to send the Result Descriptor R/D to the System 10.

e2 (b) The LCP steps from STC 4 over to STC 14 if the last word transferred at STC 4 is to be followed by an ending code in the AB digits of a word. If the LCP progresses through STC 14 without receiving the TERM level, the ending code is transferred to the System 10, and the LCP steps over to STC 12 to send a longitudinal parity word LPW. If the TERM level is now received at the STC 12, the LCP takes no action upon its receipt, but steps to STC 7 to send a Result Descriptor R/D to the System 10.

e2 (c) If a transmission from the peripheral terminal unit 50 consists of a single character (ending code), then at STC 4, the LCP enables the setting of the character end flip-flop (CHARENF) and steps over to STC 12 to send a longitudinal parity word LPW. At STC 12, if the TERM level is now received, the LCP will remain in STC 12 for an additional strobe time. If during the second strobe time, the TERM level is still active, this indicates that only the first half of the word containing the ending code was transferred and the System Memory Address was not incremented to the next word address. The LCP steps over to STC 7 to send a Result Descriptor R/D to the System 10. If the TERM level is inactive during the second strobe time, this indicates that the System Memory Address was incremented to the next word address and requires decrementing. The set state of the character end flip-flop (CHARENF) and the inactive state of the terminate level (TERM) cause the LCP to step over to STC 9 to initiate decrementing of the System Memory Address. From STC 9, the LCP steps over to STC 7 to send a Result Descriptor R/D to the System 10.

Error Conditions

During the course of a "Read" operation, certain error conditions may occur which will be acted upon by the LCP, as follows:

(a) Access Error: after transmitting the emergency request (EMRREQ) level, if the LCP has not received a reconnection to the System 10 prior to receiving a second character in the UART 31, the UART 31 generates a level called overrun error level (OE). The OE level causes the enabling of the access error flip-flop (ACCERF) and of the end flip-flop (ENDF). The LCP then initiates a request for reconnection to the System 10 to terminate the Read operation and to send an error Result Descriptor R/D to the System 10.

(b) Terminal Vertical Parity Error: during transfer of data from the UART 31 to the LCP buffer 25_{00}, if the parity error level (PE) is generated by the UART 31, the terminal vertical parity error flip-flop (TVPERF) is set to indicate existence of a vertical parity error. This flip-flop, TVPERF, has a logic state which is controlled by an output from the LCP vertical parity generator/-checker 48, or from the parity error output of the UART 31 (FIG. 6D). The set state of the flip-flop indicates that a vertical parity error occurred. during transfer of data between the LCP and the peripheral terminal unit 50. This flip-flop is located on a Terminal Control Card.

(c) Block Check Character Error: during the transfer of data (FIG. 6D) from the UART 31 to the LCP buffer 25_{00}, if the block check character OK level (BCCOK) is not active after the block check character has been checked, the block check character error flip-flop (BCCERF) is set to indicate the existence of a block check character error. The BCCOK level is provided by decoder 34 of the block check character register 33 in FIG. 6D.

Write Flip-Read Operations

This operation is essentially a Write operation followed by a Read operation. Basically the previous discussion regarding the "Write" operation and the "Read" operation of FIGS. 7B and 7C are applicable here. The receipt of a Command Descriptor C/D for the "write flip read" operation into the OP code and the variant registers 42 and 43 (FIG. 6D) respectively, causes a "Write" operation to be initiated and a FLIP level (Flip Level) to be generated. Data is transferred from the System 10 to the peripheral terminal unit 50 during the "Write" portion of the operation. When an ending code is recognized on the terminal bus 47 during data transfer from the LCP to the peripheral terminal unit 50 at STC 1 (FIG. 7C), then the end code level (EDCODE) is generated. The EDCODE level enables the setting of the end flip-flop (ENDF) indicating that the data transfer is complete. The set state of the end flip-flop (ENDF) and the generation of the FLIP level enable the setting of the read flip-flop (READF), the terminal receive flip-flop (TRECF), and the even flip-flop (EVNF), the resetting of the write flip-flop (WRITF), the terminal busy flip-flop (TRMBSYF), and the presetting of the buffer address to MADR 255. With these actions, the LCP is conditioned to receive data from the peripheral terminal unit 50, without reconnecting to the System 10 to receive additional instructions.

to initiate the "Read" portion of the Write-Flip-Read operation, the LCP does not reconnect to the System 10. As per FIG. 7E, from STC 1, the LCP steps over the STC 3 to await a transmission from the peripheral terminal unit 50. Receipt of the first character from the peripheral terminal unit 50 causes the DR level (Data Received) in the UART 31 to be active, enabling the setting of the reset UART flip-flop (RSUARTF) and the terminal busy flip-flop (TRMBSYF). The setting of the terminal busy flip-flop causes the LCP to return to STC 1 to receive the data. The "Read" operation progresses through to completion, subject to the same con-

tions discussed previously for a regular Read operaon.

Test Operation: The "test operation" provides the ystem 10 with the capability for determining the operional status of the LCP without requiring a transfer of ata to or from the System Memory 10_m. Located on a ata flow card is a test flip-flop (TESTF). The logic ate of this flip-flop is controlled by output levels from e OP code register 42, FIG. 6D. The set state indiites that a test instruction was received from the Sysm 10. In FIG. 7E, at STC 11, with the test flip-flop ESTF) set, the LCP has no requirement to step to TC 6 to receive a Descriptor Link D/L. It steps inead to STC 7 to return a Result Descriptor R/D to the ystem 10. From STC 7, the LCP steps over to STC 15, id then STC 3 (idle), where it remains until another ommand Descriptor C/D is received. Under normal onditions, the Result Descriptor R/D sent to the Sysm 10 for a "test operation" will have all bits equal to ero. The System 10 will recognize, by this condition, at the LCP is operational.

Test Enable Operation: The receipt of a Command escriptor C/D containing a "test enable" instruction onditions the LCP so that the peripheral terminal unit can initiate a communication with the System 10. he peripheral terminal unit 50 initiates a request for ommunication by sending an inquiry character (ENQ) the LCP. Upon receipt of the inquiry character ENQ), the "test enable" operation is terminated and e System initiates a "Read" operation to receive data om the peripheral terminal unit 50. If the terminal unit nds any other character but an ENQ inquiry characr, the character will not be recognized and the LCP ill take no action. The "test enable" operation operes (in reference to FIG. 7E) as follows:

At STC 3, upon receipt of a "test enable" instruction, e variant register flip-flop No. 3 (VAR3F) is set. The VAR (1-4) F" represents the 4 variant register levels. hese are generated on the Data Flow Card by outputs the variant register 43, FIG. 6D. The logic state of ese levels is dependent upon the numerical value coined in the variant digit 1 of the Command Descriptor /D. The setting of VAR3F inhibits the setting of the st flip-flop (TESTF) but allows the read flip-flop EADF) to be set. The LCP steps over to STC 11 to ceive the Command Descriptor longitudinal parity ord LPW from the System 10, and then steps over to TC 6 to receive the Descriptor Link D/L from the ystem. At STC 6, because the "Read" flip-flop EADF) is set, the LCP disconnects from the System and steps over to STC 1 to receive an inquiry characr (ENQ) from the peripheral terminal unit 50. At STC (unless an inquiry character (ENQ) is received immeately) the LCP steps over to STC 3 to await a transission from the peripheral terminal unit 50. When the rminal unit transmits, the terminal busy flip-flop RMBSYF) is set, causing the LCP to step over to TC 1 to receive the inquiry character (ENQ). When e ENQ is received, the set state of the variant register vel, VAR3F, inhibits the LCP from stepping over to TC 4 and also inhibiting the transfer of the character the System 10. Instead the LCP steps over to STC 7 return a Result Descriptor R/D to signify to the ystem 10 that the "test enable" operation is complete.

Conditional Cancel Operation: The "conditional canel operation" provides the System 10 with a capability cancel a previously sent Command Descriptor C/D ontaining a "Read" operation. Referring to FIG. 7E, if the LCP has initiated a "Read" or a "Write flip Read" operation, but the expected data transfer from the peripheral terminal unit 50 is not in progress, the LCP will remain at STC 3 awaiting a possible "conditional cancel" instruction. If a conditional cancel instruction is now received, the "Read" operation is cancelled and the cancel flip-flop (CANCF) is set. This cancellation will not be effectuated unless the LCP is at STC 3. The LCP then steps over to STC 11 to receive a Command Descriptor longitudinal parity word LPW from the System 10. The set state of the cancel flip-flop CANCF inhibits the LCP from stepping to STC 6. Instead, the LCP steps over to STC 7 to return a Result Descriptor R/D to the System 10, indicating that the conditional cancel operation is completed.

Echo Operation: The "echo operation" is a maintenance aid to trouble shooting of the LCP. This operation begins with a "Write" operation in which data is transferred from System Memory 10_m over to the LCP buffer 25_{00}. This is followed by a "Read" operation in which the same data is transferred back to System Memory 10_m. Assuming, for example, that less than a full buffer load of data will be transferred and that the operation will be terminated by receipt of an ending code in the last character position of a word; and since the "echo operation" is essentially a Write operation folowed by a Read operation, the following discussion will involve only those LCP actions which are unique to the echo operation. (Read and the Write operations were previously discussed in connection with FIGS. 7B and 7C). Now referring to FIG. 7E, at STC 6, and with the echo flip-flop (ECHOF) set, the LCP steps over to STC 8 to accept data from the System 10. Beginning at STC 8, the LCP operates as previously discussed during a regular "Write" operation up to the point that the LCP receives an ending code and then steps over to STC 12. At STC 12, although no data is to be transferred from the LCP to the peripheral terminal unit 50, the LCP disconnects from the System 10 by stepping momentarily over to STC 1. When disconnected at STC 1, the LCP initiates a request for reconnection to the System 10 by enabling the setting of: the LCP request flip-flop (LCPRQF); the I/O send flip-flop (IOSF): and by the presetting of the buffer address to MADR 253 (Descriptor Link, FIG. 6C). The LCP then steps over to STC 5 to send the Descriptor Link D/L to the System 10. At STC 5, the LCP transfers the Descriptor Link D/L to the System 10. The set state of the echo flip-flop (ECHOF) then causes the LCP to step over to STC 4 to return data in the buffer 25_{00} back to the System Memory 10_m. Beginning at STC 4, data is transferred from the LCP over to the System 10. The actions performed by the LCP are as those previously described during a regular "Read" operation up to the point that the LCP identifies an ending code on the terminal bus 47 and then steps over to STC 12. At STC 12, the Read operation is completed and the set state of the echo flip-flop (ECHOF) causes the LCP to step over to STC 7 to return a Result Descriptor R/D over to the System 10. Return of Result Descriptor R/D: FIG. 7D is a simplified logic flow diagram regarding the return of the Result Descriptor R/D. The LCP steps over to STC 7 to return a Result Descriptor R/D to the System 10 under any of the following conditions listed as (a), (b), (c), (d):

a. At STC 12 or STC 9 when a "Read" or an echo operation is completed.
b. At STC 5 when a "Write" operation is completed.

c. At STC **11** when any one of the following conditions occur:

(c1) A descriptor error occurred;

(c2) A test operation is specified by the Command Descriptor C/D being executed;

(c3) The conditional cancel flip-flop (CANCF) is set.

(d) At STC **6** if a vertical or a longitudinal parity error has occurred.

At STC **7**, if the transmit flip-flop (XMITF) is not set, it is set at this time to activate the LCP Read module. The terminal bus multiplexor select A level (SLARAM) and the terminal bus multiplexor select B level (SLBRAM) are both active, which allows the terminal bus multiplexor network (24_{x2} of FIG. **6D**) to select a word made up of Result Descriptor levels for transmission to the System **10**. When the Result Descriptor word is placed in the data latches, **49**, it is also applied to the LPW register 24_w to generate an LPW for the Result Descriptor transfer. The LCP then steps over to STC **15** to send the R/D LPW to the System **10**.

At STC **15**, the terminal bus multiplexor select A level (SLARAM) is inactive and the terminal bus multiplexor select B level (SLBRAM) is active, which allows the terminal bus multiplexor network (24_{x2} of FIG. **6D**) to select outputs of the LPW register 24_w for transmission to the System **10**. (The SLBRAM is used in conjunction with the SLARAM to select one of four inputs to the terminal bus multiplexor network). These are generated on the System Logic Card from outputs of the STC decoder **54** of FIG. **6D**. The LCP transfers the LPW, resets selected logic levels to a beginning state, and then steps over to STC **3**. The LCP remains at STC **3** until another Command Descriptor C/D is received.

In summary the LCP operates in two "modes" — the "off-line" mode and the "on-line" mode.

OFF-line Mode:

Operation of the LCP/Terminal Unit combination in an off-line mode is for the purpose of performing maintenance functions. In the field, a variety of operations can be performed to verify the condition of the LCP or for simple trouble shooting. These operations can be carried out without effecting the normal operation of other LCP's in the same Base Module.

On-line Mode

The two basic operations controlled by the LCP in the on-line mode operations are (1) a Write operation in which data is received from the System by the LCP and which data is transferred to the peripheral terminal unit; and (2) a Read operation in which data is received from the terminal unit by LCP and is transferred to the System Memory 10_m.

In addition to these basic operations, the LCP can change from a "Write" to a "Read" operation with a single instruction, and can also perform selected test operations. The following items represent the specific operations which the LCP can perform by means of program instructions from the Main System **10**. This is done by means of Command Descriptors (C/D) and herein follows a brief summary of what is accomplished by each operation.

Table XII here below summarizes the specific operations which the LCP can perform:

TABLE XII

a. Write	d. Test
b. Read	e. Test Enable
c. Write Flip Read	f. Conditional Cancel

TABLE XII-continued

g. Echo

Command Descriptors

The Command Descriptors (C/D) are instructions from the Main System **10** to the LCP regarding certain operations to be performed. The following items will summarize briefly the Command Descriptors associated with each of the instructions (of Table XII) from the Main System **10**:

(a) Write

The "Write" Command Descriptor is an instruction to transfer data from System Memory 10_m to the peripheral terminal unit desired, for example, such as peripheral terminal unit **50**. The LCP accepts data from the System **10** until the LCP buffer 25_{00}, for example, is full, or until the data transfer is stopped by the receipt of an "ending code" or a "terminate" signal from the Main System **10**. When the LCP buffer 25_{00} is full, or when an "ending code" is received, the LCP transfers the contents of the buffer 25_{00} to the peripheral terminal unit **50**. The "Write" Command Descriptor is identified as shown in Table XIII below:

TABLE XIII

Data Lines	(Write C/D) Digit Value	
A8	0	OP Digit
A4	1	
A2	0	
A1	0	
B8	0	Variant Digit 1
B4	0	
B2	0	
B1	0	

(b) Read

The "Read" Command Descriptor is an instruction to transfer data from the peripheral terminal unit involved, such as unit **50**, over to the System Memory 10_m. The LCP first accepts data from the peripheral terminal unit **50** until the LCP buffer 25_{00} is full, or until the data transfer is stopped by the receipt of an "ending code" from the peripheral terminal unit. When the LCP buffer 25_{00} is full, (or when the ending code is received), the LCP transfers the contents of the buffer 25_{00} over to the System Memory 10_m, unless the Main System **10** sends a "terminate" signal to stop the Read operation because System Memory space is not available to store any more data. If, after initiating a Read operation, the LCP receives no data for a period of one second, the LCP "times out" and sends a Result Descriptor (R/D) to the Main System **10**. The 1-second timing interval can be inhibited by setting a bit (**B1**) of the variant digit 1 of the Command Descriptor equal to 1. Table XIV below shows the "Read" C/D.

TABLE XIV

Data Lines	(Read C/D) Digit Value	
A8	1	OP Digit
A4	0	
A2	0	
A1	0	
B8	0	Variant Digit
B4	0	
B2	0	

TABLE XIV-continued

Data Lines	(Read C/D) Digit Value
B1	see note

If B1 is equal to 1, the one-second time-out period, allowed to the terminal unit to respond, is inhibited.

(c) Write Flip Read

The "Write flip Read" Command Descriptor is an instruction to the LCP to accomplish a Write operation, at the conclusion of which an immediate Read operation is performed without any intervention from the Main System **10**. Data is accepted from the Main System **10** and transferred to the peripheral terminal unit until an "ending code" is received. Upon receipt of the ending code from the Main System **10**, the LCP transfers the ending code to the peripheral terminal unit and then changes to the Read Mode. The LCP then accepts data from the peripheral terminal unit and transfers it to the System Memory $\mathbf{10}_m$ until an ending code is received from the peripheral terminal units, or until a terminate signal is received from the Main System **10**. If, after beginning of the Read portion of the operation, the LCP receives no data for a period of 1-second, then the LCP "times-out" and sends a Result Descriptor (R/D) to the Main System **10**. Of course, the one-second time interval can be inhibited if desired, by setting the bit B1 of the variant digit 1 of the Command Descriptor equal to one. Table XV below illustrates the "Write flip Read" Command Descriptor.

TABLE XV

Data Lines	(Write flip Read C/D) Digit Value	
A8	0	OP Digit
A4	1	
A2	0	
A1	0	
B8	1	Variant Digit
B4	0	
B2	0	
B1	see note	

If B1 is equal to 1, the one-second time-out period, allowed to the terminal unit to respond, is inhibited.

(d) Test

The "test" Command Descriptor is an instruction to the LCP to indicate its "operational status" by returning a Result Descriptor (R/D) to the Main System **10**. If the LCP is present and available, the Result Descriptor will be equal to all "0's". Table XVI below shows the Test Command Descriptor:

TABLE XVI

Data Lines	(Test C/D) Digit Value	
A8	0	OP Digit
A4	0	
A2	1	
A1	0	
B8	0	Variant Digit 1
B4	0	
B2	0	
B1	0	

(e) Test Enable

The "test enable" Command Descriptor is an instruction to the LCP to monitor incoming data from the peripheral terminal unit, and upon receipt of an Inquiry Character (ENQ), to form and transmit a Result Descriptor (R/D) to the System **10**. This instruction is used to allow the peripheral terminal unit to initiate a communication with the Main System **10**. Table XVII below illustrates this Command Descriptor.

TABLE XVII

Data Lines	(Test Enable C/D) Digit Value	
A8	0	OP Digiti
A4	0	
A2	1	
A1	0	
B8	0	Variant Digit
B4	1	
B2	0	
B1	see note	

If B1 is equal to 1, the one-second time-out period, allowed to the terminal unit to respond, is inhibited.

(f) Conditional Cancel

The "Conditional Cancel" Command Descriptor is an instruction to the LCP to initiate cancellation of another Command Descriptor under certain conditions. When the Conditional Cancel Command Descriptor is received by the LCP, and, if data is not being received from the peripheral terminal unit during the applicable portion of a Read operation, then the previous Command Descriptor will be cancelled. This C/D is shown in Table XVIII:

TABLE XVIII

Data Lines	(Conditional Cancel C/D) Digit Value	
A8	0	OP Digit
A4	0	
A2	1	
A1	0	
B8	1	Variant Digit 1
B4	0	
B2	0	
B1	0	

(g) Echo

The "Echo" Command Descriptor is an instruction to the LCP to accept a full buffer of data (or less) from the Main System **10** and then to return the same data back to the Main System **10** to be stored. This provides a maintenance check and trouble shooting diagnosis cycle for the System-LCP operations. Table XIX illustrates this Echo Command Descriptor.

TABLE XIX

Data Lines	(Echo C/D) Digit Value	
A8	0	OP Digit
A4	0	
A2	0	
A1	1	
B8	0	Variant Digit 1
B4	0	
B2	0	
B1	0	

Having described a digital data processing system involving a plurality of I/O Subsystems for managing data transfer operations and including certain modular units, such as the Line Control Processor, the Base Module, the Input-Output Translator and the interrelationships thereof, the following claims are made:

What is claimed is:

1. In a digital system for the transfer and control of digital information between a first main system which includes a processor with main memory having an I/O translator interface unit, and a plurality of remote peripheral terminal units, wherein each peripheral terminal unit is connected to the main system via a corresponding line control processor, and wherein groups of said line control processors at a given site are organized into Base Modules, a Base Module unit comprising:

(a) a plurality of line control processors, each of which has its own dedicated connection to a corresponding peripheral terminal unit and wherein each of said line control processors is adapted to handle the particular discipline required by its corresponding peripheral terminal unit,

each of said line control processors including a data buffer memory having sufficient memory space for storing at least one complete message block of information data and control data;

(b) a single backplane common to each of said plurality of line control processors within said Base Module, said backplane including transmission lines from each line control processor to a distribution-control means, said transmission lines being selectively activated by said distribution-control means;

(c) connection means for connecting of each said plurality of line control processors with a dedicated transmission line to its corresponding peripheral terminal so that each peripheral terminal has an unimpeded exclusive line connection to its corresponding line control processor;

(d) a distribution-control means including:

(d1) message level interface means including logic means to connect a data path having a standard transmission discipline to said I/O translator interface of said main system, said message level interface operating to transfer a stream of characters without interruption until a complete message block is transferred;

(d2) means to signal said main system if an addressed line control processor is busy;

(d3) means to signal said main system if an addressed line control processor is available, and to connect said addressed line control processor to said main system;

(d4) means, when a line control processor requests access, to connect said requesting line control processor to said main system, by activation of selected backplane transmission lines to said line control processor requesting access;

(d5) means for setting a local base priority value number for each line control processor within a given base module, and using said value number for determining priority of access when simultaneous requests for access occur between line control processors in the same Base Module.

2. The Base Module of claim 1, wherein said distribution-control means includes:

(d6) means for setting a global priority value number for each line control processor to provide said input/output translator interface of said main system with data to determine precedence of access to memory among competing line control processors in different Base Modules.

3. The Base Module of claim 2, wherein said distribution-control means includes:

(d7) emergency request interrupt means when a line control processor is requesting access to memory, to provide a signal to said input/output translator with the topmost priority value number for that line control processor when lack of such access would result in an incomplete message transmission causing an access error.

4. The Base Module of claim 1, wherein said distribution-control means includes:

means for setting a global priority number signal for each line control processor;

means for connecting, when two or more line control processors in a Base Module simultaneously request access, the line control processor with the highest local priority number to said distribution-control means and then transmitting, to the said input/output translator interface, the global priority number for the connected line control processor.

5. The Base Module of claim 1, operating in connection with an additional second main system wherein each of said first and second main systems include an input/output translator interface which generates a line control processor address signal and a channel address signal for each I/O operation, said distribution-control means further including:

means, after receipt of a line control processor address signal and a channel address signal from said first or second main system to check for availability of the selected line control processor;

means to decode the line control processor address signal for activating the backplane transmission lines to said addressed line control processor;

means to signal the input/output translator interface of said main system that the addressed line control processor is now connected for data transfer;

second message level interface connection means to said second main system.

6. The Base Module of claim 5, wherein said distribution-control means includes:

means, when the said input/output translator interface requests connection to a specifically addressed line control processor, to check parity of the request signal and parity of the address signal, to check whether the addressed line control processor is available and thence to signal a response to the said input/output translator interface.

7. In a digital system for the transfer and control of digital information between a main system, which includes a processor with main memory having an I/O translator interface unit, and a plurality of remote peripheral terminal units, wherein each peripheral terminal unit is connected to the main system via a corresponding line control processor, and wherein groups of said line control processors at a given site are organized into Base Modules, a Base Module unit comprising:

(a) a plurality of line control processors, each of which has its own dedicated connection to a corresponding peripheral terminal unit and wherein each of said line control processors is adapted to handle the particular discipline required by its corresponding peripheral terminal unit, each of said line control processors including a data buffer

memory having sufficient memory space for storing at least one complete message block of information data and complete control data, said control data including a command/descriptor word to control the type of operation, a descriptor link word to identify the particular input/output command involved, and a result descriptor word to inform the main system of completion of the command or error therein;

(b) a single backplane common to each of said plurality of line control processors within said Base Module, said backplane including:

(b1) transmission lines from each line control processor to a distribution-control means,

(b2) transmission lines from each line control processor to a maintenance card unit and a termination card unit;

(b3) a first exclusively dedicated line from each line control processor to said distribution-control means for emergency requests for access to the main system,

(b4) a second exclusively dedicated line from each line control processor to said distribution-control means for signaling the particular line control processor that is addressed by said main system;

(c) connection means for connecting of each said plurality of line control processors with a dedicated transmission line to its corresponding peripheral terminal so that each peripheral terminal has an unimpeded exclusive line connection to its corresponding line control processor.

(d) a distribution-control means including:

(d1) register means to receive the address of a selected line control processor from said main system;

(d2) decoding means to enable connection, via said backplane, of said selected line control processor to said main system;

(d3) gating means for controlling the connection or disconnection of said selected line control processor to said main system;

(e) maintenance card means connected to said backplane including:

(e1) a clock for supplying timing signals to each line control processor;

(e2) means for providing on-line diagnostic signals to any selected line control processor and for simulating the peripheral unit corresponding to each line control processor;

(e3) means to access said data buffer memory to check its operation;

(f) termination card means connected to said backplane including:

(f1) power supply means for each line control processor and said distribution control means;

(f2) termination impedance means to balance the transmission line impedances of each of the lines on said backplane;

(g) means for setting a local priority number signal for each line control processor in said Base Module, for selecting priority of access when access-requests occur simultaneously by two or more line control processors in that Base Module:

(h) message level interface means whereby said distribution control means operates to transfer a stream of characters to or from said main system without interruption until a complete message block is transferred.

8. The Base Module of claim 7, wherein said distribution-control means includes:

first logic means to signal the main system of a previously addressed line control processor that the addressed line control processor is available or non-available, and to connect the addressed line control processor, when available, to said main system;

second logic means to signal the main system when a line control processor requests access and to connect, when said main system grants access, said line control processor to said main system.

9. The Base Module of claim 8, wherein said main system provides a command descriptor word to said line control processor and said line control processor includes:

execution logic means to execute said command descriptor word received from said main system; and

a status count register having a count number which determines the sequence of logical operations to be followed by said execution logic means.

10. In a digital system for the transfer and control of digital information between a main system, which includes a processor with main memory having an I/O translator interface unit, and a plurality of remote peripheral terminal units, wherein each peripheral terminal unit is connected to the main system via a corresponding line control processor, and wherein groups of said line control processors at a given site are organized into Base Modules, a Base Module unit comprising:

(a) a plurality of line control processors each of which is connected by a dedicated transmission line to its corresponding peripheral terminal unit and wherein each of said line control processors includes:

(a1) a data buffer memory having storage for a complete message block of data and a complete set of control words including I/O commands received from main memory;

(a2) logic means to execute said I/O commands and to control data transfers between said data buffer memory and said main system and/or its corresponding peripheral terminal unit;

(a3) register means for developing status condition signals to cause said logic means to actuate the sequential switching required to execute said I/O commands;

(b) distribution-control means including:

(b1) means to receive the channel number and the address of a selected line control processor;

(b2) means to check the availability of a selected line control processor, and if available, to connect said line control processor to said main system;

(b3) gating means to control the connection or disconnection of a selected line control processor to said main system;

(b4) means to provide an internal base priority signal for each line control processor and to connect the highest priority line control processor to the main system when two or more line control processors simultaneously request access;

(b5) means to provide a global priority signal, to said main system, for each line control processor requesting access to said main system;

(b6) message level interface means to said main system and operating to transfer a complete message block of data without interruption;

(c) a common backplane means providing transmission lines to and from each line control processor to said distribution-control means and including:

(c1) a dedicated address line from each line control processor to said distribution-control means;

(c2) a dedicated line for access-requests from each line control processor to said distribution-control means;

(c3) connection means from each line control processor to a maintenance unit means and to a termination unit means;

(d) a common maintenance unit means connected to each line control processor for simulating each peripheral terminal and testing the operation of each line control processor;

(e) a common termination unit means connected to said backplane means for providing line termination impedances, and power to each of said line control processors.

11. The Base Module of claim **10**, wherein said line control processor includes:

(a) means, when said data buffer memory is either full or empty, for signaling an emergency request for access to said distribution-control means whereupon said distribution-control means will generate the top priority number access-request signal for transmittal to said main system.

* * * * *

UNITED STATES PATENT OFFICE

CERTIFICATE OF CORRECTION

Patent No. 4,074,352 Dated February 14, 1978

Inventor(s) Darwen J. Cook and Donald A. Millers, II

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Col. 5, line 29, delete "in";
line 53, change "10c" to —10_c—.

Col. 8, line 10, between "units" and "grouped", insert —(Line Control Processors)—;
line 11, delete "(Line Control Processors)";
line 57, change "Basic" to —Base—.

Col. 11, lines 27-30, delete Table IV and substitute the following:

TABLE II : Data Word

Digits —— A B C D

A8	B8	C8	D8
A4	B4	C4	D4
A2	B2	C2	D2
A1	B1	C1	D1

One Digit = 4 bits

One Character = 8 bits = AB

One Word = ABCD = 16 bits—

lines 50-56, delete in entirety.

Col. 17, line 64, change "ASCIII/EBCDIC" to —ASCII/EBCDIC—;

Col. 19, line 35, before "and" insert —(—.

Col. 24, line 53, change "of" to —to—.

Col. 28, line 36, change "on" to —in—.

Col. 29, line 9, change "an" to —and—.

UNITED STATES PATENT OFFICE

CERTIFICATE OF CORRECTION

Patent No. 4,074,352 Dated February 14, 1978

Inventor(s) Darwen J. Cook and Donald A. Millers, II

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Col. 30, line 2, change "partly" to —parity—.
Col. 31, line 9, change "if" to —it—.
Col. 39, line 25, change "CP" to —LCP—.
Col. 40, line 36, change "ST-1," to —ST-1/,—.
Col. 41, line 4, change "(State 1 + 7)" to —(State = 1 + 7)—.
Col. 42, line 18, change "acknowledges" to —acknowledged—;
line 21, change "St" to —ST—;
line 33, change "in", second occurrence, to —is—.
Col. 43, line 2, change "St" to —ST—.
Col. 44, line 61, change "either" to —eight—.
Col. 45, line 20, change "is" to —in—;
line 29, change "26" to —36—.
Col. 47, line 8, change "digits", second occurrence, to —digit—.
Col. 50, line 17, change "and" to —an—.
Col. 52, line 64, change "(WESVS)" to —(WESYS)—;
line 65, delete "(WESYS)".
Col. 54, line 32, delete "the", second occurrence;
line 50, change "work" to —word—.
Col. 56, line 49, change "not" to —no—;
Col. 62, line 67, after "begins" insert —.—;
"(c) Request for Emer-" should be on next line.
Col. 63, line 55, delete "Main";
line 56, between "the" and "System", insert —Main—.

UNITED STATES PATENT OFFICE

CERTIFICATE OF CORRECTION

Patent No. 4,074,352 Dated February 14, 1978

Inventor(s) Darwen J. Cook and Donald A. Millers, II

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Col. 66, line 57, change "to" to —To—;
line 60, change "the", first occurrence, to —to—.
Col. 68, line 27, change "folowed" to —followed—.

Signed and Sealed this

Twenty-seventh Day of March 1984

[SEAL]

Attest:

GERALD J. MOSSINGHOFF

Attesting Officer

Commissioner of Patents and Trademarks